Limits of In-context Learning II

AntNLP 2022 Fall Seminar

Liu Yanting

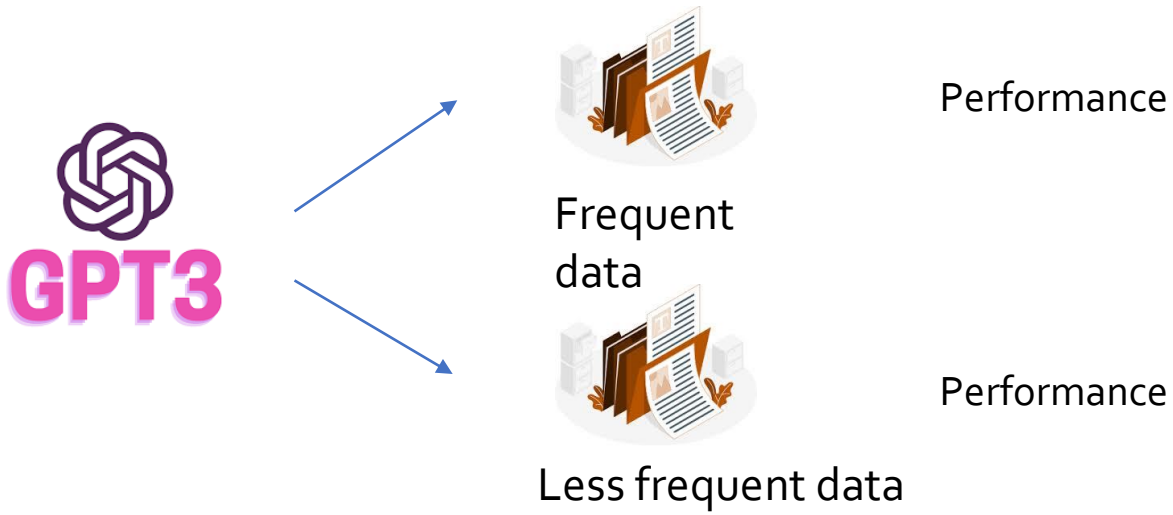
Impact of Pretraining Term Frequencies on Few-Shot Reasoning

Yasaman Razeghi [1] **Robert L. Logan IV** [1] **Matt Gardner** [2] **Sameer Singh** [1 3]

Background

current evaluation schemes for the reasoning of large language models, often neglect or underestimate the impact of **data leakage**

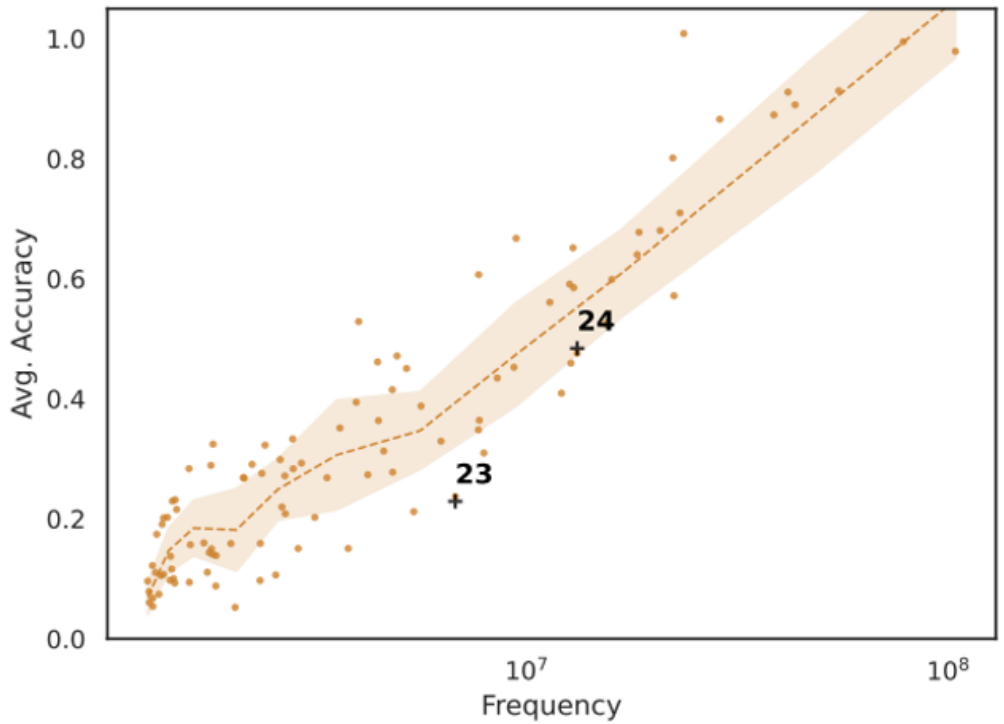
A model that has learned to reason in the training phase should be able to generalize outside of the narrow context that it was trained in.



✍: Vicky Zeng, Neha Verma, Lingfeng Shen

Problems

Q: What is 24 times 18? A: ___ *Model:* 432 ✓
Q: What is 23 times 18? A: ___ *Model:* 462 ✗



: Vicky Zeng, Neha Verma, Lingfeng Shen

Frequency

For numerical reasoning tasks, its instances consist of input terms, $x = (x_1, \ldots x_i, \ldots x_n)$, and a derived output term y, where the $x_i's$ are either positive integers or units of time, and y is a positive integer.

- $\omega_{\{x_1\}}$: the number of times that x_1 (e.g., 23) appears in the pretraining data.
- $\omega_{\{x_1,x_2\}}$: the number of times that the input terms x_1 (e.g., 23) and x_2 (e.g., 18) appear in the pretraining data within a specific window size.
- $\omega_{\{x_1,y\}}$: the number of times that the first input term x_1 (e.g., 23) and the output term y (e.g., 414) appear in the pretraining data within a specific window size.

Performance Gap

Difference in accuracy between top 10% of instances and bottom 10% of instances (by frequency)

$$\Omega = \{(\omega_X^{(n)}, a^{(n)})\}$$
$$\Delta(\Omega) = \mathrm{Acc}(\Omega_{>90\%}) - \mathrm{Acc}(\Omega_{<10\%})$$

✍: Vicky Zeng, Neha Verma, Lingfeng Shen

Experimental setup

1. Models
    1. GPT-J-6B
    2. GPT-Neo-1.3B
    3. GPT-Neo-2.7B
2. Corpus
    1. Pile dataset
3. Prompt counts
    1. k = 0,2,4,8,16

: Vicky Zeng, Neha Verma, Lingfeng Shen

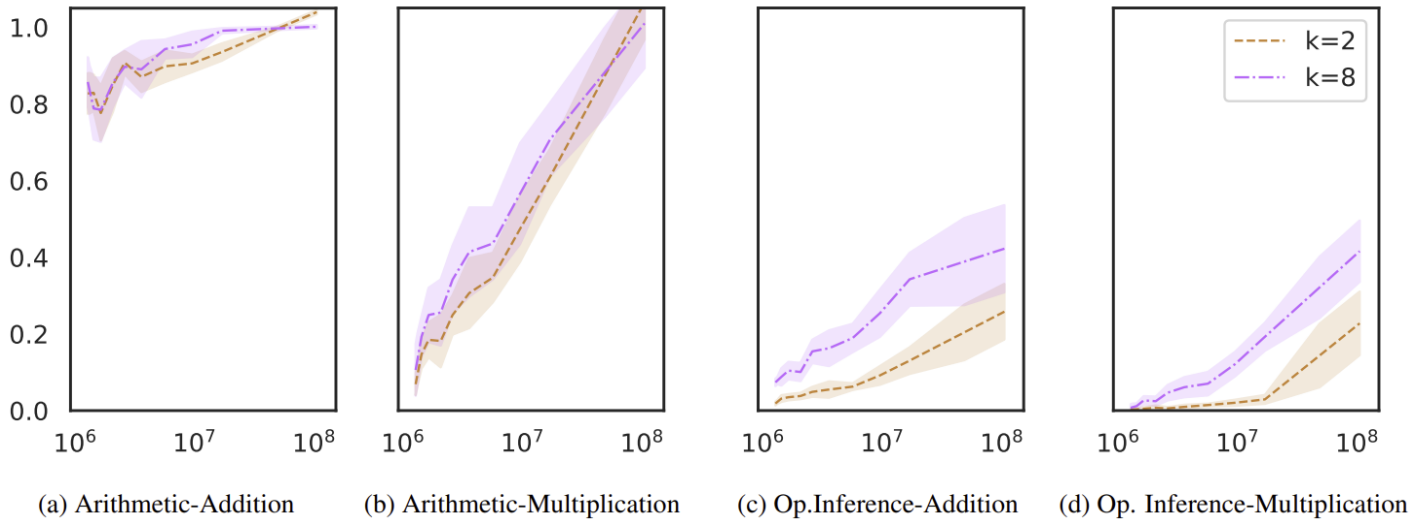
Numerical Reasoning Tasks

Table 1. Prompt templates and the number of test cases investigated for each numerical reasoning task.

Task	Prompt Template	#Test Cases
Arithematic		
Multiplication	*Q:What is* x_1 *times* x_2*? A:* y	5000
Addition	*Q:What is* x_1 *plus* x_2*? A:* y	5000
Operation Inference		
Mult. #	*Q:What is* x_1 # x_2*? A:* y	5000
Add. #	*Q:What is* x_1 # x_2*? A:* y	5000
Time Unit Inference		
Min→Sec	*Q:What is* x_1 *minutes in seconds? A:* y	79
Hour→Min	*Q:What is* x_1 *hours in minutes? A:* y	100
Day→Hour	*Q:What is* x_1 *days in hours? A:* y	100
Week→Day	*Q:What is* x_1 *weeks in days? A:* y	100
Month→Week	*Q:What is* x_1 *months in weeks? A:* y	100
Year→Month	*Q:What is* x_1 *years in months? A:* y	100
Decade→Year	*Q:What is* x_1 *decades in years? A:* y	100

Main Finding: Heavy dependence on pretraining frequency

Strong positive correlation between test performance and pretraining term frequency



(a) Arithmetic-Addition (b) Arithmetic-Multiplication (c) Op.Inference-Addition (d) Op. Inference-Multiplication

✍: Vicky Zeng, Neha Verma, Lingfeng Shen

Additional Support: Performance Gap, Inference vs Arithmetic Gap

- Performance gap **increases** as number of *k* shots increases
- Inference task performance is much **lower** than arithmetic task performance

k	Addition				Multiplication				Addition (#)				Multiplication (#)			
	Acc.	Δ_1	$\Delta_{1,2}$	$\Delta_{1,y}$	Acc.	Δ_1	$\Delta_{1,2}$	$\Delta_{1,y}$	Acc.	Δ_1	$\Delta_{1,2}$	$\Delta_{1,y}$	Acc.	Δ_1	$\Delta_{1,2}$	$\Delta_{1,y}$
0	1.6	8.4	6.9	8.0	5.4	18.0	20.6	30.8	-	-	-	-	-	-	-	-
2	88.2	16.8	21.7	21.9	35.9	77.6	79.3	89.9	7.8	18.1	25.3	28.3	3.1	14.1	13.7	14.2
4	91.4	15.0	24.8	26.4	39.2	70.8	76.4	83.5	9.8	24.8	30.1	30.4	5.7	20.9	21.3	23.4
8	89.6	16.3	26.5	29.6	42.9	74.6	80.8	86.0	19.8	31.0	44.8	45.2	9.4	31.3	33.2	34.7
16	88.6	16.4	27.3	31.0	40.9	73.3	77.7	82.6	26.2	38.5	47.2	49.9	11.0	39.6	38.7	42.6

✍: Vicky Zeng, Neha Verma, Lingfeng Shen

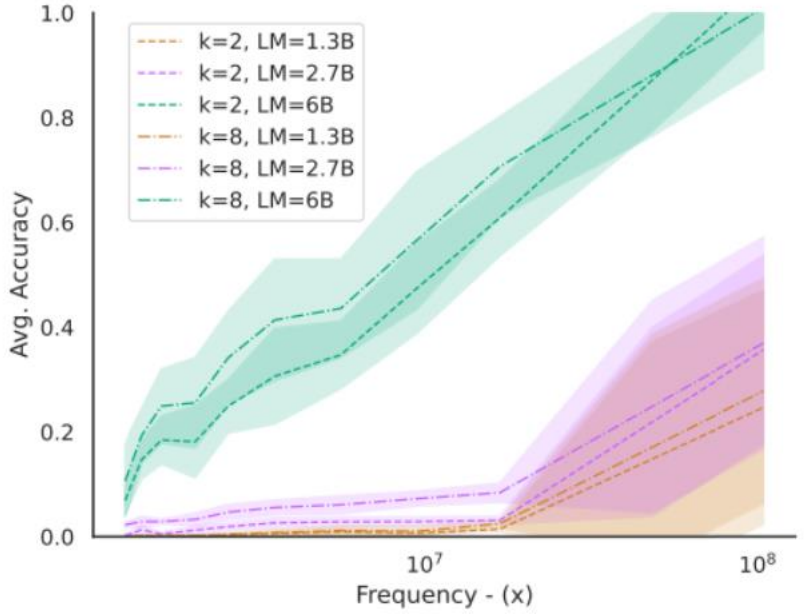
Results on Time-Unit Conversion

k	Min→Sec				Hour→Min				Day→Hour				Week→Day			
	Acc.	$\Delta_{1,2}$	$\Delta_{1,2,3}$	$\Delta_{1,2,y}$	Acc.	$\Delta_{1,2}$	$\Delta_{1,2,3}$	$\Delta_{1,2,y}$	Acc.	$\Delta_{1,2}$	$\Delta_{1,2,3}$	$\Delta_{1,2,y}$	Acc.	$\Delta_{1,2}$	$\Delta_{1,2,3}$	$\Delta_{1,2,y}$
0	1.3	0.0	0.0	12.5	1.0	0.0	0.0	5.0	1.0	0.0	0.0	10.0	1.0	0.0	0.0	10.0
2	25.5	62.5	67.5	67.5	19.4	58.0	40.5	44.0	12.1	28.9	24.0	28.0	13.1	43.5	50.0	54.0
4	35.5	60.0	71.7	63.1	29.1	76.4	50.5	59.0	22.7	46.4	45.0	47.5	19.2	40.9	43.3	47.0
8	49.9	72.1	79.0	52.7	36.3	74.6	52.5	63.0	31.0	59.1	52.5	54.5	28.6	70.6	62.0	67.0
16	58.4	82.7	74.4	48.5	42.8	80.1	49.0	62.5	43.3	62.8	56.0	54.8	28.0	22.1	31.4	33.2

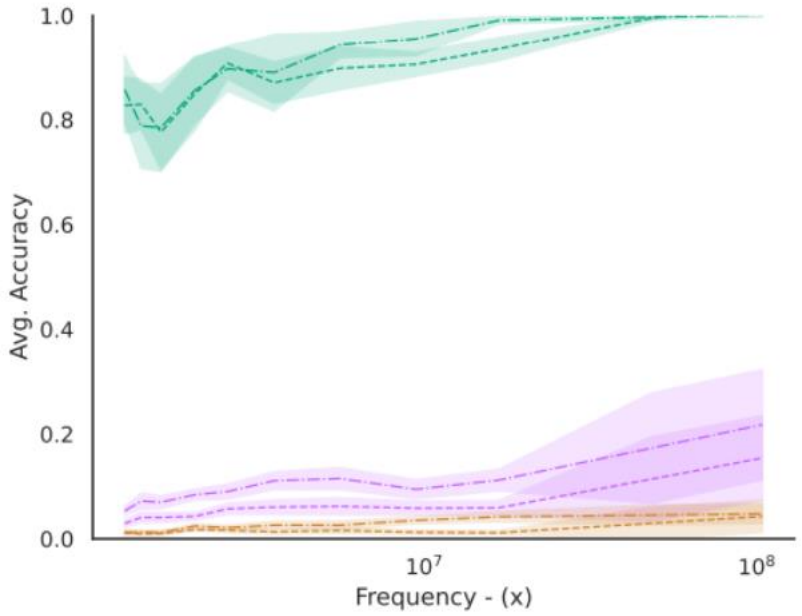
Shots, k	Month→Week				Year→Month				Decade→Year			
	Acc.	$\Delta_{1,2}$	$\Delta_{1,2,3}$	$\Delta_{1,2,y}$	Acc.	$\Delta_{1,2}$	$\Delta_{1,2,3}$	$\Delta_{1,2,y}$	Acc.	$\Delta_{1,2}$	$\Delta_{1,2,3}$	$\Delta_{1,2,y}$
0	1.0	0.0	0.0	10.0	1.0	0.0	0.0	10.0	3.1	14.3	14.3	28.6
2	30.1	8.5	9.3	21.0	21.8	58.0	64.0	53.0	76.5	38.8	47.1	43.1
4	63.3	22.9	26.2	10.5	31.9	64.8	69.5	66.8	96.7	2.9	0.0	2.9
8	80.9	33.8	30.8	24.0	45.4	55.0	72.0	50.0	99.6	0.0	0.0	0.0
16	84.5	43.4	57.0	30.3	56.7	58.7	65.3	61.3	100.0	0.0	0.0	0.0

Model Size on Performance

Models perform better on high-frequency terms across all model sizes



(a) Arithmetic-Multiplication



(b) Arithmetic-Addition

✍: Vicky Zeng, Neha Verma, Lingfeng Shen

Overview of the paper

present analysis on these numerical reasoning tasks for three sizes of the EleutherAI/GPT models pretrained

show a consistently large performance gap between highest-frequency terms and lowest-frequency terms in all of our experiments

Call for a revisit evaluation of language models with respect to their pretraining data on numerical reasoning.

: Vicky Zeng, Neha Verma, Lingfeng Shen

Strengths

- Stress the importance of pre-training dataset
- Experiment setup is intuitive
- Consistent results over 11 tasks (arithmetic, operational, time unit conversion)
- Reproducible methods and code
- Does well in tying work to related research

Weakness

- Limited by numerical reasoning tasks
- Hard to explain the performance gap:
  - Does the gap come from memorization?
  - Other confounders?
    - Sentence length
    - Context (numbers occurs in arithmetic context during training?)
    - ...

What Makes Good In-Context Examples for GPT-3?

Jiachang Liu[1]*, **Dinghan Shen**[2], **Yizhe Zhang**[3], **Bill Dolan**[3], **Lawrence Carin**[1], **Weizhu Chen**[2]
[1]Duke University [2]Microsoft Dynamics 365 AI [3]Microsoft Research
[1]{jiachang.liu, lcarin}@duke.edu
[2,3]{dishen, yizzhang, billdol, wzchen}@microsoft.com

Opportunities and Challenges for GPT-3

- In-context learning:
  - No need to fine-tuned the model or store checkpoints
  - Can be applied to many NLP tasks with a few examples
- However, performance of GPT-3 tends to fluctuate with different choices of in-context examples

Trial	1	2	3	4	5
Accuracy	94.6	95.0	95.8	93.9	86.9

Five different examples are randomly selected from the SST-2 training set for each trial. Different contexts induce different accuracies on the test set.

- Our work aims to carefully examine this sensitivity issue and improve GPT-3's performance without fine-tuning.

KATE:

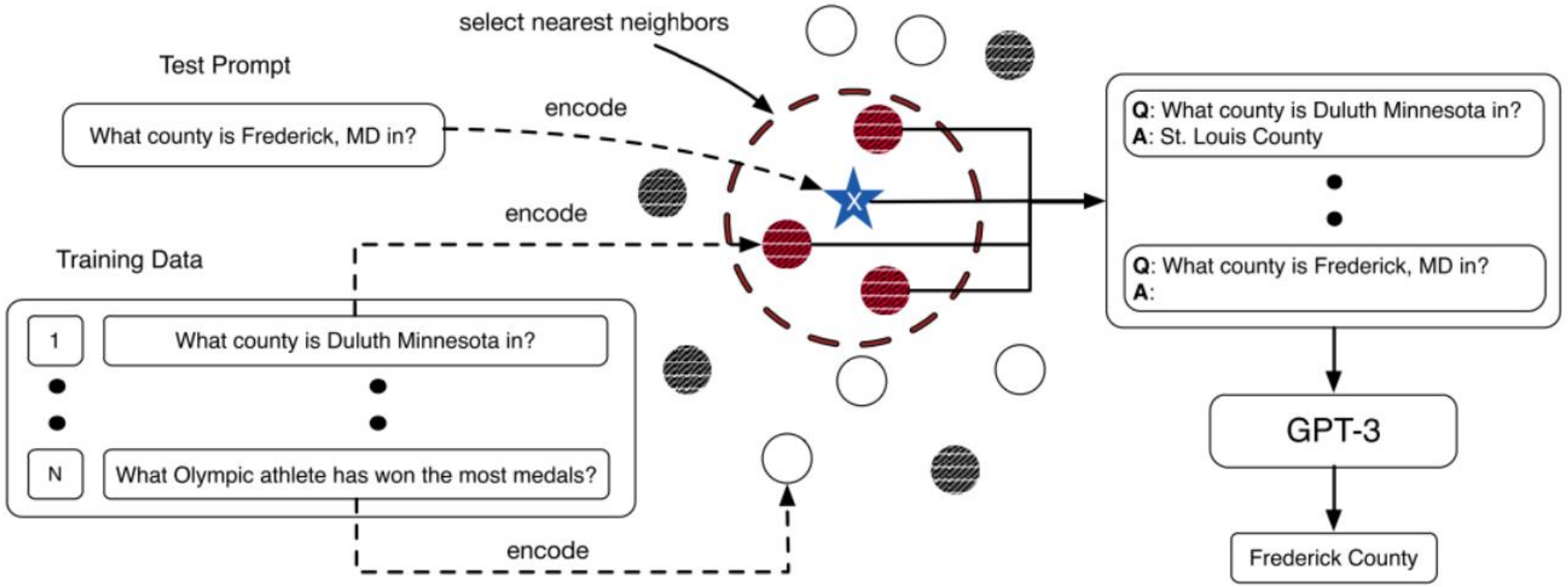


Figure 2: In-context example selection for GPT-3. White dots: unused training samples; grey dots: randomly sampled training samples; red dots: training samples selected by the k-nearest neighbors algorithm in the embedding space of a sentence encoder.

Choices of Retrieval Module

A core step for our context selection approach is mapping sentences into a latent semantic space

- pretrained sentence encoders such as a pre-trained BERT, RoBERTa, or XLNet models

  The original pre-trained RoBERTa-large model (Liu et al., 2019), which is abbreviated as KATEroberta

- sentence encoders fine-tuned on specific tasks or datasets

  i) fine-tuned on the SNLI and MultiNLI dataset (KATEnli);

  ii) first fine-tuned on the SNLI and MultiNLI dataset and then on the STS-B dataset (KATEnli+sts-b).

Baseline Methods

Random Sampling

randomly select in-context examples from the training set

K- Nearest Neighbor

To ensure fair comparison, we compare the baseline kNN and KATE under the same embedding space of a pre-trained RoBERTa-large model. This baseline is abbreviated as kNNroberta

Experimental Results-Sentiment Analysis

For sentiment classification, we select in-context examples under the transfer setting, where one dataset is treated as the training set and the evaluation is made on another dataset.

Method	Accuracy
Random	87.95 ± 2.74
$k\text{NN}_{\text{roberta}}$	50.20
$\text{KATE}_{\text{roberta}}$	91.99
KATE_{nli}	90.40
$\text{KATE}_{\text{nli+sts-b}}$	90.20
$\text{KATE}_{\text{sst-2}}$	**93.43**

Table 4: Accuracy of sentiment prediction for GPT-3 on IMDB with different choices of in-context examples. In-context examples are from the SST-2 dataset.

Experimental Results-Table-to-Text Generation

Test Table	**Table**: <page_title>Trey Johnson <section_title>College <table><cell>32 <col_header> GP <cell>4.8 <col_header>RPG <cell>2.3 <col_header>APG <cell>23.5 <col_header>PPG
Retrieved Examples	**Table**: <page_title>Dedric Lawson <section_title>College <table><cell>9.9 <col_header> RPG <cell>3.3 <col_header>APG <cell>19.2 <col_header>PPG **Sentence**: Dedric Lawson averaged 19.2 points, 9.9 rebounds and 3.3 assists per game. **Table**: <page_title>Carsen Edwards <section_title>College <table><cell>3.8 <col_header> RPG <cell>2.8 <col_header>APG <cell>18.5 <col_header>PPG **Sentence**: Edwards averaged 18.5 points, 3.8 rebounds and 2.8 assists per game.
Predictions	**Ground-truth**: Trey Johnson averaged 23.5 points, 4.8 rebounds, and 2.3 assists in 32 games. **Random**: Trey Johnson averaged 23.5 points per game in his senior year at the University of Texas. **KATE**: Johnson averaged 23.5 points, 4.8 rebounds and 2.3 assists per game.

Method	**Overall**		**Overlap Subset**		**Nonoverlap Subset**	
	BLEU	**PARENT**	**BLEU**	**PARENT**	**BLEU**	**PARENT**
Random	28.4 ± 2.1	39.3 ± 2.6	31.2 ± 2.5	41.8 ± 3.0	25.6 ± 1.8	37.0 ± 2.3
kNN$_{\text{roberta}}$	14.1	12.6	20.1	17.9	8.0	7.52
KATE$_{\text{roberta}}$	**40.3**	**49.7**	**47.8**	**55.0**	**32.9**	**44.6**
KATE$_{\text{nli}}$	39.1	48.5	46.5	53.7	31.9	43.6
KATE$_{\text{nli+sts-b}}$	38.1	47.2	45.2	52.2	31.1	42.4

Table 5: Table-to-text generation results on the ToTTo dev dataset.

Experimental Results-Questing Answering

use the **Exact Match (EM)** score to measure the performance of GPT3 on open-domain QA tasks.

Method	NQ	WQ	TriviaQA*
RAG (Open-Domain)	44.5	45.5	68.0
T5+SSM (Closed-Book)	36.6	44.7	60.5
T5 (Closed-Book)	34.5	37.4	50.1
GPT-3 (64 examples)	29.9	41.5	-
Ours			
Random	28.6 ± 0.3	41.0 ± 0.5	59.2 ± 0.4
$k\text{NN}_{\text{roberta}}$	24.0	23.9	26.2
$\text{KATE}_{\text{roberta}}$	40.0	47.7	57.5
KATE_{nli}	40.8	**50.6**	60.9
$\text{KATE}_{\text{nli+sts-b}}$	**41.6**	50.2	**62.4**

Table 7: QA results on various datasets. (*) On TriviaQA, we used 10 examples. On NQ and WQ, we used 64 examples.

Analysis and Ablation Study

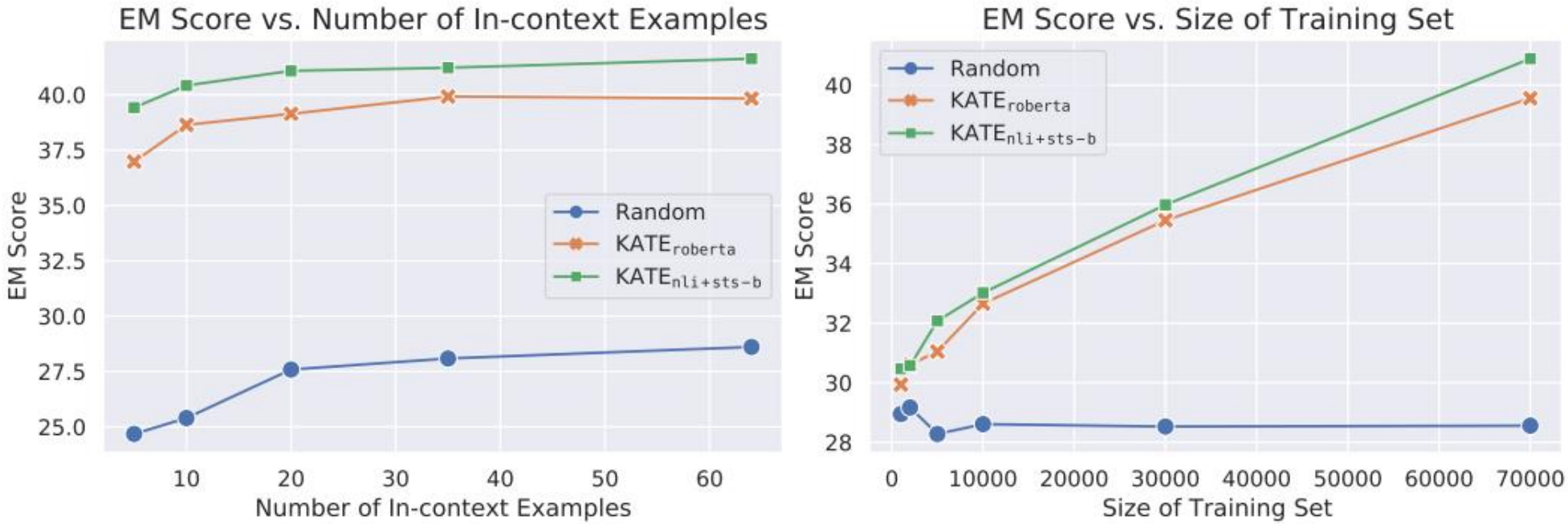


Figure 3: Left: Ablation study on the effect of number of in-context examples for GPT-3 for different selection methods. Right: Ablation study on the effect of the size of training set for retrieval on KATE. Two representative sentence encoders are used in the ablation study.

Analysis and Ablation Study

Trial	1	2	3	Default	Reverse
EM Score	42.0	42.5	42.0	41.6	42.8

Table 9: Ablation study on the effect of in-context example orders for GPT-3 on the NQ dataset using $\text{KATE}_{\text{nli+sts-b}}$. For the default order, the example A is to the left of example B if A is closer to the test prompt x than B in the embedding space. For the reverse order, the example A is to the right of example B.

However, we also did the experiments on the WQ and TriviaQA and find that the default order performs slightly better than the reverse order. Hence, the choice of orders is data-dependent.

The example order does not have a significant impact on KATE's performance.

Do Prompt-Based Models Really Understand the Meaning of Their Prompts?

Albert Webson[1,2] and Ellie Pavlick[1]

{albert_webson, ellie_pavlick}@brown.edu
[1]Department of Computer Science, Brown University
[2]Department of Philosophy, Brown University

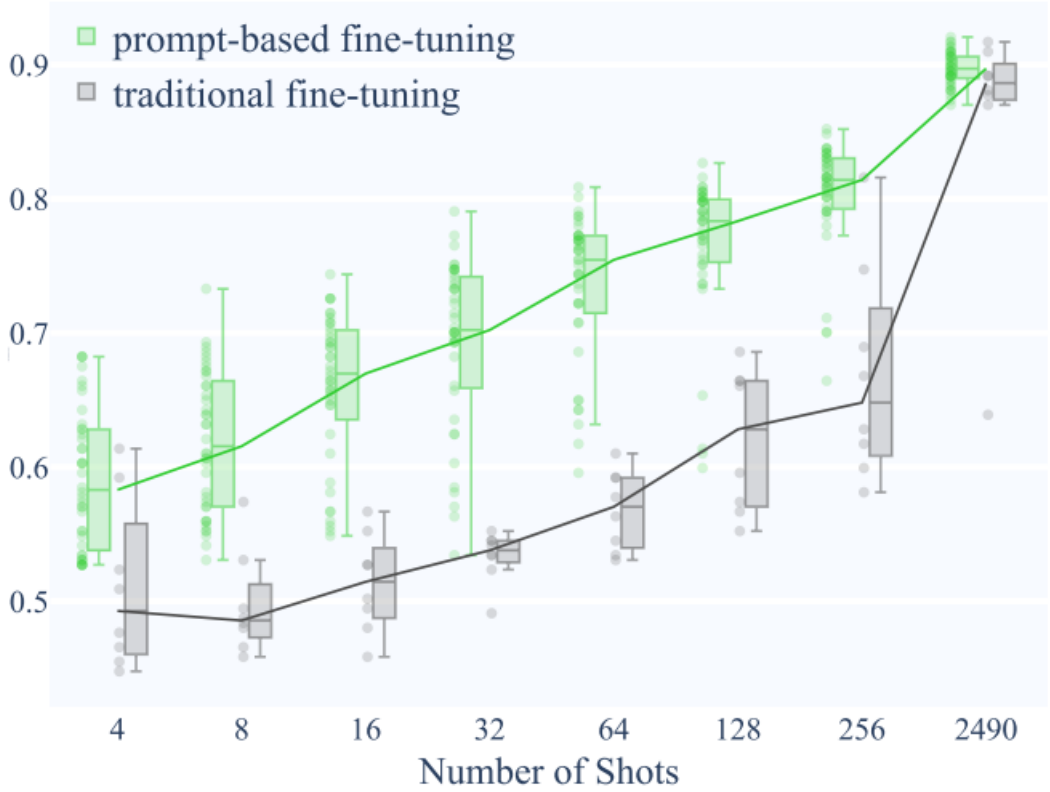
Suppose a human is given a prompt such as: Given that “no weapons of mass destruction found in Iraq yet.”, is it definitely correct that “weapons of mass destruction found in Iraq.”?

Overall Setup

Baseline Model - ALBERT

Instruction-Tuned Model - T0(3B and 11B) and T5 LM-Adapted (their noninstruction-tuned version)

Very Large Model - GTP-3



Prompt = Template + Target

Category	Examples
instructive	{prem} Are we justified in saying that "{hypo}"? Suppose {prem} Can we infer that "{hypo}"?
misleading-moderate	{prem} Can that be paraphrased as: "{hypo}"? {prem} Are there lots of similar words in "{hypo}"?
misleading-extreme	{prem} is the sentiment positive? {hypo} {prem} is this a sports news? {hypo}
irrelevant	{prem} If bonito flakes boil more than a few seconds the stock becomes too strong. "{hypo}"?
null	{premise} {hypothesis} {hypothesis} {premise}

Table 1: Example templates for NLI.

Category	Target Words
yes-no	yes;no
yes-no-like	true;false
yes-no-like	positive;negative
yes-no-like	right;wrong
yes-no-like	correct;incorrect
yes-no-like	agree;disagree
yes-no-like	good;bad
reversed	no;yes
reversed	false;true
reversed	negative;positive
arbitrary	B;C
arbitrary	cat;dog
arbitrary	she;he

Result - Irrelevant Templates & Misleading Templates

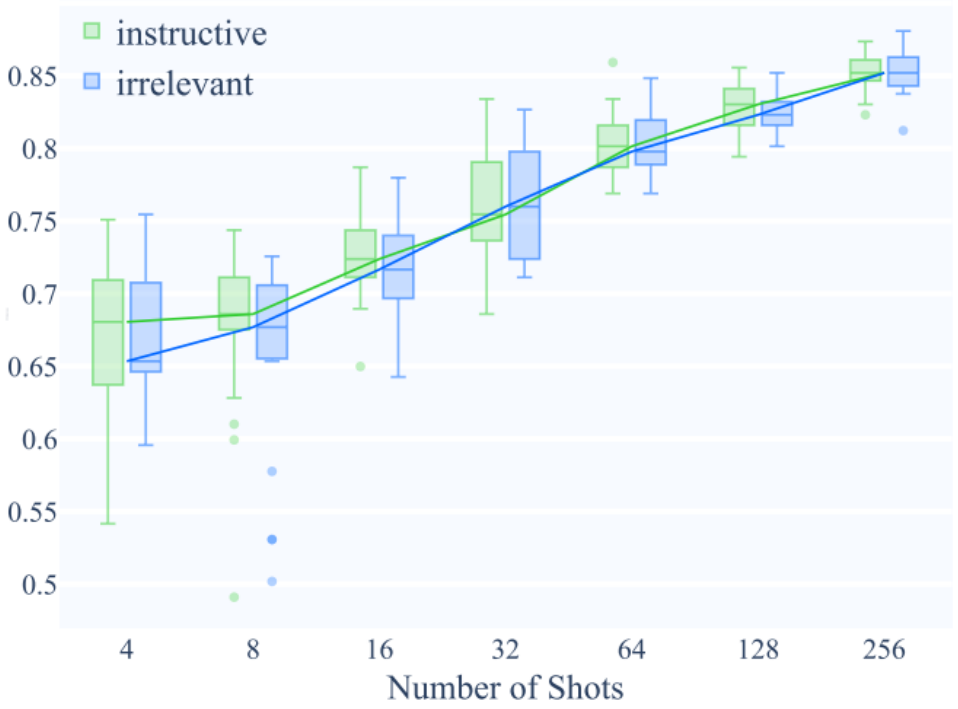


Figure 2: T0 (3B) on RTE. There is no practical difference between the performance of the models trained with instructive templates vs. those trained with irrelevant templates at any number of shots.

Figure 3: T0 (3B) on RTE. There is no practical difference between models trained with instructive and misleading-moderate templates at any number of shots. But models trained with misleading-far templates are statistically significantly worse from 8 to 128 shots.

Result – Null Templates & Zero shot

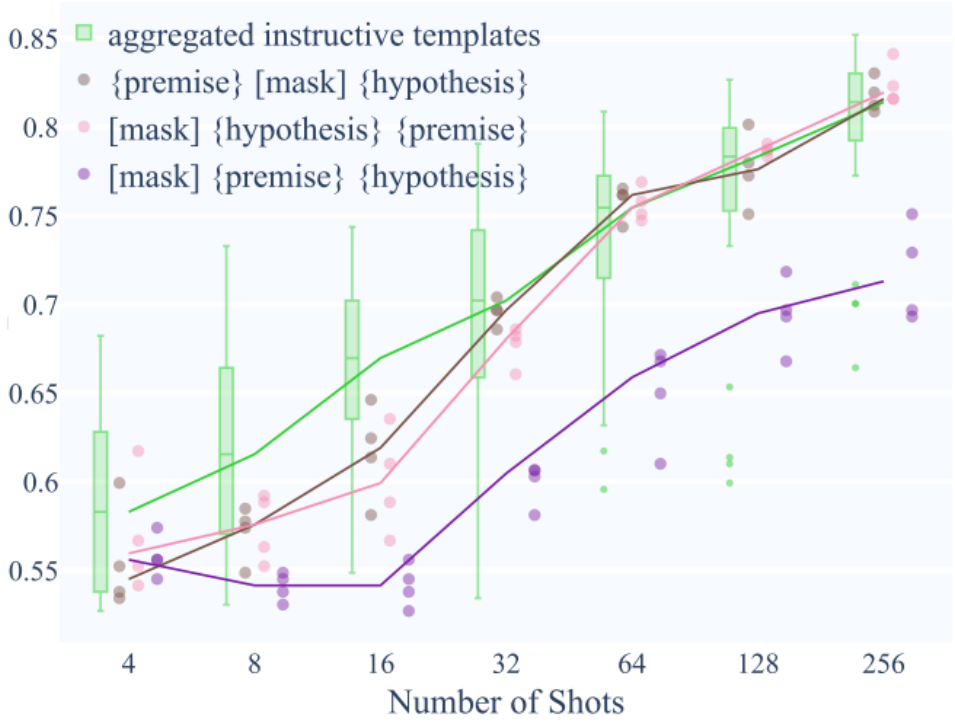


Figure 4: ALBERT on RTE. After 32 shots, models trained with 2 null templates learn just as fast as the instructive templates, but models trained with other null templates (e.g., purple) are much worse.

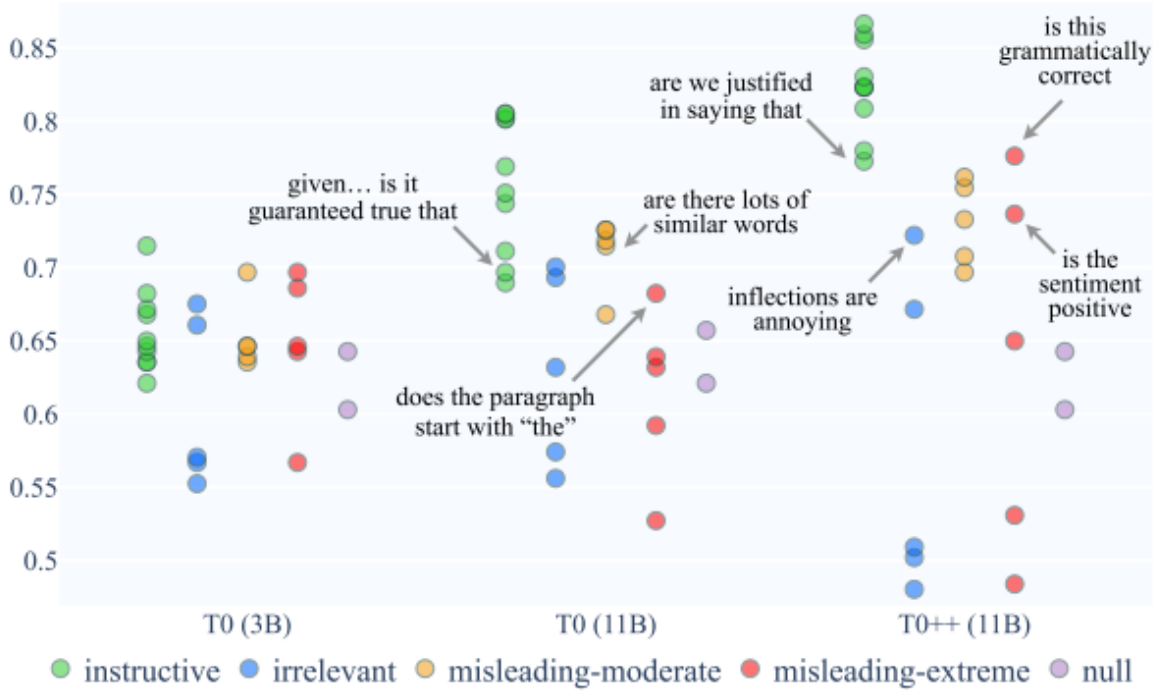
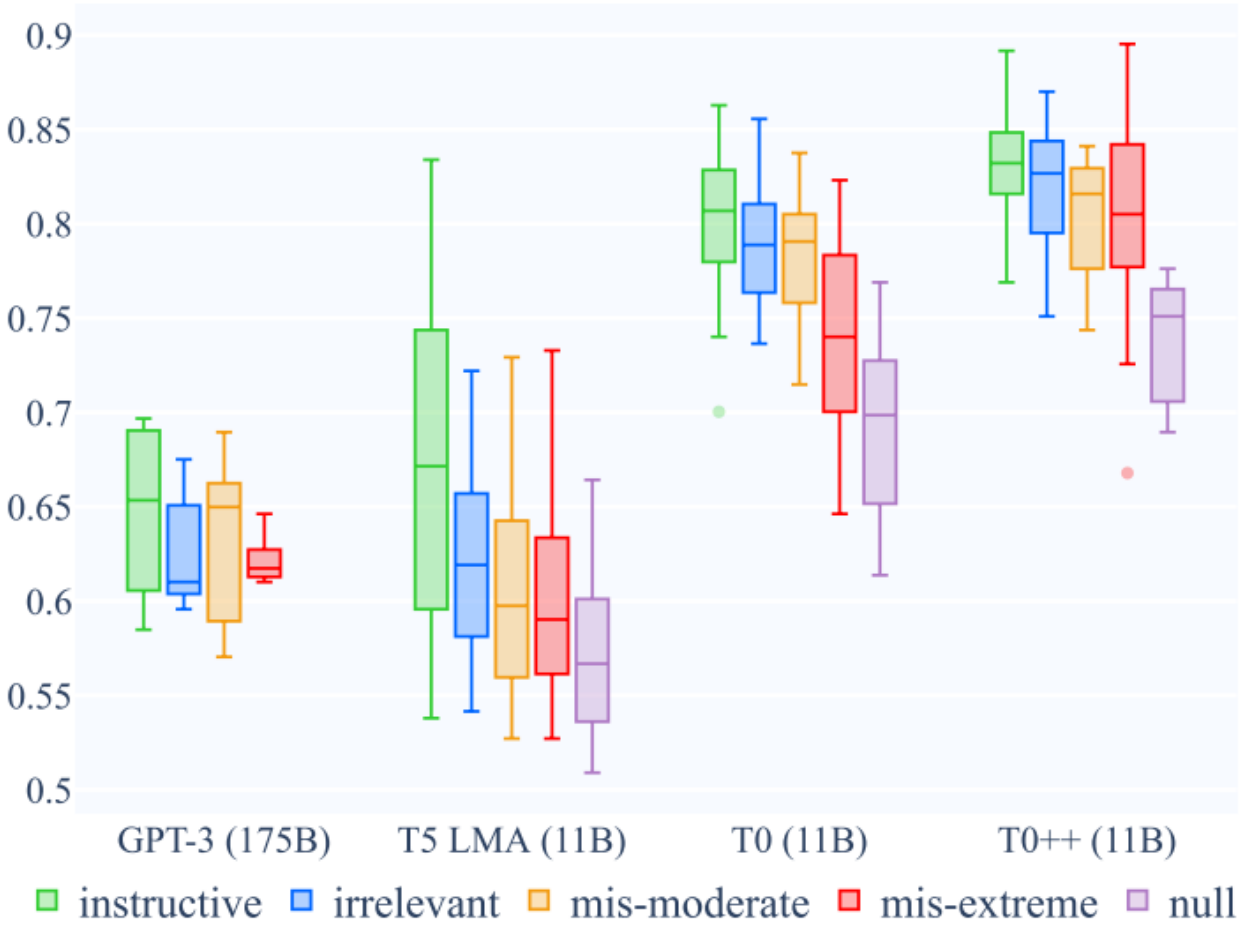


Figure 5: Zero-shot accuracy of instruction-tuned models on RTE. Each prompt's performance is a single point (unlike the few-shot figures where each prompt is approximated by multiple points with multiple samplings of few-shot examples.) Arrows highlight some prompts with their excerpts. See Appendix I for the full results.

Discussion


0.9
0.85
0.8
0.75
0.7
0.65
0.6
0.55
0.5
GPT-3 (175B)
T5 LMA (11B)
T0 (11B)
T0++ (11B)
instructive
irrelevant
mis-moderate
mis-extreme
null

Effect of Target Words

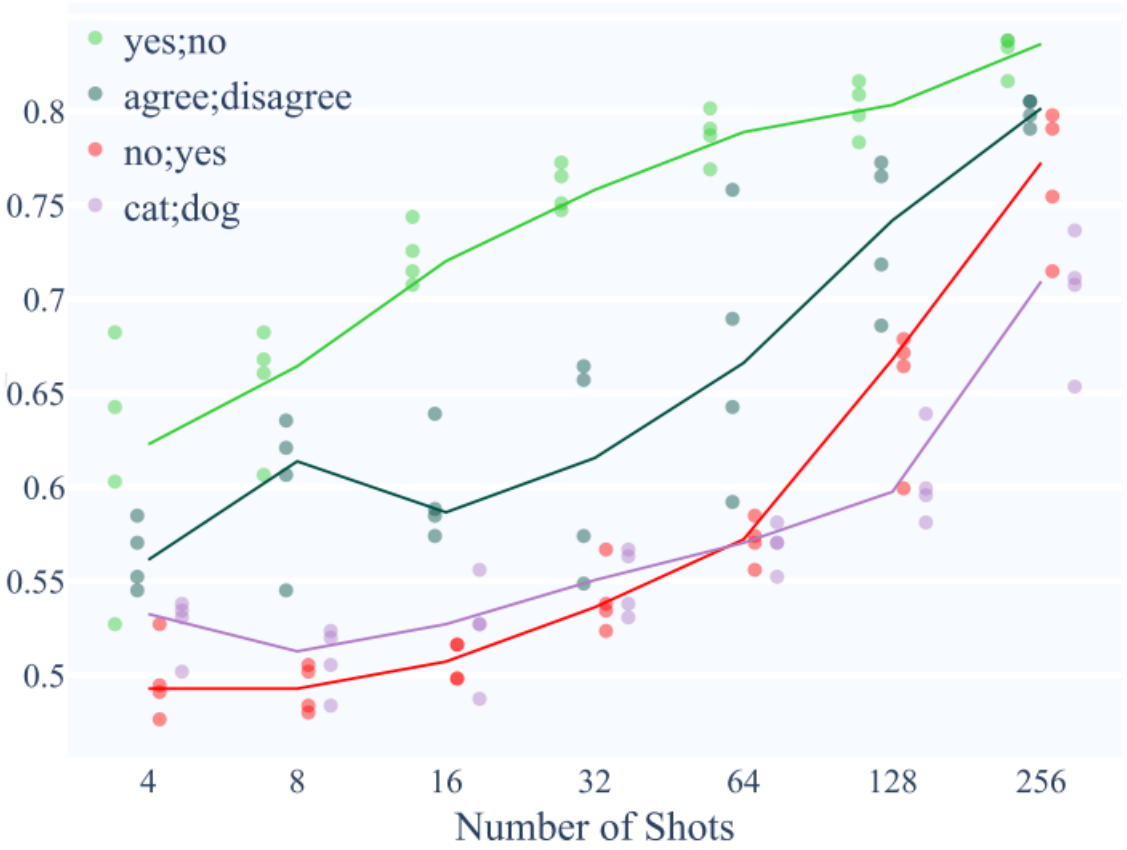


Figure 7: The best-performing instructive template for ALBERT on RTE, `{prem} Are we justified in saying that "{hypo}"?` with select LM targets from each category.

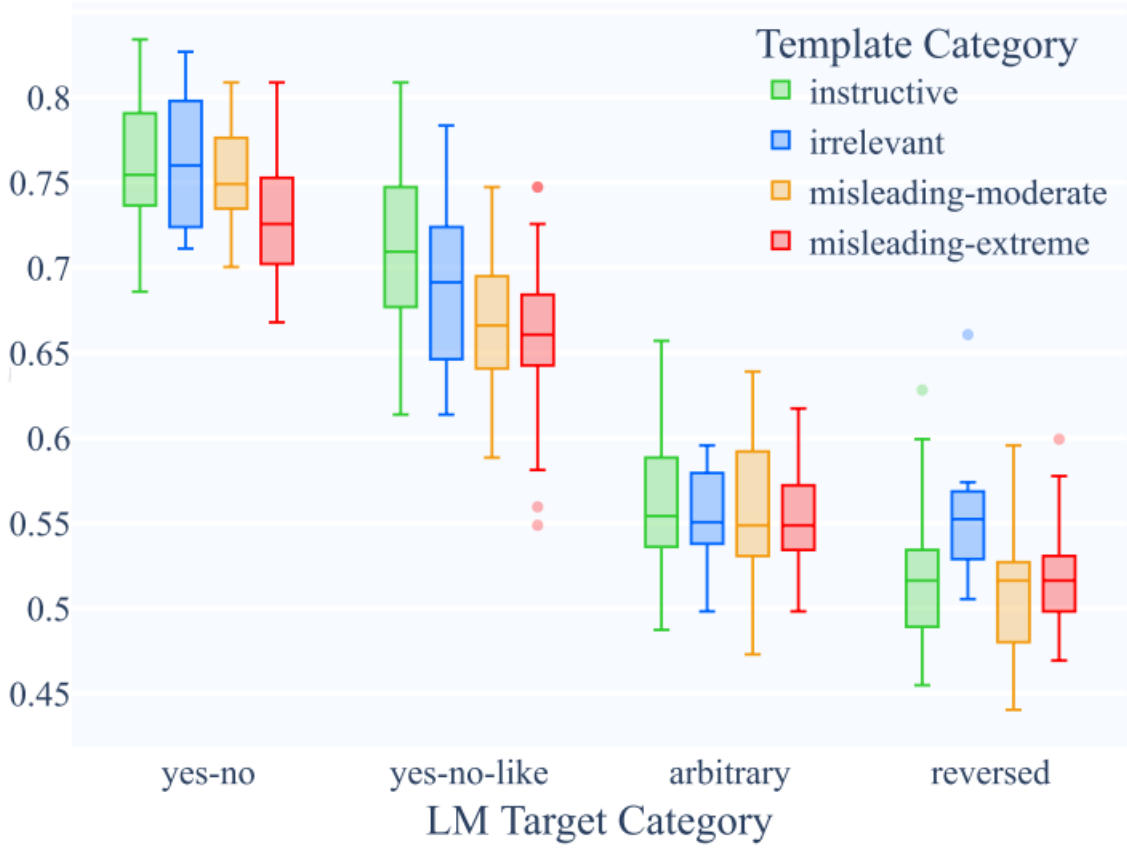


Figure 8: T0 (3B)'s 32-shot accuracy with of all template-target combinations on RTE. In general, the choice of target words (x-axis groups) matters much more than the choice of templates (colors).

Discussion

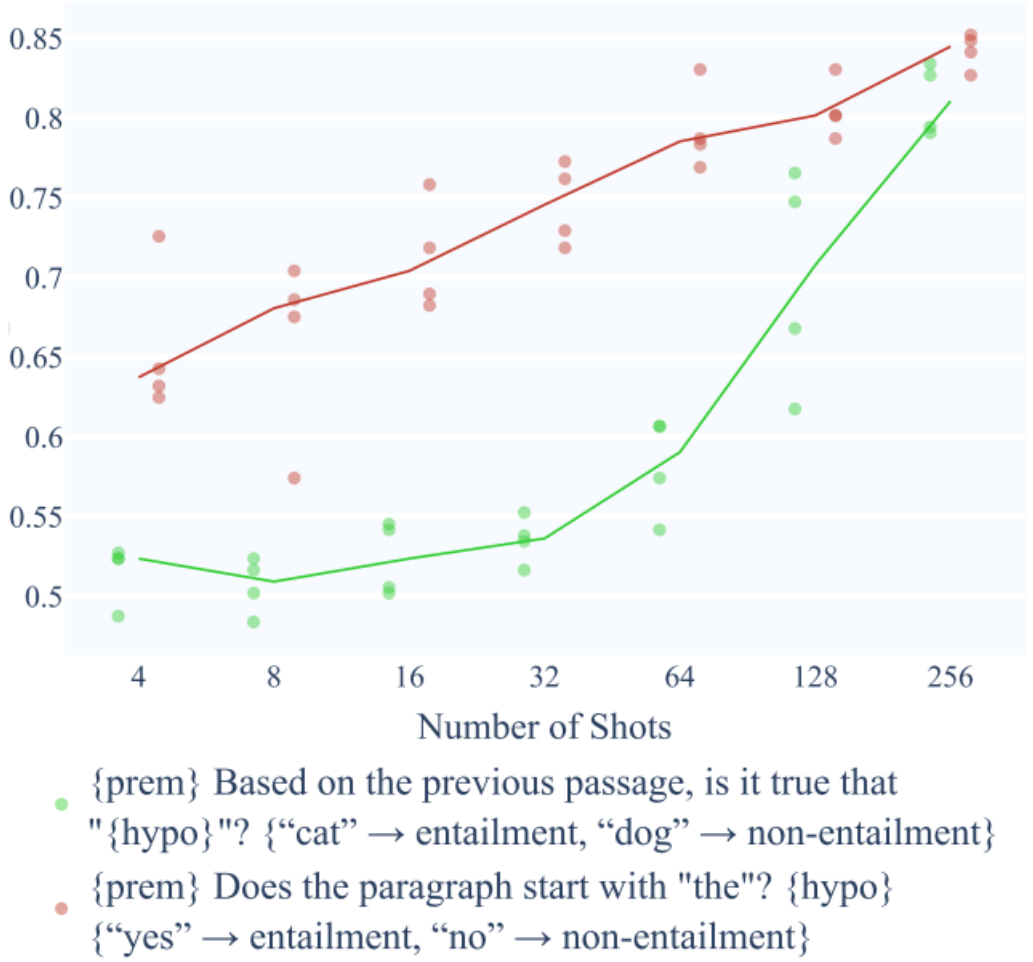


Figure 10: T0 (3B) on RTE. Misleading templates + yes-no targets (red) learn substantially faster than instructive templates + arbitrary targets (green), which is the opposite of what we expect from humans.

Effect of Punctuation

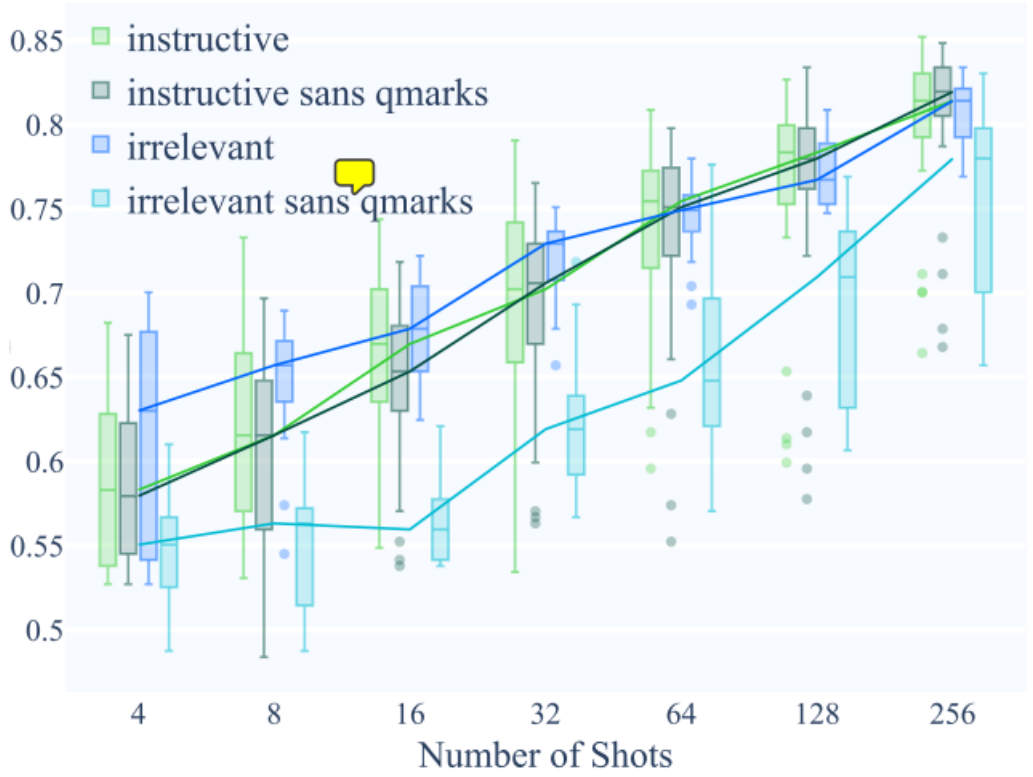


Figure 11: ALBERT on RTE. Note that (1) irrelevant templates slightly outperform the instructive templates, albeit without statistical significance. (2) Irrelevant templates are far worse without quotation and question marks. (3) But there is no significant difference between instructive templates with or without qmarks.

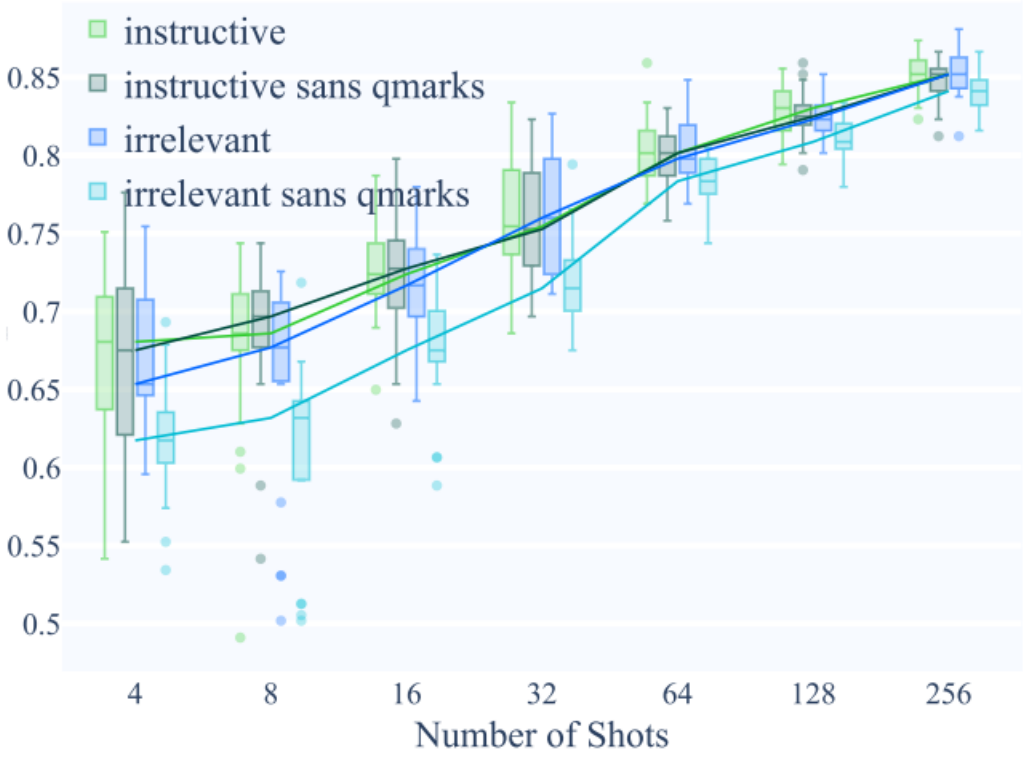


Figure 12: T0 (3B) on RTE. Like ALBERT, irrelevant sans qmarks are significantly worse than irrelevant at each and every shot, but there is no significant difference between instructive with or without qmarks.

Thanks