# WHAT MAKES NEGATION HARD FOR LARGE LANGUAGE MODELS TO UNDERSTAND?

**Yuhui Zhang, Zhengping Zhou, Michihiro Yasunaga, Jeff Z. HaoChen, James Zou, Percy Liang, Serena Yeung**
Department of Computer Science
Stanford University
Stanford, CA 94305, USA
{yuhuiz, zpzhou, myasu, jhaochen, jamesz, pliang, syyeung}@cs.stanford.edu

## ABSTRACT

Larger language models often perform better in various downstream tasks. In this work, we present NeQA, a task on which larger language models perform worse. The task is to answer multiple choice questions with negation, constructed by adding negation words to the original questions. We find that the stronger the normal scaling is shown by the language model on the original questions, the stronger its inverse scaling is on the negated question. We find that this is because adding negation has minimal impact on the predictions of language models. To understand why, we perform a geometric analysis of the contextual embedding space of the large language model and find that adding negation approximates adding a constant vector orthogonal to the question embedding subspace. This unique geometric property makes the language model ignore negation, which we provide a theoretical explanation for this. We further provide intuition why the language modeling process makes the model learn this geometry, therefore unable to understand negation. Our work sheds light on a fundamental problem of large language models and provides new insights into understanding their properties.

## 1 TASK

| | **Multi-choice Joint** |
|---|---|
| Question | The following are multiple choice questions (with answers) about common sense.<br>Question: If a cat has a body temp that is below average, it isn't in<br>A. danger<br>B. safe ranges<br>Answer: |
| Choices | [A, **B**] |
| | **Multi-choice Separate** |
| Question | If a cat has a body temp that is below average, it isn't in |
| Choices | [danger, **safe ranges**] |

Table 1: Example of our curated dataset, NeQA, representing negated question answering. For each question, we use two templates to evaluate multi-choice question answering. Top: we include choices in the question and predict single token as the answer. Bottom: we do not include choices in the question and predict multiple tokens as the answer. Correct choices are bolded.

### 1.1 TASK DESCRIPTION

*Negation* is a common linguistic phenomenon that can completely alter the semantics of a sentence by changing just a few words. This task evaluates whether language models can understand negation, which is an important step towards true natural language understanding. Specifically, we focus on negation in multi-choice questions, considering its wide range of applications and the simplicity of evaluation. We collect a multi-choice question answering dataset, *NeQA*, that includes questions

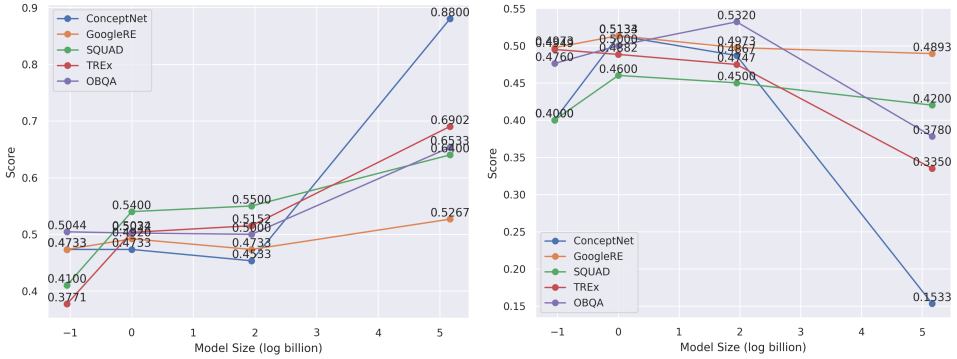


Figure 1: Inverse scaling trend of five subsets of data in NeQA. We find that the stronger the normal scaling is on original questions (left), the stronger the inverse scaling is on negated questions (right). X-axis shows four GPT-3 models: Ada (350M), Babbage (1B), Curie (7B), Davinci (175B). This demonstrates a fundamental challenge for large language models to understand negations.

with negations. When negation is presented in the question, the original correct answer becomes wrong, and the wrong answer becomes correct. We use the *accuracy* metric to examine whether the model can understand negation in the questions and select the correct answer given the presence of negation. We observe a clear inverse scaling trend on GPT-3, demonstrating that larger language models can answer more complex questions but fail at the last step to understanding negation.

### 1.2 Task Importance

This task is important because it demonstrates that current language models cannot understand negation — a very common linguistic phenomenon and a real-world challenge to natural language understanding.

### 1.3 Why Inverse Scaling

For question answering, larger language models usually achieve better accuracy because more factual and commonsense knowledge is stored in the model parameters and can be used as a knowledge base to answer these questions without context.

A higher accuracy rate means a lower chance of choosing the wrong answer. Can we change the wrong answer to the correct one? A simple solution is to negate the original question. If the model is insensitive to negation, it will still predict the same answer and, therefore, will exhibit an inverse scaling trend.

We expect that the model cannot understand negation because negation introduces only a small perturbation to the model input. It is difficult for the model to understand that this small perturbation leads to completely different semantics.

In Section 2, we provide a detailed analysis about why inverse scaling appears under negation.

### 1.4 Novelty and Surprisingness

The closest work to us is Kassner & Schütze (2020), which shows that Elmo / BERT cannot understand negation. However, to the best of our knowledge, no work has extended the analysis to recent advances of enormous language models. BERT only has hundreds of millions of parameters, but GPT-3 Davinci has hundreds of billions of parameters. This finding should be surprising to the community, as large language models show an incredible variety of emergent capabilities (Bommasani et al., 2021), but still fail to understand negation, which is a fundamental concept in language.

### 1.5 Data Generation Procedure

For ConceptNet, GoogleRE, SQUAD, TREx, we leverage the existing benchmark Negated-LAMA (Kassner & Schütze, 2020) and perform the following processings:

The original input contains three raw fields, masked negations (negated question), masked misprimed (misprimied question), and obj label (answer).

- masked negations: Child does not want [MASK].
- masked misprimed: Lab? Child wants [MASK].
- obj label: Love

We use the misprime in the question as the second choice ("lab" in the example). Combined with the answer ("love" in the example), we form two-choice questions using two templates introduced in Table 1. We randomly sample at most 50 questions from each json files in the NegatedLAMA dataset, and we balance the label distributions such that the first and the second choices are equally presented as the correct choice.

NegatedLAMA already includes different types of negations, but these metadata are not available in the dataset. To further understand the effect of different negation types, we apply rules to filter and transform questions in OBQA (Mihaylov et al., 2018), a widely-used commonsense dataset.

Specifically, we define six types of negation: notional verb negation (e.g., cause → does not/doesn't cause), linking verb negation (e.g., is → is not / isn't), modal verb negation (e.g., can → can not / can't), conjunction negation (e.g., because → not because), other negation words (e.g., hardly, unable, incapable), and negation prompt (e.g., choose the wrong answer).

For each type, we first include existing questions in the OBQA training and validation dataset that satisfy this type, keep the correct answer and uniformly sample an incorrect answer. If there are less than 50 questions in the OBQA dataset, we additionally apply a rule-based transformation, and sample an incorrect answer as the correct answer and treat the correct answer as the incorrect answer. Then, we balance the label distributions and negation abbreviation distributions if applicable (e.g., do not / don't). Finally, we verify the correctness of each question by manual inspection. In total, OBQA contains 500 high-quality questions representing different types of negation.

In summary, our dataset includes 150 questions from ConceptNet, 374 questions from GoogleRE, 100 questions from SQUAD, 594 questions from TREx, and 500 questions from OBQA, which represents diverse negation types, data distributions, and prompts. We believe that our dataset is an important benchmark to test whether language models understand negation.

For the inverse scaling prize submission, we only include subsets of questions that cause strong inverse scaling, which are 150 questions from ConceptNet, 594 questions from TREx, and 200 questions from OBQA with negation type linking verb negation and conjunction negation. For each question, we use two templates to evaluate multi-choice question answering shown in Table 1. We analyze why these subsets lead to stronger inverse scaling in Section 2.

## 2 Inverse Scaling Result and Analysis

### 2.1 Main Result: Stronger Normal Scaling, Strong Inverse Scaling on Negated Questions

We use five datasets for experiments: ConceptNet, GoogleRE, SQUAD, TREx, OBQA. We show GPT-3 performance curves on these five datasets in Figure 1.

The main finding is that, *the stronger the normal scaling is on original questions, the stronger the inverse scaling is on negated questions*.

Therefore, we only include ConceptNet, TREx, and OBQA in the final submission, because GPT-3 does not show strong normal scaling on the original GoogleRE and SQUAD datasets.

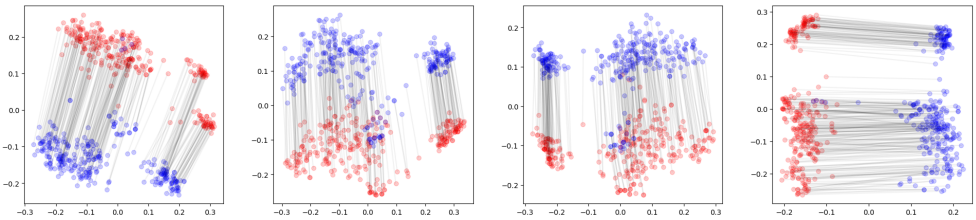

Figure 2: Embedding analysis of four GPT-3 models: Ada (350M), Babbage (1B), Curie (7B), Davinci (175B). Embeddings are obtained by feeding original sentences and negated sentences into OpenAI GPT-3 embedding APIs, and visualized using PCA dimension reductions. Red points represent embeddings of original sentences, while blue points represent embeddings of negated sentences. Lines connect pairs of original sentence and negated sentence. From the PCA view, we can find that negation approximates a constant vector in the embedding space.

### 2.2 Why Negation is Hard for Large Language Models?

To understand why negation is hard for large language models to understand, we provide a two-step explanation. First, we show that negation is represented with a unique geometry in the contextualized representation space, and this unique geometry makes language model ignore the negation. Then, we provide intuition about why this geometry is formed during language model pre-training.

#### 2.2.1 Negation Approximates a Constant Vector in the Contextualized Representation Space

We randomly sample 300 questions with their corresponding negated questions from ConceptNet, GoogleRE, SQUAD, TREx. For each original question and negated question, we generate embeddings from OpenAI embedding APIs. We visualize these embeddings using PCA. We find that, negation approximates a constant vector in the contextualized representation space. More specifically, for any sentence, given its original embedding $x$ and corresponding negated embedding $y$, we find that $x - y \approx c$, where $c$ is a constant vector. This is shown in Figure 2. Further analysis reveals that this vector is also orthogonal to the embedding space of $x$ and $y$.

The language modeling task can be viewed as a multi-class classification task. Given a contextualized sentence embedding $x$, we builds a decoder $h(x) = Wx$ that maps embedding to the probability distribution of tokens, where $W \in \mathbb{R}^{D\times|V|}$ with $D$ being the embedding dimension and $|V|$ being the vocabulary size.

The above findings with respect to the geometry of negation indicate that the decoder inputs of original questions and negated questions only differs by a constant $c$, i.e., $h(x) \approx h(y + c)$. Intuitively, since $c$ is an orthogonal constant to the span of embeddings $x$ and $y$, the weight matrix of the learned decoder should also be orthogonal to $c$. Hence the prediction of the decoder is not affected by $c$. This intuition explains why we observe that top predicted tokens remain the same when adding negation to the original question.

In the following proposition, we further adapt the theorem proposed by Zhang et al. (2022) to show that a decoder trained with a regularized quadratic loss is guaranteed to be orthogonal to the negation. Therefore, negation will not affect language model predictions.

**Proposition 2.1.** *Suppose there exists a constant vector $\boldsymbol{c} \in \mathbb{R}^d$ such that every pair of original sentence embedding $\boldsymbol{x}$ and negated sentence embedding $\boldsymbol{y}$ satisfies $\boldsymbol{c} = \boldsymbol{x} - \boldsymbol{y}$. Suppose $\boldsymbol{c}$ is orthogonal to the span of sentence embeddings (i.e., $\boldsymbol{c}^T\boldsymbol{x} = \boldsymbol{c}^T\boldsymbol{x}'$ for two sentence embeddings $\boldsymbol{x}$ and $\boldsymbol{x}'$), and the sentence embeddings have zero mean in the subspace orthogonal to $\boldsymbol{c}$ (i.e., $\mathbb{E}_x[\Pi_c(\boldsymbol{x})] = \boldsymbol{0}$ where $\Pi_c(\boldsymbol{x})$ projects the vector $\boldsymbol{x}$ to the subspace orthogonal to $\boldsymbol{c}$). Then, for any $\lambda > 0$ and linear function $h_W(\boldsymbol{u}) = \boldsymbol{W}\boldsymbol{u}$ that minimizes the regularized quadratic loss $\mathbb{E}_{x,t}[\mathcal{L}_{quad}(h_W(\boldsymbol{x}), t)] + \lambda\|\boldsymbol{W}\|_F^2$, where $\mathcal{L}_{quad}(h_W(\boldsymbol{u}), t) = \|h_W(\boldsymbol{u}) - \tilde{\boldsymbol{e}}_t\|_2^2$, $\tilde{\boldsymbol{e}}_t = \boldsymbol{e}_t - \mathbb{E}_{t'}[\boldsymbol{e}_{t'}]$, $\boldsymbol{e}_t \in \{0,1\}^{|V|}$ be a one-hot vector such that the $t$-th dimension is $1$ and other dimensions are $0$, we*

*have that*

$$h_W(\boldsymbol{x}) = h_W(\boldsymbol{y})$$

*Thus, negation will not affect language model predictions.*

*Proof of Proposition 2.1.* Since $\boldsymbol{c}^T\boldsymbol{x} = \boldsymbol{c}^T\boldsymbol{x}'$ for all image features $\boldsymbol{x}$ and $\boldsymbol{x}'$, we can find a $\tau \in \mathbb{R}$ such that $\boldsymbol{x} = \Pi_c(\boldsymbol{x}) + \tau\boldsymbol{c}$. Notice that

$$\begin{aligned}
\mathbb{E}_{x,t}[\mathcal{L}_{\text{quad}}(h_W(\boldsymbol{x}), t)] &= \mathbb{E}_{x,t}[\|\boldsymbol{W}\boldsymbol{x} - \tilde{\boldsymbol{e}}_t\|_2^2] \\
&= \|\mathbb{E}_x[\boldsymbol{W}\boldsymbol{x}] - \mathbb{E}_t[\tilde{\boldsymbol{e}}_t]\|_2^2 + \mathbb{E}_{x,t}[\|(\boldsymbol{W}\boldsymbol{x} - \tilde{\boldsymbol{e}}_t) - (\mathbb{E}_x[\boldsymbol{W}\boldsymbol{x}] - \mathbb{E}_t[\tilde{\boldsymbol{e}}_t])\|_2^2] \\
&= \|\mathbb{E}_x[\boldsymbol{W}\boldsymbol{x}] - \mathbb{E}_t[\tilde{\boldsymbol{e}}_t]\|_2^2 + \mathbb{E}_{x,t}[\|\boldsymbol{W}\Pi_c(\boldsymbol{x}) - \tilde{\boldsymbol{e}}_t\|_2^2] \\
&= \|\boldsymbol{W}\mathbb{E}_x[\Pi_c(\boldsymbol{x})] + \tau\boldsymbol{W}\boldsymbol{g} - \mathbb{E}_t[\tilde{\boldsymbol{e}}_t]\|_2^2 + \mathbb{E}_{x,t}[\|\boldsymbol{W}\Pi_c(\boldsymbol{x}) - \tilde{\boldsymbol{e}}_t\|_2^2].
\end{aligned}$$

Since $\mathbb{E}_x[\Pi_c(\boldsymbol{x})] = \boldsymbol{0}$ and $\mathbb{E}_t[\tilde{\boldsymbol{e}}_t] = \boldsymbol{0}$, the first term reduces to $\tau^2\|\boldsymbol{W}\boldsymbol{c}\|_2^2$. Notice that the second term in the loss decomposition only involves $\boldsymbol{W}$'s components that are orthogonal to $\boldsymbol{c}$. Thus the minimization of the second term is independent of the minimization of the first term. As a result, any $\boldsymbol{W}$ that minimizes the regularized quadratic loss must satisfy $\boldsymbol{W}\boldsymbol{c} = \boldsymbol{0}$.

For a pair of original sentence embedding and negated sentence embedding $\boldsymbol{x}, \boldsymbol{y}$, since $\boldsymbol{x} - \boldsymbol{y} = \boldsymbol{c}$ and $\boldsymbol{W}\boldsymbol{c} = \boldsymbol{0}$, we have $h_W(\boldsymbol{x}) = h_W(\boldsymbol{y})$. Hence we know their quadratic losses must be the same, which finishes the proof. □

Indeed, we observe the top-1 predicted token by various GPT-3 models is not affected when adding negation to the original sentences. For GPT-3 Ada, Babbage, Curie, Davinci, 87%, 91%, 86%, 93% of model predictions stay unchanged when adding negations, respectively.

**Note:** OpenAI's embedding interface uses a variant of GPT-3 that was fine-tuned in a contrastive learning fashion (Neelakantan et al., 2022). Therefore, it is different from the original GPT-3. To also confirm this finding on non-finetuned, vanilla language models, we also experimented with the embeddings from the publicly available GPT-2. We observed similar findings (Appendix Figure 4).

#### 2.2.2 Bag-of-word Property of the Text Encoder Explains Negation as a Constant Vector

The finding that negation approximates an orthogonal constant vector in various language models' contextualized embedding spaces explains why negation is hard to be captured by various language models.

To further understand why negation is a constant vector in the embedding space, we provide the following intuition. Let's think about a bag-of-word encoder; when we negate the sentence by adding the word "not", the sentence embedding will be pushed forward to the word "not" direction. Larger and more complex language models such as BERT have also been shown to exhibit analogous bag-of-word behaviors, especially at the initialization stage (Sinha et al., 2021). Therefore, this can explain why negation is approximately a constant vector in the contextualized embedding space after large language models are trained.

**Discussions:** The bag-of-word insight also suggests that other types of word additions / deletions may similarly cause model misbehaviors, such as modifiers, etc.

#### 2.2.3 Summary of Inverse Scaling with Negation

To summarize, because large language model exhibits bag-of-word behavior, negation approximates a constant vector in the contextualized representation space. This unique geometry makes the learned language models ignore negation. Therefore, since larger language models often perform better on normal tasks and show normal scaling trend, adding negation will cause inverse scaling.

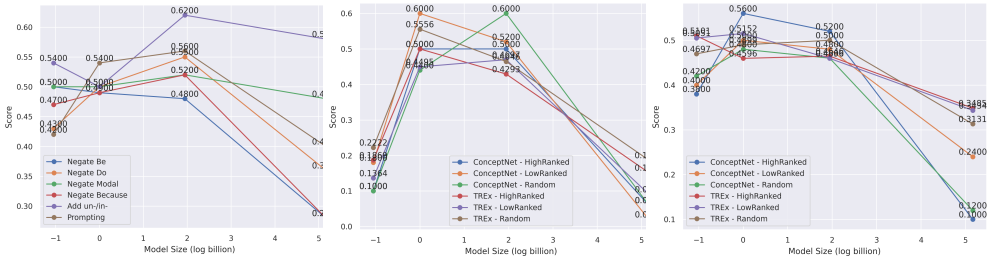


Figure 3: The effect of negation cateogry, wrong choice confusion, mispriming to inverse scaling trend. X-axis shows four GPT-3 models: Ada (350M), Babbage (1B), Curie (7B), Davinci (175B). High-ranked, low-ranked, random indicates the confusion of the wrong choice.

## 3 Other Analyses

In this section, we show that how our analysis framework enables us to predict inverse scaling in a diverse settings.

**Few-shot instead of zero-shot.** Based on our analysis, negation will still be a constant vector in the few-shot learning settings. Therefore, it will still show inverse scaling. This is also empirically demonstrated from the feedback of our first-round submission.

**Reinforcement learning from human feedback (RLHF) models.** Negation is barely optimized during the RLHF process. Therefore, negation will still be a constant vector in the few-shot learning setting and thus inverse scaling holds. This is also empirically demonstrated from the feedback of our first-round submission.

**Negation category.** Negating linking verbs ("be") leads to stronger inverse scaling, because these questions show stronger normal scaling. See Figure 3.

**Wrong choice confusion.** This experiment aims to understand whether more confusing choices will change the inverse scaling. For example, given the question "Apple is not made by", the wrong choice can be "Microsoft" (high ranked, more confusing), or can be "air" (low ranked, less confusing), or random word "China" (random). Based on our analysis, because negation is still a constant vector and therefore model predictions will not change. Therefore, it has no impact. See Figure 3.

**Mispriming.** Following Kassner & Schütze (2020), we put the wrong choice (i.e., the correct choice before negation) before the question (e.g., "Apple? iPhone is not made by"). Mispriming makes inverse scaling stronger on negated questions, because mispriming makes the normal scaling stronger. Interestingly, we also note a phase change happens in small-size models. See Figure 3. While this is a very interesting finding, mispriming might not be frequent in real-world applications of language models, so we are not including this in our main submission.

## Reproducibility Statement

We provide our datasets and implementations at `https://github.com/yuhui-zh15/InverseScaling`. The implementations will enable researchers to reproduce datasets and results described here, as well as apply our negation transformations to other datasets and run their own analyses.

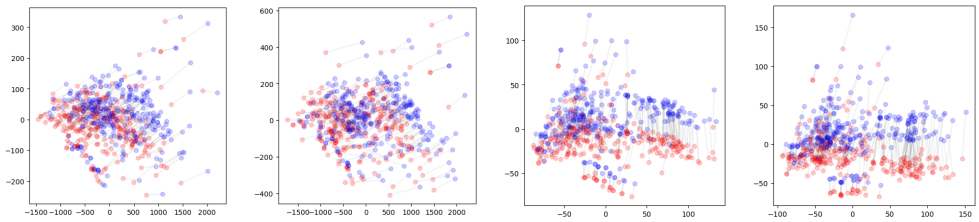

Figure 4: Embedding analysis of four GPT-2 models: small, medium, large, xl. From the PCA view, we can find that negation approximates a constant vector in the embedding space.

## REFERENCES


Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Nora Kassner and Hinrich Schütze. Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly. In *ACL*, 2020.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*, 2018.

Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, Nikolas Tezak, Jong Wook Kim, Chris Hallacy, et al. Text and code embeddings by contrastive pre-training. *arXiv preprint arXiv:2201.10005*, 2022.

Koustuv Sinha, Robin Jia, Dieuwke Hupkes, Joelle Pineau, Adina Williams, and Douwe Kiela. Masked language modeling and the distributional hypothesis: Order word matters pre-training for little. In *EMNLP*, 2021.

Yuhui Zhang, Jeff Z HaoChen, Shih-Cheng Huang, Kuan-Chieh Wang, James Zou, and Serena Yeung. DrML: Diagnosing and rectifying vision models using language. In *NeurIPS DistShift Workshop*, 2022. `https://openreview.net/pdf?id=ZpN2EOEUnr`.