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

Abstract

Recently, a boom of papers have shown extraordinary progress in few-shot learning with various prompt-based models. Such success can give the impression that prompts help models to learn faster in the same way that humans learn faster when provided with task instructions expressed in natural language. In this study, we experiment with over 30 prompts manually written for natural language inference (NLI). We find that models learn just as fast with many prompts that are intentionally irrelevant or even pathologically misleading as they do with instructively "good" prompts. Additionally, we find that model performance is more dependent on the choice of the LM target words (a.k.a. the "verbalizer" that converts LM vocabulary predictions to class labels) than on the text of the prompt itself. In sum, we find little evidence that suggests existing prompt-based models truly understand the meaning of their given prompts.

1 Introduction

Suppose a human is given two sentences: "No weapons of mass destruction found in Iraq yet." and "Weapons of mass destruction found in Iraq." They are then asked to respond 0 or 1 and receive a reward if they are correct. In this setup, they would likely need a large number of trials and errors before figuring out what they are really being rewarded to do. This setup is akin to the pretrain-and-fine-tune setup which has dominated NLP in recent years, in which models are asked to classify a sentence representation (e.g., a CLS token) into some arbitrary dimensions of a one-hot vector. In contrast, 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."?*[1] Then it would be no surprise that they are able to perform the task more accurately and without needing many examples to figure out what the task is.

Similarly, reformatting NLP tasks with prompts such as the underlined text above has dramatically improved few-shot learning performance over traditionally fine-tuned models (Schick and Schütze, 2021b; Le Scao and Rush, 2021). Such results naturally give rise to the hypothesis that the extra prompt text included within each input example serves as semantically meaningful task instructions which help models to learn faster, in the way task instructions help humans to learn faster. This hypothesis is implicitly assumed by many and explicitly argued by Mishra et al. (2021), Schick and Schütze (2021a), and Brown et al. (2020).

While last year saw a gold rush of papers (summarized in §2) that proposed automatic methods of optimizing prompts, comparing a representative sample of the newly proposed methods, Logan et al. (2021) reported that Schick and Schütze (2021b)'s manually written prompts still on average outperform the automatically searched prompts across a range of SuperGLUE tasks (Wang et al., 2019). Such findings suggest that expert-crafted prompts are among the best, if not *the* best, which reinforces the above hypothesis that models benefit from meaningful instructions.

In this paper, we test this hypothesis by training a masked language model on RTE (Dagan et al., 2006) in a few-shot setting, using more than 30 manually written templates and 10 sets of LM target words for a total of over 300 prompts. We find that, in many cases, models learn identically as fast when given irrelevant, misleading, or even empty (null) templates as they do when given instructively good templates (§4). Additionally, models are overly sensitive to the choice of the LM targets: The mapping {"yes" → entailment, "no" →

[1]This prompt is adapted from MultiNLI (Williams et al., 2018, p. 3)'s instructions to crowdsourced workers, while the example is the first one in RTE's validation set. We italicize prompts to avoid ambiguity of nested quotes.

non-entailment} substantially outperforms all other word-to-label mappings, even when "yes"/"no" is not syntactically or semantically well-formed with the overall prompt (§5). In sum, despite prompt-based models' impressive improvement in few shot-learning, we find little evidence that such improvement is derived from models understanding task instructions in a way that is analogous to humans' use of task instructions. Although exisiting models are far from this goal, we nonetheless agree that learning from instructions is an important research direction, and discuss future directions of investigating models' understanding of the meaning of prompts (§6).

2 Related Work

2.1 Prompt-Based Models

At the time of writing, the term "prompt-based" is used to refer to any one or combination of roughly three general approaches described below:

Discrete Prompts reformat each example with some template text. For example, in a sentiment analysis task, the template can be *{sent} In summary, the restaurant is [prediction]*, where the predicted mask word is then converted to a class prediction by a predefined mapping, e.g., {"great" $\rightarrow$ positive, "terrible" $\rightarrow$ negative}. The prompts can be manually written (Schick and Schütze, 2021a; Bragg et al., 2021) or automatically generated (e.g., Gao et al., 2021; Shin et al., 2020). This approach typically tunes all parameters of the model, but its few-shot performance can exceed that of GPT-3 despite using a 3 orders of magnitude smaller LM (Schick and Schütze, 2021b; Tam et al., 2021).

Priming (a.k.a. in-context learning) prepends k priming examples to the evaluation example, where each example is wrapped in a minimal prompt such as *Question: {sent$_1$} True or false? {label$_1$}* ... *Question: {sent$_k$} True or false? {label$_k$} Question: {eval_sent} True or false? [prediction]*". Notably, although models see labeled examples, their parameters do not receive gradient updates based on those examples. While this approach is compelling, Brown et al. (2020) reported that it only performs well on the largest GPT-3 model, which is not widely available for academic research.

Continuous Prompts prepend or append examples with special tokens optionally initialized with word embeddings, but during learning, those tokens can be updated arbitrarily such that the final embeddings do not necessarily correspond to any real word in the vocabulary (e.g., Lester et al., 2021; Li and Liang, 2021; Qin and Eisner, 2021; see Liu et al., 2021 for a survey.) This approach often efficiently tunes a much smaller set of model parameters, but these methods have not yet reported success in few-shot settings. Moreover, foregoing prompts as expressed in natural language makes it much harder to study their semantics. Thus, it is not clear if continous prompts serve as task-specific instructions or simply more model parameters.

2.2 Analyses of Prompts

In this paper, we focus on discrete prompts because we can manually write and control their wording and semantics. We measure the effect of prompt semantics by the model's k-shot performance where $k = \{3, 5, 10, 20, 30, 50, 100, 250\}$. This setup resembles that of Le Scao and Rush (2021), but their study focuses on comparing Schick and Schütze (2021b)'s existing small set of prompts against traditional fine-tuning over the training trajectories of entire training sets, whereas our study focuses on the few-shot learning trajectories among a much more diverse set of prompts designed to test specific hypotheses about the effect of prompt semantics on few-shot learning speed.

At a high-level, our findings contradict Mishra et al. (2021)'s claim that models benefit from elaborate instructions adapted from crowdsourcing annotation guides. But note that they define "instructions" more broadly as including priming examples, and they found that "GPT-3 benefits the most from positive examples, mildly from definition, and deteriorates with negative examples." (p. 10). In other words, if we ablate priming and narrow "instructions" to just the description and explanation of a task, we actually have the same finding that instructions are only moderately beneficial over no instructions (our null prompts in §4). However, we further show that good instructions have almost no benefit over irrelevant instructions, thus raising questions of whether models' use of prompts can fairly be described as "understanding".

2.3 Cloze-Style Diagnoses of LMs

Before prompts were adopted for downstream tasks in general, directly using LM's word prediction without fine-tuning was first used for knowledge-base retrieval (Petroni et al., 2019; Jiang et al., 2020) as well as knowledge-related analyses of LMs (e.g., Ettinger, 2020; Kassner et al., 2020; Do and Pavlick, 2021; Carlini et al., 2020). Such work has revealed valuable insights on using LMs in the style of cloze. For example, Ettinger (2020) demonstrated LMs' near complete inability to respond correctly to negation (e.g., "a robin is not a [mask]" where BERT predicts "bird"). Kassner et al. (2020) showed that models lack various symbolic reasoning abilities such as asymmetry, inversion, and composition (e.g., "sun bigger than earth" $\rightarrow$ "earth bigger than [mask]" where BERT predicts "sun".) Do and Pavlick (2021) showed that models especially lack commonsense reasoning when infrequent lexical association is involved even though they are easy for humans (e.g., "Aaron says that the apple is rotten. The apple is [mask]." where BERT predicts "edible" with higher probability than "inedible".) All three works concluded that LMs' predictions heavily rely on memorizing patterns seen in pretraining.

A subset of these findings has been already been reproduced using newer prompt methods. For example, Mishra et al. (2021) found that models cannot understand negation in the form of instructions which specify "things to avoid": providing negative examples, albeit helpful for humans, significantly hurts model performance. Meanwhile, Cao et al. (2021) found that KB prompts selected based on the LAMA dataset (which features many London-born people; Petroni et al., 2019) would still predict that most people are born in London even on a novel dataset with different distributions of birthplaces, thus calling into question whether prompts really represent instructions such as "retrieve the birthplace of x" or rather, a fixed set of words that commonly co-occur with "x was born in".

3 Experimental Setup

We test the hypothesis that models understand prompts as meaningful task instructions by comparing good prompts to different types of bad prompts and observing the effects on their few-shot learning speed. We use "prompt" to mean a unique combination of a template and a predefined LM target for each class label. For example, {"must" $\rightarrow$ entailment, "might" $\rightarrow$ non-entailment} are the default targets for the template *{premise} It [mask] be true that {hypothesis}*.

Data Recognizing Textual Entailment (RTE, Dagan et al., 2006, inter alios) is a series of expert-annotated NLI datasets where a model is asked to classify whether one piece of text (the "premise") entails another (the "hypothesis"). Specifically, we use the SuperGLUE collection of RTE (i.e., RTE1, 2, 3, and 5; all converted to binary classification) for comparability with prior work on prompts.

Model We implement a discrete prompt model which in essence is the same as that of Schick and Schütze (2021b), except their implementation includes several augmentations such as self-labeling and ensembling of multiple prompts for competitive results. In order to focus on measuring the effect of prompts themselves, our implementation does not include those augmentations. In preliminary experiments, we fine-tuned and prompt-tuned our model with bert-large-uncased, distilbert-base-uncased, roberta-large, and albert-xxlarge-v2 (Devlin et al., 2018; Sanh et al., 2019; Liu et al., 2019; Lan et al., 2019; all provided by Wolf et al., 2020). Like prior work (Schick and Schütze, 2021b; Tam et al., 2021), we find albert-xxlarge-v2 consistently yields the best performance, so we use it for our main experiments. All experiments use the same hyperparameters documented in our public GitHub repository.[2]

Random Seeds & Example Sampling All experiments are run over the same set of 8 random seeds. Within a given seed, all models see the same set of examples. For instance, under seed 1, the 5-shot models see examples 550–555, the 10-shot models see examples 550–560, and so on. Across different seeds, a different starting example index is drawn. The exact training example indices are also recorded in our GitHub repository for reproducibility.

Comparison to Prior Work To verify that our implementation is comparable with prior work, Figure 1 reports the RTE validation accuracy of our model at different numbers of shots. At 30 shots, our minimal manual prompt model reaches a median accuracy of 70.76% (mean = 68.8%, stddev = 5.9%), which is largely comparable to the 32-shot performance reported by Schick and Schütze (2021b) (69.8%) and Gao et al. (2021) (73.9% $\pm$ 2.2%). Further, Figure 1 confirms Le Scao and Rush (2021)'s finding that, while both fine-tuning and prompt-tuning converge to the same results when fully trained on the entire set ($n = 2490$ for RTE), prompt-tuning yields dramatic improvement in few-shot settings. Going forward, we focus on studying the few-shot learning trajectory between 5 and 100 examples.

[2] https://github.com/awebson/prompt_semantics

Template	Default LM Targets	Template Category
{premise} Are we justified in saying that "{hypothesis}"? [mask]	yes/no	instructive
{premise} It [mask] be true that {hypothesis}	must/might	instructive
{premise} Can that be paraphrased as: {hypothesis}? [mask]	yes/no	misleading
{hypothesis} Do most of the above words appear in the following passage? [mask] "{premise}"	yes/no	misleading
{premise} Inflections are annoying and thank god that Middle English got rid of most of them. {hypothesis}? [mask]	yes/no	irrelevant
{premise} [mask] {hypothesis}	yes/no	null

Table 1: Example templates from each category. See Appendix A for the full list.

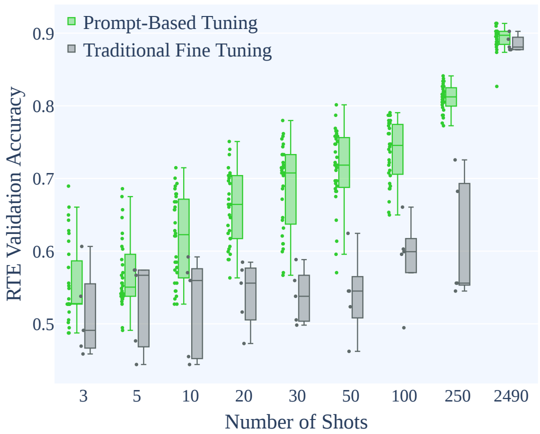

Figure 1: Comparison of few-shot learning speed between traditional fine tuning and prompt tuning on RTE. Boxes span from the first quartile to the third quartile, while lines inside boxes mark the medians.

4 Effect of Template Semantics on Few-Shot Learning

4.1 Method

In this experiment, we study how template text affects few-shot performance, controlling for the LM target words. We manually write 4 categories of templates (see Table 1 for examples):

- Instructive: How we would describe the NLI task to a human who has never seen the task before.
- Irrelevant: Concatenate the premise, a sentence entirely unrelated to NLI, and the hypothesis.
- Misleading: Instruct the models to perform a task (related or tangential to NLI, e.g., semantic similarity, paraphrasing, summarization, lexical overlap) such that, if the model were to perform the task as explicitly instructed, it would perform poorly on NLI in general.
- Null: A naive concatenation of the premise, the hypothesis, and the mask in different orders without any additional text.

We write at least 5 templates for each category. Each interrogative template (a question sentence as opposed to a declarative one) has "yes"/"no" as its default LM targets, whereas each declarative template has template-specific default targets such as "must"/"might". In this section, because each template performs best with its default targets, a template's performance is always reported with its default targets. The next section studies the differences among different targets controlling for the template.

In preliminary experiments (Appendix A.2), we also write several minimal pairs to test for sensitivity to non-semantic features of the templates. We find no obvious patterns between the following conditions: Whether a template contains question or quotation marks; whether the template is a declarative or an interrogative sentence; whether the mask immediately follows the interrogative sentence; and the overall grammatical acceptability of the template-target combination. However, a comprehensive investigation of spurious features in prompts is outside the scope of this paper, so our experiments on these are not exhaustive.

After preliminary experiments, to avoid cherry picking, all prompts reported in this section were written prior to evaluation, i.e., we do not allow retroactively excluding or including templates from the categories defined above.

4.2 Results

Irrelevant Templates Figure 2 shows the aggregate performance of models trained with instructive templates vs. those trained with irrelevant templates. We find no statistically significant difference[3] between the two settings at any number of

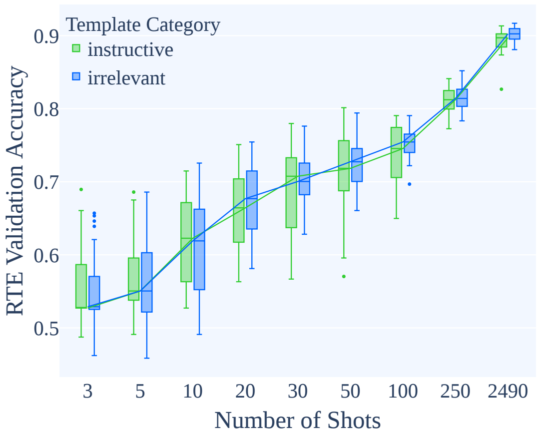

Figure 2: In aggregate, models trained with irrelevant templates learn just as fast as models trained with instructive templates.

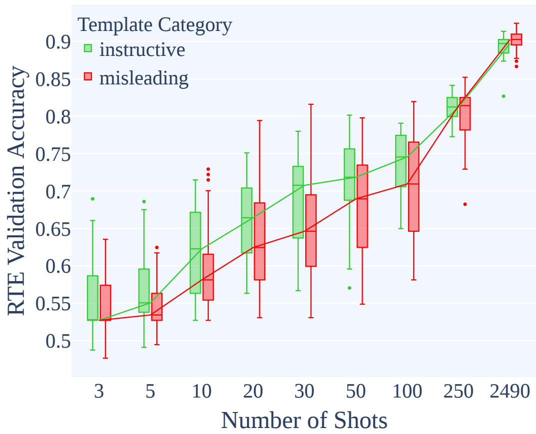

Figure 4: In aggregate, models trained with misleading templates learn slower than models trained with instructive templates.

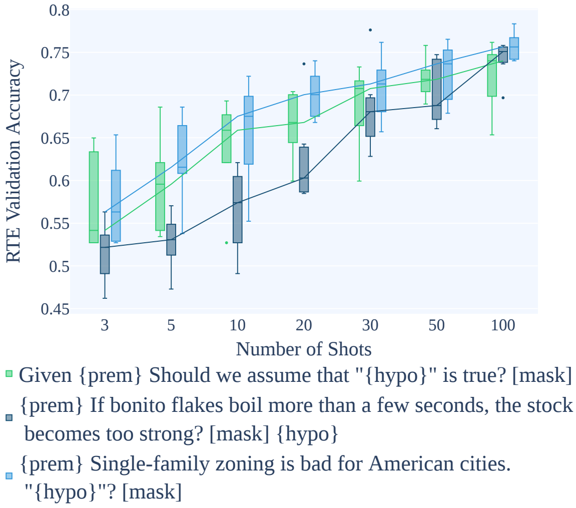

Figure 3: The top-performing irrelevant template (light blue) enables the models to learn faster than an average-performing instructive template (green), while the worst-performing irrelevant template (dark blue) starts slow but catches up by 30 shots.

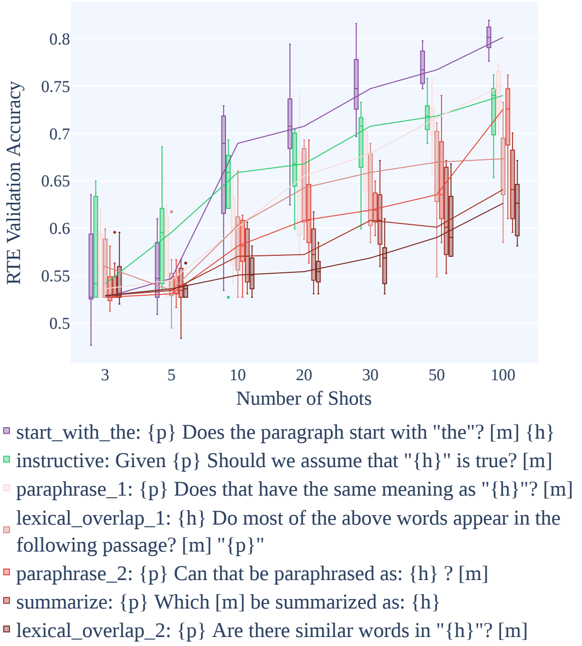

Figure 5: With the exception of one outlier (purple), all misleading templates (shades of red) learn slower than the instructive template (green).

shots. This trend is confirmed when we compare individual templates as well. For example, given the top-performing irrelevant template *{premise} Single-family zoning is bad for American cities. "{hypothesis}"? [yes/no]* (Figure 3, light blue), the model actually learns faster than it does when given an average-performing instructive prompt, *Given {premise} Should we assume that "{hypothesis}" is true? [yes/no]* (Figure 3, green). Even when given the worst-performing irrelevant template, *{premise} If bonito flakes boil for more than a few seconds, the stock becomes too strong? [yes/no] {hypothesis}* (Figure 3, dark blue), its performance largely catches up to the average instructive template by 30 shots, with only a 2% gap in median accuracies.

Misleading Templates While the above results suggest models are not sensitive to the difference between instructive and irrelevant templates, the results on misleading templates tell a different story. In aggregate (Figure 4), models trained with misleading templates do learn slower than those trained with instructive templates. Comparing individual

[3] We use the non-parametric Mann–Whitney–Wilcoxon test when comparing two categories of prompts and the Kruskal–Wallis test coupled with Dunn's post hoc test when comparing among multiple categories. Significance is always computed using models at the same number of shots. We set $\alpha = 0.05$ and use the Bonferroni correction to account for multiple comparisons (i.e., at each number of shots).

templates, however, yields many counterintuitive results: Some models learn faster when instructed to do a less relevant task than a more relevant task. For example, summarization is a task closer to RTE than paraphrasing. Specifically, based on a manual analysis,[4] performing summarization could lead one to reach 60% accuracy on training, while performing paraphrasing would lead one to achieve only 46.67%. However, a paraphrasing judgment prompt (Figure 5, lightest pink) causes the model to learn only modestly slower than the average instructive NLI prompt (green); in contrast, a summarization judgment prompt causes the model to learn dramatically slower than the average instructive prompt.

While most misleading templates make the model perform worse than the instructive ones do, we observe one outlier: A prompt that instructs the model to say whether or not the premise begins with the word "the" (Figure 5, purple), while clearly pathological,[5] is the best performing prompt among all of those in our experiments (see Appendix A for the full results). If this outlier template is removed, then the performance of models trained with misleading templates in aggregate is statistically significantly lower than that of the the instructive ones at each and every shot.

Null Templates Finally, we look at how models learn when given null prompts, i.e., naive concatenations of the premise and the hypothesis with no intervening text. In RTE, a premise that entails a hypothesis rarely has the hypothesis also entailing the premise. However, with null templates, the prompt does not provide models with any information about which sentence is the premise and which is the hypothesis and, when [mask] is not inserted in the middle,[6] does not even provide information about how the input paragraph should be segmented into a premise and a hypothesis. Despite these pathological conditions that would make NLI difficult for humans, models trained with the top-performing null template *{premise} {yes/no} {hypothesis}* (Figure 7, yellow) learn similarly fast

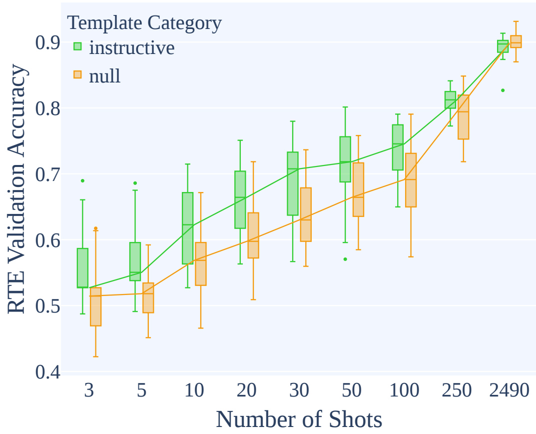

Figure 6: In aggregate, models trained with null templates learn slower than models trained with instructive templates.

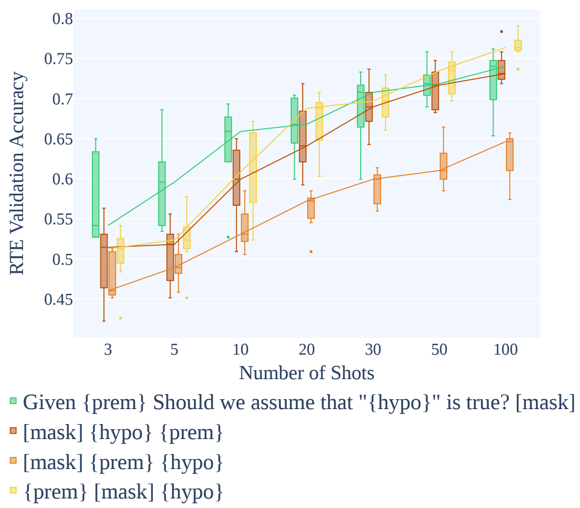

Figure 7: After 10 shots, models trained with 2 null templates (e.g., lemon and maple syrup) learn comparably fast as the average-performing instructive template (mint), but models trained with other 4 null templates (e.g., orange) lag behind. See Figure 14 for a full comparison of all null templates.

to the average instructive template after 10 shots.

Further, note that Schick and Schütze (2021b) and many subsequent papers' prompts for NLI (e.g., *"{hypothesis}" ? | [yes/no]. "{premise}"*) are basically null templates with some variation in punctuation between the hypothesis and the premise. We find that models learn poorly with the vanilla *{hypothesis} [yes/no] {premise}*, but they learn as fast as the instructive templates with Schick & Schütze's punctuated version (see Appendix A.2 for detail results). However, punctuation alone cannot explain the performance gap, since models

[4] An author manually labeled the 30 training examples seen by models under random seed 1 (example nos. 550–580), among which we find 17 pairs of entailment, 5 or 8 pairs (depending on how strictly one judges their acceptability) of summarizations, and only one pair of paraphrase.

[5] If a model were to literally label whether the premise starts with "the", it would only score 49.6% on the RTE training set.

[6] Our tokenizer does not insert special [SEP] tokens between premises and hypotheses.

trained with *[yes/no] {hypothesis} {premise}* (Figure 7, dark orange) perform second to best, even though no [mask] or any template punctuation segment their premises and hypotheses. Yet, swapping their premises and hypotheses (light orange) makes it the worst performing among all null templates.

5 Effect of Target Word Semantics on Few-Shot Learning

5.1 Method

In this experiment, we study the effect of different LM targets given a fixed template. We write 4 categories[7] of targets (see Table 2 for examples):

1. Yes-no: Model is expected to predict the word "yes" for entailment and "no" for non-entailment.

2. Yes-no-like: Semantically equivalent to yes-no but using superficially different words, e.g., "true"/"false", "positive"/"negative".

3. Arbitrary: Model is expected to predict arbitrary words that have no semantic relation to the entailment task, e.g., "cat" for entailment, "dog" for non-entailment.

4. Reversed: Model is expected to predict the opposite of the (intuitive) yes-no and yes-no-like labels, e.g., "no" for entailment, "yes" for non-entailment.

Within the arbitrary category, in addition to common anglophone first names as Le Scao and Rush (2021) tested, we also include word pairs with high semantic similarity, low similarity, and pairs which are highly frequent in the English language, but we find no consistent difference among these various subcategories of the arbitrary category.

5.2 Results

We find that models trained with yes-no targets learn drmatically faster than those trained with arbitrary and reversed targets. For example, Figure 8 shows the top-performing instructive template trained with different target words. The large effect sizes are particularly noteworthy, e.g., at 30 shots, the difference between the median accuracies of "yes"/"no" vs. "no"/"yes" is $73.3\% - 54.5\% = 18.8\%$. Similarly, Table 3 confirms the same trends when aggregating over all templates.

[7] With declarative templates, another category is their template-specific targets. They are excluded from experiments in this section because combining declarative templates with other target categories yield ungrammatical prompts.

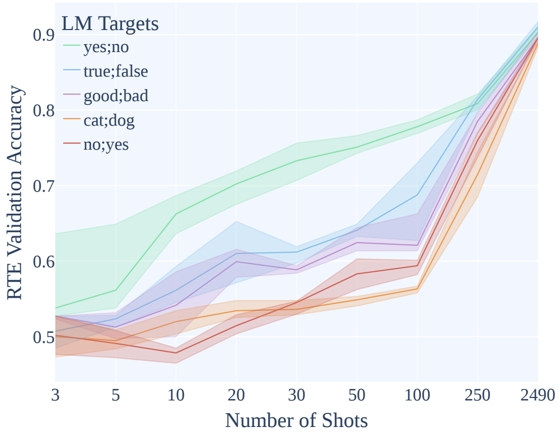

Figure 8: The best-performing instructive prompt, *{premise} Are we justified in saying that "{hypothesis}"? [mask]* with select LM targets from each category. See Appendix D for results with other templates.

Target Words	Category
yes; no	yes-no
true; false	yes-no-like
good; bad	yes-no-like
no; yes	reversed
false; true	reversed
cat; dog	arbitrary (semantically similar)
cake; piano	arbitrary (semantically dissimilar)
the; a	arbitrary (highly frequent)
she; he	arbitrary (highly frequent)

Table 2: Example Sets of LM targets.

The above seems to be a positive result—models are sufficiently sensitive to the semantics of the target words such that they consistently learn slower when the target words are unintuitive. However, there is a major caveat: The effect of the target words overrides the semantics of the overall prompt. Consider two kinds of template-target combinations:

1. An irrelevant or misleading template + yes-no targets, e.g., *{premise} Does the paragraph start with "the"? [yes/no] {hypothesis}*

2. An instructive template + arbitrary targets, e.g., *{premise} Are we justified in saying that "{hypothesis}"? [cat/dog]*

Figure 9 shows that that combinations such as (1) often dramatically outperform (2), which is the opposite of what we expect because (1) is a pathological condition under which we would expect that a

	instructive	irrelevant	null	misleading
yes-no	71.12%	70.58%	62.63%	66.43%
yes-no-like	59.93%	62.45%	58.84%	58.30%
arbitrary	55.23%	56.50%	54.51%	55.60%
reversed	54.87%	56.50%	53.07%	55.42%

Table 3: Median accuracies of all template-target combinations at 30 shots. In general, the choice of target words (rows) matters much more than the choice of templates (columns), which is counterintuitive because humans would care much more about the templates (the task instructions) than the targets (which words they need to respond with).

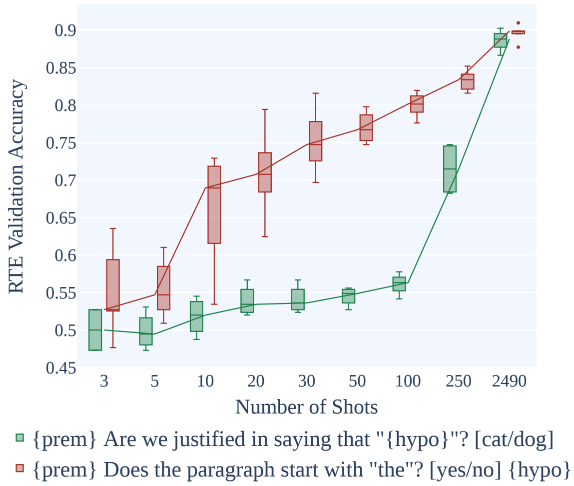

Figure 9: Misleading templates + yes-no targets (red) learn substantially faster than instructive templates + arbitrary targets (green), which is the opposite of what we expect from humans.

human would be confused and would still need a large number of trials and errors to figure out what is the actual task. In contrast, (2) simply requires figuring out a mapping: "Reply 'cat' if entailed and reply 'dog' if not entailed".[8] For humans, this can be learned in a few shots, e.g., Ferrigno et al. (2017) showed that adults can reach 60% accuracy in 18 trials[9] for an arbitrary map of {more numerous → star shape, less numerous → diamond shape} without receiving any language instructions. In contrast, models under many arbitrary LM targets struggle to reach 60% median accuracy even by 100 shots with instructive templates (Figure 9 green; Figure 8 yellow, orange, red).

Further, even given intuitive yes-no-like targets such as "true"/"false" and "positive"/"negative", models learn drastically slower compared to when given "yes"/"no". As Figure 8 (green vs. blue) and Table 3 (first vs. second row) show, there exists a large performance gap between yes-no and yes-no-like targets which is not closed until 250 shots, whereas for humans, the difference between answering "yes"/"no" vs. answering "true"/"false" should be trivial and likely would not require more than 100 examples to close any gap.

6 Discussion

6.1 Summary and Interpretation

Our main research question is whether models understand prompts as meaningful task instructions analogous to how humans would. To answer this question, consider how a human annotator would respond to different scenarios. Suppose an experimenter provides a human annotator with an informative instruction of a reasonably easy task. If the annotator understands the instruction, we expect them to perform better than when the experimenter provides no instructions, makes irrelevant chit-chat in lieu of giving instructions, or deliberately instructs to perform a task other than the intended one. Thus, if models understand the meaning of prompts, at a minimum, we would generally expect that helpful prompts will outperform uninformative and deceptive prompts.

Helpful > Deceptive Prompts? If models understand the meaning of prompts like humans do, we would expect models to learn much faster when trained with helpful prompts (i.e., those with instructive templates plus yes-no or yes-no-like targets) than with deceptive ones (those containing misleading templates or reversed targets). Some of our results appear to suggest that this is the case: In aggregate, instructive > misleading templates (Figure 4) and yes-no > yes-no-like > arbitrary targets (Figure 8). However, on a closer look, we also find many contradictory results that make it difficult to conclude that models really respond to instructive vs. misleading templates in a human-like way. For example, it is not clear that instructions "closer" to NLI make models learn faster. For instance, models learn faster with a paraphrasing instruction than with a summarization instruction, even though the latter has more overlap with the gold labels of RTE (§4.2). In other words, human

[8] In preliminary experiments, we tried adding this kind of "reply 'cat' if entailed and reply 'dog' if not entailed" hint to the templates, but we found no significant difference.

[9] And this comparison is heavily charitable to the models because "18 trials" means that humans see 18 examples for 18 times in total, whereas "20-shot" means that models can see the same 20 examples over and over again for many epochs.

semantic intuition hardly predicts prompt performance. Likewise, although yes-no > arbitrary targets is expected, yes-no > yes-no-like targets (e.g., "true"/"false", "positive"/"negative", Table 3) is not expected, since humans would likely consider responding with "true"/"false" to be just as easy as responding with "yes"/"no". The fact that models are overly sensitive to the choice of target words suggests semantics alone is not the primary explanation of models' behavior.

Helpful > Uninformative Prompts? If models understand the meaning of prompts, we would also expect models to learn much faster when trained with helpful prompts than with uninformative ones (irrelevant templates or arbitrary targets). Here, we find an unequivocally negative result that helpful templates $\approx$ uninformative templates (Figure 2), which provides our strongest evidence that models do not truly understand their prompts. Meanwhile, for the target words, we see a partially positive result that yes-no > arbitrary targets, but again with the caveat that the preference for yes-no targets persists even when the targets do not make sense in the context of the overall prompt (Table 3). For example, irrelevant template + yes-no targets > instructive template + arbitrary targets, even though humans would likely show the opposite direction of inequality (§5.2).

Finally, it is hard to argue with the simple observation that we can always find an irrelevant, null, or misleading template that performs just as well as the instructive ones. Thus, our main argument shares the same logic as a recent line of studies (Sinha et al., 2021; Pham et al., 2021; Gupta et al., 2021) which argue that the fact that LMs achieve good performance under ideal conditions is insufficient to establish language understanding because they also succeed under pathological conditions (e.g., sentences with shuffled word order) where humans fail catastrophically. In other words, the fact that models are so good at inferring the gold labels from pathological inputs casts major doubts on whether models make inferences in any way that resembles how humans make inferences.

6.2 Alternative Interpretations

As with any extrinsic evaluation, accuracy cannot directly measure understanding, and thus it is worth asking whether the observed negative results might be attributable to something other than a lack of understanding of the prompt. For example, a human could perfectly understand an instruction but still, e.g., have the same accuracy with instructive vs. irrelevant templates because the task is too hard (a lack of competence) or because they for some reason ignore the instructions (a lack of compliance).

For our experiments, a lack of compliance is a greater concern than a lack of competence,[10] since our observed trend (that instructive $\approx$ irrelevant) could be explained by models simply ignoring the prompts. Such an explanation would be consistent with prior work on generative tasks which found that models often generate text unrelated to, and thus possibly ignore, their prompts (Efrat and Levy, 2020; Mishra et al., 2021). However, we don't find lack of compliance alone provides a satisfying explanation for our results overall. If models truly ignore the prompts, we should not see any systematic differences between categories of prompts. Instead, we do see consistent patterns that (1) instructive and irrelevant templates make models learn faster than do misleading and null templates and (2) yes-no targets make models learn faster than with arbitrary and reversed targets.

Perhaps a more nuanced argument is not that models ignore their prompts entirely, but rather that it "takes less effort" for models to use the superficial or spurious features for predictions as opposed to the more complex syntactic or semantic features (Lovering et al., 2021; Warstadt et al., 2020) required to properly comply with the instructions. However, spurious features alone cannot explain the observed performance gaps either. Recall that, within each random seed, all models see exactly the same training examples (with the same spurious features). Thus, to the extent that models perform differently with some prompts compared to others, it may be due to some complex interactions between the (spurious or semantic) features in prompts and the spurious features in inputs. Understanding the nature of this interaction is an interesting direction for future work, and suggests a way in which the semantics of the prompt might matter, e.g., by affecting the models' inductive biases, even if the models fall short of fully understanding the instructions they are given.

[10] An argument can be made that the observed negative results are due to models' lack of competence in NLI, rather than a lack of understanding NLI instructions. However, such an argument hinges on whether NLI as a task is a good measure of language understanding for NLP in general, which is beyond the scope of this paper.

6.3 Open Questions

In this paper, we show that models are far from understanding manually written prompts like humans do, but do other approaches of prompts fare better in understanding? Recall that, by construction, only discrete prompt methods allow learning from instructions: Priming methods supply extra examples, not instructions, while continuous prompts forgo natural language prompts altogether (§2.1).

In principle, automatically searched discrete prompts may support understanding instructions, but in practice, these methods often yield unnatural language prompts like *{premise} [agency/##ponents/nobody] concretepathic workplace {hypothesis}* (Shin et al., 2020, p. 7), the success of which is reminiscent of our irrelevant prompts. Gao et al. (2021)'s method arguably generates among the most natural-language-looking automatic prompts, e.g., *{premise} [yes/maybe/no], this time {hypothesis}* and *{premise} ? [Alright/ Watch/Except] {hypothesis}* (p. 8). However, these appear to be more like a kind of relaxed RegEx patterns that match premise and hypothesis pairs naturally occurred in the pretraining corpora. Words in these prompt (e.g., "this time", "except") plausibly capture some distributional signal of the general premise-hypothesis relation, but these are not instructions of an NLI task. If, similar to prior work (§2.3), memorization of patterns from pretraining is the primary driver of prompt-based models' predictions, it would imply that the performance of prompts might be specific to particular pretrained model checkpoints and particular evaluation datasets, as opposed to an assumption that there exists a set of, e.g., "good" NLI instructions that generalize to most NLI datasets and most pretrained models.

In sum, it appears that other approaches of prompts are also far from understanding instructions. That being said, their extraordinary improvement in few-shot learning is real and laudable; still, such improvement appears to be due to significantly improved pattern matching as opposed to a fundamental shift to learning from instructions, as is sometimes claimed (§1). Nevertheless, while our experimental results show that existing models are far from this goal, we agree with Mishra et al. (2021), Schütze (2021), and Weller et al. (2020) that this new line of research on developing models that learn from instructions is important. There has not previously been a generic way inform the model about the requirements of the task it needs to perform. As such, model have had to rely on trial and error with a large amount of training data in order to figure out the task at hand. In contrast, discrete prompt-based models may have the potential to enable a generic, task-agnostic method that learns from instructions expressed in natural language. Achieving this would require not only progress on a currently popular topic of (1) following instructions to few-shot learn novel tasks, but also an understudied topic of (2) following instructions to make different predictions even when given the same training data: This paper's tests of whether models make different predictions when given irrelevant, misleading, and null instructions are only a first step toward this direction. Ideally, we would like to see models make different predictions based on nuanced insutrctions (e.g., strictly logical entailment vs. pragmatic and natural inference) even when given the same data. Further, additional analyses are needed to investigate (3) pattern matching between task data and pretraining corpora as well as (4) interactions between spurious features in prompts and spurious features in task data (§6.2). If successful in clearing these hurdles, prompt-based models may go beyond an efficency improvement of learning from data and enable a fundamentally different mode of learning (Shettleworth, 2012).

7 Conclusion

In this study, we train a representative discrete prompt model with over 30 manually written templates and 10 sets of LM targets for NLI. While models learn faster with instructive templates over misleading prompts (with many caveats), they learn equally fast with instructive templates and irrelevant templates. We further show that the choice of the target words overrides the meaning of the overall prompts. These results contradict a hypothesis commonly assumed in the literature that prompts serve as semantically meaningful task instructions and that writing high-performing prompts requires domain expertise. Although we find that existing models are far from genuinely understanding the meaning of their prompts, we agree that learning from instructions is an important research direction, and we propose several future directions of investigating models' understanding of the meaning of prompts.

Acknowledgments

We are grateful to Stephen Bach, Colin Raffel, Roman Feiman, Teven Le Scao, Pierre Colombo, Ryan Teehan, Debajyoti Datta, Nihal Nayak, Roma Patel, Charles Lovering, and Aaron Traylor for comments and discussions on early drafts of this paper. Special thanks to Teven for technical clarifications.

References

Jonathan Bragg, Arman Cohan, Kyle Lo, and Iz Beltagy. 2021. FLEX: Unifying evaluation for few-shot NLP. *arXiv preprint arXiv:2107.07170*.

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. Curran Associates, Inc.

Boxi Cao, Hongyu Lin, Xianpei Han, Le Sun, Lingyong Yan, Meng Liao, Tong Xue, and Jin Xu. 2021. Knowledgeable or educated guess? revisiting language models as knowledge bases. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1860–1874, Online. Association for Computational Linguistics.

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom B. Brown, D. Song, Ú. Erlingsson, Alina Oprea, and Colin Raffel. 2020. Extracting training data from large language models. *ArXiv*, abs/2012.07805.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The pascal recognising textual entailment challenge. In *Machine Learning Challenges Workshop*, pages 177–190. Springer.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Nam Do and Ellie Pavlick. 2021. Are rotten apples edible? challenging commonsense inference ability with exceptions. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2061–2073, Online. Association for Computational Linguistics.

Avia Efrat and Omer Levy. 2020. The turking test: Can language models understand instructions? *arXiv preprint arXiv:2010.11982*.

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

Stephen Ferrigno, Julian Jara-Ettinger, Steven T Piantadosi, and Jessica F Cantlon. 2017. Universal and uniquely human factors in spontaneous number perception. *Nature communications*, 8(1):1–10.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830, Online. Association for Computational Linguistics.

Marvin J Greenberg. 1974. *Euclidean and non-Euclidean Geometries: Development and history*. W. H. Freeman and Company.

Ashim Gupta, Giorgi Kvernadze, and Vivek Srikumar. 2021. Bert & family eat word salad: Experiments with text understanding.

Zhengbao Jiang, Frank F. Xu, J. Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Nora Kassner, Benno Krojer, and Hinrich Schütze. 2020. Are pretrained language models symbolic reasoners over knowledge? In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 552–564, Online. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Teven Le Scao and Alexander Rush. 2021. How many data points is a prompt worth? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2627–2636, Online. Association for Computational Linguistics.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *ArXiv*, abs/2104.08691.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the*

Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 4582–4597, Online. Association for Computational Linguistics.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Robert L Logan, Ivana Balažević, Eric Wallace, Fabio Petroni, Sameer Singh, and Sebastian Riedel. 2021. Cutting down on prompts and parameters: Simple few-shot learning with language models. *arXiv preprint arXiv:2106.13353*.

Charles Lovering, Rohan Jha, Tal Linzen, and Ellie Pavlick. 2021. Predicting inductive biases of pre-trained models. In *International Conference on Learning Representations*.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2021. Natural instructions: Benchmarking generalization to new tasks from natural language instructions. *arXiv preprint arXiv:2104.08773*.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Thang M. Pham, Trung Bui, Long Mai, and Anh Nguyen. 2021. Out of order: How important is the sequential order of words in a sentence in natural language understanding tasks?

Plato. c. 399 BC. *Euthyphro*. Penguin Books.

Guanghui Qin and Jason Eisner. 2021. Learning how to ask: Querying LMs with mixtures of soft prompts. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5203–5212, Online. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Timo Schick and Hinrich Schütze. 2021a. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021b. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352, Online. Association for Computational Linguistics.

Hinrich Schütze. 2021. Humans learn from task descriptions and so should our models. Keynote Speech of the 2021 Annual Conference of the North American Chapter of the Association for Computational Linguistics.

Sara J Shettleworth. 2012. Modularity, comparative cognition and human uniqueness. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 367(1603):2794–2802.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235, Online. Association for Computational Linguistics.

Koustuv Sinha, Prasanna Parthasarathi, Joelle Pineau, and Adina Williams. 2021. Unnatural language inference.

Derek Tam, Rakesh R Menon, Mohit Bansal, Shashank Srivastava, and Colin Raffel. 2021. Improving and simplifying pattern exploiting training. *arXiv preprint arXiv:2103.11955*.

Shizuo Tsuji and Mary Sutherland. 1980. *Japanese Cooking: A Simple Art*. Kodansha International.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *NeurIPS*.

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

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.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. 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.

A All Prompts

A.1 Main Experiments

template	default targets	template category	target category	median	mean	std. dev.
traditional fine-tuning				53.79%	53.84%	3.38%
{premise} Are we justified in saying that "{hypothesis}"? {mask}	yes;no	instructive	yes-no	73.29%	73.06%	3.53%
Given {premise} Is it guaranteed true that {hypothesis}{mask}	yes;no	instructive	yes-no	71.30%	69.67%	4.73%
Given {premise} Should we assume that "{hypothesis}" is true? {mask}	yes;no	instructive	yes-no	70.76%	68.86%	4.53%
{premise} Question: {hypothesis}Yes or no? {mask}	yes;no	instructive	yes-no	61.01%	63.58%	6.96%
{premise} We {mask} infer that "{hypothesis}"	can;cannot	instructive	template-specific	66.79%	68.50%	5.26%
{premise} It is {mask} to be true that {hypothesis}	likely;unlikely	instructive	template-specific	69.86%	69.99%	5.48%
{premise} It is {mask} that {hypothesis}	likely;questionable	instructive	template-specific	64.62%	63.49%	4.26%
{premise} It {mask} be true that {hypothesis}	should;might	instructive	template-specific	64.62%	65.21%	5.23%
{premise} It {mask} be true that {hypothesis}	must;might	instructive	template-specific	69.67%	68.73%	5.30%
{premise} Single-family zoning is bad for American cities. "{hypothesis}"? {mask}	yes;no	irrelevant	yes-no	71.30%	70.80%	3.42%
{premise} Inflections are annoying and thank god that Middle English got rid of most of them. {hypothesis}? {mask}	yes;no	irrelevant	yes-no	73.83%	72.97%	2.27%
{premise} When Bolyai sent Gauss his discovery of non-Euclidean geometry, Gauss replied that he arrived at the same results 30 years ago. {hypothesis}? {mask} (Greenberg, 1974, p. 141)	yes;no	irrelevant	yes-no	70.40%	70.40%	1.68%
{premise} If bonito flakes boil more than a few seconds, the stock becomes too strong? {mask} {hypothesis} (Tsuji and Sutherland, 1980, p. 148)	yes;no	irrelevant	yes-no	68.05%	68.28%	4.62%
{premise} Is the pious loved by the gods because it is pious? Or is it pious because it is loved by the gods? {mask} {hypothesis} (Plato, c. 399 BC, 10a)	yes;no	irrelevant	yes-no	68.77%	69.00%	3.79%
{hypothesis} Do most of the above words appear in the following passage? {mask} "{premise}"	yes;no	misleading	yes-no	65.88%	64.44%	4.20%
{premise} Are there similar words in "{hypothesis}"? {mask}	yes;no	misleading	yes-no	56.86%	56.50%	2.62%
{premise} Does the paragraph start with "the"? {mask} {hypothesis}	yes;no	misleading	yes-no	74.73%	75.18%	3.84%
{premise} Does that have the same meaning as "{hypothesis}"? {mask}	yes;no	misleading	yes-no	67.87%	67.15%	3.75%
{premise} Can that be paraphrased as: {hypothesis}? {mask}	yes;no	misleading	yes-no	61.91%	62.09%	2.01%
{premise} Which {mask} be summarized as: {hypothesis}	can;cannot	misleading	template-specific	60.83%	61.06%	3.78%
{premise} {mask} {hypothesis}	yes;no	null	yes-no	69.67%	69.54%	2.35%
{premise} {hypothesis}{mask}	yes;no	null	yes-no	60.83%	60.33%	2.98%
{mask} {premise} {hypothesis}	yes;no	null	yes-no	59.93%	58.98%	2.09%
{hypothesis}{mask} {premise}	yes;no	null	yes-no	62.81%	63.36%	2.21%
{hypothesis}{premise} {mask}	yes;no	null	yes-no	59.39%	61.46%	4.38%
{mask} {hypothesis}{premise}	yes;no	null	yes-no	68.95%	68.95%	2.97%

Table 4: 30-shot accuracy of all templates used in the main paper. To avoid cherry picking, all of them were written prior to analysis and no retroactive exclusion or inclusion of templates from the categories defined above is allowed. We intentionally aimed to write more minimal pairs, while more diverse prompts will be reported in a future version of this paper. For experiments in Section 5, models are trained by permutating these templates with all sets of LM targets (see Appendix D) in addition to their default targets.

A.2 Preliminary Experiments

template	targets	template category	target category	median	mean	std. dev.
{premise} Can we infer that "{hypothesis}"? {mask}	yes;no	instructive	yes-no	75.09%	75.09%	2.30%
{premise} Can we infer that "{hypothesis}"? {mask}	positive;negative	instructive	yes-no-like	64.98%	65.97%	6.76%
{premise} Can we infer that "{hypothesis}"? {mask}	no;yes	instructive	reversed	54.87%	54.96%	0.80%
{premise} Can we infer that "{hypothesis}"? {mask}	cat;dog	instructive	arbitrary	52.16%	52.89%	3.52%
{premise} Can we infer that "{hypothesis}"? Answer B if we can and C if we cannot. {mask}	B;C	hinted	arbitrary	57.04%	56.86%	3.26%
{premise} Can we infer that "{hypothesis}"? Cat or dog? {mask}	cat;dog	hinted	arbitrary	54.15%	53.88%	1.48%
{premise} Can we infer that "{hypothesis}"? Say cat if we can and dog if we cannot. {mask}	cat;dog	hinted	arbitrary	53.43%	53.52%	0.80%
{premise} Can we infer that "{hypothesis}"? Say cat if we can and dog if we cannot. {mask}	yes;no	misleading	yes-no	65.88%	66.25%	5.34%
{premise} It {mask} be true that {hypothesis}	might;must	instructive	reversed	59.38%	60.74%	5.96%
{premise} It is {mask} to be true that {hypothesis}	unlikely;likely	instructive	reversed	66.97%	65.61%	4.60%
{premise} It {mask} be true that {hypothesis}	yes;no	ungrammatical	yes-no	70.58%	70.85%	2.33%
{premise} We {mask} infer that "{hypothesis}"	yes;no	ungrammatical	yes-no	62.27%	62.19%	4.15%
Given {premise} Is it guaranteed true that {hypothesis}{mask}	positive;negative	instructive	yes-no-like	62.81%	62.27%	3.31%
Given {premise} Is it guaranteed true that {hypothesis}{mask}	Mary;John	instructive	arbitrary	60.29%	59.48%	3.63%
Given {premise} Is it guaranteed true that {hypothesis}{mask}	0;1	instructive	arbitrary	58.13%	57.49%	3.39%
Given {premise} Is it guaranteed true that {hypothesis}{mask}	B;C	instructive	arbitrary	59.38%	60.20%	5.08%
{premise} It is {mask} to be true that {hypothesis}	positive;negative	instructive	yes-no-like	71.66%	71.66%	3.80%
{premise} It is {mask} to be true that {hypothesis}	Mary;John	instructive	arbitrary	59.56%	59.39%	3.08%
{premise} It is {mask} to be true that {hypothesis}	0;1	instructive	arbitrary	56.68%	56.32%	2.67%
{premise} It is {mask} to be true that {hypothesis}	B;C	instructive	arbitrary	61.37%	60.92%	3.48%
{premise} It {mask} be true that {hypothesis}	positive;negative	ungrammatical	yes-no-like	69.14%	68.68%	3.01%
{premise} It {mask} be true that {hypothesis}	Mary;John	ungrammatical	arbitrary	63.18%	62.63%	2.79%
{premise} It {mask} be true that {hypothesis}	0;1	ungrammatical	arbitrary	69.49%	70.13%	3.33%
{premise} It {mask} be true that {hypothesis}	B;C	ungrammatical	arbitrary	68.78%	68.23%	4.24%
"{hypothesis}" ?| {mask}. "{premise} " (Schick and Schütze, 2021b)	yes;no	null	yes-no	72.74%	72.83%	2.95%
{hypothesis}{mask} {premise}	yes;no	null	yes-no	63.18%	62.91%	2.13%
"{premise} " ?| {mask}. "{hypothesis}"	yes;no	null	yes-no	73.11%	73.47%	2.87%
{premise} ? {mask} {hypothesis}	yes;no	null	yes-no	72.75%	72.47%	1.68%
{premise} Does the next sentence start with "the"? {mask} {hypothesis}	yes;no	misleading	yes-no	75.63%	76.08%	1.98%
{premise} Do all of the above words appear in "{hypothesis}" ? {mask}	yes;no	misleading	yes-no	58.30%	59.29%	3.47%
{premise} Do all of the above words appear in "{hypothesis}" ? {mask}	no;yes	misleading	reversed	53.25%	53.70%	2.63%
{premise} Do all of the above words appear in "{hypothesis}" ? {mask}	mary;john	misleading	arbitrary	55.78%	55.78%	1.33%

Table 5: 30-shot accuracy of all prompts used in preliminary experiments.

B Comparison of All Four Template Categories in Aggregate

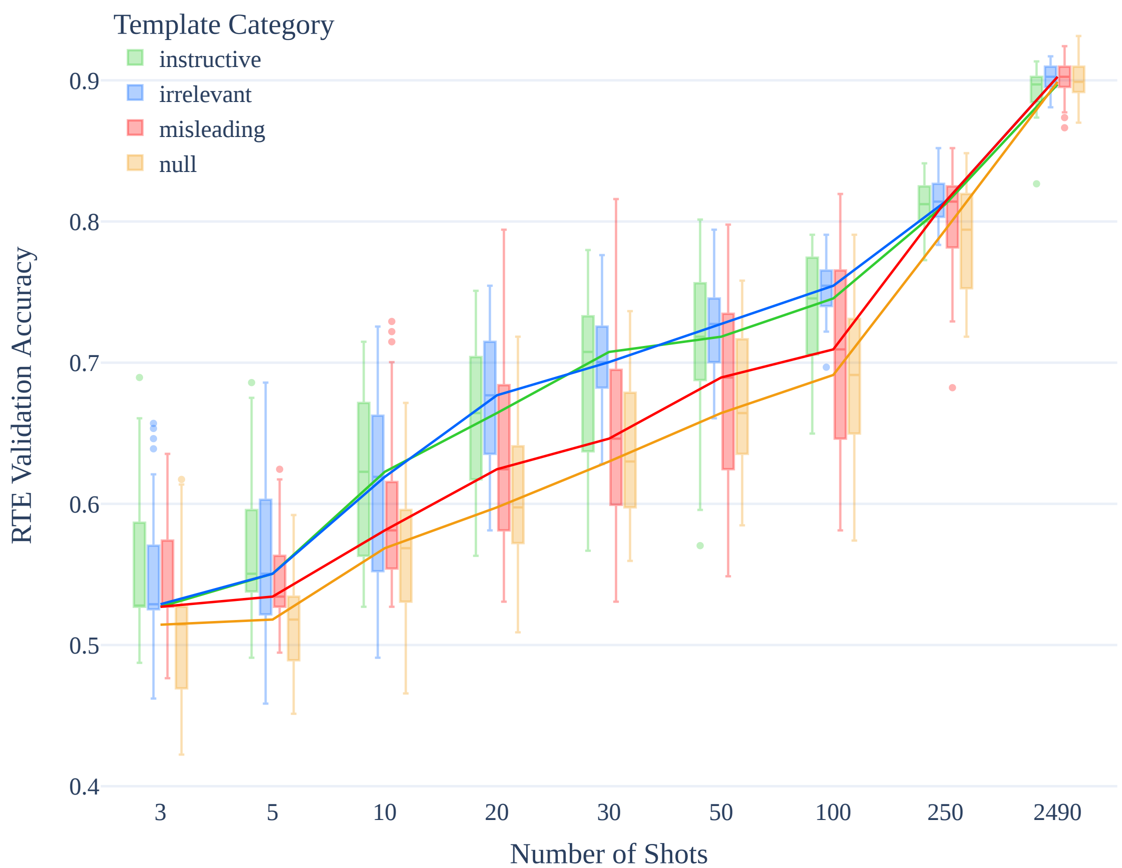

Figure 10

C Comparison of Individual Templates within the Same Category

C.1 All Instructive Templates

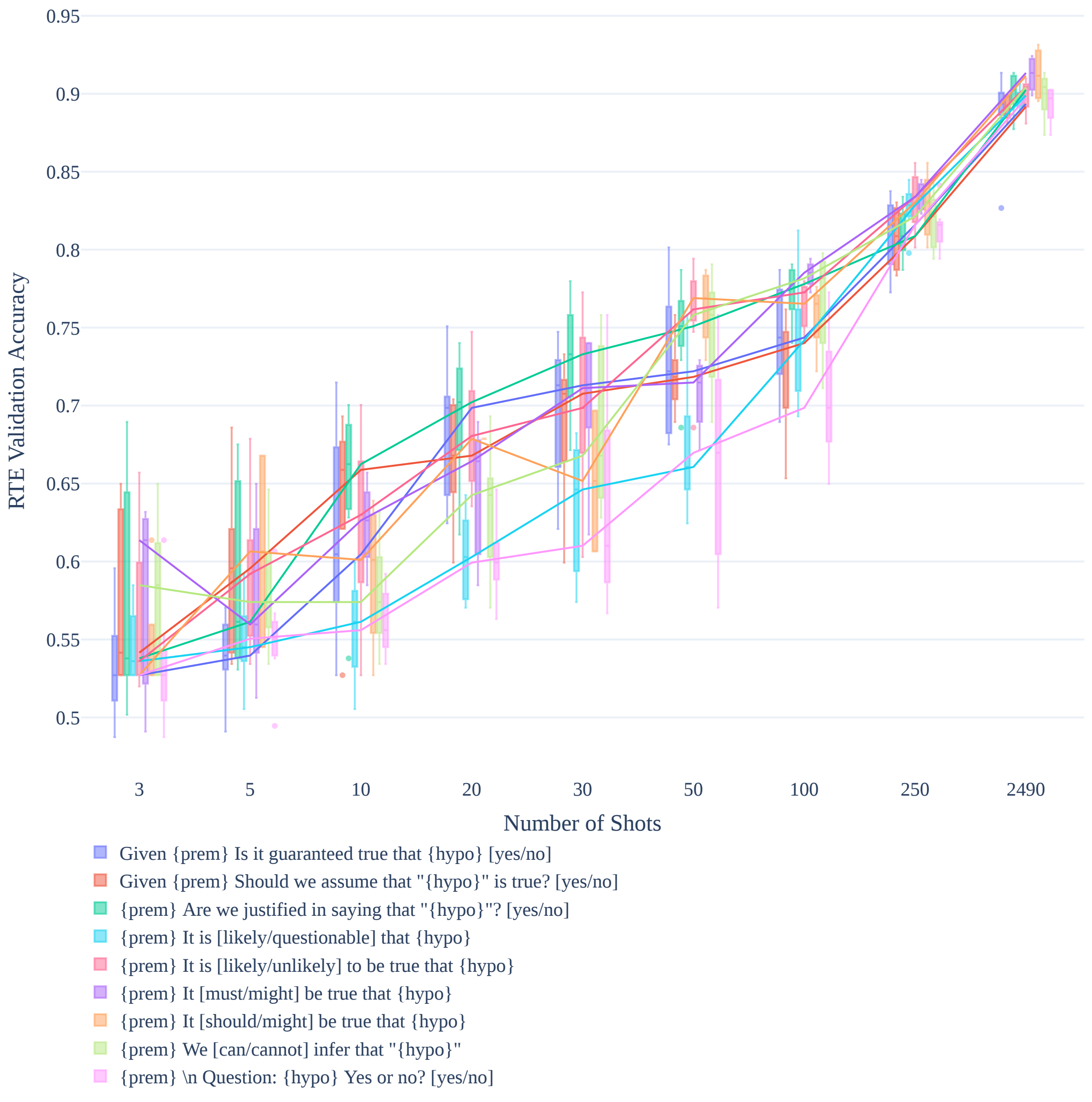

Figure 11: All instructive templates with default LM targets, plus an average-performing instructive prompt for comparison.

C.2 All Irrelevant Templates

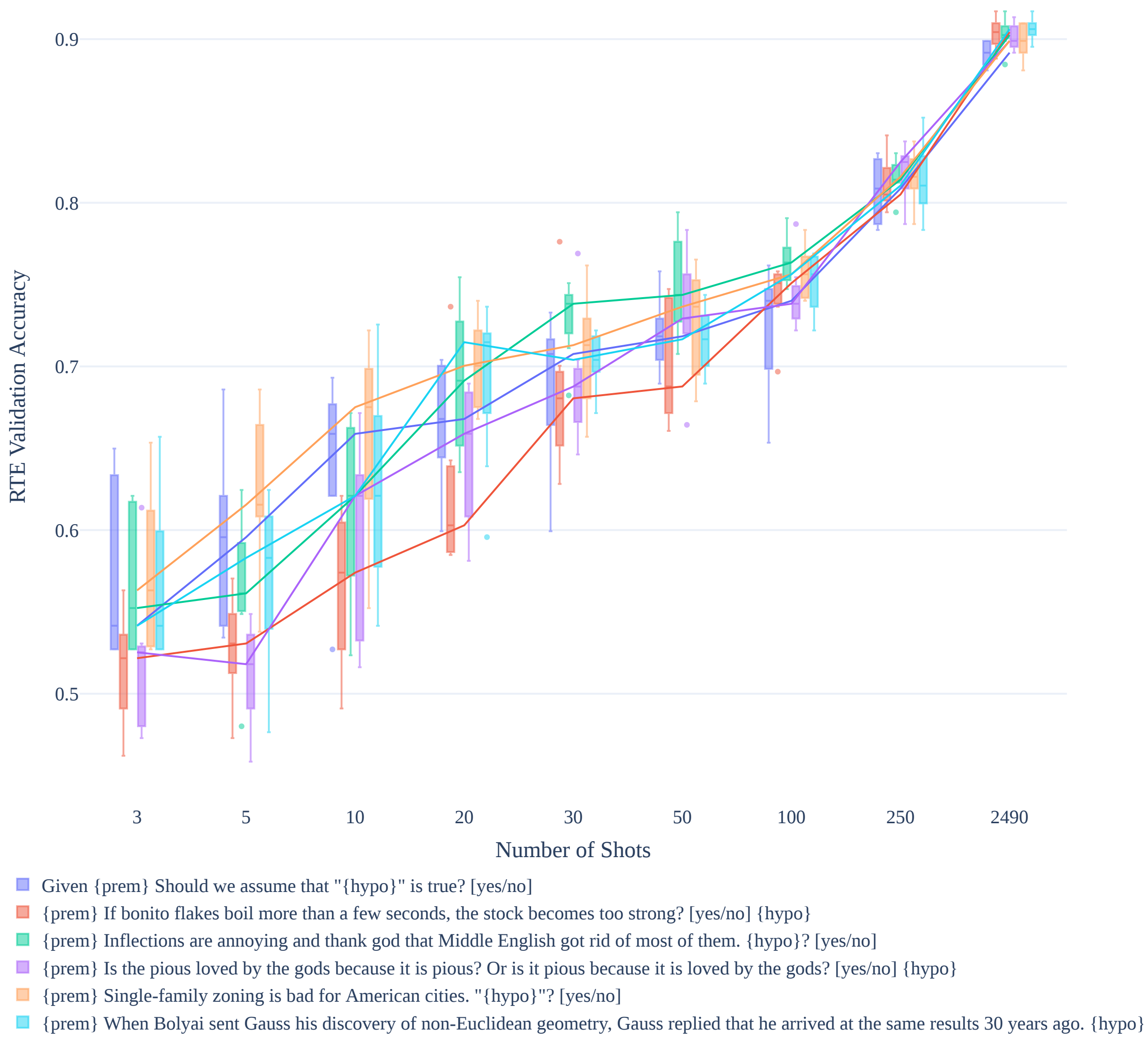

Figure 12: All irrelevant templates with default LM targets, plus an average-performing instructive prompt for comparison.

C.3 All Misleading Templates

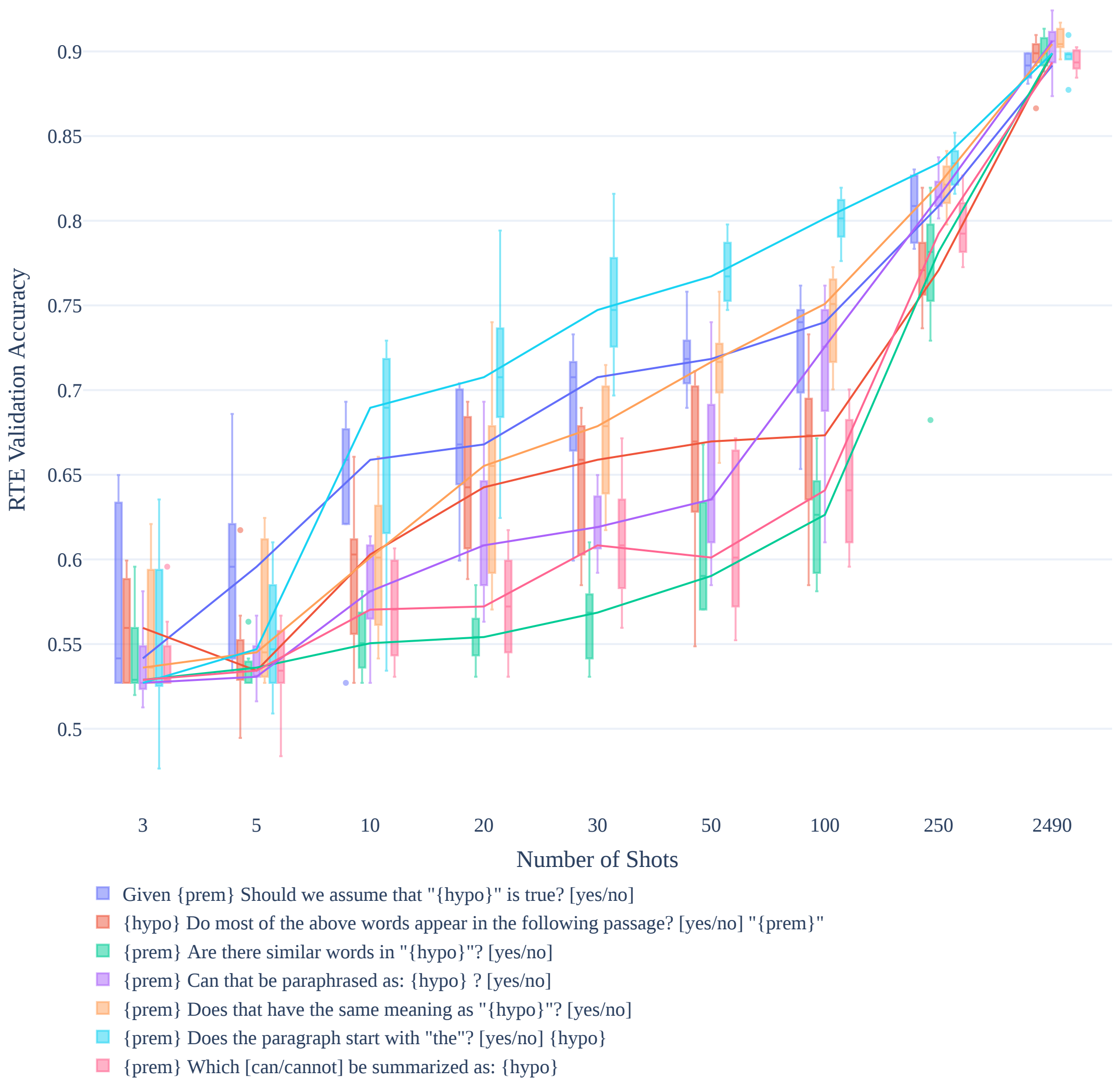

Figure 13: All misleading templates with default LM targets, plus an average-performing instructive prompt for comparison.

C.4 All Null Templates

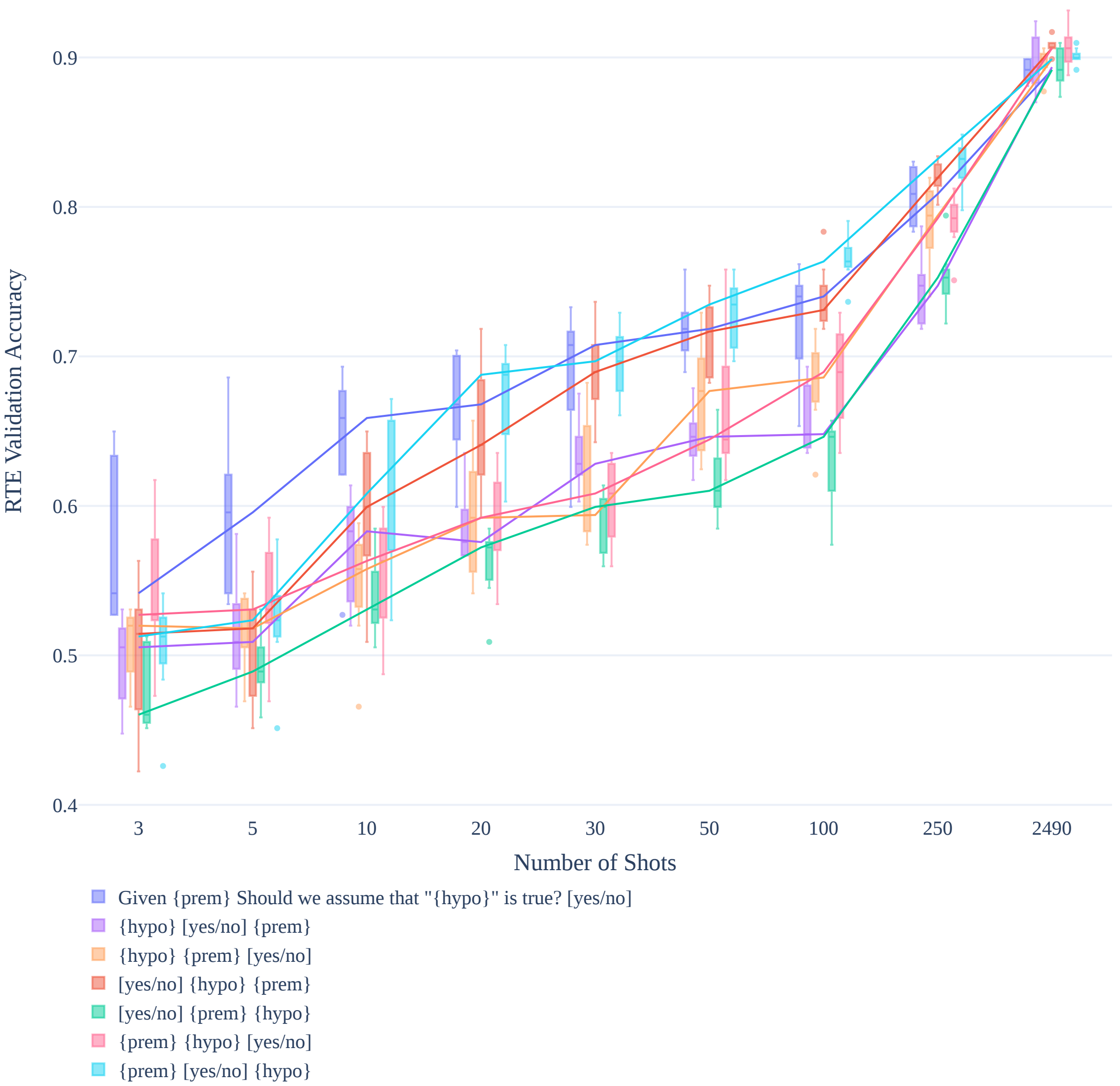

Figure 14: All null templates with default LM targets, plus an average-performing instructive prompt for comparison.

D Comparison of All LM targets, Controlling for the Template

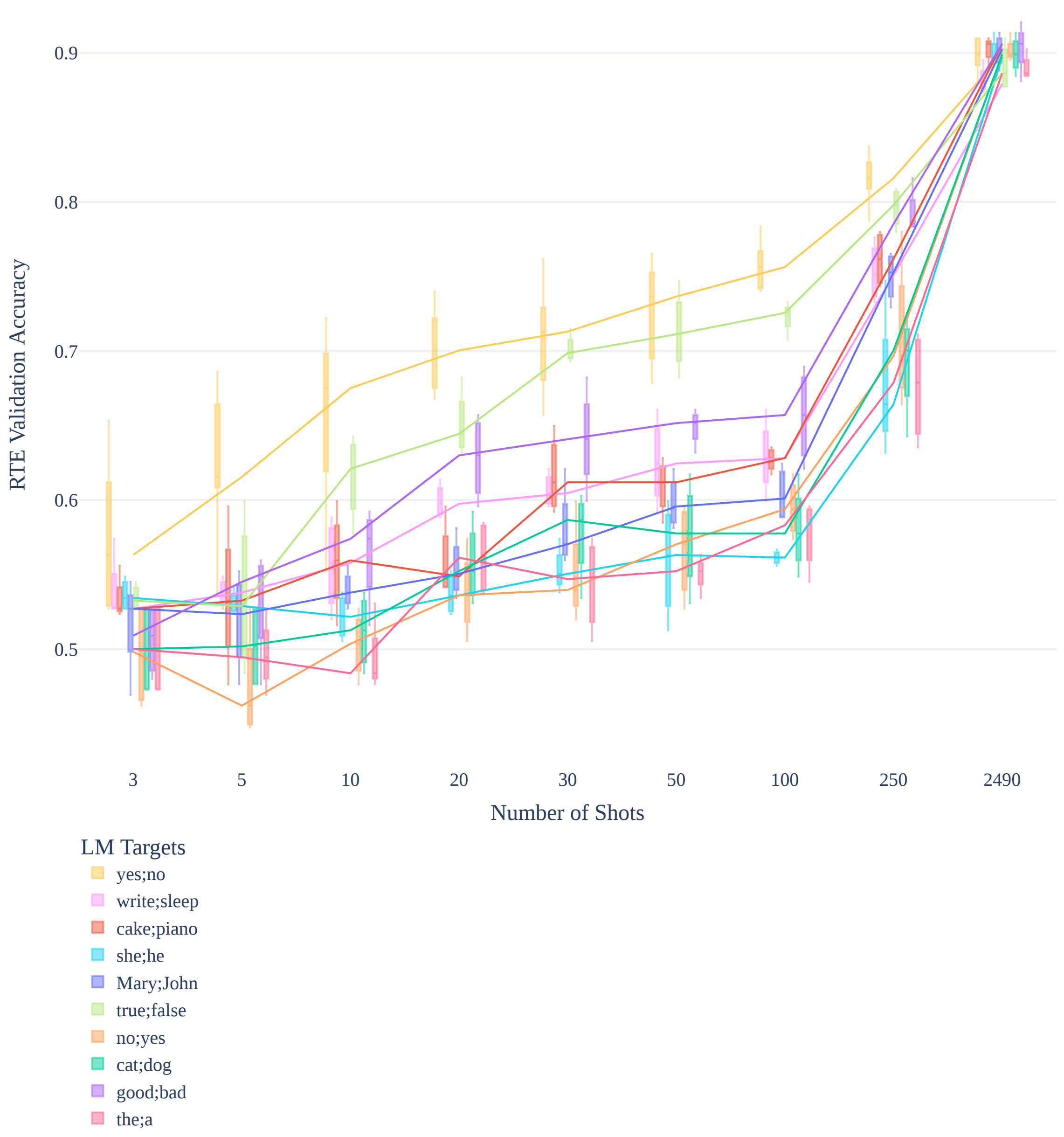

Figure 15: The best performing irrelevant prompt, *{premise} Single-family zoning is bad for American cities. "{hypothesis}"? [mask]* with all LM targets.

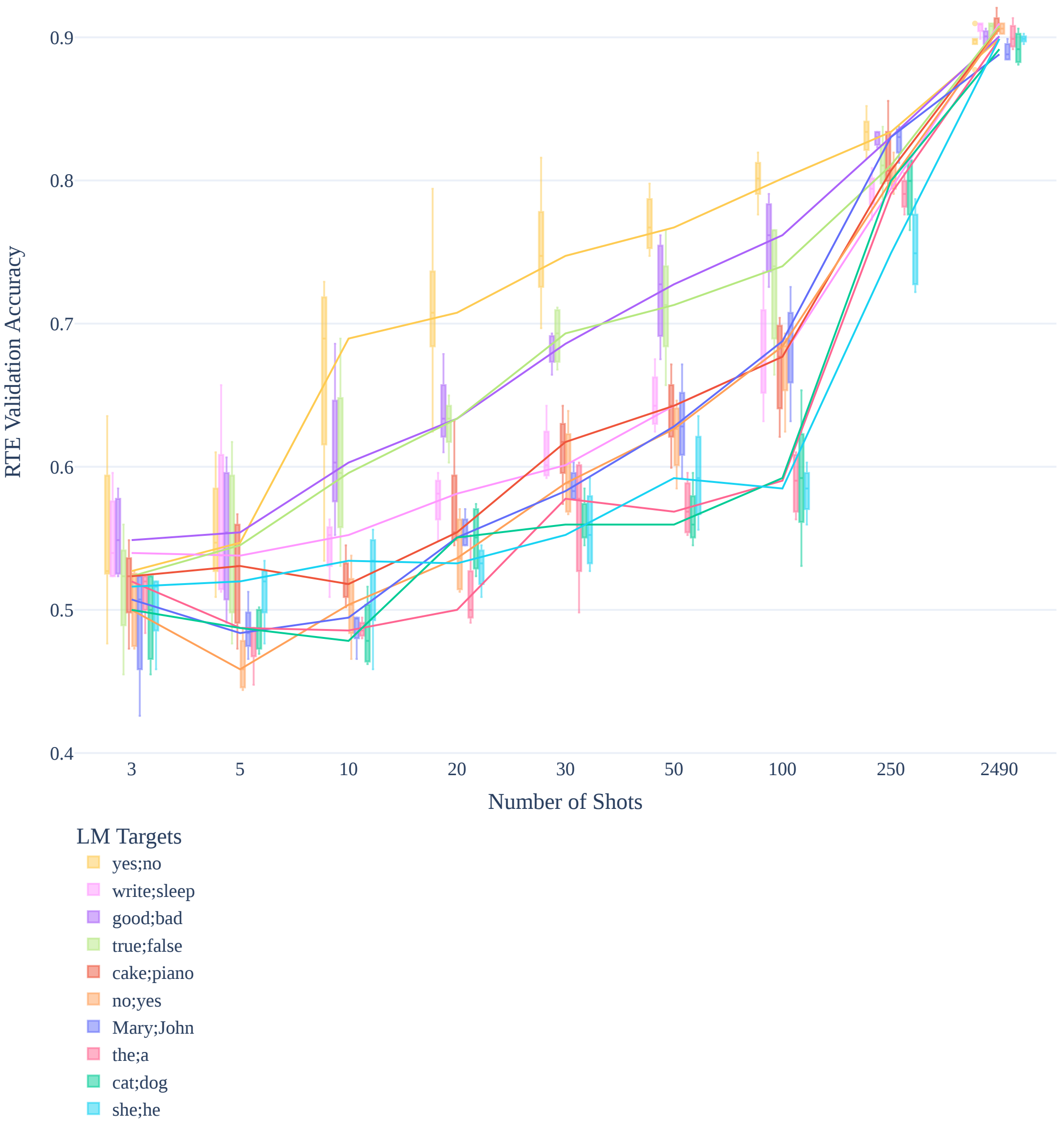

Figure 16: The best-performing misleading prompt, *{premise} Does the paragraph start with "the"? [mask] "{hypothesis}"* with all LM targets.

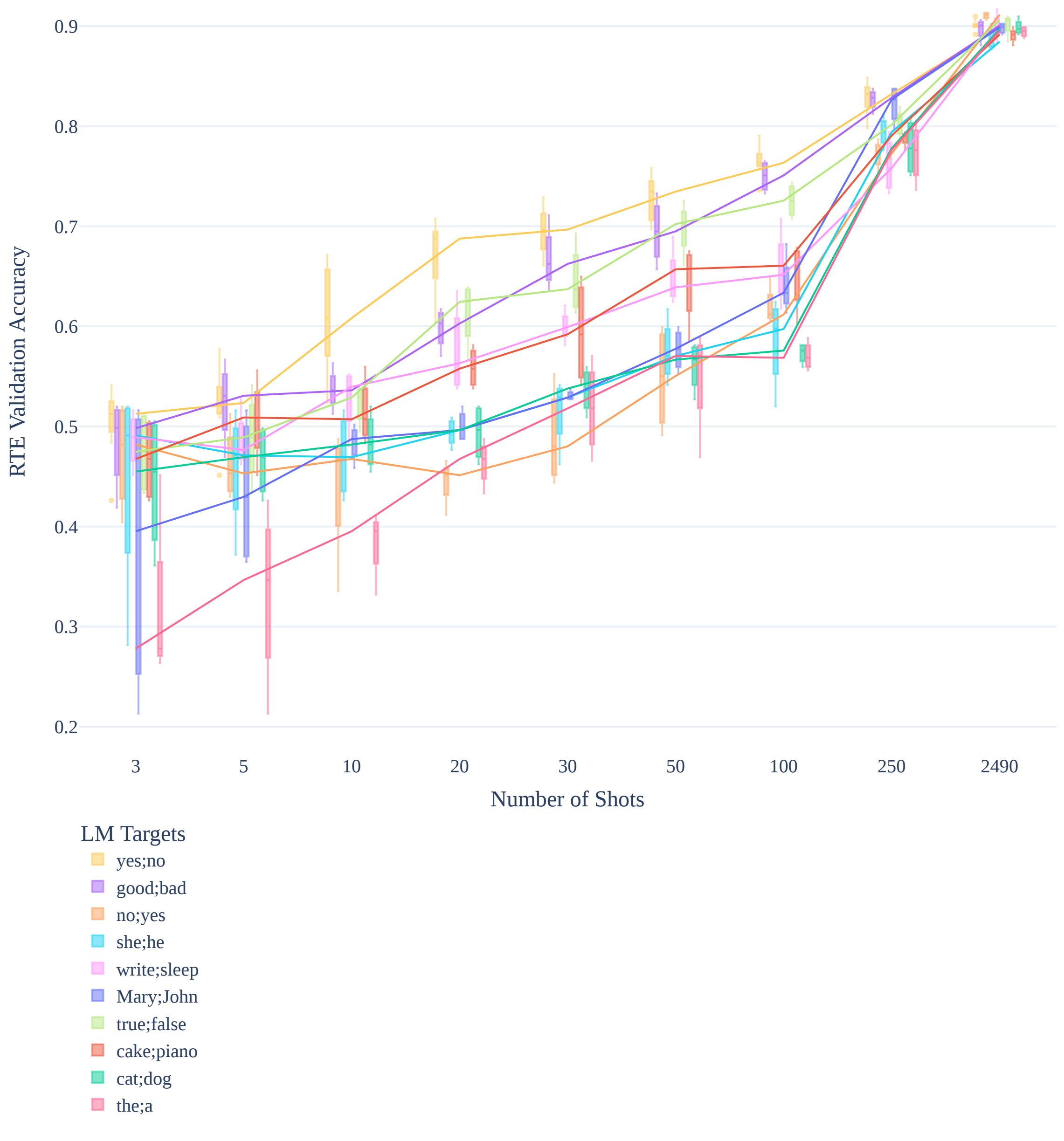

Figure 17: The best-performing null prompt, *{premise} [mask] "{hypothesis}"* with all LM targets.