

Factual Probing Is [MASK]: Learning vs. Learning to Recall

Zexuan Zhong* Dan Friedman* Danqi Chen

Department of Computer Science

Princeton University

{zzhong, dfriedman, danqic}@cs.princeton.edu

Abstract

Petroni et al. (2019) demonstrated that it is possible to retrieve world facts from a pre-trained language model by expressing them as cloze-style prompts and interpret the model’s prediction accuracy as a lower bound on the amount of factual information it encodes. Subsequent work has attempted to tighten the estimate by searching for better prompts, using a disjoint set of facts as training data. In this work, we make two complementary contributions to better understand these factual probing techniques. First, we propose OPTIPROMPT, a novel and efficient method which directly optimizes in continuous embedding space. We find this simple method is able to predict an additional 6.4% of facts in the LAMA benchmark. Second, we raise a more important question: Can we really interpret these probing results as a lower bound? Is it possible that these prompt-search methods learn from the training data too? We find, somewhat surprisingly, that the training data used by these methods contains certain regularities of the underlying fact distribution, and all the existing prompt methods, including ours, are able to exploit them for better fact prediction. We conduct a set of control experiments to disentangle “learning” from “learning to recall”, providing a more detailed picture of what different prompts can reveal about pre-trained language models.¹

1 Introduction

Pre-trained language models like BERT are optimized to predict the distribution of words in an Internet corpus (Devlin et al., 2019). Naturally, this distribution encodes information about world facts. Recently, researchers have taken an interest in measuring how much factual information language models acquire from pre-training. Petroni et al. (2019) formally define this project in the LAMA

*The first two authors contributed equally.

¹The code is publicly available at <https://github.com/princeton-nlp/OptiPrompt>.

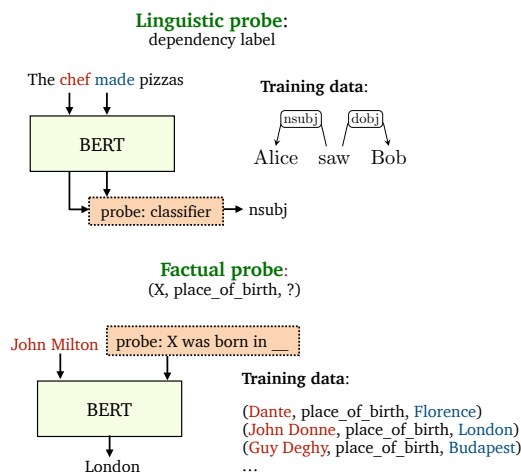


Figure 1: A linguistic probe is trained to predict linguistic annotations given the representations returned by a language model, and evaluated on a held-out set of sentences. A factual probe is trained to predict an *object* for a *subject* and a *relation* using a pre-trained language model, and evaluated on a held-out set of subject-object pairs that express the same relation.

benchmark, which consists of (*subject*, *relation*, *object*) triples along with human-written templates that express each relation. They show that BERT can predict objects given cloze-style prompts—for example, “Dante was born in [MASK]”—and they present their result as a lower bound on the amount of factual information BERT encodes. Subsequent work has attempted to tighten this bound by finding better prompts. Jiang et al. (2020) use text mining and paraphrasing to find a set of candidates and select the prompts that lead to the highest accuracy on a training set. Shin et al. (2020) train a model to generate prompts automatically by searching for the sequence of tokens that maximizes expected likelihood of the gold object label. Both of these methods collect additional triples from Wikidata to use for tuning their prompts.

In this paper, we first take a natural next step in the search for better prompts: rather than confining our search space to discrete input tokens, we di-

rectly optimize in the input embedding space, finding the real-valued input vectors that are most effective at eliciting facts. We also find that initializing with manual prompts can provide a better starting point for the search process. Our approach, OPTIPROMPT, is simple and compute-efficient, and improves accuracy on the LAMA benchmark from 42.2% to 48.6%, compared to previous discrete alternatives. On the more difficult LAMA-UHN split (Poerner et al., 2019), which filters out easy-to-guess entity names, OPTIPROMPT improves accuracy from 31.3% to 38.4%.

At the same time, we observe that prompts that are optimized on training data may exploit some regularities in the underlying distribution of facts. How can we make sure our prompts are recovering information solely from the language model? An analogous question has been explored recently in linguistic probing, which aims to explore the linguistic properties encoded in contextualized word representations (Belinkov et al., 2017; Tenney et al., 2019; Lin et al., 2019)—for example, by seeing if a classifier can predict that “*chef*” is the nominal subject of “*made*” given the representations returned from a language model (Figure 1). Recent work has attempted to disentangle the information encoded in the representations from the information learned by the probe (Hewitt and Liang, 2019; Pimentel et al., 2020; Voita and Titov, 2020; Zhu and Rudzicz, 2020). However, this question has not been yet explored in factual probing, in part because it is assumed that there is no way to predict a knowledge fact simply from observing a non-overlapping set of facts about other entities.² For example, learning that *Dante* was born in *Florence* should tell you nothing about the birthplace of *John Donne*.

We analyze our training data and find that this assumption is not warranted. Even though the training data was collected independently of the LAMA benchmark, there are sufficient regularities in the underlying distribution of Wikidata relations that a naive classifier fit to the training data can achieve surprisingly good performance. Furthermore, our experiments reveal that all the data-driven prompt-search methods, including previous methods and our proposed OPTIPROMPT, are able to exploit this

²In knowledge base completion or link prediction, researchers study how to predict a fact (*Barack Obama, nationality, ?*) from other triples such as (*Barack Obama, place_of_birth, Honolulu*) and (*Honolulu, city_of, USA*). In knowledge probing, the underlying assumption is that one can’t predict facts from the other facts of the *same* relation.

information to achieve better prediction accuracy. Given some training data, a good search algorithm can find prompts that recover a non-trivial number of “facts” from a neural network with randomly initialized parameters, exploiting both simple class statistics and higher order lexical regularities.

This finding makes it challenging to interpret relative accuracy scores on the knowledge probing task. We show how our control experiments allow us to form a more detailed understanding of the behavior of different probes. For example, by partitioning the test set into “easy” examples, which can be predicted by random controls, and “hard” examples, we can form some conclusions about which facts are less likely to have been learned from training data. OPTIPROMPT outperforms prior methods in both subsets, suggesting it is both better at learning from training data and better at eliciting facts from a language model. We conclude with suggestions for future work that might be less susceptible to the confounding effect of training data.

2 Background: Prompting for Facts

2.1 LAMA

The factual probing setting was introduced by the LAMA benchmark (Petroni et al., 2019), which is designed to measure the amount of factual information encoded in a pre-trained language model (LM). In LAMA, a fact is defined as a triple $\langle s, r, o \rangle$, where s is a subject (e.g., *Dante*), r is a relation from a fixed set of relations \mathcal{R} (e.g., *place of birth*), and o is an object (*Florence*). LAMA facts are drawn from a number of sources, including Wikidata, ConceptNet (Speer and Havasi, 2012), and SQuAD (Rajpurkar et al., 2016). We follow recent factual probing work (Jiang et al., 2020; Shin et al., 2020) in focusing on the T-REx split (Elsahar et al., 2018), which contains up to 1000 $\langle s, r, o \rangle$ triples for each of 41 Wikidata relation types. The relation types are divided into three categories: 1-1 includes relations like *capital of*; N-1 includes relations like *place of birth*; and N-M includes relations like *shares border with*. In the LAMA evaluation, each relation is associated with a human-written prompt that contains a single [MASK] token—for example, “[X] was born in [MASK].” To accommodate masked language models such as BERT, LAMA is restricted to facts for which the object label is a single token in a predefined vocabulary

Method	Prompt	Data-driven?
LAMA (Petroni et al., 2019)	[X] is [MASK] citizen	✗
LPAQA (Jiang et al., 2020)	[X] is a citizen of [MASK]	✓
AUTOPROMPT (Shin et al., 2020)	[X] m ³ badminton pieces internationally representing [MASK]	✓
OPTIPROMPT	[X] [V] ₁ [V] ₂ [V] ₃ [V] ₄ [V] ₅ [MASK]	✓
OPTIPROMPT (manual)	[X] [V] ₁ := is [MASK] [V] ₂ := citizen	✓

Table 1: Comparison of prompts for the relation *country of citizenship*. [X] denotes the name of the subject and [MASK] is single-token object label to be predicted. In our OPTIPROMPT approach, we optimize a sequence of learned embeddings $[V]_i \in \mathbb{R}^d$ for each relation type. $[V]_i := w$ indicates that the vector is learned but initialized by the pre-trained embedding of word w and OPTIPROMPT (manual) indicates that we use a manual prompt as initialization (see Section 3 for more details).

\mathcal{V} .³ Given a subject s , a relation prompt t_r , and a masked language model, we can identify the word $\hat{o} \in \mathcal{V}$ to which the LM assigns the highest probability of $P([\text{MASK}] = \hat{o} \mid t_r(s))$, where $t_r(s)$ represents the prompt template with the subject placeholder [X] replaced by s . If \hat{o} is the same as the gold object o , we conclude that the LM encodes information about the fact.

LAMA is an evaluation benchmark, so there is no training data. It is constructed so that a pre-trained language model can be evaluated “off-the-shelf” with no additional fine-tuning. Petroni et al. (2019) remark that their benchmark provides only a lower-bound estimate of the amount of factual information stored in an LM, because their manually written prompts might not be optimal for eliciting facts. Accordingly, subsequent work has focused on tightening this bound by using additional training data to find more optimal prompts.

2.2 LPAQA

Jiang et al. (2020) use a range of text-mining and paraphrasing techniques to generate a set of candidate prompts for each relation. They collect a training dataset from Wikidata, ensuring that there is no overlap with subject-object pairs in the LAMA benchmark, and select prompts by measuring accuracy on this training data. They consider a number of rules for selecting prompts, including top- K baselines and an “optimized ensemble”, which consists of multiple prompts per relation with weights tuned on the training data. Their prompt dataset, LPAQA, is available online.⁴

³Subject names are usually longer, with an average length of 3.7 tokens using the BERT-base-cased vocabulary.

⁴<https://github.com/jzbjyb/LPAQA>

2.3 AUTOPROMPT

Shin et al. (2020) take prompt optimization one step further by training a statistical model, AUTOPROMPT, to search over the space of input tokens for prompts that elicit correct predictions. They collect 1000 $\langle s, r, o \rangle$ triples for each relation type, either from the original T-REx dataset (Elsahar et al., 2018) or from Wikidata, with no triples that appear in the LAMA benchmark. They define a prompt for a given relation r as the subject followed by a fixed number of “trigger” tokens:

$$t_r = [\text{X}] [\text{T}]_1 [\text{T}]_2 \dots [\text{T}]_m [\text{MASK}],$$

where [X] is replaced by the subject, $[\text{T}]_i$ represents a “trigger” token which can be any token in the vocabulary, and the number of [T] tokens is set as a pre-defined number m . The tokens are initialized as [MASK] tokens and then iteratively updated, at each step using a gradient-based searching algorithm (Wallace et al., 2019) to replace one of the trigger tokens with the token that is estimated to maximize the likelihood of the gold label on the training set.

3 Our Approach: OPTIPROMPT

Our approach is motivated by the view that restricting the search to the space of vocabulary tokens is a suboptimal and artificial constraint. In the case of AUTOPROMPT, optimizing over a discrete subspace is also inefficient: at each step we have to enumerate a set of candidate tokens, replace the selected trigger token, and re-run the model (Shin et al., 2020). The examples in Table 1 also illustrate that optimized textual prompts can be opaque, despite consisting of tokens from the English vocabulary. This undermines one argument in favor of natural language prompts, which is that they are

Method	1-1	N-1	N-M	All	UHN
Majority	1.8	23.9	22.0	22.0	23.8
LAMA (manual)	68.0	32.4	24.7	31.1	21.8
LPAQA (manual + paraphrased)	65.0	35.9	27.9	34.1	28.7
AUTOPROMPT (5 [T] s)	58.0	46.5	34.0	42.2	31.3
OPTIPROMPT (5 [V] s)	49.6	53.1	39.4	47.6	37.5
OPTIPROMPT (10 [V] s)	60.7	53.2	39.2	48.1	37.9
OPTIPROMPT (manual)	59.6	54.1	40.1	48.6	38.4

Table 2: Micro-averaged results (top-1) on the LAMA benchmark using the BERT-base-based model, averaged over relations. UHN stands for UnHelpfulNames (Poerner et al., 2019), which is a subset of LAMA where questions with helpful entity names were deleted. The LAMA results are broken down by relation category. Examples from each category are *capital of* (1-1), *place of birth* (N-1), and *shares border with* (N-M).

human readable so might be easier to interpret.

OPTIPROMPT In this view, we propose OPTIPROMPT, a method for continuous prompt optimization. Rather than limiting the search to the space of discrete tokens, OPTIPROMPT searches for optimal prompts directly, composing prompts using any vector in the embedding space. We first follow AUTOPROMPT and define a prompt in the following form:

$$t_r = [X] [V]_1 [V]_2 \dots [V]_m [\text{MASK}],$$

where each $[V]_i \in \mathbb{R}^d$ is a dense vector with the same dimension as the LM’s input embedding (e.g., 768 for BERT-base) and the number of $[V]$ vectors is set to a pre-defined number m .

Treating prompts as dense vectors allows us to search for optimal prompts much more efficiently. Given some initial values for $[V]_i$, we keep all other model parameters fixed and use gradient-descent to minimize the negative log-likelihood of a training set:

$$\mathcal{L}_r = -\frac{1}{|D_r|} \sum_{(s,o) \in D_r} \log P([\text{MASK}] = o | t_r(s)),$$

where D_r is the set of (*subject*, *object*) pairs with relation r and t_r represents the prompt template for relation r with subject tokens s substituted for the placeholder $[X]$.

In this basic form, we pick a fixed value for m (treated as a hyperparameter) and randomly initialize all the $[V]$ tokens. We also consider a more sophisticated form of using manual prompts (we use the prompts provided in the LAMA benchmark) to decide the number as well as the *position* of the $[V]$ tokens for each relation and initialize each

$[V]_i$ with the pre-trained input embedding for the corresponding tokens in the manual prompt. As shown in Table 1, we can convert a manual prompt “[X] is [MASK] citizen” into

$$t_r = [X] [V]_1 [\text{MASK}] [V]_2,$$

and use the embeddings of *is* and *citizen* to initialize $[V]_1$ and $[V]_2$ respectively. Our motivation is that a good initialization is likely to be important in this challenging non-convex optimization problem.

Setup We train OPTIPROMPT using the data collected by Shin et al. (2020), which contains 800 training examples with 200 held out for development. For our main experiments, we probe the BERT-base-based model and we compare other pre-trained language models in Appendix C. We report top-1 micro-averaged accuracy:

$$\frac{1}{|\mathcal{R}|} \sum_{r \in \mathcal{R}} \frac{1}{|D_r|} \sum_{(s,o) \in D_r} \mathbb{1}[\hat{o} = o],$$

where \mathcal{R} is the set of relations, D_r is the set of (*subject*, *object*) pairs with relation r , and $\hat{o} = \arg \max_o P([\text{MASK}] = o | t_r(s))$. More implementation details can be found in Appendix B.1.

LAMA results Our results are in Table 2. Overall, OPTIPROMPT outperforms the previous reported results in terms of accuracy on the LAMA benchmark. Compared to AUTOPROMPT⁵, our

⁵For AUTOPROMPT, we obtain a slightly different accuracy 42.2% by evaluating their released prompts, instead of 42.9% reported in their paper. We suspect that this is due to a discrepancy in the vocabulary used in different papers. We use the vocabulary provided in the LAMA benchmark for all the evaluation: <https://github.com/facebookresearch/LAMA#unified-vocabulary>.

models perform 5.4%–6.4% higher on LAMA and 6.2%–7.1% on the more-difficult LAMA-UHN benchmark. The improvement is consistent across all categories, with the exception of the “1-1” category, which contains two relations, *capital* and its inverse, *capital of*. Interestingly, the prompt that yields the best results in this category is the manual prompt, with LPAQA and AUTOPROMPT prompts performing steadily worse. We speculate that there are very few prompts that elicit this relation with high accuracy and they are difficult to find via stochastic, non-convex optimization.

We also find that initializing the prompt vectors using the manually written prompts improves performance consistently. This confirms our intuition that the manual initialization provides a good prior for finding a good solution in the non-convex optimization problem. The results are broken down by relation in Table 8 in the Appendix.

4 Can We Trust Optimized Prompts?

Our factual probing results confirm that OPTIPROMPT is an effective approach, outperforming the best previous method by 6.4% on the LAMA benchmark. However, can we conclude that BERT encodes 6.4% more facts than was previously known? Our prompts, like LPAQA and AUTOPROMPT, are optimized on in-distribution Wikidata relations, which raises the possibility that they exploit some regularities in the underlying fact distribution. In this section we aim to answer two questions. First, are there patterns in the Wikidata fact distribution that statistical model could theoretically exploit to predict unseen facts? Second, are optimized prompts capable of exploiting these patterns in practice?

4.1 Facts can be predicted from training data

We first examine whether it is possible to predict any facts by just looking at the training data. The simplest pattern is the class prior $P(o \mid r)$: if one or two object labels dominate the relation r , it is easier to guess them regardless of the subject entity. A more sophisticated pattern is to find a correlation between subject tokens and object labels—that is, to estimate $P(o \mid r, w_1, \dots, w_{|s|})$, where $w_1, \dots, w_{|s|} \in \mathcal{V}$ are the tokens of the subject name. To see whether such patterns exist, we fit two simple probabilistic models to the Wikidata training set collected by Shin et al. (2020). The first model always predicts the majority class, with class priors learned from the training data, and the sec-

Relation	Class Prior	Naive Bayes
All	17.3	24.6
1-1	0.2	0.3
N-1	23.2	28.6
N-M	11.0	21.8
<i>member of</i>	2.2	59.6
<i>manufacturer</i>	8.9	62.0

Table 3: Results for simple classifiers fit to the Wikidata training data and evaluated on the LAMA test set. We highlight two relations for which object labels are correlated with particular subject tokens: In the *member of* category, the model appears to learn that any subject with “football” in its name, such as *Ghana Football Association*, is likely to be a member of *FIFA*. In the *manufacturer* category, the model learns to predict that *Chevrolet* manufactures the *Chevrolet Impala*, *BMW* manufactures the *BMW M Coupe*, and so on.

ond is a Naive Bayes classifier (bag-of-words) with add-one smoothing (see details in Appendix B.2). Table 3 shows the accuracy of these models on the LAMA benchmark, averaged over relations. The majority class model performs well because, on some relations, well over half of the examples are from the majority class.⁶ The Naive Bayes baseline performs even better in all categories by learning correlations between subject tokens and object labels. This analysis complements an observation of Poerner et al. (2019), who point out that BERT can exploit superficial information in a cloze prompt to “guess” the correct answer—for example, predicting that people with stereotypically Italian names were likely born in *Rome*. Our results show that it is possible to learn these correlations even without prior information about entity names, and there might be other, subtler patterns in the Wikidata distribution.

4.2 Prompts can exploit training data

We have shown that the training data clearly encodes certain regularities and simple statistical models can learn to fit the training data. In the following, we study whether a prompt optimization method built with pre-trained language models, is expressive enough to exploit these regularities in practice. We attempt to answer this question by means of two random controls, inspired by similar proposals from linguistic probing. In our *Random Model* (RM) baseline, we optimize prompts to elicit

⁶These include *native language* (60% French) and *continent* (72% Antarctica).

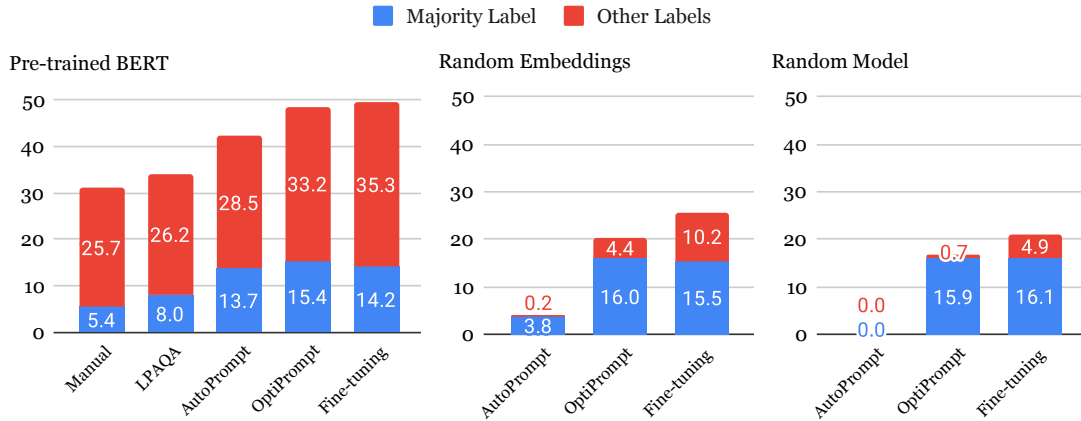


Figure 2: Accuracy on LAMA obtained by prompting BERT-base-based, either the pre-trained model, reinitializing the input embeddings, or reinitializing all parameters. Each bar represents total accuracy micro-averaged over relations and divided into two categories: accuracy obtained by predicting the training set majority class label, and accuracy obtained by predicting other object labels. We also fine-tune BERT, which, in the random control settings, can be thought of as a better lower bound on the entropy of the task distribution.

facts from a neural network with the same architecture as the pre-trained LM but with randomly initialized parameters. This is analogous to a *control function* Pimentel et al. (2020), a function that removes information from a linguistic representation. Any successful predictions in this setting must be the result of optimizing on training data. We also consider a *Random Embeddings* (RE) baseline, where we reinitialize only the input embeddings.⁷ This is analogous to a *control task* (Hewitt and Liang, 2019), a variant of the probing task in which word types are associated with random labels.⁸ Our motivation is that the Random Model setting is more difficult to optimize, so might underestimate the ways a prompt model could exploit information from the training data. Finally, we directly fine-tune a reinitialized BERT model on the training data with the goal of getting a better estimate of the number of LAMA facts that could be predicted from the training data.

The results are shown in Figure 2 (see implementation details and more results in Appendix B.1 and Table 8). In the Random Embeddings setting, both AUTOPROMPT and OPTIPROMPT are capable of finding prompts that elicit some correct predictions. In the Random Model setting, AUTOPROMPT gets

⁷In the RE setting, the classifier head of the model is also reinitialized, as the output embeddings are tied to the input embeddings.

⁸Hewitt and Liang (2019) consider tasks like part-of-speech tagging, where each word type can be associated with a randomly selected tag. We randomize the inputs rather than the labels, which preserves most of the statistical correlations between subject token types and object labels but removes lexical information from the embeddings.

0% of predictions correct, presumably because it is more difficult to optimize, but OPTIPROMPT is still capable of finding successful prompts. Most successful predictions are obtained by finding a prompt that elicits the majority class label, although OPTIPROMPT also makes a number of correct predictions that cannot be attributed to this strategy. Our qualitative analysis suggests that these prompts exploit both class statistics and correlations between objects and subject tokens (Appendix A.2).

Fine-tuning BERT results in even higher accuracy, indicating that there are patterns that prompts fail to exploit. The random controls represent a challenging setting for prompt optimization, and it is possible that the prompts are better exploiting the training data when they have access to full pre-trained BERT model. We find evidence that this is the case by calculating how often each prompt elicits the training class majority label on LAMA, plotting the results in Figure 3. Both AUTOPROMPT and OPTIPROMPT are prone to over-predicting the majority class label. For example, although AUTOPROMPT gets 0% accuracy in the RM setting, it finds a prompt that elicits the majority label more than 95% of the time for six relations when optimized on the pre-trained BERT model.⁹

LPAQA prompts predict the majority class less often, possibly because they are less effective at

⁹Shin et al. (2020) attempt to prevent the model from using this strategy by filtering out prompts that contain proper nouns or gold object labels, but this evidently is not enough. For example, the prompt for the *position held* relation is “[X] explorers voting municipal → consecrated [MASK].”, which elicits *bishop* for 100% of LAMA examples.

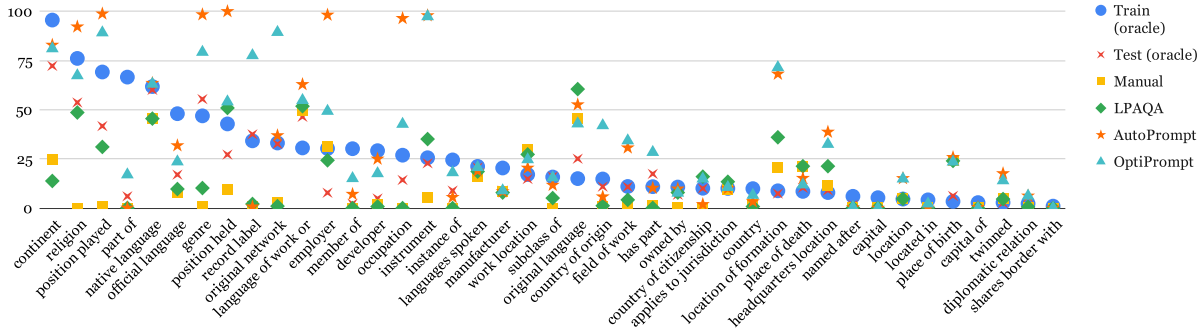


Figure 3: The percentage of LAMA examples for which a prompt elicits the training set majority label, compared with the percentage of training and test facts with that label. Optimized prompts show a strong tendency to over-predict the majority class relative to manual prompts and the ground truth. “Train (oracle)” is calculated from the set of Wikidata facts collected by Shin et al. (2020), which is used to train AUTOPROMPT and OPTIPROMPT.

Method	All (34,039)	Easy (10,546)	Hard (23,493)
Manual	31.1	41.5	24.3
LPAQA	34.1	47.0	25.6
AUTOPROMPT	42.2	68.2	26.7
OPTIPROMPT	48.6	75.6	33.0

Table 4: Accuracy on LAMA partitioned into easy examples or hard examples, micro-averaged over relations. Easy facts are the facts that can be predicted by fine-tuning a BERT model with either randomly initialized parameters or randomly initialized token embeddings, or by the Naive Bayes model described in Section 4. Hard examples are everything else. The numbers in parentheses denote the size of each subset.

fitting the training distribution. However, it is still clear that LPAQA prompts also encode distribution of the training data. For instance, the highest ranked *occupation* prompts discovered by LPAQA include prompts such as “[MASK] and actors [X]” and “[MASK] and player [X].”,¹⁰ which reflect several of the most common occupations in Wikidata. We also discuss examples in Appendix A.2 of cases where LPAQA finds subtle changes to the prompt template that leads the model to predict the majority label more often than the manual prompt and the true test distribution. All the above evidence shows that optimized prompts can learn new facts to some extent.

5 How to Interpret Probing Results?

Our analysis in Section 4.2 shows that optimized prompts can predict new facts from training data. How can we interpret our factual probing results in

¹⁰<https://github.com/jzbyb/LPAQA/blob/master/prompt/paraphrase/P106.jsonl>

this light? In order to get another perspective of the relative improvement, we partition LAMA into an easy subset and a hard subset (examples from each subset can be found in Table 5). The easy subset consists of the facts that can be correctly predicted by any of three models fit to the training data: the Naive Bayes model described in Section 4.2 and a fine-tuned BERT model with either token embeddings reinitialized or all parameters reinitialized. The easy subset serves as an estimate of the set of facts that can be predicted from training data. The hard subset consists of the remain facts. Table 4 shows the results of each prompt on these two subsets of LAMA (the per-relation results are given in Table 9). First, we observe that all the probing methods achieve a much higher accuracy on the easy subset. Using more sophisticated prompt optimization techniques tends to result in big improvements on the easy subset of LAMA and smaller improvements on the hard subset. OPTIPROMPT outperforms AUTOPROMPT by 7.4% on the easy examples; while on the hard examples, where we filtered out facts that we know can be predicted from the training data, OPTIPROMPT also yields a big improvement (+6.3%). This suggests that OPTIPROMPT is both better at learning from training data and better at eliciting facts from an LM.

For a more qualitative analysis, we randomly sample ten facts from each subset, keeping only facts that are predicted correctly by at least one model and exclude examples that have the majority class label. The examples, shown in Table 5, give a better idea of the types of predictions elicited by different prompts. For example, both AUTOPROMPT and OPTIPROMPT appear to be exploiting the training data in some cases. In the easy subset, they elicit more accurate predictions on cases when

Rel.	Fact (manual template)	NB	Manual	LPAQA	Auto	Opti
P103	The native language of <i>Jan van Krimpen</i> is <u>Dutch</u> .	<u>Dutch</u>	<u>Dutch</u>	<u>Dutch</u>	<u>Dutch</u>	<u>Dutch</u>
P279	<i>edible mushroom</i> is a subclass of <u>mushroom</u> .	protein	<u>mushroom</u>	category	<u>mushroom</u>	<u>mushroom</u>
P1001	<i>Governor of Tasmania</i> is a legal term in <u>Tasmania</u> .	Canada	Australia	Australia	<u>Tasmania</u>	<u>Tasmania</u>
P106	<i>Dude Harlino</i> is a <u>actor</u> by profession .	<u>actor</u>	lawyer	wrestler	politician	<u>actor</u>
P27	<i>Jens Evensen</i> is <u>Norway</u> citizen .	<u>Norway</u>	Danish	Sweden	Norway	Norway
P176	<i>Porsche Cayenne</i> is produced by <u>Porsche</u> .	Honda	<u>Porsche</u>	<u>Porsche</u>	<u>Porsche</u>	<u>Porsche</u>
P279	<i>United States H-class submarine</i> is a subclass of <u>submarine</u> .	protein	submarines	<u>submarine</u>	<u>submarine</u>	<u>submarine</u>
P138	<i>Milwaukee Mitchell International Airport</i> is named after <u>Milwaukee</u> .	Peter	Mitchell	Mitchell	<u>Milwaukee</u>	<u>Milwaukee</u>
P176	<i>BMW E9</i> is produced by <u>BMW</u> .	<u>BMW</u>	<u>BMW</u>	<u>BMW</u>	<u>BMW</u>	<u>BMW</u>
P1412	<i>Tom Mann</i> used to communicate in <u>English</u> .	French	<u>English</u>	<u>English</u>	<u>English</u>	<u>English</u>
P937	<i>Francis Hagerup</i> used to work in <u>Oslo</u> .	London	London	London	Copenhagen	<u>Oslo</u>
P127	<i>Apple Store Online</i> is owned by <u>Apple</u> .	Germany	Apple	Apple	Apple	Apple
P1412	<i>Berengaria of Castile</i> used to communicate in <u>Spanish</u> .	French	<u>Spanish</u>	Latin	<u>Spanish</u>	<u>Spanish</u>
P176	<i>SNES-CD</i> is produced by <u>Sony</u> .	Honda	Sega	Sony	IBM	IBM
P47	<i>Honduras</i> shares border with <u>Guatemala</u> .	Lyon	<u>Guatemala</u>	<u>Guatemala</u>	<u>Guatemala</u>	<u>Guatemala</u>
P937	<i>David Ben-Gurion</i> used to work in <u>Jerusalem</u> .	London	<u>Jerusalem</u>	<u>Jerusalem</u>	<u>Jerusalem</u>	<u>Jerusalem</u>
P19	<i>Peter I of Serbia</i> was born in <u>Belgrade</u> .	Paris	<u>Belgrade</u>	<u>Belgrade</u>	<u>Belgrade</u>	<u>Belgrade</u>
P31	<i>Dally M Medal</i> is a <u>award</u> .	album	prize	prize	award	award
P30	<i>Snowdon</i> is located in <u>Europe</u> .	Antarctica	Wales	Europe	Antarctica	Antarctica
P937	<i>William Lyon Mackenzie King</i> used to work in <u>Ottawa</u> .	London	Canada	London	Montreal	<u>Ottawa</u>

Table 5: Randomly sampling 10 examples each from LAMA-easy (the first block) and LAMA-hard (the second block), only keeping examples that are predicted correctly by at least one model and that do not have the majority label. NB: the Naive Bayes model (Section 4.2), Auto: AUTOPROMPT, Opti: OPTIPROMPT. The correct predictions are underlined.

the answer is a token in the subject name. In the hard subset, they show signs of having over-fit to the training distribution, incorrectly predicting the most common object labels for *continent* (*Antarctica*) and *manufacturer* (*IBM*). OPTIPROMPT performs better than the other prompts on some facts in both categories. On an easy *profession* example, while AUTOPROMPT incorrectly predicts the majority label (*politician*), OPTIPROMPT—along with our Naive Bayes model—apparently encodes a lexical correlation between some aspect of the subject’s name and the correct label, *actor*. On the other hand, OPTIPROMPT out-performs the other prompts on two more difficult examples: “*Francis Hagerup* used to work in *Oslo*” and “*William Lyon Mackenzie King* used to work in *Ottawa*.” In both cases, LPAQA predicts the training majority label (*London*), AUTOPROMPT gets geographically closer (*Copenhagen* and *Montreal*), and OPTIPROMPT predicts the correct city.

We note that we cannot conclude that there is no way to predict these “hard” facts from training data. A more general limitation of this analysis is that it does not allow us to say which strategy a model uses to make a particular prediction. Many facts can be predicted either by learning the class prior; by learning a lexical correlation between subject tokens and objects; by exploiting lexical information from the LM; or because the LM genuinely encodes information about a particular entity. Still,

the qualitative examples reveal interesting patterns in the behavior of the different prompt models that could not be observed from the summary accuracy results on the LAMA benchmark, and looking at specific predictions across a number of prompts gives us more evidence for deciding what kind of information the LM encodes about a particular fact.

6 Discussion

Our experiments show that OPTIPROMPT is an effective optimization algorithm, outperforming prior work at the task of eliciting facts from a pre-trained language model. However, our results are complicated by the fact that any data-driven optimization can find prompts that encode new information from the training data. This leaves open the question of which method we should select if we are interested in factual probing.

Continuous vs. discrete prompts We find that both continuous and discrete optimization are capable of finding prompts that exploit the training data. Even when the prompt is discrete, it is rarely clear why a prompt elicits a particular prediction.¹¹ Hence, we believe that continuous prompting is more preferable, because it is easier and more efficient to optimize, and makes better predictions (in both easy and hard subsets). On the other hand,

¹¹For an illustration, see Appendix A.2 for a list of the AUTOPROMPT templates that elicit the majority class label more than 95% of the time.

one drawback of OPTIPROMPT (which is shared by AUTOPROMPT) is that we need white-box access to the LM to compute the gradients. Discrete prompts will still be necessary in cases where the model parameters are not available, for example in the case of very large language models that are provided over an API.

Learning vs. learning to recall Regardless of how we choose to optimize prompts, it remains difficult to say why a model made a particular prediction—whether it was learned from training data or encoded in the LM. Some avenues for future work might be to consider techniques for attributing predictions to specific training instances, with the goal of developing a causal understanding of how facts are acquired during pre-training or prompt optimization. More generally, our real goal is to understand how pre-trained language models learn and represent information. Prompt-based probing might provide some insight into this question, but we hope that future research will eventually be able to provide more mechanistic explanations for neural network behavior. For example, it would be interesting to understand how information about entities is laid out in neural network parameters and later retrieved in response to an input prompt.

7 Related Work

Our work follows from the line of factual probing experiments initiated by [Petroni et al. \(2019\)](#), who introduced the LAMA benchmark for cloze-style factual probing. Subsequent work on LAMA has introduced data-driven methods for optimizing prompts ([Jiang et al., 2020](#); [Shin et al., 2020](#)). [Poerner et al. \(2019\)](#) point out that many facts in LAMA can be predicted using lexical clues, and they introduce a new benchmark, LAMA-UHN, that is less susceptible to these heuristics. Our work follows these projects by introducing (a) more effective techniques for optimizing prompts, and (b) a more comprehensive approach for accounting for the role of train/test overlap. Concurrently with this work, other authors explore continuous prompt optimization: [Haviv et al. \(2021\)](#) use an encoder to map a manually written prompt to a sequence of continuous vectors, which are then replaced with the discrete tokens that are nearby in embedding space; [Li and Liang \(2021\)](#) propose Prefix-Tuning, which fine-tunes the left-most hidden representations in auto-regressive language models; [Liu et al. \(2021\)](#) use an LSTM to generate a sequence of prompt

vectors. Prompting has been explored more generally as a method for achieving “few-shot” learning with language models ([Brown et al., 2020](#); [Schick and Schütze, 2020](#); [Gao et al., 2020](#)).

Linguistic probing is an extensive area of research that we do not attempt to summarize here (see [Rogers et al., 2020](#) for an overview). Our work is most related to recent proposals about how to measure whether a probe is extracting information from a representation or learning to predict the annotation from probe training data. These include random baselines ([Hewitt and Liang, 2019](#)) and information-theoretic measurements ([Voita and Titov, 2020](#)). We adopt the notion of control functions from [Pimentel et al. \(2020\)](#). Our study also relates to a larger category of work diagnosing “shortcut learning” ([Geirhos et al., 2020](#)) in neural NLP models. [McCoy et al. \(2019\)](#) discover that models like BERT are often “right for the wrong reason”, exploiting shallow heuristics rather than underlying linguistic structure, and similar effects have been discovered in many other tasks ([Sugawara et al., 2018](#); [Wallace et al., 2019](#)).

8 Conclusion

We introduce OPTIPROMPT, an effective continuous method for optimizing prompts. Applied to factual probing, OPTIPROMPT outperforms the best previous prompt method by 6.4% on the LAMA benchmark. We find that the typical training data used for prompt optimization reveals useful information about the underlying task distribution, to the point that search algorithms can find prompts that recover “facts” even from a randomly initialized model. By comparing the predictions of different prompt methods across our different controls we can form a more detailed understanding of how different prompts behave and what they can reveal about pre-trained language models.

Ethical Considerations

Our experiments illustrate that the “facts” recovered from a pre-trained language model should not be considered real facts. Optimizing any kind of statistical model for factual prediction is likely to devolve into stereotype-learning as the model learns lexical correlations between entity names and object labels. This problem is more pronounced if our training distribution comes from a source like Wikidata, which we find to be imbalanced. More generally, language models that are trained on the Internet will model the toxic and harmful language that is found there, a well-documented finding for pre-trained language models like BERT (e.g., Gehman et al., 2020; Nadeem et al., 2020). Using such models for factual prediction is liable to amplify those biases. OPTIPROMPT is intended to be a diagnostic tool and general-purpose optimization method, not a way to use BERT as a knowledge base.

Acknowledgement

We thank Zhengbao Jiang for answering questions about LPAQA. We thank the members of the Princeton NLP group and the anonymous reviewers for their valuable comments and feedback. This work is supported in part by a Graduate Fellowship at Princeton University.

References

- Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017. What do neural machine translation models learn about morphology? In *Association for Computational Linguistics (ACL)*, pages 861–872.
- Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 4171–4186.
- Hady Elsahar, Pavlos Vougiouklis, Arslan Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. 2018. T-REx: A large scale alignment of natural language with knowledge base triples. In *Eleventh International Conference on Language Resources and Evaluation (LREC)*.
- Tianyu Gao, Adam Fisch, and Danqi Chen. 2020. Making pre-trained language models better few-shot learners. *arXiv preprint arXiv:2012.15723*.
- Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A Smith. 2020. Realtocixityprompts: Evaluating neural toxic degeneration in language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 3356–3369.
- Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. 2020. Shortcut learning in deep neural networks. *arXiv preprint arXiv:2004.07780*.
- Adi Haviv, Jonathan Berant, and Amir Globerson. 2021. BERTese: Learning to speak to BERT. In *European Chapter of the Association for Computational Linguistics (EACL)*.
- John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 2733–2743.
- Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association of Computational Linguistics (TACL)*.
- 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. In *International Conference on Learning Representations (ICLR)*.
- Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.
- Yongjie Lin, Yi Chern Tan, and Robert Frank. 2019. Open sesame: Getting inside bert’s linguistic knowledge. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 241–253.
- Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021. GPT understands, too. *arXiv preprint arXiv:2103.10385*.
- 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*.
- Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448.

- Moin Nadeem, Anna Bethke, and Siva Reddy. 2020. StereoSet: Measuring stereotypical bias in pre-trained language models. *ArXiv*, abs/2004.09456.
- Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 2463–2473.
- Tiago Pimentel, Josef Valvoda, Rowan Hall Maudslay, Ran Zmigrod, Adina Williams, and Ryan Cotterell. 2020. Information-theoretic probing for linguistic structure. In *Association for Computational Linguistics (ACL)*, pages 4609–4622.
- Nina Poerner, Ulli Waltinger, and Hinrich Schütze. 2019. BERT is not a knowledge base (yet): Factual knowledge vs. name-based reasoning in unsupervised qa. *arXiv preprint arXiv:1911.03681*.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 2383–2392.
- Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in bertology: What we know about how bert works. *Transactions of the Association of Computational Linguistics (TACL)*, 8:842–866.
- Timo Schick and Hinrich Schütze. 2020. It’s not just size that matters: Small language models are also few-shot learners. *arXiv preprint arXiv:2009.07118*.
- Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Automatic prompt construction for masked language models. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235.
- Robyn Speer and Catherine Havasi. 2012. Representing general relational knowledge in conceptnet 5. In *LREC*, pages 3679–3686.
- Saku Sugawara, Kentaro Inui, Satoshi Sekine, and Akiko Aizawa. 2018. What makes reading comprehension questions easier? In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 4208–4219.
- Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel Bowman, Dipanjan Das, et al. 2019. What do you learn from context? probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations (ICLR)*.
- Elena Voita and Ivan Titov. 2020. Information-theoretic probing with minimum description length. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 183–196.
- Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing NLP. In *Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, page 2153–2162.
- Thomas Wolf, Julien Chaumond, Lysandre Debut, Victor Sanh, Clement Delangue, Anthony Moi, Pierric Cistac, Morgan Funtowicz, Joe Davison, Sam Shleifer, et al. 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.
- Zining Zhu and Frank Rudzicz. 2020. An information theoretic view on selecting linguistic probes. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 9251–9262.

A Detailed Results

A.1 Breakdown Accuracy for LAMA

Table 7 shows the per-relation accuracy for each prompting method. In many cases, we can better understand the probing results by examining the specific predictions each method makes.

A.2 Exploiting Training Data

Majority class baseline Figure 3 shows that all optimized prompts have a tendency to over-predict the majority class label. This behavior is most pronounced in the gradient-based methods (AUTOPROMPT and OPTIPROMPT). It is not always clear why a particular prompt elicits these predictions. For example, Shin et al. (2020) attempt to prevent AUTOPROMPT from “cheating” by filtering out prompts that contain proper nouns or gold object labels, but there are still six relations for which AUTOPROMPT elicits the majority label more than 95% of the time. The AUTOPROMPT prompts for these relations are:

- *genre = jazz*: “[X] freaking genre orchestra fiction acid [MASK].”
- *position played = midfielder*: “[X] played colors skier \u2194 defensive [MASK].”
- *occupation = politician*: “[X] supporters studied politicians musician turned [MASK].”
- *employer = IBM*: “[X] 1987adeNBC computing succeeded [MASK].”
- *instrument = piano*: “[X] playingdrum concertoative electric [MASK].”
- *position held = bishop*: “[X] explorers voting municipal \u2192 consecrated [MASK].”

This illustrates that even discrete prompts are capable of finding prompts that elicit a specific label from an LM, and the mechanism by which these prompts elicit the prediction is often obscure.

Perhaps more surprisingly, even LPAQA occasionally finds prompts that are more likely to elicit the majority label compared to the manual prompt. The changes in these cases are often very subtle. For example, the manual prompt for the *position of* relation is “[X] has the position of [MASK]” and the LPAQA prompt is “[X] has the position of a [MASK]”. Simply inserting the determiner “a” into the prompt leads BERT to predict the majority label, *bishop*, more than five times as often

compared to the manual prompt (50.9% vs. 9.5%), and almost twice as often relative to the true distribution in the LAMA benchmark (27.3%). This suggests that even simple data-driven methods can find prompts that encode some regularities in the training data and result in over-estimates of the number of facts in the language model.

Control result details Table 8 shows the accuracy of optimized prompts under our random controls (Section 4.2) and also shows how much accuracy can be attributed to predict the majority class label. AUTOPROMPT cannot predict any facts in the Random Model setting but performs decently on several relations in the Random Embeddings setting by predicting the majority class. For reasons we cannot entirely explain, there is one relation, *occupation*, for which AUTOPROMPT’s performance cannot be attributed to the class prior. The correct predictions in this category are all a result of predicting *actor*, which AUTOPROMPT predicts 23.3% of the time. (The most frequent label in the training data is *politician*.) Other high frequency predictions for this relation include *jet*, *wool*, and *smart*. Notably, even when AUTOPROMPT finds a prompt that can draw out the class prior, it typically does not elicit the class prior 100% of the time.

OPTIPROMPT is more successful at exploiting the training data. In the Random Model setting, virtually all correct predictions can be attributed to the majority class, which OPTIPROMPT can frequently elicit for all inputs. One noteworthy exception is *languages spoken*, where OPTIPROMPT is able to successfully classify subjects as speaking either *English* or *French* in some cases. It is not immediately clear what decision rule the model learns for these predictions—for example, it could be that the model predicts either *English* or *French* at random, in rough proportion to the training distribution; or the model is able to use the correlations between names and spoken languages. In any case, the results illustrate that optimized prompts can learn more sophisticated strategies than simply predicting the majority class, even given a Transformer that contains no prior information at all.

A.3 LAMA-easy and LAMA-hard

Table 9 shows the accuracy of different prompts on the easy and hard subset of LAMA described in Section 5. All of the optimized models tend to perform better on LAMA-easy compared to LAMA-hard, and OPTIPROMPT out-performs AU-

Method	BERT (110M)		BERT (330M)		RoBERTa (330M)		ALBERT (235M) [†]	
	Pre-T.	Rand M.	Pre-T.	Rand M.	Pre-T.	Rand M.	Pre-T.	Rand M.
Manual	30.6	-	32.2	-	23.6	-	27.4	-
LPAQA	35.6	-	36.2	-	29.3	-	29.8	-
AUTOPROMPT	44.6	0.0	44.5	0.1	38.6	0.0	33.2	0.0
OPTIPROMPT	50.8	19.9	52.7	19.3	47.8	19.6	44.6	16.9
Fine-tuning	51.9	19.8	54.9	19.6	52.3	21.4	52.8	21.1

Table 6: Comparison of different pre-trained LMs. We downsample the LAMA test set to make sure that in each sample, the object is a single token for all the models. [†]: there is parameter sharing in ALBERT models so the actual models are much bigger. *Pre-T.*: pre-trained language models. *Rand M.*: randomly initialized models.

TOPROMPT in both categories. For example, on the *shares border* relation, OPTIPROMPT achieves an improvement on both easy questions (*Campagnano di Roma, Rome*) and hard ones (*Chiapas, Veracruz*). But note that high accuracy on LAMA-easy does not necessarily mean that a prompt encodes information about the fact distribution. For example, all prompts, including the manually written prompts, perform well on the easy examples in the *capital* relation. This category includes such facts as “The capital of Sarajevo Canton is Sarajevo,” which evidently do not require very much tuning to predict.

B Implementation Details

B.1 Prompt Optimization

We implement OPTIPROMPT based on the HuggingFace’s Transformers (Wolf et al., 2020) library. During training, we use an Adam optimizer and a scheduler with a warmup ratio of 0.1. We use an Adam optimizer and a linear scheduler with a warmup ratio of 0.1. We train our OPTIPROMPT model for 10 epochs with a learning rate of 3e-3 and a batch size of 16. For fine-tuning, we use an Adam optimizer and a linear scheduler with a warmup ratio of 0.1. We fine-tune the language models for 10 epochs with a learning rate of 2e-6 and a batch size of 2.

We report AUTOPROMPT’s performance based on the prompts released by Shin et al. (2020). When we apply AUTOPROMPT to a control task (e.g., the Random Embeddings model), or compare AUTOPROMPT with different language models on a different dataset (see Appendix C), we run AUTOPROMPT for 1000 iterations for each model to search for the prompt of a relation.

B.2 LAMA Classifiers

In Section 4.2 we fit two simple probabilistic models to the Wikidata training data collected

by (Shin et al., 2020). Given a relation r and a subject s consisting of tokens $w_1, \dots, w_{|s|} \in \mathcal{V}$, the Class Prior model predicts $\hat{o} = \arg \max_o P(o | r)$, the object label that is most frequently associated with relation r in the training data. The Naive Bayes model predicts $\hat{o} = \arg \max_o P(o | s, r)$, with

$$P(o | s, r) = P(o | r) \prod_{i=1}^{|s|} P(o | w_i).$$

The probabilities are estimated from the corpus with add-one smoothing:

$$P(o | w_i) = \frac{\text{count}(o, w_i) + 1}{\sum_{w \in \mathcal{V}} (\text{count}(o, w) + 1)}.$$

C Comparing Pre-trained Language Models

We compare different pre-trained language models (BERT, RoBERTa (Liu et al., 2019), and ALBERT (Lan et al., 2019)) with different probing methods. We collect at most 1000 training samples for each relation from the TRE-x dataset and constrain the object of each sample to be a token for all the models¹². During testing, we downsample the LAMA test set to make sure that the object in each sample is a single token for all the models. In Table 6 shows the results of different probing methods applied to four pre-trained language models, along with our Random Model baseline. We make the following observations:

- **Base vs. Large:** The larger version of BERT performs better on LAMA than BERT base

¹²The original data collected by Shin et al. (2020) is not applicable when we compare different language models, because some object tokens are not in the vocabulary of RoBERTa or ALBERT.

in the OPTIPROMPT probe. We might hypothesize that BERT-large is simply more capable of finding patterns in the training data, but our baseline result does not indicate that this is the case—on the contrary, BERT-large performs marginally worse on the Random Model baseline. This could lead us to believe that BERT-large truly does store information about 1 or 2% more LAMA facts compared to BERT-base.

- **BERT vs. RoBERTa vs. ALBERT:** [Shin et al. \(2020\)](#) find that RoBERTa performs significantly worse on LAMA than BERT. We find this is true for our prompts as well (comparing with BERT-large), but the magnitude of the difference decreases in the fine-tuning setting. Our baseline result gives a possible hint as to why: RoBERTa performs better in the RM setting with fine-tuning, indicating that part of the difference between OPTIPROMPT and fine-tuning might be due to better exploitation of training data. This change is even more dramatic in ALBERT. Perhaps these models store less factual information due to pre-training on a wider variety of genres.

We believe that further comparisons along these lines are a promising area of future work—for example, if we could show that probing results are correlated with downstream task performance and use probes to guide model selection.

Relation	Type	Name	#	Manual	LPAQA	Auto	Opti
P1376	1-1	capital of	233	73.8	67.8	56.2	56.7
P36	1-1	capital	702	62.1	62.1	59.7	61.3
P103	N-1	native language	977	72.2	72.2	79.7	86.8
P127	N-1	owned by	687	34.8	32.5	44.3	49.6
P131	N-1	located in the administrative territorial entity	881	23.3	22.8	28.9	41.4
P136	N-1	genre	931	0.8	16.8	55.3	63.6
P138	N-1	named after	642	61.4	59.5	70.7	73.4
P140	N-1	religion	473	0.6	59.8	60.5	76.5
P159	N-1	headquarters location	967	32.4	35.6	35.7	37.4
P17	N-1	country	930	31.3	39.8	51.0	57.8
P176	N-1	manufacturer	973	85.5	81.5	87.5	87.3
P19	N-1	place of birth	944	21.1	21.1	19.5	20.6
P20	N-1	place of death	953	27.9	27.9	29.8	33.8
P264	N-1	record label	429	9.6	6.3	4.2	45.5
P276	N-1	location	958	41.5	41.5	43.0	47.1
P279	N-1	subclass of	964	30.7	14.7	54.9	64.7
P30	N-1	continent	975	25.4	16.9	78.6	86.3
P361	N-1	part of	932	23.6	31.4	37.0	46.4
P364	N-1	original language of film or TV show	856	44.5	43.9	45.0	51.3
P37	N-1	official language	966	54.6	56.8	52.7	58.6
P407	N-1	language of work or name	877	64.2	65.2	68.4	71.0
P413	N-1	position played on team / speciality	952	0.5	23.7	41.7	44.0
P449	N-1	original network	880	20.9	9.1	33.1	36.0
P495	N-1	country of origin	909	28.7	32.2	35.8	40.8
P740	N-1	location of formation	936	8.9	13.7	13.1	15.0
P1001	N-M	applies to jurisdiction	701	70.5	72.8	80.5	85.2
P101	N-M	field of work	696	9.9	5.3	12.1	14.1
P106	N-M	occupation	958	0.6	0.0	13.6	35.7
P108	N-M	employer	383	6.8	5.7	7.8	11.2
P1303	N-M	instrument	949	7.6	18.0	23.1	23.6
P1412	N-M	languages spoken, written or signed	969	65.0	64.7	71.5	76.1
P178	N-M	developer	591	62.9	59.4	64.3	67.9
P190	N-M	twinned administrative body	992	2.2	1.7	2.4	3.1
P27	N-M	country of citizenship	966	0.0	41.5	45.8	47.1
P31	N-M	instance of	922	36.7	36.7	53.6	64.9
P39	N-M	position held	892	8.0	16.1	27.2	42.8
P463	N-M	member of	225	67.1	57.3	64.0	64.0
P47	N-M	shares border with	920	13.7	13.7	19.2	22.2
P527	N-M	has part	976	11.2	10.6	22.1	34.8
P530	N-M	diplomatic relation	996	2.8	3.9	2.8	3.3
P937	N-M	work location	954	29.8	39.1	34.4	43.3

Table 7: The accuracy of different prompts on LAMA for each relation using BERT-base-cased. Manual: the manually written prompts included in LAMA; LPAQA: manually written + paraphrased prompts from Jiang et al. (2020); Auto: the five-token AUTOPROMPT prompts released by Shin et al. (2020). Opti: OPTIPROMPT initialized using the manually written templates.

Relation	Type	Name	Random Embeddings			Random Model		
			Auto	Opti	FT	Auto	Opti	FT
P1376	1-1	capital of	0.0/0.0	0.0/0.0	0.0/0.0	0.0/0.0	0.0/0.0	0.0/0.0
P36	1-1	capital	0.0/0.1	0.0/11.8	0.0/0.1	0.0/0.0	0.0/0.0	0.0/0.0
P103	N-1	native language	26.5/26.5	60.1/60.1	57.6/63.5	0.0/0.0	60.1/60.1	60.1/60.5
P127	N-1	owned by	0.0/0.0	6.8/7.0	6.7/17.2	0.0/0.0	6.8/6.8	6.4/12.4
P131	N-1	located in...	0.0/0.0	0.0/2.3	0.2/1.5	0.0/0.0	0.2/0.2	0.6/0.6
P136	N-1	genre	27.6/27.6	55.4/55.4	54.7/56.1	0.0/0.0	55.3/55.3	55.4/55.4
P138	N-1	named after	0.0/0.0	1.2/3.9	0.8/50.8	0.0/0.0	1.6/1.6	1.6/1.6
P140	N-1	religion	0.0/0.0	53.7/53.7	53.7/53.7	0.0/0.0	53.7/53.7	53.7/53.7
P159	N-1	headquarters location	0.0/0.0	9.0/9.0	7.7/7.7	0.0/0.0	9.0/9.0	9.0/9.0
P17	N-1	country	0.0/0.1	2.8/2.8	1.6/7.7	0.0/0.0	0.5/1.9	2.6/2.6
P176	N-1	manufacturer	0.2/0.2	8.9/8.9	8.8/80.0	0.0/0.0	8.9/8.9	8.8/70.2
P19	N-1	place of birth	0.0/0.0	2.9/3.2	5.3/5.4	0.0/0.0	0.0/0.3	6.0/6.4
P20	N-1	place of death	0.0/0.0	7.3/11.4	7.8/12.7	0.0/0.0	10.2/10.2	9.7/11.6
P264	N-1	record label	0.0/0.0	37.5/37.5	37.5/37.5	0.0/0.0	37.5/37.5	37.5/37.5
P276	N-1	location	0.2/0.2	2.7/2.7	3.8/5.6	0.0/0.0	0.7/0.9	4.2/4.4
P279	N-1	subclass of	0.0/0.1	15.7/26.2	14.4/36.9	0.0/0.0	15.9/15.9	16.0/16.0
P30	N-1	continent	57.5/57.5	72.3/72.3	72.3/72.3	0.0/0.0	72.3/72.3	72.3/72.3
P361	N-1	part of	4.9/4.9	6.0/6.0	6.0/7.6	0.0/0.0	6.0/6.0	6.0/6.0
P364	N-1	original language...	0.0/0.0	21.1/25.1	23.5/27.8	0.0/0.0	24.1/25.5	22.4/25.5
P37	N-1	official language	0.0/0.0	17.0/17.0	14.7/16.0	0.0/0.0	17.0/17.0	17.0/17.0
P407	N-1	language of work...	0.7/0.9	46.4/46.4	45.8/46.4	0.0/0.0	46.4/46.4	46.4/46.4
P413	N-1	position played...	30.3/30.3	41.6/41.6	41.7/41.7	0.0/0.0	41.7/41.7	41.7/41.7
P449	N-1	original network	3.2/3.2	25.9/31.0	23.9/32.2	0.0/0.0	28.9/31.9	21.5/29.8
P495	N-1	country of origin	1.9/1.9	8.9/10.8	9.6/12.3	0.0/0.0	10.9/10.9	8.6/13.1
P740	N-1	location of formation	0.0/0.0	7.4/7.4	6.8/7.6	0.0/0.0	4.5/5.6	7.4/7.4
P1001	N-M	applies to jurisdiction	0.1/0.1	8.0/42.8	7.4/54.9	0.0/0.0	9.6/9.6	9.6/9.7
P101	N-M	field of work	0.0/0.0	9.6/10.1	10.3/10.8	0.0/0.0	10.5/10.5	10.5/10.8
P106	N-M	occupation	0.0/8.9	6.2/27.5	5.2/30.8	0.0/0.0	14.3/14.3	6.8/26.4
P108	N-M	employer	0.0/0.0	7.6/8.9	3.4/9.1	0.0/0.0	4.2/6.3	5.7/9.4
P1303	N-M	instrument	0.0/0.0	22.8/22.8	21.9/22.7	0.0/0.0	10.6/10.6	22.8/22.8
P1412	N-M	languages spoken...	0.0/0.0	13.9/27.7	10.5/28.0	0.0/0.0	7.3/25.3	12.0/28.2
P178	N-M	developer	0.0/0.0	2.7/11.3	4.2/29.4	0.0/0.0	4.7/5.4	4.6/8.8
P190	N-M	twinned admin...	0.0/0.0	0.0/1.1	1.7/2.1	0.0/0.0	1.4/2.1	0.3/2.4
P27	N-M	country of citizenship	1.3/1.3	9.7/9.9	8.8/13.4	0.0/0.0	10.0/10.0	9.9/10.1
P31	N-M	instance of	0.3/0.3	8.0/11.2	8.4/24.9	0.0/0.0	8.8/8.8	8.9/8.9
P39	N-M	position held	0.0/0.0	20.9/28.7	23.8/32.4	0.0/0.0	27.2/27.2	18.5/30.4
P463	N-M	member of	1.3/1.3	2.2/45.8	2.2/60.4	0.0/0.0	2.2/2.2	2.2/56.4
P47	N-M	shares border with	0.0/0.0	0.0/0.1	0.2/0.9	0.0/0.0	0.0/0.0	0.0/1.1
P527	N-M	has part	0.0/0.0	17.3/17.3	12.0/18.0	0.0/0.0	14.4/14.4	17.4/17.4
P530	N-M	diplomatic relation	0.0/0.0	0.3/0.5	0.0/1.2	0.0/0.0	0.9/0.9	0.0/0.6
P937	N-M	work location	0.0/0.0	14.6/14.6	12.1/16.8	0.0/0.0	13.2/13.2	14.8/14.8

Table 8: Control result details. The value in each cell is *Maj./Acc.*, where *Acc.* is the percentage of facts of relation *r* that the model predicts correctly and *Maj.* is the percentage of facts $\langle s, r, o \rangle$ such that (a) the model predicts *o* correctly, and (b) *o* is the most frequent object for relation *r* in the training data. We probe the BERT-base-based model, reinitializing either the token embeddings or all of the parameters.

Relation	Type	Name	LAMA-easy					LAMA-hard				
			#	Man.	LPAQA	Auto	Opti	#	Man.	LPAQA	Auto	Opti
P1376	1-1	capital of	1	100.0	0.0	100.0	0.0	232	73.7	68.1	56.0	56.9
P36	1-1	capital	2	50.0	50.0	50.0	50.0	700	62.1	62.1	59.7	61.3
P103	N-1	native language	668	75.7	75.7	94.0	95.7	309	64.4	64.4	48.9	67.6
P127	N-1	owned by	131	53.4	87.0	89.3	90.8	556	30.4	19.6	33.6	39.9
P131	N-1	located in...	72	16.7	23.6	56.9	63.9	809	23.9	22.7	26.5	39.4
P136	N-1	genre	535	1.1	15.7	96.1	92.9	396	0.3	18.2	0.3	24.0
P138	N-1	named after	341	84.2	85.0	94.7	95.3	301	35.5	30.6	43.5	48.5
P140	N-1	religion	254	0.0	80.7	100.0	96.9	219	1.4	35.6	14.6	53.0
P159	N-1	headquarters location	97	45.4	57.7	82.5	75.3	870	30.9	33.1	30.5	33.2
P17	N-1	country	131	44.3	52.7	71.8	86.3	799	29.2	37.7	47.6	53.2
P176	N-1	manufacturer	791	94.7	90.8	95.6	95.8	182	45.6	41.2	52.2	50.0
P19	N-1	place of birth	101	77.2	77.2	77.2	75.2	843	14.4	14.4	12.6	14.0
P20	N-1	place of death	183	78.7	78.7	78.7	74.9	770	15.8	15.8	18.2	24.0
P264	N-1	record label	195	6.2	6.7	2.6	81.5	234	12.4	6.0	5.6	15.4
P276	N-1	location	90	60.0	60.0	68.9	81.1	868	39.6	39.6	40.3	43.5
P279	N-1	subclass of	374	32.9	28.3	79.1	92.2	590	29.3	6.1	39.5	47.3
P30	N-1	continent	705	34.0	19.1	98.0	99.9	270	3.0	11.1	27.8	50.7
P361	N-1	part of	73	1.4	11.0	16.4	98.6	859	25.5	33.2	38.8	41.9
P364	N-1	original language...	373	69.4	71.8	76.9	82.3	483	25.3	22.4	20.3	27.3
P37	N-1	official language	197	43.7	53.8	88.8	87.3	769	57.3	57.6	43.4	51.2
P407	N-1	language of work...	460	84.6	85.9	92.8	91.3	417	41.7	42.4	41.5	48.7
P413	N-1	position played...	397	1.3	56.7	100.0	99.5	555	0.0	0.2	0.0	4.3
P449	N-1	original network	428	26.4	11.9	54.7	64.5	452	15.7	6.4	12.6	9.1
P495	N-1	country of origin	187	41.2	44.9	51.9	80.2	722	25.5	28.9	31.6	30.6
P740	N-1	location of formation	79	43.0	57.0	74.7	86.1	857	5.7	9.7	7.5	8.4
P1001	N-M	applies to jurisdiction	408	79.9	84.1	90.7	95.1	293	57.3	57.0	66.2	71.3
P101	N-M	field of work	104	9.6	12.5	38.5	36.5	592	10.0	4.1	7.4	10.1
P106	N-M	occupation	386	0.0	0.0	29.8	74.1	572	1.0	0.0	2.6	9.8
P108	N-M	employer	54	27.8	22.2	53.7	72.2	329	3.3	3.0	0.3	1.2
P1303	N-M	instrument	243	7.8	37.9	87.7	88.1	706	7.5	11.2	0.8	1.4
P1412	N-M	languages spoken...	473	86.7	81.8	89.4	90.7	496	44.4	48.4	54.4	62.1
P178	N-M	developer	259	79.5	81.1	90.3	95.4	332	50.0	42.5	44.0	46.4
P190	N-M	twinned admin...	44	2.3	2.3	18.2	11.4	948	2.2	1.7	1.7	2.7
P27	N-M	country of citizenship	217	0.0	74.7	54.8	75.1	749	0.0	31.9	43.1	39.0
P31	N-M	instance of	316	48.1	48.1	79.7	94.0	606	30.7	30.7	39.9	49.7
P39	N-M	position held	421	11.2	28.5	55.3	63.2	471	5.1	5.1	2.1	24.6
P463	N-M	member of	139	87.1	71.9	91.4	95.0	86	34.9	33.7	19.8	14.0
P47	N-M	shares border with	35	5.7	5.7	14.3	22.9	885	14.0	14.0	19.4	22.1
P527	N-M	has part	296	9.1	3.7	29.1	61.8	680	12.1	13.5	19.1	23.1
P530	N-M	diplomatic relation	18	5.6	5.6	5.6	0.0	978	2.8	3.9	2.8	3.4
P937	N-M	work location	258	77.1	86.4	75.6	88.0	696	12.2	21.6	19.1	26.7

Table 9: The accuracy by relation on LAMA-easy and LAMA-hard. LAMA-easy consists of the facts that are predicted correctly by any of three models: the Naive Bayes model described in Section 4; BERT-base-cased with randomly initialized token embeddings; and BERT-base-cased with all parameters reinitialized. LAMA-hard contains all the remaining facts.