Attentiveness to Answer Choices Doesn't Always Entail High QA Accuracy

Sarah Wiegreffe, Matthew Finlayson, Oyvind Tafjord, Peter Clark, Ashish Sabharwal

Allen Institute for AI, Seattle, U.S.A.
wiegreffesarah@gmail.com, mfinlays@usc.edu
{oyvindt,peterc,ashishs}@allenai.org

Abstract

When large language models (LMs) are applied in zero- or few-shot settings to discriminative tasks such as multiple-choice questions, their attentiveness (i.e., probability mass) is spread across many vocabulary tokens that are not valid choices. Such a spread across multiple surface forms with identical meaning is thought to cause an underestimation of a model's true performance, referred to as the "surface form competition" (SFC) hypothesis. This has motivated the introduction of various probability normalization methods. However, many core questions remain unanswered. How do we measure SFC or attentiveness? Are there direct ways of increasing attentiveness on valid choices? Does increasing attentiveness always improve task accuracy? We propose a mathematical formalism for studying this phenomenon, provide a metric for quantifying attentiveness, and identify a simple method for increasing it—namely, in-context learning with even just one example containing answer choices. The formalism allows us to quantify SFC and bound its impact. Our experiments on three diverse datasets and six LMs reveal several surprising findings. For example, encouraging models to generate a valid answer choice can, in fact, be detrimental to task performance for some LMs, and prior probability normalization methods are less effective (sometimes even detrimental) to instruction-tuned LMs. We conclude with practical insights for effectively using prompted LMs for multiple-choice tasks.

1 Introduction

Large pre-trained autoregressive language models (LMs) have shown success not only on generation, but also on classification and multiple-choice tasks with pre-specified answer choices (Dai and Le, 2015; Howard and Ruder, 2018; Raffel et al., 2020; Lewis et al., 2020; Brown et al., 2020, *inter alia*). To succeed on such tasks, one must pay attention to what's a valid answer choice and what's not, i.e., understand the *task format*. This is accomplished relatively easily in the pre-train-and-finetune paradigm (Dai and Le, 2015; Howard and Ruder, 2018; Raffel et al., 2020; Lewis et al., 2020), via task-specific fine-tuning.[1] However, enforcing the generation of valid answer choices is more difficult in the prompting and in-context learning paradigms, in which the model is provided only a description or a handful of examples of the target task (Radford et al., 2019; Brown et al., 2020). The model's attentiveness (i.e., probability mass) instead tends to be spread across many vocabulary tokens that are not valid answer choices.

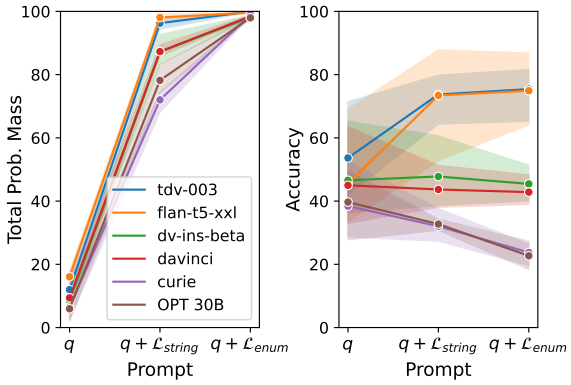


Figure 1: Higher probability mass on valid answer choices (left; Eq. (4)) does not always translate to better accuracy (right; Eq. (1)), as shown here for three different prompt formats (§6.3) each with one in-context example. Results are averaged across MMLU, OpenbookQA, and CommonsenseQA (§6.2) for six LMs (§6.1). Including answer choices in the prompt substantially increases probability mass on (i.e., attentiveness to) valid answer choices. However, attentiveness is surprisingly **not always** associated with increased accuracy; in fact, it can lead to a **substantial drop** in performance (e.g., for OPT 30B and GPT-3 `curie`).

Most prior work tries to circumvent this issue

[1] For example, a T5 (Raffel et al., 2020) model fine-tuned on question-answering tasks (UnifiedQA; Khashabi et al., 2020) generates a valid answer choice string on the OpenbookQA validation set 99.4% of the time.

by ignoring generated predictions and instead selecting a valid answer choice that has the highest probability under the model ("sequence scoring"; Trinh and Le, 2018; Radford et al., 2019; Brown et al., 2020, *i.a.*). This helps to some extent by ignoring any attention the models pays to tokens unrelated to valid answer choices. However, the problem persists as the model's attentiveness can still be split among various strings or surface forms that are *semantically equivalent* to a valid answer choice.

Holtzman et al. (2021) propose that this phenomenon can result in underestimates of model performance, and refer to it as the **surface form competition** ("SFC") hypothesis. Motivated by this, they propose to use a *probability normalization* method, PMI_{DC}, as a way to address the SFC issue, thereby (according to the SFC hypothesis) increasing model performance. In the same spirit, other probability normalization methods have also been proposed (Zhao et al., 2021; Malkin et al., 2022), and their merit assessed in terms of end task accuracy.

However, accuracy improvements may be attributable to multiple sources. Thus, without a metric to directly measure SFC or, more generally, a model's attentiveness to valid answer choices,[2] it is difficult to assess whether the increased accuracy of these methods is, in fact, a consequence of reduced SFC or increased attentiveness.

To address this gap, we propose a mathematical formalism for studying this phenomenon and use it to investigate four research questions:

1. **How can we measure SFC?** We propose total probability mass on valid answer choices as a metric for a model's attentiveness to answer choices, and use it to upper bound the extent and impact of SFC (§4).

2. **How can a model's attentiveness be increased?** Lack of attentiveness is a consequence of an inherently *under-constrained* output space that arises from the model failing to understand the task format. We use this observation to explain a simple way of increasing attentiveness: using in-context learning with prompts containing answer choices (§5.1 and §5.2). We demonstrate its empirical success across 6 LMs and 3 datasets (§7.1).

3. **Does increasing attentiveness improve accuracy?** Surprisingly, not always! We provide an upper bound on the maximum effect increase in attentiveness can have on accuracy (§4.1). We find empirically (Fig. 1 and §7.2) that the alignment between attentiveness and accuracy is heavily dependent on the model. This is an important finding because it demonstrates that encouraging models to produce valid answer choices by showing them in the prompt can counter-intuitively be **detrimental** to task performance for many LMs trained only on the next-token prediction objective.

4. **When do probability-normalization methods improve accuracy?** While the direct effect of PMI_{DC} on SFC is not easy to measure (§3.4), we extend prior work by studying when PMI_{DC}, which is complimentary to our approach, improves accuracy on a wider set of prompts and models. We find that it always has a positive effect on accuracy when models are not shown answer choices, which generally also corresponds with low attentiveness. However, when models benefit from seeing answer choices, which results in high attentiveness, PMI_{DC} has a generally negative effect on accuracy. This indicates that as instruction-tuned LMs become more commonplace, PMI-based scoring may provide less utility.

We conclude by leveraging these insights to provide practical recommendations on how to maximize LM accuracy on multiple-choice tasks when using zero- and few-shot prompting.

2 Related Work

While various methods have been proposed to improve the accuracy of sequence scoring using probability normalization methods (Brown et al., 2020; Zhao et al., 2021; Holtzman et al., 2021; Malkin et al., 2022), most do not investigate surface form competition as a metric and whether their methods alleviate it. To the best of our knowledge, we are the first to systematically study the role of in-context examples and prompt format on a model's attentiveness, as well as the relationship between attentiveness and accuracy.

Holtzman et al. (2021) show PMI_{DC} improves over sequence scoring accuracy in most cases for GPT-2 and GPT-3 base models of various sizes

[2]For brevity, we will henceforth write 'attentiveness' to mean 'attentiveness to valid answer choices'.

in a 0-shot and 4-shot setting. Somewhat contradictorily, Brown et al. (2020) find that using a version of Eq. (3) where the denominator is $P_\theta(x|\text{“Answer : ”or“A : ”})$ improves task performance on the validation set for only 5 out of 17 datasets investigated. Zhao et al. (2021) propose to fit a linear weight matrix and bias vector for classification tasks with a shared label set, such that the labels all have equal probability prior to observing x. Malkin et al. (2022) add hyperparameters to Eq. (3) that are fit on a dataset's validation set, showing further gains at test-time. Min et al. (2022) propose to score inputs given answer choices ($p(x|\ell)$), which is mathematically equivalent to PMI_{DC} (§3.4). This results in lower variance and better *worst-case* accuracy on multiple-choice tasks in 0- and few-shot settings for GPT-2.

Liang et al. (2022) investigate the effect of showing answer choices in the prompt and applying PMI based scoring (though not the combination of the two) and find that the success of one method over the other tends to vary by dataset and model. Our findings agree that PMI-based scoring improves over sequence scoring for OpenbookQA and MMLU when answer choices are not given. They also find that in 5/6 models, PMI-based scoring with no answer choices in the prompt results in the highest accuracy for these datasets; we find this for 4/6 (different models). Our results elucidate further that overall capability and/or instruction tuning may be a key factor in whether this finding holds or is in fact the inverse for a model.

3 Background

We first provide a formalism for attentiveness and the Surface Form Competition (SFC) hypothesis, and then discuss the associated solution, PMI_{DC}, of Holtzman et al. (2021).

3.1 Problem Statement

Given a set of answer choices $\mathcal{L}$ and correct answer $y^* \in \mathcal{L}$ for a task input x, the goal of a multiple-choice classification task is to correctly select y^*. The task input x is often specified by a question q and, optionally, valid answer choices $\mathcal{L}$ concatenated to q as one string.[3]

Let M be a generative model architecture with learned parameters θ. We can use M_θ to solve the task via a standard *sequence scoring* approach:

$$\hat{y} = \underset{\ell \in \mathcal{L}}{\text{argmax}}\, P_\theta(\ell|x) \quad (1)$$

This is the most common approach for performing classification with generative LMs, as it ensures the prediction is always a valid answer choice. It is the prediction setup we will use for analysis.

3.2 Surface Form Competition Hypothesis

The SFC hypothesis (Holtzman et al., 2021) posits that an LM's vocabulary can contain many different strings, or *surface forms*, for representing the same (or similar) semantic concept, but only one of them is a **valid answer choice** for the task. When LMs distribute probability mass across surface forms such that the valid surface form for the correct answer is assigned a lower probability than the valid surface form for an incorrect answer, the model's prediction will be considered incorrect even if the *total* probability placed by the model on the correct *concept* is higher than that of the incorrect concept (Fig. 2, left).

Formally, there exists a set of possible synonyms $\mathcal{G}_\ell$ for each answer choice $\ell \in \mathcal{L}$ that may be "stealing" probability mass from ℓ, because ℓ is the only *valid surface form* when doing prediction. Kuhn et al. (2023) refer to $\mathcal{G}_\ell$ as a *semantic equivalence class*, and we use this terminology. For example, if $\mathcal{L} = \{A, B, C\}$, then there exist semantic equivalence classes $\mathcal{G}_A, \mathcal{G}_B$, and $\mathcal{G}_C$ containing all synonyms of A, B, and C, respectively. If $A = \mathit{True}$, $\mathcal{G}_A$ might be $\{\mathit{True}, \mathit{true}, \mathit{yes}, \mathit{verified}, \ldots\}$, which are all token generations the language model may use to express the semantically-equivalent concept. By definition, classes $\mathcal{G}_\ell$ are disjoint, and the valid surface form ℓ is always a member of its class (i.e., $\forall \ell \in \mathcal{L} : |\mathcal{G}_\ell| \geq 1$).

The original paper provides an example of how SFC may lead to incorrect predictions, which we extend and adapt in Fig. 2 (left).

3.3 Inference Under SFC

Sequence scoring for predicting the answer (Eq. (1)) does not take into account other surface forms semantically equivalent to the valid answer choices. A solution that would fully alleviate surface form competition and compute a fully-"SFC-free" prediction is to take these other surface forms into account and compute the most likely option

[3] For instance, if x is a true/false question, $\mathcal{L}$ may be $\{\mathit{True}, \mathit{False}\}$. For a multiple choice question, $\mathcal{L}$ may be the set of (string) answers, their labels such as A/B/C/D, or both, depending on the format used to pose the task to an LM.

among semantic equivalence classes, rather than among specific surface forms:

$$y^{\text{SFC-free}} = \underset{\ell \in \mathcal{L}}{\text{argmax}}\, P_\theta(\mathcal{G}_\ell | x) \tag{2}$$

where $P_\theta(\mathcal{G}_\ell|x) = \sum_{z \in \mathcal{G}_\ell} P_\theta(z|x)$. A limitation of this formulation is that it is only possible to compute if the full membership of each $\mathcal{G}_\ell$ is known, which is rarely the case. Language model vocabularies typically contain many tens of thousands of tokens, many of which may be partial synonyms. This motivates the need for practical workarounds.

3.4 PMI_{DC} as a Workaround

Holtzman et al. (2021) propose the following alternative selection method, PMI_{DC}:[4]

$$y^{\text{PMI-DC}} = \underset{\ell \in \mathcal{L}}{\text{argmax}}\, \frac{P_\theta(\ell|x)}{P_\theta(\ell)} \tag{3}$$

Intuitively, PMI_{DC} measures the causal effect[5] of the input x on the probability assigned to each label ℓ, and selects $\hat{y}$ as the label on which x had the largest effect. The method can be seen as computing predictions using an alternative scoring function, without changing the underlying probabilities of the language model, P_θ.

It is unclear when $y^{\text{PMI-DC}} = y^{\text{SFC-free}}$, i.e., when Eqs. (2) and (3) lead to the same, SFC-free prediction. Holtzman et al. note that PMI_{DC} is mathematically equivalent to $\text{argmax}_{\ell \in \mathcal{L}}\, P_\theta(x|\ell)$. This, in turn, should intuitively not be far from $\text{argmax}_{\ell \in \mathcal{L}}\, P_\theta(x|\mathcal{G}_\ell)$ when ℓ is not directly mentioned in the question (which is the setting used in PMI_{DC}). In this case, the competition among surface forms within $\mathcal{G}_\ell$ would be alleviated. However, there is still no a priori reason for either $\text{argmax}_{\ell \in \mathcal{L}}\, P_\theta(x|\ell)$ or $\text{argmax}_{\ell \in \mathcal{L}}\, P_\theta(x|\mathcal{G}_\ell)$ to be the same as $y^{\text{SFC-free}}$. Moreover, this view reveals a different competition, namely, among various questions x whose answer (according to the model) is ℓ. Specifically, a valid choice that the model thinks is the answer to *many* questions will receive an artificially low PMI_{DC} score relative to a choice that is the answer to only a few questions. In other words, now different questions (rather than different surface forms) compete for each answer choice.

[4] W.l.o.g., we ignore their use of a "domain context" string in the denominator.

[5] In the sense that it measures the multiplicative factor by which the probability of ℓ increases upon observing x.

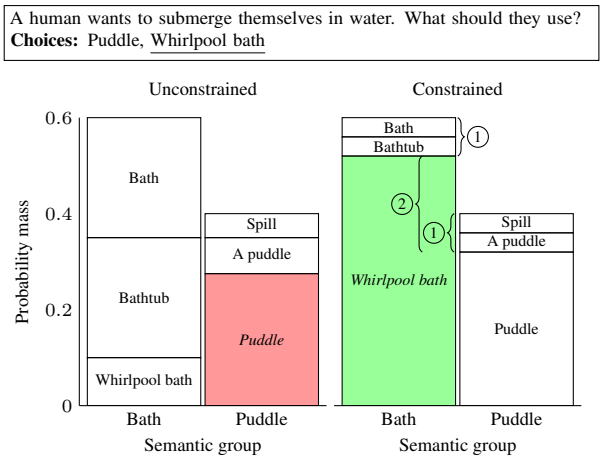


Figure 2: Left: A visualization of the surface form competition hypothesis (§3.2): invalid answer choices can "steal" probability from the correct answer choice, leading to an incorrect prediction. Right: LMs can be constrained to place more probability mass on the valid answer choices (§5). So long as the probability mass on invalid choices (1) is less than the difference in probability mass between valid choices (2), surface form competition cannot affect the model's prediction (§4.1).

4 How can we measure SFC?

Prior work has solely tested approaches geared towards resolving SFC by using task accuracy as the metric. However, it is unclear whether task accuracy is an effective measure of the amount of SFC present. In fact, as we will show later, task accuracy is often *not* correlated with the amount of SFC.

Can we measure the extent to which a model M_θ suffers from SFC when operating on input x and valid answer choices $\mathcal{L}$? We propose to do so in terms of M_θ's **probability mass on valid answer choices** or PMV, defined as follows:

$$\text{PMV}_\theta(\mathcal{L}, x) = \sum_{\ell \in \mathcal{L}} P_\theta(\ell|x) \tag{4}$$

where $P_\theta(y)$ is the probability M_θ assigns to output y. Intuitively, if a model is properly trained or instructed, it would place all probability mass on choices in $\mathcal{L}$, resulting in $\text{PMV}_\theta(\mathcal{L}, x) = 1$. However, if surface form competition exists, we would observe $\text{PMV}_\theta(\mathcal{L}, x) < 1$.

The probability mass that the model does *not* place on $\mathcal{L}$ must be placed on either the synonyms of $\mathcal{L}$ (i.e., "stolen" by these synonyms as a result of surface form competition) or on other tokens outside $\mathcal{L}$'s semantic equivalence classes. Thus,

the amount of surface form competition present for M_θ on this data instance is bounded as follows:

$$\text{SFC}_\theta(\mathcal{L}, x) \leq 1 - \text{PMV}_\theta(\mathcal{L}, x) \quad (5)$$

For a model trained to fit a probability distribution to only the elements of $\mathcal{L}$, we would observe $\text{SFC}_\theta(\mathcal{L}, x) = 0$. In practice, given that answer choices may be > 1 token in length, we compute $\text{PMV}_\theta(\mathcal{L}, x)$ using the set of unique first tokens of the answer choices in $\mathcal{L}$.

4.1 When Can SFC Impact Accuracy?

The formulation of SFC as a measurable quantity also allows us to quantify the maximum amount by which this phenomenon may be impacting a prediction. Specifically, the amount of probability mass that does *not* fall on $\mathcal{L}$ cannot affect the model's final prediction if it is less than the difference in probability between the highest-probability answer choice, $\hat{y}$, and the second-highest-probability answer choice, $y_2 \in \mathcal{L}$. The righthand side of Fig. 2 illustrates this principle. Formally, surface form competition simply *cannot* affect the output of M_θ on input x if the following holds:

$$1 - \text{PMV}_\theta(\mathcal{L}, x) < P_\theta(\hat{y}|x) - P_\theta(y_2|x) \quad (6)$$

In other words, one can completely remove the impact of surface form competition on a model's accuracy by raising PMV high enough (relative to the gap between the probabilities of $\hat{y}$ and y_2); it doesn't have to be fully resolved (i.e., one does not need PMV = 1).

Further, since $P_\theta(\hat{y}|x) + P_\theta(y_2|x) \leq 1$, the above condition is satisfied if we observe that $P_\theta(\hat{y}|x)$ is large enough, specifically, if:

$$P_\theta(\hat{y}|x) > 1 - \frac{\text{PMV}_\theta(\mathcal{L}, x)}{2} \quad (7)$$

This gives us a simpler empirical test that ensures SFC is not impacting a model's accuracy. Not surprisingly, if $P_\theta(\hat{y}|x) > 0.5$, then this condition is satisfied and SFC cannot impact model accuracy. In §7, we will report the tighter bound (Eq. (6)).

5 How can SFC be reduced?

Note that the quantities used to make predictions in PMI_{DC} do not directly represent a valid probability distribution in that they are no longer upper-bounded by 1,[6] making it difficult to compute our proposed metric $\text{PMV}_\theta(\mathcal{L}, x)$ for measuring the extent of SFC. Is there a more straightforward way to equate Eqs. (1) and (2)?

5.1 Using In-Context Examples

A clear way to do so is to somehow directly constrain the model M_θ such that:

$$P_\theta(\mathcal{G}_\ell|x) = P_\theta(\ell|x) \quad \forall \ell \in \mathcal{L} \quad (8)$$

This holds if and only if ℓ is the only member of $\mathcal{G}_\ell$ to which M_θ assigns non-zero probability mass. This, we posit, will occur naturally when language models are properly constrained or instructed (see Fig. 2, right).

One means to achieve this is to condition the predictions of M_θ on not only x but also on some in-context examples $e_0, \ldots, e_k$:

$$y^{\text{ICE}} = \underset{\ell \in \mathcal{L}}{\text{argmax}}\, P_\theta(\ell|x; e_0, \ldots, e_k) \quad (9)$$

Given that Eq. (9) is already widely used in practice, this technique is simple and straightforward to implement. It is additionally straightforward to empirically measure the effectiveness of this method for reducing SFC and quantify its error bounds. In §6, we demonstrate empirically that with effective conditioning (prompt format and number of in-context examples), using in-context examples can significantly reduce surface form competition. In fact, we often find that an effective prompt can condition models to place *all* of the probability mass on valid answer choices (§7.1).

5.2 Prompting With Answer Choices

A key design decision when choosing which format to use to specify x (and optionally in-context examples $e_0, \ldots, e_k$) is whether to provide the model only the question q or also the answer choices $\mathcal{L}$. Our PMV metric can be used to provide insight into this, by helping disentangle the contribution that each of q and $\mathcal{L}$ makes to the task accuracy as well as to reducing surface form competition.

Intuitively, conditioning the prediction on $\mathcal{L}$ makes the model aware of what's a valid answer choice and what's not. It can thus push the model towards the specific surface forms contained in $\mathcal{L}$, without necessarily affecting model accuracy.

[6] The quantity $\frac{P_\theta(\ell|x)}{P_\theta(\ell)}$ can, in principle, be viewed as the *unnormalized* probability of ℓ. However, turning it into a proper probability distribution requires computing the normalization factor $\sum_z \frac{P_\theta(z|x)}{P_\theta(z)}$, which is prohibitively expensive and also unreliable as LMs are generally not well-calibrated on the long tail of low-probability tokens.

This, by definition, directly increases the probability mass over valid answer choices. One can empirically quantify the effect of exposure to $\mathcal{L}$ by considering the gain one observes in PMV and in accuracy when going from $P_\theta(\ell)$ to $P_\theta(\ell|\mathcal{L})$.

On the other hand, one would expect that conditioning the prediction on q pushes the model towards the correct semantic concept, i.e., the semantic equivalence class $\mathcal{G}^*$ of the correct answer. However, not knowing which specific surface form ℓ^* appears in both $\mathcal{G}^*$ and $\mathcal{L}$, the model has no reason to prefer ℓ^* over other equivalent surface forms $\ell \in \mathcal{G}^* \setminus \{\ell^*\}$. Thus, conditioning on q alone can increase accuracy by increasing the probability mass on $\mathcal{G}^*$, but it does not resolve SFC within $\mathcal{G}^*$. We can, again, measure this by considering the gain in PMV and accuracy when going from either $P_\theta(\ell)$ to $P_\theta(\ell|q)$ or from $P_\theta(\ell|\mathcal{L})$ to $P_\theta(\ell|q, \mathcal{L})$.

6 Experiments

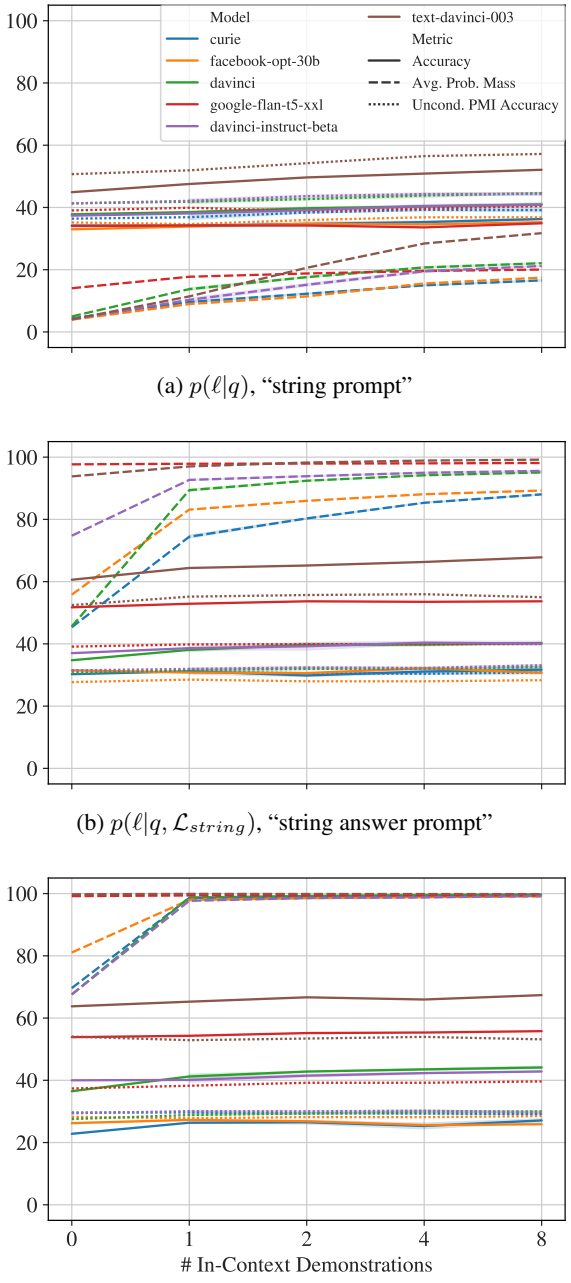
We run experiments on 6 models and 3 datasets/benchmarks, described below.

6.1 Models

Vanilla Language Models These are models that are (to the best of publicly-available knowledge) only trained on the next-token prediction task. We experiment on two GPT-3 base models (Brown et al., 2020)—`curie` (~6.7B parameters) and `davinci` (~175B parameters)— and one model whose weights are publicly available, OPT 30B (Zhang et al., 2022).[7]

Language Models with Further Fine-Tuning We study two instruction-tuned (Weller et al., 2020; Mishra et al., 2022; Aribandi et al., 2022; Sanh et al., 2022; Wei et al., 2022) models: FLAN-T5 XXL (~11B parameters; Chung et al., 2022), and the "original" InstructGPT model, GPT-3 `davinci-instruct-beta` (~175B parameters; Ouyang et al., 2022). We additionally test one "state of the art" model, GPT-3 `text-davinci-003` (unknown # parameters; OpenAI, 2022). FLAN-T5-XXL is based on the T5 architecture (Raffel et al., 2020) and its weights are publicly available. It has demonstrated comparable performance to GPT-3 `davinci` despite being ~16x smaller. We include `davinci-instruct-beta` to study the effect of supervised instruction tuning on a model of identical scale to `davinci-base` that is also associated with a publicly-available research paper, unlike later OpenAI model re-

[7] Sizes and corresponding citations for GPT-3 models are approximate, using the most up-to-date information available (OpenAI, 2022).



(a) $p(\ell|q)$, "string prompt"

(b) $p(\ell|q, \mathcal{L}_{string})$, "string answer prompt"

(c) $p(\ell|q, \mathcal{L}_{enum})$; "enumerated answer prompt"

Figure 3: MMLU test subset accuracy (Eq. (1); solid lines), PMI accuracy (Eq. (3); dotted lines), and amount of probability mass on first tokens of valid answer choices (Eq. (4); dashed lines) as a function of **number** (x-axis) and **format** (a,b,c subgraphs) of in-context examples, for six pretrained language models. Tabular version of these results in Table 8 and results for CommonsenseQA/OpenbookQA in Figs. 8 and 9 (Appendix).

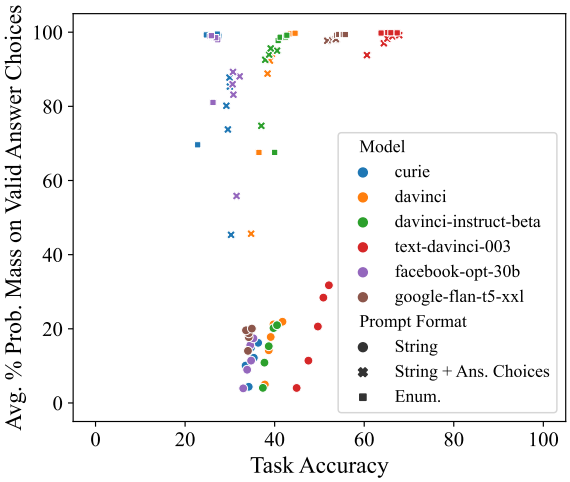


Figure 4: A scatterplot showing the relationship between task accuracy and average total probability mass on valid answer choices for the MMLU test subset (for 0, 1, 2, 4 and 8 in-context examples). See Figures 10-11 in Appendix A.4 for CommonsenseQA and OpenbookQA.

leases.[8] GPT-3 `text-davinci-003` is (along with `text-davinci-002`) a state-of-the-art model at the time of writing, according to the HELM benchmark (Liang et al., 2022).

More implementation details are given in Appendix A.1.

6.2 Tasks

We test on three challenging multiple-choice tasks that are open-vocabulary (i.e., each instance has a unique set of answer choices). We select out-of-domain tasks for the models studied (though we can't be sure for GPT-3 models; more details in Appendix A.2). Examples of the tasks are given in Appendix A.3.

OpenbookQA (Mihaylov et al., 2018) is a 4-way multiple-choice science question-answering task. The questions are elementary-level. It was explicitly included in the training data of FLAN-T5. Random accuracy is 25%. The test set has 500 instances.

CommonsenseQA v1.11 (Talmor et al., 2019) is a 5-way multiple-choice commonsense reasoning task. It was explicitly included in the training data of FLAN-T5. Random accuracy is 20%. The test set is not publicly available; we use the first 500 instances of the validation set.

[8] It has been hypothesized that `davinci-instruct-beta` has been tuned directly from the `davinci` checkpoint (Fu, 2022), though this is unconfirmed.

MMLU (Hendrycks et al., 2021), or the "Massive Multitask Language Understanding" benchmark, contains dev, validation, and test questions for 57 different topical areas. The questions are 4-way multiple-choice spanning subjects in social sciences, STEM, and humanities that were manually scraped from practice materials available online for exams such as the GRE and the U.S. Medical Licensing Exam. This is considered a challenge benchmark as many state-of-the-art models have near-random accuracy (random is 25%). We evaluate on the first 20 test questions from each category (1140 instances total).

6.3 Prompts

In-Context Examples We experiment with $k = 0, 1, 2, 4$ and 8 in-context demonstrations, which are the same for each instance, and selected as the first k examples from a fixed set of 8. For `curie`, `davinci`, and `davinci-instruct-beta` models, we report the mean and standard error over 3 random seeds used to select the set of 8 demonstrations, since the choice of in-context demonstrations can significantly affect performance (Liu et al., 2022; Lu et al., 2022; Liu et al., 2023, *i.a.*). We select in-context examples from each dataset's associated training set (combined dev and validation sets for MMLU due to the lack of training set).

Prompt Format We experiment with three prompt formats, which correspond to the format of x in §5.2.

The first, "string prompt", only contains q and is thus most similar to next-word prediction. The question is given as a string, such as:

```
kinetics change stored energy into motion and
```

The model is expected to generate the subsequent token(s) as the string answer, here, `warmth`. This is identical to the evaluation protocol used in Brown et al. (2020). We notate the resulting probability assigned to a label using this prompt as $p(\ell|q)$.

The second, "string answer prompt", presents the question as well as answer choices as a string list, such as:

```
question: kinetics change stored energy into
motion and
answer choices: snacks, naps, kites, or warmth
The correct answer is:
```

The model is expected to generate the complete string output, here, `warmth`. This prompt is similar to formats included in PromptSource (Bach et al.,

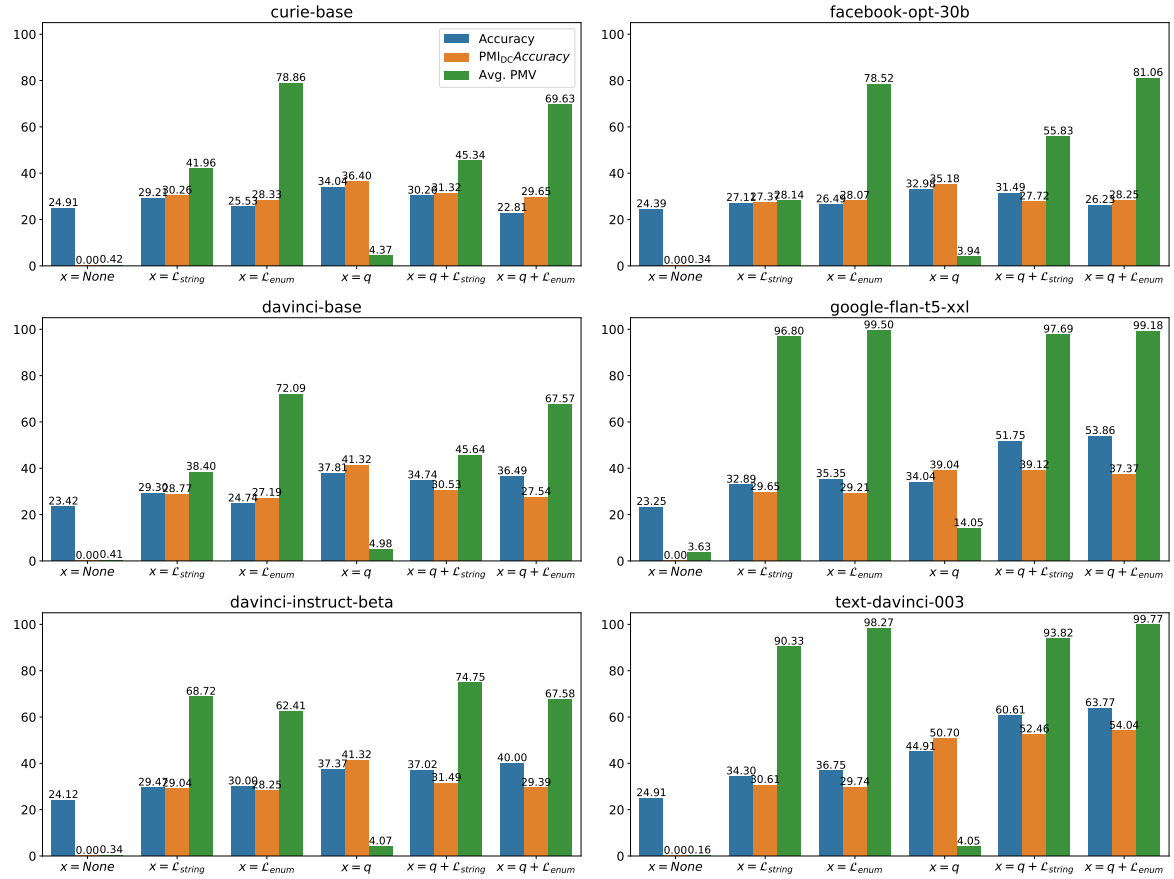


Figure 5: Zero-shot results on a subset of the MMLU test set for various LLMs: sequence-scoring accuracy (Eq. (1); blue), PMI accuracy (Eq. (3); orange), and average total probability mass on first tokens of valid answer choices (Eq. (4); green). Random accuracy is 25%. The x-axis indicates the input x given to the model (§5.2): q = the test question, $\mathcal{L}_{enum/string}$ = the answer choices in enumerated or string format (§6.3), and the addition symbol represents string concatenation. $c = None$ corresponds to the uncontextual probability of the answer, or the denominator in Eq. (3) (PMI accuracy is undefined in this case). Results for CommonsenseQA (Fig. 12) and OpenbookQA (Fig. 13) in Appendix A.4. Observing answer choices in the prompt contributes far more to attentiveness than observing the question, confirming our hypothesis in §5.2. Even without observing the question, all models place a substantial amount of probability mass on valid answer choices after observing them in the prompt.

2022) used to train FLAN-T5 and other models. We notate the resulting probability assigned to a label using this prompt as $p(\ell|q, \mathcal{L}_{string})$.

The third format, "enumerated answer prompt", presents answer choices in a list with symbolic representations for each answer:

```
Question: kinetics change stored energy into
Choices:
 A: snacks
 B: naps
 C: kites
 D: warmth
Answer:
```

The model is expected only to generate the (single-token) symbol of the predicted answer, here, `D`. This format is similar to that used for zero-shot evaluations in FLAN (Wei et al., 2022) and FLAN-T5. We notate the resulting probability assigned to a label using this prompt as $p(\ell|q, \mathcal{L}_{enum})$. The full prompts (as well as more details) are given in Appendix A.3.

7 Results and Discussion

7.1 On Reducing Surface Form Competition

Fig. 1 (left), demonstrates the effect of choice of prompt format on PMV in the one-shot setting. Across datasets, showing answer choices in the "string" format leads to a substantial increae in PMV, which reaches near-100% for all models using the "enumerated" format. Zooming in on the role of in-context examples in Fig. 3 (dashed lines), we observe that all models place

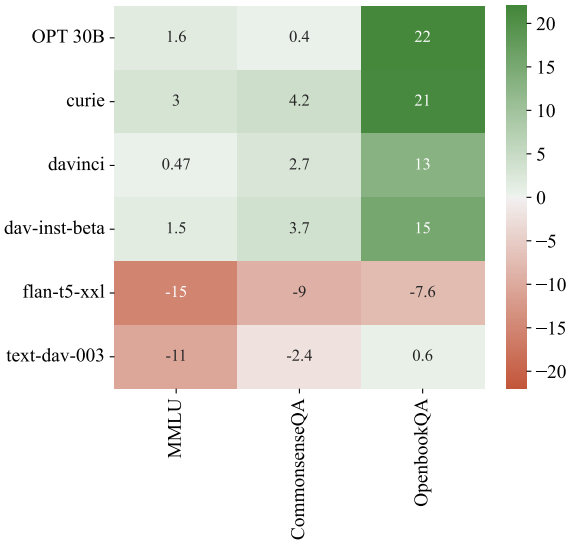
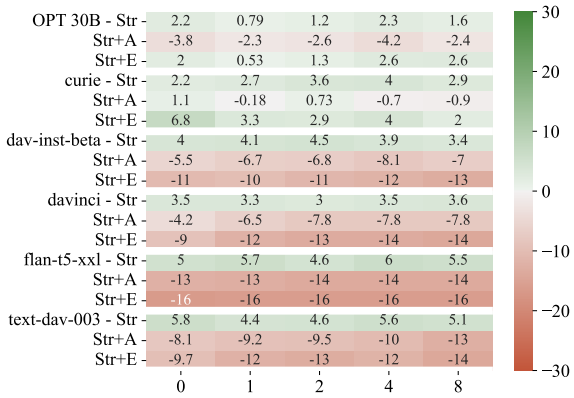


Figure 6: Differences in the best accuracy achieved by PMI scoring (Eq. (3)) and the best achieved by sequence scoring (Eq. (1)) across prompt settings for each model and dataset. Results for all prompt settings are in Fig. 7.

significantly more probability mass on valid answer choices after seeing only one in-context example *that includes the answer choices*, and stronger models such as `text-davinci-003` and FLAN-T5-XXL exhibit this behavior zero-shot. Trends also hold for CommonsenseQA and OpenbookQA (Figs. 8 and 9). For `curie`, `davinci`, and `davinci-instruct-beta`, we present standard error over 3 random seeds for selecting prompt examples and find the effects of random seed are generally negligible. The number of instances for which the bound is satisfied and SFC is fully alleviated (Eq. (6)) for each method can be found in Table 7 (Appendix A.4).

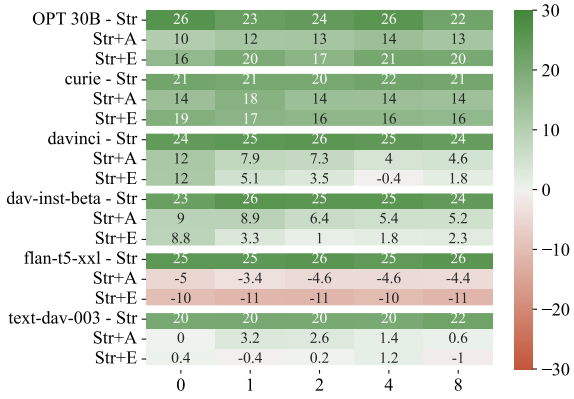
7.2 Relationship between Surface Form Competition and Accuracy

Fig. 1 (right), demonstrates the effect of choice of prompt format on accuracy in the one-shot setting. While gains in PMV are consistent across models, this is not the case for accuracy. Certain models (`curie`, OPT 30B) actually achieve their best task performance when their probability mass on valid answer choices is the *lowest*. We hypothesize that this is due to the string prompt being the closest to the next-token prediction objective. For others (`davinci`, `davinci-instruct-beta`), accuracy is relatively consistent across prompts, even while PMV substantially increases. Seeing the answer choices in the prompt is crucial also to



(a) MMLU

(b) CommonsenseQA



(c) OpenbookQA

Figure 7: Accuracy changes achieved by using PMI scoring (Eq. (3)) over standard sequence scoring (Eq. (1)) for all models and tasks investigated. Prompt formats labeled as: "Str"= string prompt (no answer choices), "Str + A"= string answer prompt, "Str+E" = enumerated answer prompt (detailed in §6.3). The full accuracy scores are in Tables 8 to 10.

achieving good accuracy with `text-davinci-003` and FLAN-T5-XXL, likely due to their instruction tuning. These results confirm that showing the answer choices does **not guarantee** improved ac-

curacy, especially in the case of vanilla language models.

We can also observe this lack of positive correlation from the angle of in-context examples (Figs. 3, 8 and 9). While PMV increases as a function of the number of in-context examples, accuracy is relatively stable across all models and prompt formats.

We plot each model result as a datapoint on a shared scatterplot in Fig. 4. This graph further illustrates the lack of correlation that exists between increases in the x-axis (accuracy) and increases on the y-axis (PMV). There are a number of examples in the bottom half of the plot where PMV is increasing without any shift in x-axis position.

7.2.1 Role of Different Parts of the Input

In Fig. 5, we follow the methodology proposed in §5.2 and break down the zero-shot contributions to attentiveness and accuracy of question q vs. answer choices $\mathcal{L}$ when included in the prompt.

We find that conditioning $P_\theta(\ell)$ on $\mathcal{L}$ (i.e., considering $P_\theta(\ell|\mathcal{L})$) substantially increases the probability mass the model places over $\mathcal{L}$ (65.77% vs. under 1% PMV on average for MMLU; the accuracy of both is similar, at 24.05% and 29.29%, resp.). On the other hand, conditioning either of these probabilities further on q (i.e., considering $P_\theta(\ell|q)$ or $P_\theta(\ell|q,\mathcal{L})$) provides a very small gain on PMV (4.20% (absolute) PMV gain as opposed to 10.32% accuracy gain on average for MMLU). This indicates that conditioning on q is not an effective way to increase PMV (or decrease SFC). Overall, observing q plays a larger role on accuracy while observing $\mathcal{L}$ plays a larger role on in increasing probability mass. Results hold for CommonsenseQA and OpenbookQA (Figs. 12 and 13).

7.3 When does PMI_{DC} improve accuracy?

Our experiments provide more insight into when PMI_{DC} or related normalization methods may be successful. In Fig. 6 and Fig. 7, we illustrate the amount by which PMI_{DC} increases or decreases accuracy for each dataset, also illustrated in Fig. 3 and Table 8.

Whether PMI_{DC} improves accuracy seems generally tied to the largest amount of probability mass on valid answers that can be achieved by any prompt as well as overall model performance (lower probability mass and lower accuracy, higher gains from PMI_{DC}). Indeed, PMI_{DC} *always* improves accuracy when answer choices are not observed in the prompt (Fig. 7), and the size of the gains are fairly consistent for each dataset across # of in-context examples and models. However, as established earlier, not observing answer choices is often the prompt format with the worst accuracy for strong models. Fig. 6 plots the difference between the best achievable accuracies using each method; with the exception of OpenbookQA, gains are relatively muted. Additionally, PMI_{DC} consistently (though not always) leads to significant drops in accuracy for the strongest models (`text-davinci-003` and FLAN-T5-XXL).

8 Conclusion

We take a novel approach to studying the effects of prompt formatting, # of in-context examples, and model choice on attentiveness and its relationship with end task performance, by proposing a new formalization of model attentiveness and a quantifiable metric (PMV). This is an important step towards understanding and improving the use of LMs for discriminative tasks. Our findings inform discussions about the role of attentiveness in model performance. They also challenge intuitive assumptions such as showing answer choices for multiple-choice tasks is always beneficial, which is a common practice (Hendrycks et al., 2021; Rae et al., 2021; Hoffmann et al., 2022, *i.a.*). We show that this strategy is effective only for certain LMs.

Practical Insights: We find that the best way to use vanilla LMs in multiple-choice settings is to provide a string prompt *without* answer choices and apply probability normalization. For instruction-tuned models, on the other hand, answer choices *should* be shown and in an enumerated prompt format, and probability normalization should *not* be used. More generally, our results reveal that efforts to increase model attentiveness via prompting methods can have surprisingly negative effects, and that scoring methods can drastically affect the conclusions we reach about an underlying language model's fundamental capabilities. We advocate future work to look into length normalization as another understudied scoring mechanism.

Acknowledgements

We thank members of the Aristo team at AI2, Peter West, Hanna Hajishirzi, and the H2 lab at the University of Washington (Akari Asai, Yanai Elazar, Jiacheng Liu, Sewon Min, Yizhong Wang) for insightful feedback.

References

Vamsi Aribandi, Yi Tay, Tal Schuster, Jinfeng Rao, Huaixiu Steven Zheng, Sanket Vaibhav Mehta, Honglei Zhuang, Vinh Q. Tran, Dara Bahri, Jianmo Ni, Jai Gupta, Kai Hui, Sebastian Ruder, and Donald Metzler. 2022. Ext5: Towards extreme multi-task scaling for transfer learning. In *International Conference on Learning Representations*.

Stephen Bach, Victor Sanh, Zheng Xin Yong, Albert Webson, Colin Raffel, Nihal V. Nayak, Abheesht Sharma, Taewoon Kim, M Saiful Bari, Thibault Fevry, Zaid Alyafeai, Manan Dey, Andrea Santilli, Zhiqing Sun, Srulik Ben-david, Canwen Xu, Gunjan Chhablani, Han Wang, Jason Fries, Maged Alshaibani, Shanya Sharma, Urmish Thakker, Khalid Almubarak, Xiangru Tang, Dragomir Radev, Mike Tian-jian Jiang, and Alexander Rush. 2022. PromptSource: An integrated development environment and repository for natural language prompts. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 93–104, Dublin, Ireland. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901, Online. Curran Associates, Inc.

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

Andrew M Dai and Quoc V Le. 2015. Semi-supervised sequence learning. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Yao Fu. 2022. How does gpt obtain its ability? tracing emergent abilities of language models to their sources. Blogpost.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. 2022. Training compute-optimal large language models. ArXiv:2203.15556.

Ari Holtzman, Peter West, Vered Shwartz, Yejin Choi, and Luke Zettlemoyer. 2021. Surface form competition: Why the highest probability answer isn't always right. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7038–7051, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. UNIFIEDQA: Crossing format boundaries with a single QA system. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1896–1907, Online. Association for Computational Linguistics.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. 2021. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online

and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. 2022. Holistic evaluation of language models. ArXiv:2211.09110.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9).

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics.

Nikolay Malkin, Zhen Wang, and Nebojsa Jojic. 2022. Coherence boosting: When your pretrained language model is not paying enough attention. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8214–8236, Dublin, Ireland. Association for Computational Linguistics.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

Sewon Min, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Noisy channel language model prompting for few-shot text classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5316–5330, Dublin, Ireland. Association for Computational Linguistics.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-task generalization via natural language crowdsourcing instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3470–3487, Dublin, Ireland. Association for Computational Linguistics.

OpenAI. 2022. Model index for researchers. Blogpost.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. ArXiv:2203.02155.

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

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. 2021. Scaling language models: Methods, analysis & insights from training gopher. ArXiv:2112.11446.

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

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal 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 Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. 2022. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. 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 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Trieu H Trinh and Quoc V Le. 2018. A simple method for commonsense reasoning. ArXiv:1806.02847.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Orion Weller, Nicholas Lourie, Matt Gardner, and Matthew E. Peters. 2020. Learning from task descriptions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*

(EMNLP), pages 1361–1375, Online. 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:2205.01068.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706. PMLR.

A Appendix

A.1 Implementation Details

We use Huggingface Datasets (Lhoest et al., 2021) and Huggingface Transformers (Wolf et al., 2020) for implementation. All GPT-3 models were queried via the OpenAI API (`https://beta.openai.com`) between January and May 2023.

A.2 Nature of Datasets for Each Model

For models trained only on the autoregressive next-token prediction objective (`curie`, `davinci`, and OPT 30B (Zhang et al., 2022)), in theory the OpenbookQA and CommonsenseQA datasets have not been seen in during training. However, guarantees would require access and indexing of the training corpora, which are not publicly available for the GPT-3 models. Additionally, due to the fact that training data was scraped for these models up to and including 2019 (Brown et al., 2020), it is possible there is some leakage in the training corpus.

For the instruction-tuned models, the authors of FLAN-T5 (Chung et al., 2022) explicitly report the datasets which are used and not used during training, and we report these details in §6.2. As for InstructGPT `instruct-davinci-beta` (Ouyang et al., 2022), the following details are given about its supervised instruction tuning training dataset (emphasis ours):

> "...The SFT dataset contains **about 13k training prompts (from the API and labeler-written)**...To give a sense of the composition of our dataset, in Table 1 we show the distribution of use-case categories for our API prompts (specifically the RM [reward modeling] dataset) as labeled by our contractors. Most of the use-cases have (sp) are **generative, rather than classification or QA**. These prompts are **very diverse and include generation, question answering, dialog, summarization, extractions, and other natural language tasks** (see Table 1)."

In Table 1, generation makes up 45.6% of the dataset, followed by open QA at 12.4%. Closed QA is a relatively small percentage of the training set, at 2.6%, and classification 3.5%, providing some possibility that the tasks we study are out-of-domain/zero-shot (though these exact numbers are reported on the reward modeling dataset, not the one used for instruction tuning, and these are not guarantees due to the proprietary nature of the dataset). No details are given about the datasets used to train `text-davinci-003` (OpenAI, 2022).

A.3 Prompt Details

The "string", "string answer" and "enumerated answer" prompts containing 4 in-context demonstrations (for 1 of the 3 random seeds used) are given in Tables 1 to 3 for OpenbookQA and Tables 4 to 6 for CommonsenseQA. The last instance shown is the test instance, which the model completes with an answer prediction. For each random seed, 8 demonstrations are drawn from the training set of each dataset. When fewer demonstrations (0-4) are used, the first k are taken and the prompt otherwise stays the same.

A.4 Additional Results

- Table 7 contains the tables of bound satisfaction (Eq. (6)) for all datasets.
- Table 8 contains tabular results for MMLU; Table 9 for CommonsenseQA and Table 10 for OpenbookQA.
- Figs. 8 and 9 contain line graphs for CommonsenseQA and OpenbookQA, respectively.

Bears will always have longer life cycles than a fox
If a river is rushing southwest on a sunny day, then it is safe to assume that the land gently inclines in that direction
After the moon phase where you can see nothing of the moon, what comes next? the first quarter
kinetics change stored energy into motion and warmth
A person wants to start saving money so that they can afford a nice vacation at the end of the year. After looking over their budget and expenses, they decide the best way to save money is to

Table 1: One of three "string" prompt templates used for OpenbookQA, containing 4 in-context demonstrations and one test instance.

Let's answer science questions.

question: Bears will always have longer life cycles than a
answer choices: tortoises, whales, elephants, or fox
The correct answer is: fox
###
question: If a river is rushing southwest on a sunny day, then it is safe to assume that
answer choices: southwest is a good place to be, the land gently inclines in that direction, the world is mostly land, or the land is supple
The correct answer is: the land gently inclines in that direction
###
question: After the moon phase where you can see nothing of the moon, what comes next?
answer choices: the full moon, the last quarter, the first quarter, or the half moon
The correct answer is: the first quarter
###
question: kinetics change stored energy into motion and
answer choices: snacks, naps, kites, or warmth
The correct answer is: warmth
###
question: A person wants to start saving money so that they can afford a nice vacation at the end of the year. After looking over their budget and expenses, they decide the best way to save money is to
answer choices: make more phone calls, quit eating lunch out, buy less with monopoly money, or have lunch with friends
The correct answer is:

Table 2: One of three "string answer" prompt templates used for OpenbookQA, containing 4 in-context demonstrations and one test instance.

- Figs. 10 and 11 contain scatterplots for CommonsenseQA and OpenbookQA, respectively.
- Figs. 12 and 13 show CommonsenseQA and OpenbookQA barcharts for accuracy and probability mass on valid answer choices conditioned on various combinations of independent variables in the prompt.

The following are elementary-level multiple-choice questions about science. For the question below, select the most suitable answer from the 4 options given.

Question: Bears will always have longer life cycles than a
Choices:
A: tortoises
B: whales
C: elephants
D: fox
Answer: D

Question: If a river is rushing southwest on a sunny day, then it is safe to assume that
Choices:
A: southwest is a good place to be
B: the land gently inclines in that direction
C: the world is mostly land
D: the land is supple
Answer: B

Question: After the moon phase where you can see nothing of the moon, what comes next?
Choices:
A: the full moon
B: the last quarter
C: the first quarter
D: the half moon
Answer: C

Question: kinetics change stored energy into motion and
Choices:
A: snacks
B: naps
C: kites
D: warmth
Answer: D

Question: A person wants to start saving money so that they can afford a nice vacation at the end of the year. After looking over their budget and expenses, they decide the best way to save money is to
Choices:
A: make more phone calls
B: quit eating lunch out
C: buy less with monopoly money
D: have lunch with friends
Answer:

Table 3: One of three "enumerated answer" prompt templates used for OpenbookQA, containing 4 in-context demonstrations and one test instance.

Fabric is cut to order at what type of seller? tailor shop
Where are you if your reading magazines while waiting for a vehicle on rails? train station
What would need oil to be used? combustion engines
What is person probably feeling that plans on stopping being married to their spouse? detachment
A revolving door is convenient for two direction travel, but it also serves as a security measure at a what?

Table 4: One of three "string" prompt templates used for CommonsenseQA, containing 4 in-context demonstrations and one test instance.

Let's answer commonsense reasoning questions.

question: Fabric is cut to order at what type of seller?
answer choices: hardware store, curtains, tailor shop, clothing store, or sewing room
The correct answer is: tailor shop
###
question: Where are you if your reading magazines while waiting for a vehicle on rails?
answer choices: bookstore, vegetables, market, doctor, or train station
The correct answer is: train station
###
question: What would need oil to be used?
answer choices: service station, ground, human body, repair shop, or combustion engines
The correct answer is: combustion engines
###
question: What is person probably feeling that plans on stopping being married to their spouse?
answer choices: wrong, detachment, bankruptcy, sad, or fights
The correct answer is: detachment
###
question: A revolving door is convenient for two direction travel, but it also serves as a security measure at a what?
answer choices: new york, bank, library, department store, or mall
The correct answer is:

Table 5: One of three "string answer" prompt templates used for CommonsenseQA, containing 4 in-context demonstrations and one test instance.

The following are multiple-choice questions about everyday situations. For the question below, select the most suitable answer from the 5 options given.

Question: Fabric is cut to order at what type of seller?
Choices:
A: curtains
B: tailor shop
C: clothing store
D: sewing room
E: hardware store
Answer: B

Question: Where are you if your reading magazines while waiting for a vehicle on rails?
Choices:
A: vegetables
B: market
C: doctor
D: train station
E: bookstore
Answer: D

Question: What would need oil to be used?
Choices:
A: ground
B: human body
C: repair shop
D: combustion engines
E: service station
Answer: D

Question: What is person probably feeling that plans on stopping being married to their spouse?
Choices:
A: detachment
B: bankruptcy
C: sad
D: fights
E: wrong
Answer: A

Question: A revolving door is convenient for two direction travel, but it also serves as a security measure at a what?
Choices:
A: bank
B: library
C: department store
D: mall
E: new york
Answer:

Table 6: One of three "enumerated answer" prompt templates used for CommonsenseQA, containing 4 in-context demonstrations and one test instance.

Model Name	Prompt Format	Dataset	# In-Context Demonstrations 0	1	2	4	8
OPT 30B	$p(\ell\|q)$	MMLU	0.18	0.88	0.61	1.05	2.19
		CommonsenseQA	0.0	0.0	1.2	2.6	3.6
		OpenbookQA	0.6	0.4	0.6	0.6	0.4
	$p(\ell\|q, \mathcal{L}_{string})$	MMLU	6.23	37.54	40.53	45.26	49.21
		CommonsenseQA	2.4	29.4	38.2	47.2	52.2
		OpenbookQA	4.8	25.8	42.4	45.2	48.6
	$p(\ell\|q, \mathcal{L}_{enum})$	MMLU	3.68	78.25	78.25	82.81	88.6
		CommonsenseQA	0.0	54.6	63.4	74.6	78.2
		OpenbookQA	0.2	65.2	78.2	78.0	85.4
GPT-3 curie (~6.7B)	$p(\ell\|q)$	MMLU	0.35	$1.37_{0.56}$	$1.2_{0.94}$	$0.94_{0.56}$	$0.91_{0.13}$
		CommonsenseQA	0.0	$0.07_{0.12}$	$0.6_{0.53}$	$2.13_{0.5}$	$3.8_{0.4}$
		OpenbookQA	0.4	$0.4_{0.0}$	$0.4_{0.0}$	$0.4_{0.0}$	$0.47_{0.12}$
	$p(\ell\|q, \mathcal{L}_{string})$	MMLU	3.86	$25.11_{1.71}$	$30.12_{0.9}$	$38.01_{1.58}$	$44.53_{0.68}$
		CommonsenseQA	1.8	$28.27_{1.21}$	$32.0_{0.6}$	$37.87_{2.81}$	$40.8_{4.13}$
		OpenbookQA	1.2	$21.6_{10.89}$	$35.07_{12.7}$	$36.6_{11.48}$	$32.8_{6.92}$
	$p(\ell\|q, \mathcal{L}_{enum})$	MMLU	0.0	$78.83_{2.26}$	$86.05_{0.79}$	$91.08_{0.82}$	$93.62_{0.99}$
		CommonsenseQA	0.0	$71.87_{11.7}$	$82.87_{11.96}$	$98.27_{0.12}$	$98.0_{0.53}$
		OpenbookQA	0.0	$76.87_{10.85}$	$84.53_{11.27}$	$91.27_{4.96}$	$91.67_{1.17}$
GPT-3 davinci (~175B)	$p(\ell\|q)$	MMLU	0.53	$2.75_{0.28}$	$2.49_{0.98}$	$3.33_{0.46}$	$4.62_{0.4}$
		CommonsenseQA	0.0	$1.8_{1.71}$	$4.53_{0.31}$	$6.13_{0.58}$	$7.33_{1.3}$
		OpenbookQA	0.2	$0.73_{0.31}$	$0.4_{0.2}$	$0.4_{0.35}$	$0.6_{0.2}$
	$p(\ell\|q, \mathcal{L}_{string})$	MMLU	2.89	$51.87_{1.4}$	$59.07_{1.7}$	$64.06_{1.72}$	$66.81_{1.25}$
		CommonsenseQA	6.8	$61.8_{6.12}$	$62.33_{14.87}$	$67.53_{12.08}$	$74.4_{6.97}$
		OpenbookQA	4.0	$48.47_{13.27}$	$67.73_{12.91}$	$61.73_{11.71}$	$64.07_{6.74}$
	$p(\ell\|q, \mathcal{L}_{enum})$	MMLU	1.14	$88.19_{1.63}$	$94.21_{0.09}$	$96.7_{0.73}$	$98.36_{0.42}$
		CommonsenseQA	0.0	$87.67_{3.14}$	$96.2_{0.2}$	$97.8_{0.2}$	$98.4_{0.4}$
		OpenbookQA	0.0	$92.8_{1.59}$	$96.73_{0.7}$	$97.8_{0.53}$	$99.0_{0.53}$
davinci-instruct-beta	$p(\ell\|q)$	MMLU	0.88	$1.34_{0.53}$	$2.1_{0.85}$	$2.69_{0.05}$	$4.24_{0.37}$
		CommonsenseQA	0.0	$3.07_{1.36}$	$5.4_{0.6}$	$6.73_{0.42}$	$7.0_{0.87}$
		OpenbookQA	0.4	$0.87_{0.31}$	$0.67_{0.23}$	$0.6_{0.2}$	$0.73_{0.12}$
	$p(\ell\|q, \mathcal{L}_{string})$	MMLU	32.72	$65.56_{1.77}$	$68.89_{0.98}$	$70.85_{0.57}$	$72.75_{0.05}$
		CommonsenseQA	46.8	$69.07_{7.16}$	$76.0_{4.57}$	$79.6_{3.34}$	$80.8_{4.2}$
		OpenbookQA	36.4	$60.93_{16.15}$	$70.13_{10.72}$	$68.87_{9.2}$	$72.0_{5.96}$
	$p(\ell\|q, \mathcal{L}_{enum})$	MMLU	20.79	$93.71_{0.36}$	$95.94_{0.34}$	$96.35_{0.68}$	$97.92_{0.86}$
		CommonsenseQA	15.6	$94.8_{0.35}$	$97.2_{0.35}$	$98.0_{0.4}$	$98.13_{0.42}$
		OpenbookQA	33.4	$95.8_{2.03}$	$98.2_{0.35}$	$99.0_{0.53}$	$99.0_{0.4}$
FLAN-T5-XXL (11B)	$p(\ell\|q)$	MMLU	0.79	1.05	1.75	1.58	2.28
		CommonsenseQA	3.4	5.0	6.0	6.0	7.8
		OpenbookQA	0.6	0.8	0.6	0.6	0.6
	$p(\ell\|q, \mathcal{L}_{string})$	MMLU	85.96	88.86	89.74	90.18	90.79
		CommonsenseQA	98.0	99.4	99.4	99.0	98.6
		OpenbookQA	95.6	95.2	96.4	97.4	96.8
	$p(\ell\|q, \mathcal{L}_{enum})$	MMLU	98.86	98.33	98.68	98.68	98.33
		CommonsenseQA	99.2	99.2	99.4	99.4	99.0
		OpenbookQA	99.8	100.0	100.0	99.6	100.0
text-davinci-003	$p(\ell\|q)$	MMLU	0.88	3.77	7.11	9.47	10.61
		CommonsenseQA	0.0	11.6	19.6	18.8	19.6
		OpenbookQA	1.6	1.6	1.8	2.6	3.0
	$p(\ell\|q, \mathcal{L}_{string})$	MMLU	91.75	94.56	95.96	96.67	96.93
		CommonsenseQA	80.4	92.8	91.8	93.2	93.4
		OpenbookQA	82.0	94.4	93.6	94.0	95.2
	$p(\ell\|q, \mathcal{L}_{enum})$	MMLU	99.91	100.0	99.91	100.0	99.82
		CommonsenseQA	99.8	100.0	100.0	99.8	100.0
		OpenbookQA	99.8	100.0	99.8	99.8	100.0

Table 7: % of instances for which Eq. (6) is true for all models, datasets, and prompt formats.

Model Name	Prompt Format	Metric	# In-Context Demonstrations 0	1	2	4	8
OPT 30B	$p(\ell \mid q)$	Accuracy	32.98	**33.86**	34.74	34.56	35.26
		PMI Acc.	**35.18**	**34.65**	**35.96**	**36.84**	**36.84**
		Prob. Mass	3.94	8.96	11.42	15.58	17.44
	$p(\ell \mid q, \mathcal{L}_{string})$	Accuracy	31.49	30.79	30.61	32.19	30.7
		PMI Acc.	27.72	28.51	27.98	27.98	28.33
		Prob. Mass	55.83	83.14	85.96	88.09	89.27
	$p(\ell \mid q, \mathcal{L}_{enum})$	Accuracy	26.23	27.28	26.84	25.61	25.88
		PMI Acc.	28.25	27.81	28.16	28.16	28.51
		Prob. Mass	81.06	97.97	98.52	98.87	99.06
GPT-3 `curie` (~6.7B)	$p(\ell \mid q)$	Accuracy	34.21	$34.15_{0.58}$	$34.71_{1.04}$	$35.29_{0.53}$	$36.29_{0.31}$
		PMI Acc.	**36.4**	$\mathbf{36.87_{1.23}}$	$\mathbf{38.36_{0.27}}$	$\mathbf{39.24_{0.45}}$	$\mathbf{39.15_{0.84}}$
		Prob. Mass	4.37	$9.62_{0.48}$	$12.27_{0.26}$	$14.97_{0.01}$	$16.57_{0.31}$
	$p(\ell \mid q, \mathcal{L}_{string})$	Accuracy	30.26	$31.2_{1.44}$	$29.85_{0.58}$	$31.11_{0.96}$	$31.72_{1.59}$
		PMI Acc.	31.32	$31.02_{0.48}$	$30.58_{0.31}$	$30.41_{0.35}$	$30.82_{0.7}$
		Prob. Mass	45.34	$74.43_{0.73}$	$80.31_{0.19}$	$85.34_{0.09}$	$88.04_{0.25}$
	$p(\ell \mid q, \mathcal{L}_{enum})$	Accuracy	22.81	$26.37_{0.13}$	$26.52_{0.89}$	$25.32_{1.49}$	$27.1_{0.83}$
		PMI Acc.	29.65	$29.65_{0.23}$	$29.39_{0.35}$	$29.33_{0.18}$	$29.15_{0.28}$
		Prob. Mass	69.63	$98.56_{0.02}$	$98.97_{0.01}$	$99.26_{0.02}$	$99.49_{0.01}$
`davinci-instruct-beta`	$p(\ell \mid q)$	Accuracy	37.37	$38.07_{0.4}$	$39.15_{0.54}$	$40.44_{0.81}$	$40.91_{0.51}$
		PMI Acc.	**41.32**	$\mathbf{42.14_{0.89}}$	$\mathbf{43.63_{0.53}}$	$\mathbf{44.3_{0.61}}$	$\mathbf{44.33_{0.75}}$
		Prob. Mass	4.07	$10.32_{0.85}$	$15.15_{0.65}$	$19.49_{0.63}$	$21.2_{0.2}$
	$p(\ell \mid q, \mathcal{L}_{string})$	Accuracy	37.02	$38.62_{0.85}$	$39.24_{2.06}$	$40.38_{0.67}$	$40.11_{1.17}$
		PMI Acc.	31.49	$31.96_{0.48}$	$32.43_{0.48}$	$32.25_{0.48}$	$33.13_{0.31}$
		Prob. Mass	74.75	$92.69_{0.15}$	$93.87_{0.18}$	$94.98_{0.2}$	$95.59_{0.15}$
	$p(\ell \mid q, \mathcal{L}_{enum})$	Accuracy	40.0	$40.12_{1.25}$	$41.49_{0.7}$	$42.31_{0.25}$	$42.83_{0.1}$
		PMI Acc.	29.39	$30.09_{0.35}$	$30.03_{0.14}$	$30.23_{0.35}$	$29.68_{0.18}$
		Prob. Mass	67.58	$97.67_{0.1}$	$98.56_{0.11}$	$98.76_{0.05}$	$99.22_{0.05}$
GPT-3 `davinci` (~175B)	$p(\ell \mid q)$	Accuracy	37.81	$38.51_{0.23}$	$39.73_{0.57}$	$40.32_{0.5}$	$40.96_{0.83}$
		PMI Acc.	**41.32**	$\mathbf{41.84_{0.85}}$	$\mathbf{42.69_{0.33}}$	$\mathbf{43.83_{0.89}}$	$\mathbf{44.59_{0.31}}$
		Prob. Mass	4.98	$13.76_{0.67}$	$17.61_{0.32}$	$20.72_{0.41}$	$22.09_{0.16}$
	$p(\ell \mid q, \mathcal{L}_{string})$	Accuracy	34.74	$38.07_{0.84}$	$39.74_{0.75}$	$39.77_{0.73}$	$40.26_{0.27}$
		PMI Acc.	30.53	$31.58_{0.61}$	$31.99_{0.45}$	$32.02_{0.3}$	$32.51_{0.35}$
		Prob. Mass	45.64	$89.39_{0.51}$	$92.42_{0.11}$	$94.18_{0.1}$	$95.09_{0.07}$
	$p(\ell \mid q, \mathcal{L}_{enum})$	Accuracy	36.49	$\mathbf{41.23_{1.29}}$	$\mathbf{42.81_{0.09}}$	$\mathbf{43.51_{0.35}}$	$\mathbf{44.12_{0.69}}$
		PMI Acc.	27.54	$28.83_{0.28}$	$29.36_{0.18}$	$29.94_{0.49}$	$30.0_{0.31}$
		Prob. Mass	67.57	$98.73_{0.05}$	$99.23_{0.02}$	$99.54_{0.03}$	$99.73_{0.01}$
FLAN-T5-XXL (11B)	$p(\ell \mid q)$	Accuracy	34.04	34.21	34.21	33.6	34.91
		PMI Acc.	39.04	39.91	38.86	39.65	40.44
		Prob. Mass	14.05	17.74	18.79	19.59	20.06
	$p(\ell \mid q, \mathcal{L}_{string})$	Accuracy	51.75	52.89	53.68	53.51	53.68
		PMI Acc.	39.12	39.82	39.91	39.91	40.0
		Prob. Mass	97.69	97.87	97.94	98.0	98.13
	$p(\ell \mid q, \mathcal{L}_{enum})$	Accuracy	**53.86**	**54.3**	**55.18**	**55.35**	**55.79**
		PMI Acc.	37.37	38.25	39.21	39.21	39.65
		Prob. Mass	99.18	99.39	99.42	99.38	99.36
`text-davinci-003`	$p(\ell \mid q)$	Accuracy	44.91	47.54	49.65	50.88	52.11
		PMI Acc.	50.7	51.93	54.21	56.49	57.19
		Prob. Mass	4.05	11.42	20.62	28.41	31.74
	$p(\ell \mid q, \mathcal{L}_{string})$	Accuracy	60.61	**64.39**	65.18	**66.32**	**67.81**
		PMI Acc.	52.46	55.18	55.7	55.96	55.0
		Prob. Mass	93.82	97.02	98.27	98.9	99.19
	$p(\ell \mid q, \mathcal{L}_{enum})$	Accuracy	**63.77**	**65.26**	**66.67**	**65.96**	**67.37**
		PMI Acc.	54.04	52.89	53.42	53.95	53.16
		Prob. Mass	99.77	99.9	99.86	99.85	99.82

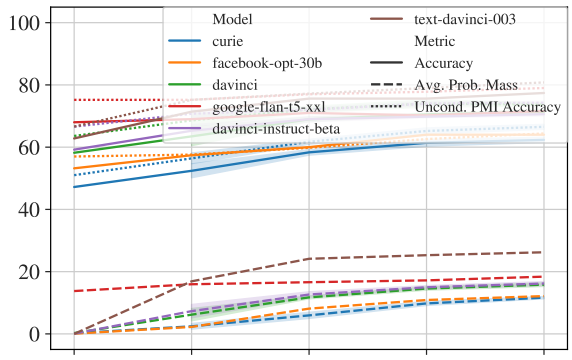
Table 8: A comparison of metrics (plotted in Fig. 3-Fig. 5) for each model and prompt type on the MMLU test subset. "Prob. Mass" is Eq. (4) averaged across instances. Models are ordered by increasing performance. The mean and standard error of using 3 random seeds to select in-context demonstrations are reported for experiments with at least 1 demonstration for the `curie`, `davinci`, and `davinci-instruct-beta` models. For each model and each column, we **bold** the prompt format and scoring metric (accuracy or PMI accuracy) that results in the highest score, as well as any scores within 1 percentage point of it. We underline the prompt format with the largest average total probability mass on first tokens of valid answer choices.

Model Name	Prompt Format	Metric	# In-Context Demonstrations 0	1	2	4	8
OPT 30B	$p(\ell\|q)$	Accuracy	53.2	**57.4**	**60.0**	**64.0**	**64.0**
		PMI Acc.	**57.0**	**57.6**	**59.8**	62.6	**64.4**
		Prob. Mass	0.08	2.24	8.12	10.89	12.15
	$p(\ell\|q, \mathcal{L}_{string})$	Accuracy	30.8	35.4	34.2	35.8	35.4
		PMI Acc.	21.4	20.2	20.8	21.6	22.0
		Prob. Mass	48.94	77.21	81.04	84.24	86.79
	$p(\ell\|q, \mathcal{L}_{enum})$	Accuracy	18.0	18.4	21.8	21.8	20.8
		PMI Acc.	17.8	17.2	18.0	17.2	17.6
		Prob. Mass	75.13	97.68	98.11	99.19	99.32
GPT-3 `curie` (~6.7B)	$p(\ell\|q)$	Accuracy	47.2(40.0)	$52.4_{3.82}$	$58.33_{1.5}$	$61.4_{1.83}$(52.3)	$62.33_{1.36}$
		PMI Acc.	**51.0**(50.3)	$\mathbf{56.4_{2.91}}$	$\mathbf{61.73_{1.3}}$	$\mathbf{65.2_{1.64}}$(56.5)	$\mathbf{66.53_{1.3}}$
		Prob. Mass	0.21	$2.44_{1.53}$	$5.95_{1.66}$	$9.79_{1.08}$	$11.64_{0.56}$
	$p(\ell\|q, \mathcal{L}_{string})$	Accuracy	30.4	$37.6_{3.8}$	$39.4_{2.03}$	$40.2_{4.39}$	$40.07_{4.63}$
		PMI Acc.	21.8	$23.2_{0.53}$	$22.87_{0.61}$	$22.93_{0.61}$	$23.13_{1.22}$
		Prob. Mass	41.0	$73.48_{1.14}$	$76.28_{1.35}$	$79.11_{0.56}$	$81.1_{1.32}$
	$p(\ell\|q, \mathcal{L}_{enum})$	Accuracy	19.2	$19.6_{1.91}$	$21.0_{0.4}$	$21.27_{0.12}$	$21.2_{0.0}$
		PMI Acc.	17.6	$18.2_{0.8}$	$18.6_{0.35}$	$19.0_{0.72}$	$18.73_{0.9}$
		Prob. Mass	69.09	$99.11_{0.14}$	$99.36_{0.05}$	$99.61_{0.06}$	$99.7_{0.02}$
GPT-3 `davinci` (~175B)	$p(\ell\|q)$	Accuracy	58.2(61.0)	$63.47_{5.13}$	$68.93_{1.29}$	$70.53_{0.5}$(69.1)	$71.33_{1.27}$
		PMI Acc.	**63.6**(66.7)	$\mathbf{68.6_{3.61}}$	$\mathbf{72.13_{1.62}}$	$\mathbf{74.0_{1.91}}$(72.0)	$\mathbf{73.73_{0.81}}$
		Prob. Mass	0.26	$6.18_{3.37}$	$11.73_{0.76}$	$14.57_{0.86}$	$15.82_{0.94}$
	$p(\ell\|q, \mathcal{L}_{string})$	Accuracy	38.0	$51.27_{0.5}$	$55.2_{3.83}$	$56.33_{3.92}$	$57.27_{1.5}$
		PMI Acc.	23.8	$29.2_{0.53}$	$31.73_{1.79}$	$33.73_{2.89}$	$33.67_{1.22}$
		Prob. Mass	55.58	$86.6_{1.72}$	$84.27_{7.41}$	$86.93_{5.35}$	$88.99_{2.28}$
	$p(\ell\|q, \mathcal{L}_{enum})$	Accuracy	34.4	$48.27_{1.72}$	$50.8_{0.87}$	$49.6_{2.31}$	$53.67_{3.61}$
		PMI Acc.	17.8	$21.73_{0.61}$	$23.13_{1.01}$	$25.07_{1.14}$	$27.47_{0.46}$
		Prob. Mass	63.61	$98.84_{0.3}$	$99.53_{0.08}$	$99.82_{0.01}$	$99.86_{0.03}$
`davinci-instruct-beta`	$p(\ell\|q)$	Accuracy	59.2	$65.27_{3.37}$	$69.0_{1.39}$	$69.8_{0.53}$	$70.6_{0.92}$
		PMI Acc.	**66.6**	$\mathbf{70.73_{2.23}}$	$\mathbf{72.0_{1.06}}$	$\mathbf{74.0_{1.04}}$	$\mathbf{74.27_{0.7}}$
		Prob. Mass	0.07	$7.31_{3.66}$	$12.65_{1.48}$	$14.99_{0.88}$	$16.19_{1.01}$
	$p(\ell\|q, \mathcal{L}_{string})$	Accuracy	52.2	$60.6_{3.3}$	$63.73_{2.01}$	$61.47_{2.47}$	$62.13_{1.22}$
		PMI Acc.	32.4	$36.6_{1.4}$	$38.07_{1.29}$	$37.6_{1.78}$	$36.33_{0.81}$
		Prob. Mass	73.89	$85.48_{5.49}$	$88.56_{2.96}$	$90.2_{2.6}$	$91.01_{1.69}$
	$p(\ell\|q, \mathcal{L}_{enum})$	Accuracy	44.2	$51.33_{1.8}$	$53.2_{0.53}$	$53.47_{2.47}$	$54.93_{2.32}$
		PMI Acc.	24.0	$27.27_{0.81}$	$26.73_{0.61}$	$28.47_{1.6}$	$30.0_{1.06}$
		Prob. Mass	58.96	$98.12_{0.45}$	$99.36_{0.09}$	$99.62_{0.08}$	$99.75_{0.09}$
FLAN-T5-XXL (11B)	$p(\ell\|q)$	Accuracy	68.0	69.0	71.0	70.2	71.6
		PMI Acc.	75.2	75.2	77.0	77.8	79.0
		Prob. Mass	13.76	15.95	16.61	17.19	18.39
	$p(\ell\|q, \mathcal{L}_{string})$	Accuracy	**86.8**	**87.8**	**88.0**	**87.6**	**87.6**
		PMI Acc.	71.8	74.0	74.2	73.6	74.0
		Prob. Mass	98.35	99.07	99.19	99.22	99.22
	$p(\ell\|q, \mathcal{L}_{enum})$	Accuracy	**87.2**	86.8	86.8	**87.6**	**88.0**
		PMI Acc.	69.2	68.0	68.2	67.6	68.8
		Prob. Mass	99.7	99.76	99.79	99.71	99.47
`text-davinci-003`	$p(\ell\|q)$	Accuracy	62.8	71.4	75.6	75.6	77.4
		PMI Acc.	66.6	75.2	77.2	79.0	80.8
		Prob. Mass	0.0	16.89	24.14	25.29	26.22
	$p(\ell\|q, \mathcal{L}_{string})$	Accuracy	76.2	**79.8**	**82.0**	81.6	**82.0**
		PMI Acc.	74.2	76.8	79.0	79.0	79.0
		Prob. Mass	75.62	94.4	94.06	94.74	95.15
	$p(\ell\|q, \mathcal{L}_{enum})$	Accuracy	**79.4**	**79.4**	**82.8**	**83.2**	**81.8**
		PMI Acc.	76.0	77.0	77.8	80.2	78.0
		Prob. Mass	99.86	99.96	99.97	99.94	99.95

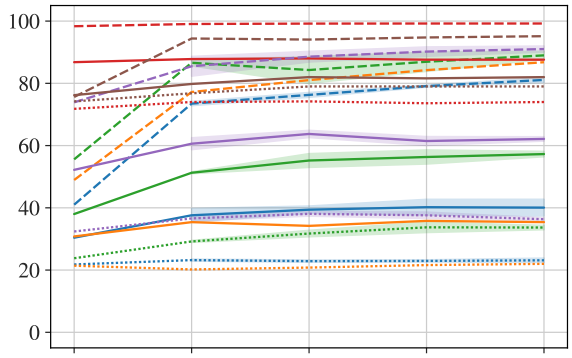
Table 9: A comparison of metrics (plotted in Fig. 8-Fig. 12) for each model and prompt type on the CommonsenseQA validation subset. See caption of Table 8 for more details. #s in parentheses are those reported in Holtzman et al. (2021), though exact model used may not be the same.

Model Name	Prompt Format	Metric	# In-Context Demonstrations 0	1	2	4	8
OPT 30B	$p(\ell\|q)$	Accuracy	29.2	27.8	29.2	29.8	34.2
		PMI Acc.	**55.6**	**51.0**	**53.2**	**55.4**	**55.8**
		Prob. Mass	6.38	6.61	7.33	8.6	9.79
	$p(\ell\|q, \mathcal{L}_{string})$	Accuracy	34.0	32.0	32.8	32.4	32.4
		PMI Acc.	44.4	44.4	45.4	46.6	45.0
		Prob. Mass	52.7	74.27	82.45	82.12	85.04
	$p(\ell\|q, \mathcal{L}_{enum})$	Accuracy	27.2	22.4	25.2	20.8	22.0
		PMI Acc.	43.0	42.0	42.2	41.6	42.2
		Prob. Mass	79.08	98.24	98.49	98.63	99.14
GPT-3 curie (~6.7B)	$p(\ell\|q)$	Accuracy	29.0(22.4)	$28.67_{1.17}$	$29.53_{2.2}$	$31.33_{0.46}$	$31.8_{1.11}$
		PMI Acc.	**50.4**(48.0)	$\mathbf{49.27_{2.53}}$	$\mathbf{49.93_{3.38}}$	$\mathbf{53.0_{3.5}}$	$\mathbf{52.73_{2.05}}$
		Prob. Mass	6.17	$7.29_{0.96}$	$7.56_{0.53}$	$8.55_{0.8}$	$8.74_{1.05}$
	$p(\ell\|q, \mathcal{L}_{string})$	Accuracy	29.2	$27.33_{3.95}$	$30.4_{1.11}$	$30.6_{2.16}$	$30.27_{0.31}$
		PMI Acc.	43.4	$45.13_{0.5}$	$44.87_{1.53}$	$44.67_{0.31}$	$44.0_{0.8}$
		Prob. Mass	41.88	$68.15_{10.47}$	$78.62_{5.19}$	$80.33_{4.73}$	$79.97_{2.62}$
	$p(\ell\|q, \mathcal{L}_{enum})$	Accuracy	24.0	$25.27_{0.92}$	$25.33_{1.29}$	$26.0_{1.04}$	$26.8_{2.09}$
		PMI Acc.	42.6	$42.53_{0.64}$	$41.67_{0.31}$	$42.2_{0.35}$	$42.67_{0.31}$
		Prob. Mass	67.55	$99.04_{0.08}$	$99.18_{0.16}$	$99.53_{0.05}$	$99.6_{0.04}$
GPT-3 davinci (~175B)	$p(\ell\|q)$	Accuracy	33.6(33.2)	$33.0_{2.42}$	$33.2_{3.03}$	$35.73_{0.95}$	$36.67_{0.81}$
		PMI Acc.	**57.8**(58.0)	$\mathbf{58.07_{2.58}}$	$\mathbf{58.73_{3.58}}$	$\mathbf{60.53_{1.45}}$	$\mathbf{60.2_{1.04}}$
		Prob. Mass	6.71	$7.97_{1.29}$	$8.61_{1.36}$	$9.51_{1.2}$	$9.83_{0.67}$
	$p(\ell\|q, \mathcal{L}_{string})$	Accuracy	34.8	$41.67_{3.14}$	$43.27_{2.91}$	$46.53_{2.32}$	$45.73_{0.7}$
		PMI Acc.	46.6	$49.6_{2.11}$	$50.53_{0.95}$	$50.53_{0.64}$	$50.33_{1.33}$
		Prob. Mass	47.31	$83.39_{4.29}$	$89.47_{2.92}$	$89.05_{3.26}$	$90.3_{1.47}$
	$p(\ell\|q, \mathcal{L}_{enum})$	Accuracy	31.2	$39.0_{0.72}$	$41.73_{2.55}$	$47.47_{2.08}$	$44.53_{1.1}$
		PMI Acc.	42.8	$44.07_{1.03}$	$45.2_{0.2}$	$47.07_{0.61}$	$46.33_{1.36}$
		Prob. Mass	65.75	$98.73_{0.28}$	$99.31_{0.24}$	$99.57_{0.12}$	$99.7_{0.1}$
davinci-instruct-beta	$p(\ell\|q)$	Accuracy	36.4	$36.4_{0.4}$	$37.73_{2.14}$	$38.27_{1.47}$	$38.4_{0.72}$
		PMI Acc.	**59.6**	$\mathbf{62.27_{0.12}}$	$\mathbf{63.07_{1.67}}$	$\mathbf{63.6_{1.25}}$	$\mathbf{62.67_{1.14}}$
		Prob. Mass	7.07	$8.26_{1.0}$	$9.05_{1.35}$	$10.37_{0.92}$	$10.39_{0.58}$
	$p(\ell\|q, \mathcal{L}_{string})$	Accuracy	41.6	$44.13_{5.62}$	$46.4_{3.22}$	$48.47_{1.03}$	$47.73_{1.97}$
		PMI Acc.	50.6	$53.07_{2.01}$	$52.8_{1.64}$	$53.87_{1.7}$	$52.93_{0.42}$
		Prob. Mass	72.15	$85.9_{5.12}$	$89.41_{2.3}$	$90.28_{2.11}$	$91.43_{1.15}$
	$p(\ell\|q, \mathcal{L}_{enum})$	Accuracy	36.2	$44.87_{2.19}$	$47.87_{2.84}$	$46.87_{2.81}$	$48.13_{0.5}$
		PMI Acc.	45.0	$48.2_{1.59}$	$48.87_{0.76}$	$48.67_{1.03}$	$50.4_{1.0}$
		Prob. Mass	69.07	$97.79_{0.45}$	$98.92_{0.25}$	$99.41_{0.01}$	$99.54_{0.17}$
FLAN-T5-XXL (11B)	$p(\ell\|q)$	Accuracy	33.2	33.4	34.6	35.2	35.8
		PMI Acc.	58.4	58.4	60.6	59.8	61.4
		Prob. Mass	12.99	14.47	14.63	14.79	13.88
	$p(\ell\|q, \mathcal{L}_{string})$	Accuracy	80.0	79.6	81.2	81.2	80.6
		PMI Acc.	75.0	76.2	76.6	76.6	76.2
		Prob. Mass	96.92	97.25	97.86	97.99	98.0
	$p(\ell\|q, \mathcal{L}_{enum})$	Accuracy	**83.0**	**83.6**	**84.2**	**83.6**	**84.2**
		PMI Acc.	73.0	72.4	72.8	73.2	72.8
		Prob. Mass	99.69	99.73	99.75	99.71	99.64
text-davinci-003	$p(\ell\|q)$	Accuracy	42.8	42.0	44.4	45.4	46.0
		PMI Acc.	63.0	62.4	64.6	65.4	68.0
		Prob. Mass	7.67	7.68	8.91	11.7	12.59
	$p(\ell\|q, \mathcal{L}_{string})$	Accuracy	77.0	77.0	78.0	80.2	81.0
		PMI Acc.	77.0	80.2	80.6	81.6	81.6
		Prob. Mass	80.62	97.41	97.39	97.78	98.52
	$p(\ell\|q, \mathcal{L}_{enum})$	Accuracy	**80.0**	**81.6**	**81.6**	83.0	**83.6**
		PMI Acc.	**80.4**	**81.2**	**81.8**	**84.2**	82.6
		Prob. Mass	99.76	99.96	99.79	99.83	99.92

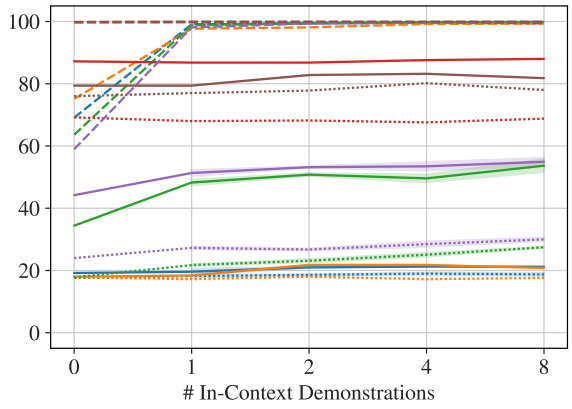
Table 10: A comparison of metrics (plotted in Fig. 9-Fig. 13) for each model and prompt type on the OpenbookQA test set. See caption of Table 8 for more details. #s in parentheses are those reported in Holtzman et al. (2021), though exact model used may not be the same.



(a) $p(\ell|q)$, "string prompt"

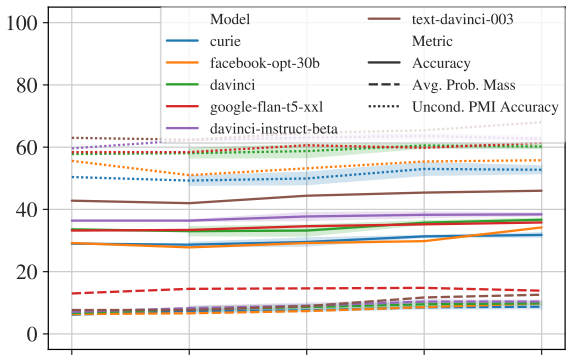


(b) $p(\ell|q, \mathcal{L}_{string})$, "string answer prompt"

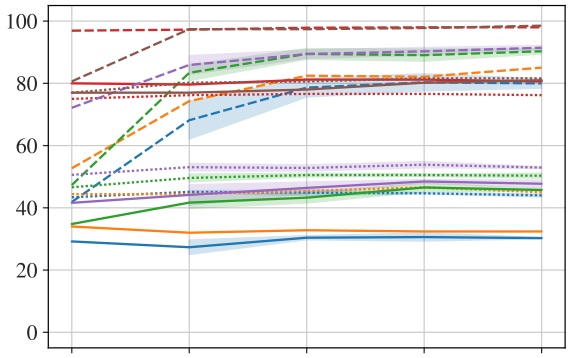


(c) $p(\ell|q, \mathcal{L}_{enum})$; "enumerated answer prompt"

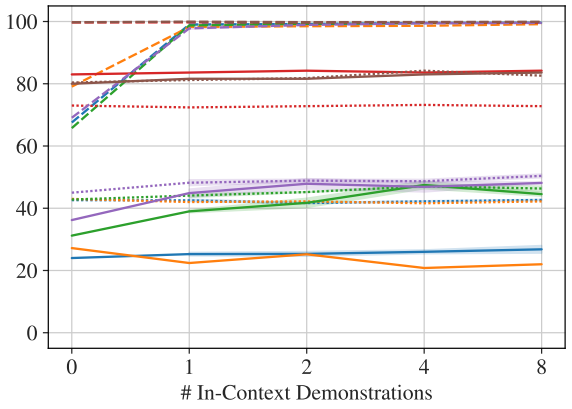
Figure 8: CommonsenseQA validation subset sequence-scoring accuracy (Eq. (1); solid lines), PMI accuracy (Eq. (3); dotted lines), and amount of probability mass on first tokens of valid answer choices (Eq. (4); dashed lines) as a function of **number** (x-axis) and **format** (a,b,c subgraphs) of in-context examples, for six pretrained language models. Random accuracy is 20%. See caption of Figure 3 for more details. See the tabular version of these results in TODO. Note this task is explicitly in-domain for FLAN-T5.



(a) $p(\ell|q)$, "string prompt"



(b) $p(\ell|q, \mathcal{L}_{string})$, "string answer prompt"



(c) $p(\ell|q, \mathcal{L}_{enum})$; "enumerated answer prompt"

Figure 9: OpenbookQA test set sequence-scoring accuracy (Eq. (1); solid lines), PMI accuracy (Eq. (3); dotted lines), and amount of probability mass on first tokens of valid answer choices (Eq. (4); dashed lines) as a function of **number** (x-axis) and **format** (a,b,c subgraphs) of in-context examples, for six pretrained language models. Random accuracy is 25%. See caption of Figure 3 for more details. See the tabular version of these results in TODO. Note this task is explicitly in-domain for FLAN-T5.

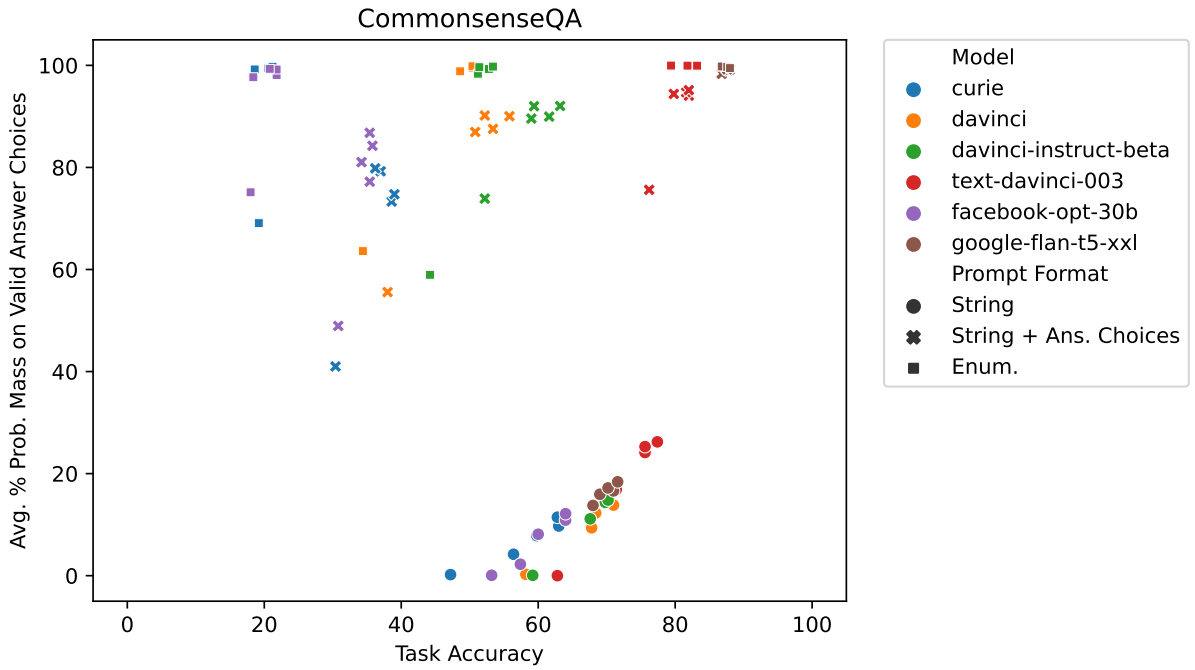


Figure 10: A scatterplot showing the relationship between task accuracy and average total probability mass on valid answer choices for the CommonsenseQA validation subset (for 0, 1, 2, 4 and 8 in-context examples). Note this task is explicitly in-domain for FLAN-T5. See Figure 4 for more info.

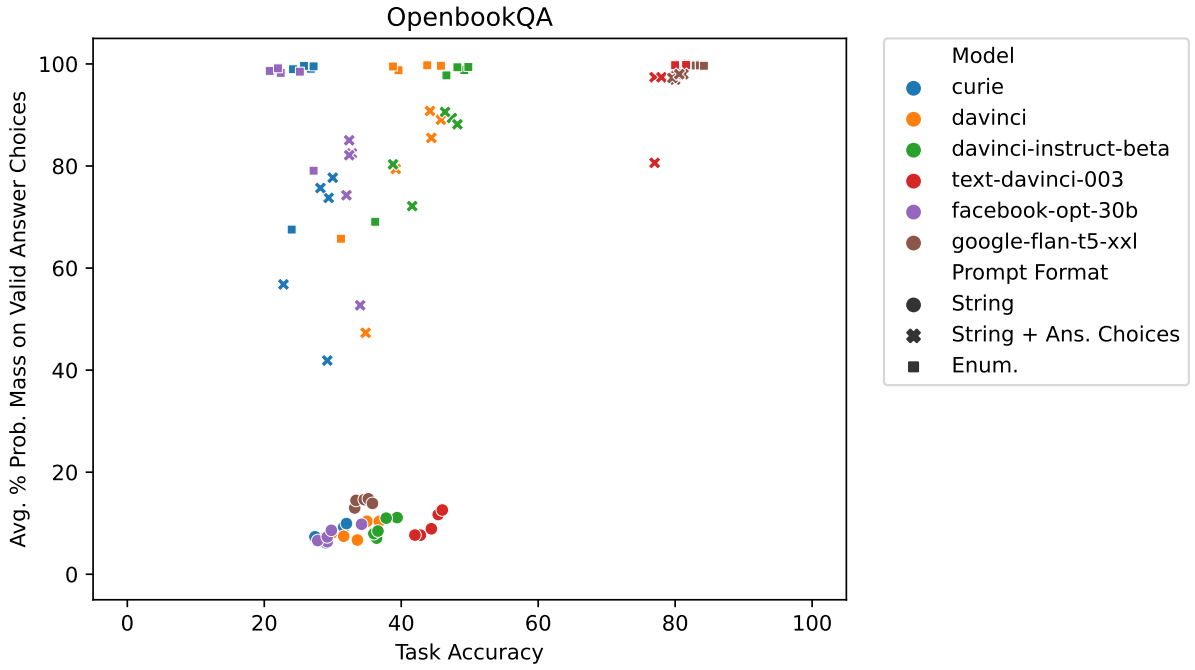


Figure 11: A scatterplot showing the relationship between task accuracy and average total probability mass on valid answer choices for the OpenbookQA test set (for 0, 1, 2, 4 and 8 in-context examples). Note this task is explicitly in-domain for FLAN-T5. See Figure 4 for more info.

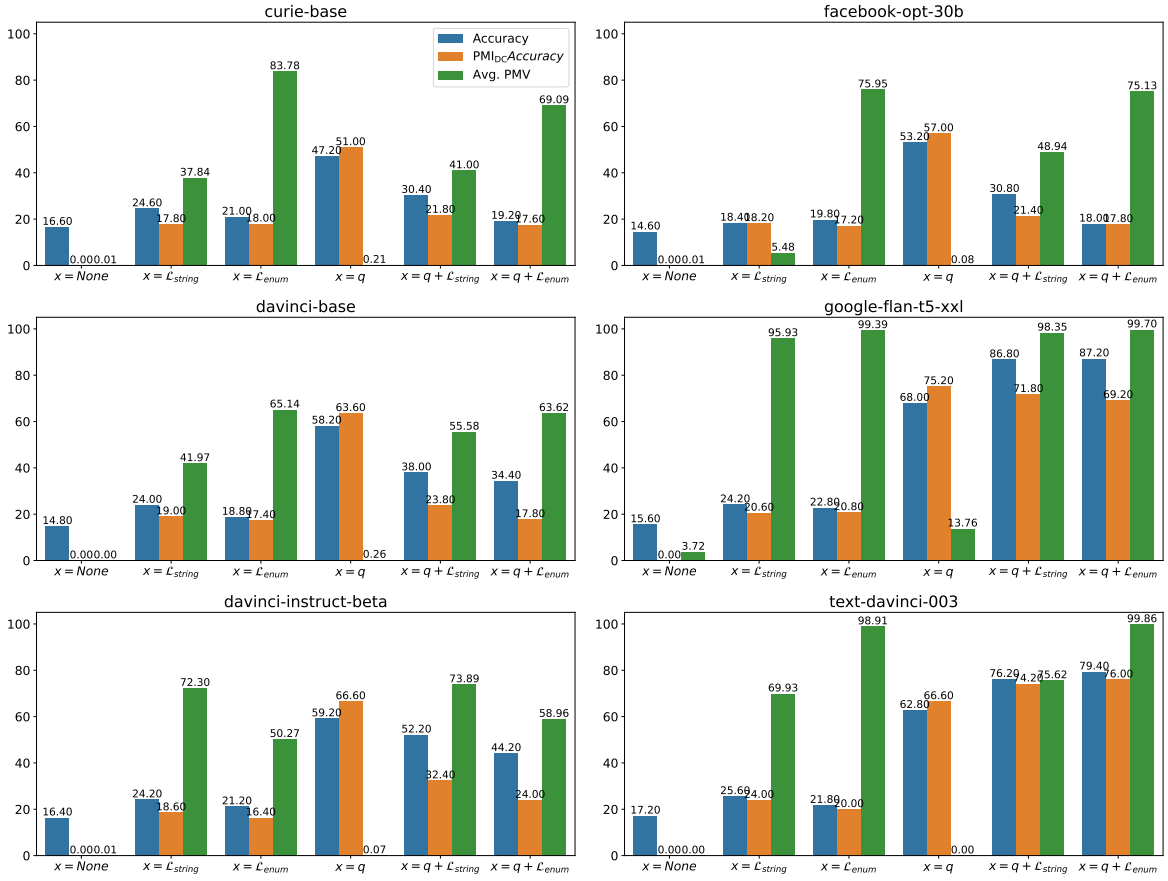


Figure 12: Zero-shot results on a subset of the CommonsenseQA validation set for various LLMs: sequence-scoring accuracy (Eq. (1); blue), PMI accuracy (Eq. (3); orange), and average total probability mass on first tokens of valid answer choices (Eq. (4); green). Random accuracy is 20%. See caption of Fig. 5 for more info. Note this task is explicitly in-domain for FLAN-T5.

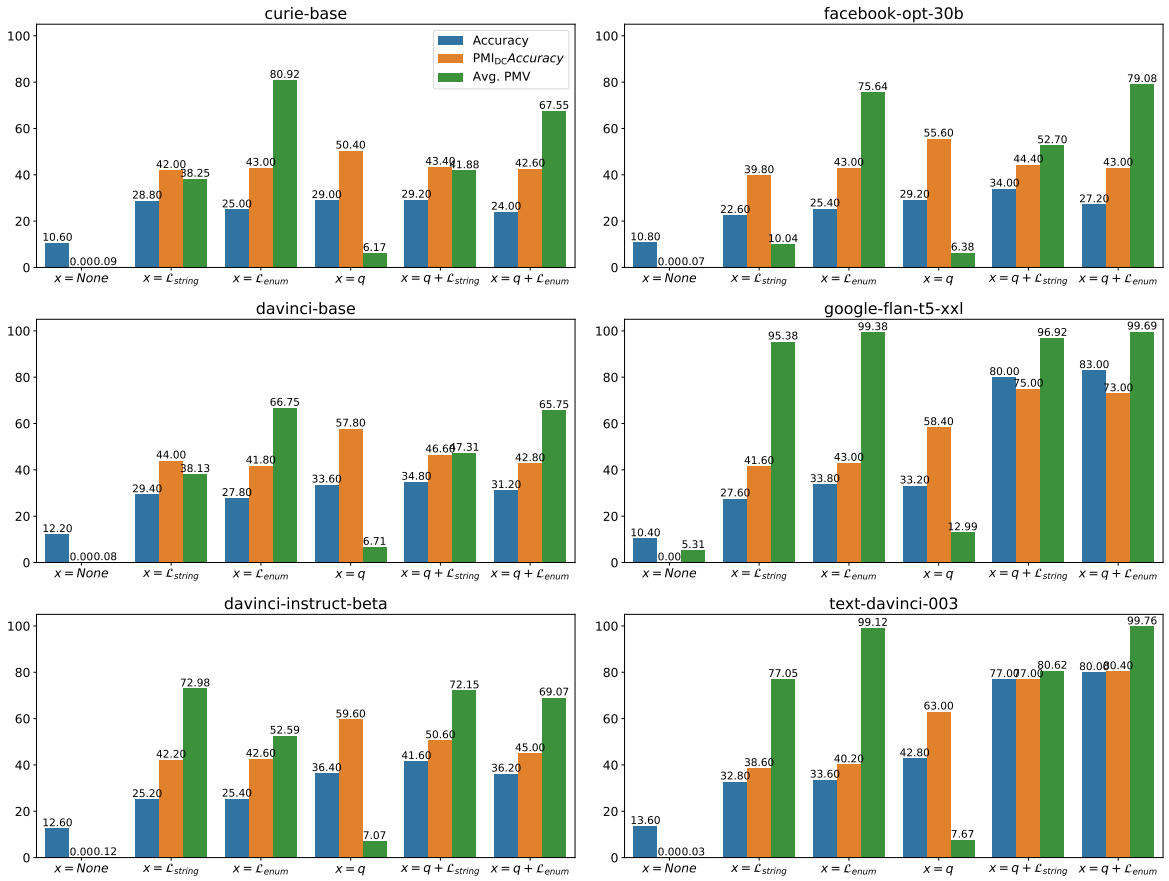


Figure 13: Zero-shot results on the OpenbookQA test set for various LLMs: sequence-scoring accuracy (Eq. (1); blue), PMI accuracy (Eq. (3); orange), and average total probability mass on first tokens of valid answer choices (Eq. (4); green). Random accuracy is 20%. See caption of Fig. 5 for more info. Note this task is explicitly in-domain for FLAN-T5.