AUTOPROMPT: Eliciting Knowledge from Language Models with Automatically Generated Prompts

Taylor Shin[*◇]   **Yasaman Razeghi**[*◇]   **Robert L. Logan IV**[*◇]
Eric Wallace[♠]   **Sameer Singh**[◇]
◇University of California, Irvine   ♠University of California, Berkeley
{tshin1, yrazeghi, rlogan, sameer}@uci.edu
ericwallace@berkeley.edu

Abstract

The remarkable success of pretrained language models has motivated the study of what kinds of knowledge these models learn during pretraining. Reformulating tasks as fill-in-the-blanks problems (e.g., cloze tests) is a natural approach for gauging such knowledge, however, its usage is limited by the manual effort and guesswork required to write suitable prompts. To address this, we develop AUTOPROMPT, an *automated* method to create prompts for a diverse set of tasks, based on a gradient-guided search. Using AUTOPROMPT, we show that masked language models (MLMs) have an inherent capability to perform sentiment analysis and natural language inference without additional parameters or finetuning, sometimes achieving performance on par with recent state-of-the-art supervised models. We also show that our prompts elicit more accurate factual knowledge from MLMs than the manually created prompts on the LAMA benchmark, and that MLMs can be used as relation extractors more effectively than supervised relation extraction models. These results demonstrate that automatically generated prompts are a viable parameter-free alternative to existing probing methods, and as pretrained LMs become more sophisticated and capable, potentially a replacement for finetuning.

1 Introduction

Pretrained language models (LMs) have had exceptional success when adapted to downstream tasks via *finetuning* (Peters et al., 2018; Devlin et al., 2019). Although it is clear that pretraining improves accuracy, it is difficult to determine whether the knowledge that finetuned LMs contain is learned during the *pretraining* or the *finetuning*

_____
[*] First three authors contributed equally.

process. How can we directly evaluate the knowledge present in pretrained LMs, be it linguistic, factual, commonsense, or task-specific?

Numerous techniques have been proposed to elicit such knowledge by analyzing pretrained LMs' internal representations. A common strategy is to use probing classifiers—shallow classifiers that predict certain attributes using an LMs' representations as features (Conneau et al., 2018; Liu et al., 2019). However, probing classifiers require additional learned parameters and are thus susceptible to false positives; high probing accuracy is *not* a sufficient condition to conclude that an LM contains a certain piece of knowledge (Hewitt and Liang, 2019; Voita and Titov, 2020). Attention visualization, another common technique, has a similar failure mode: attention scores may be correlated with, but not caused by the underlying target knowledge, leading to criticism against their use as explanations (Jain and Wallace, 2019; Wiegreffe and Pinter, 2019). Both probing and attention visualizations also struggle to evaluate knowledge that cannot be represented as simple token- or sequence-level classification tasks.

A more direct approach for eliciting knowledge from these models, since they are language models after all, is *prompting*, i.e. converting tasks into a language model format. For example, Radford et al. (2019) frame summarization as a language modeling task by appending "TL;DR:" to the end of an article and then generating from an LM. Similarly, Petroni et al. (2019) manually reformulate a knowledge base completion task as a cloze test (i.e., a fill-in-the-blank problem). Compared to existing model analysis methods, prompting is non-invasive: it does not introduce large amounts of additional parameters or require direct inspection of a model's representations. Thus prompting pro-
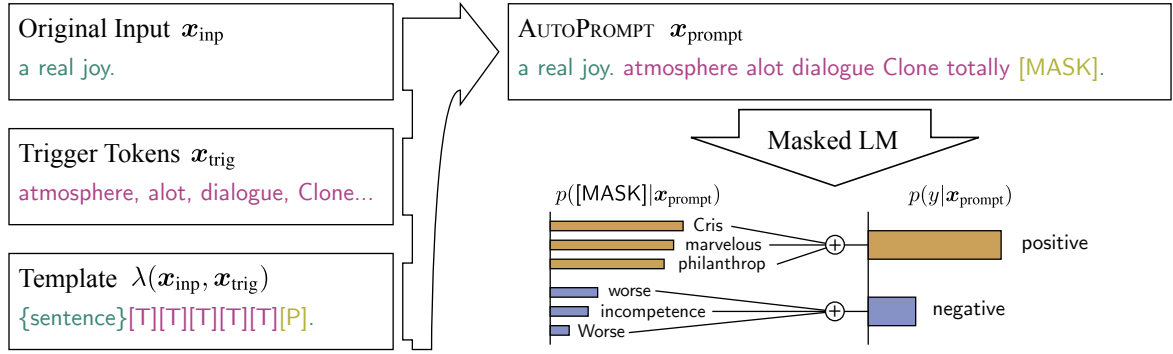
Figure 1: **Illustration of AUTOPROMPT** applied to probe a masked language model's (MLM's) ability to perform sentiment analysis. Each input, x_{inp}, is placed into a natural language prompt, x_{prompt}, which contains a single [MASK] token. The prompt is created using a template, λ, which combines the original input with a set of trigger tokens, x_{trig}. The trigger tokens are shared across all inputs and determined using a gradient-based search (Section 2.2). Probabilities for each class label, y, are then obtained by marginalizing the MLM predictions, $p([MASK]|x_{prompt})$, over sets of automatically detected label tokens (Section 2.3).

vides a lower bound on what the model "knows", and is therefore a more useful analysis tool. However, prompting unfortunately requires manually crafting the context to feed into the model. Not only is this time consuming and non-intuitive for many tasks (e.g., textual entailment), more importantly, models are highly sensitive to this context: improperly-constructed contexts cause artificially low performance (Jiang et al., 2020). Overcoming the need to manually specify prompts would make prompting a more widely useful analysis tool.

In this paper, we introduce AUTOPROMPT—an *automated* method for generating prompts for any task, illustrated in Figure 1. Given a task, e.g., sentiment analysis, AUTOPROMPT creates a prompt by combining the original task inputs (e.g. reviews) with a collection of *trigger tokens* according to a template. The same set of trigger tokens is used for all inputs, and is learned using a variant of the gradient-based search strategy proposed in Wallace et al. (2019). The LM predictions for the prompt are converted to class probabilities by marginalizing over a set of associated label tokens, which can either be learned or specified ahead of time, enabling the LM to be evaluated the same as one would any other classifier.

We validate the effectiveness of AUTOPROMPT in numerous experiments. First, we use AUTOPROMPT to construct prompts that test pretrained masked language models (MLMs) on sentiment analysis and natural language inference (NLI). Our tests reveal that, without any finetuning, MLMs perform well on both of these tasks—a properly-prompted RoBERTa achieves 91% accuracy on SST-2 (better than a finetuned ELMo model (Peters et al., 2018)), and 69% accuracy on a balanced variant of the SICK-E dataset (Marelli et al., 2014). Next, we apply AUTOPROMPT to the fact retrieval tasks of LAMA (Petroni et al., 2019), where we are able to construct prompts that more effectively elicit MLM's factual knowledge than existing prompts generated using manual and corpus-mining methods. Concretely, we achieve 43.3% precision-at-1, compared to the current best single-prompt result of 34.1% (Jiang et al., 2020). We also introduce a variant of this task, similar to relation extraction (RE), that tests whether MLMs can extract knowledge from a given piece of text. We show that MLMs can actually *outperform* existing RE models when context sentences with real facts are provided, however, they struggle when context sentences are artificially falsified.

Finally, although the goal of AUTOPROMPT is to analyze models, we find that it provides certain practical advantages over finetuning. First, AUTOPROMPT achieves higher average- and worst-case accuracy than finetuning in low-data regimes. Moreover, unlike finetuning, prompting LMs does not require large amounts of disk space to store model checkpoints; once a prompt is found, it can be used on off-the-shelf pretrained LMs. This is beneficial when serving models for multiple tasks.

2 Overview of AUTOPROMPT

A natural way to elicit knowledge from pretrained LMs is to pose tasks as fill-in-the-blank problems.

However, writing prompts is not only time consuming, but it is not clear that the same phrasing will be effective for every model, nor is it clear what criteria determine whether a particular phrasing the *best* to elicit the desired information. In light of this, we introduce AUTOPROMPT, a method that constructs customized prompts for a specific task and MLM of interest, to cause the MLMs to produce the desired knowledge.[1] An illustration of AUTOPROMPT is provided in Figure 1. The prompt is constructed by taking the original task inputs—a collection of one or more sequences of tokens (e.g., the review in Figure 1)—and mapping them to a sequence of tokens using a template. In the following sections, we describe how AUTOPROMPT uses labeled training data to construct prompts, and how it uses the output of the MLM as a prediction for the task.

2.1 Background and Notation

For the purpose of prompt construction, we distinguish the original task inputs $\boldsymbol{x}_{\text{inp}}$ (e.g., the review in Figure 1, "a real joy.") from the prompt $\boldsymbol{x}_{\text{prompt}}$ (e.g., "a real joy. atmosphere alot dialogue Clone totally [MASK].") that is fed into the MLM. The mapping from $\boldsymbol{x}_{\text{inp}}$ to $\boldsymbol{x}_{\text{prompt}}$ is performed using a template, λ. This template defines where each input sequence will be placed in the prompt, as well as the placement of any additional tokens. In particular, it must also define the placement of a special [MASK] token for the MLM to fill in (denoted by [P] in the template to distinguish it from other [MASK] tokens that might appear). Feeding the prompt into the MLM produces a probability distribution $p([\text{MASK}]|\boldsymbol{x}_{\text{prompt}})$ describing which tokens most likely fill in the blank.

If class labels naturally correspond to tokens in the vocabulary (e.g., entity names in knowledge base completion tasks), this distribution may be readily interpreted as a distribution over class labels. However, for tasks such as sentiment analysis, there may be a set of label tokens $\mathcal{V}_y$ that correspond to a particular label y. For example, in Figure 1, "Cris", "marvelous", and "philanthrop" all indicate positive sentiment. In this case, the class probability is obtained by marginalizing over the

set of label tokens:

$$p(y|\boldsymbol{x}_{\text{prompt}}) = \sum_{w \in \mathcal{V}_y} p([\text{MASK}] = w|\boldsymbol{x}_{\text{prompt}}) \tag{1}$$

2.2 Gradient-Based Prompt Search

So far, we have shown how to reformulate a classification task as a language modeling task using prompts. Here, we propose a method for *automatic prompt construction* based on Wallace et al. (2019). The idea is to add a number of "trigger" tokens that are shared across all prompts (denoted by [T] in the example template in Figure 1). These tokens are initialized to [MASK] tokens, and then iteratively updated to maximize the label likelihood (Equation (1)) over batches of examples.

Formally, at each step, we compute a first-order approximation of the change in the log-likelihood that would be produced by swapping the jth trigger token $x_{\text{trig}}^{(j)}$ with another token $w \in \mathcal{V}$. Then we identify a candidate set $\mathcal{V}_{\text{cand}}$ of the top-k tokens estimated to cause the greatest increase:

$$\mathcal{V}_{\text{cand}} = \underset{w \in \mathcal{V}}{\text{top-}k} \left[\boldsymbol{w}_{\text{in}}^T \nabla \log p(y|\boldsymbol{x}_{\text{prompt}}) \right] \tag{2}$$

where $\boldsymbol{w}_{\text{in}}$ is the input embedding of w, and the gradient is taken with respect to the input embedding of $x_{\text{trig}}^{(j)}$. Note that computing this candidate set is roughly as expensive as a single forward pass and backward pass of the model (the dot-products require the same amount of multiplications as computing the LM output projection). For each candidate in this set, we then re-evaluate Equation (1) on the updated prompt, and retain the prompt with the highest probability in the next step—this requires k forward passes of the model. An example prompt produced by this method for the task of sentiment analysis is shown in Figure 1.

2.3 Automating Label Token Selection

While in some settings the choice of label tokens is obvious (e.g., when class labels directly correspond to words in the vocabulary), it is less clear what label tokens are appropriate for problems involving more abstract class labels (e.g., NLI). In this section, we develop a general two-step approach to automate the selection of the sets of label tokens $\mathcal{V}_y$. In the first step, we train a logistic classifier to predict the class label using the contextualized embedding of the [MASK] token as input:

$$\boldsymbol{h} = \text{Transformer}_{\text{enc}}(\tilde{\boldsymbol{x}}) \tag{3}$$

---

[1]Although we focus only on MLMs in this work, our method is trivially extendable to autoregressive LMs. The only adjustment is that the predict token must occur at the end of the prompt.

We write the output of this classifier as:

$$p(y|\boldsymbol{h}^{(i)}) \propto \exp(\boldsymbol{h}^{(i)} \cdot \boldsymbol{y} + \beta_y) \qquad (4)$$

where $\boldsymbol{y}$ and β_y are the learned weight and bias terms for the label y, and i represents the index of the [MASK] token.

In the second step, we substitute $\boldsymbol{h}^{(i)}$ with the MLM's output word embeddings $\boldsymbol{w}_{\text{out}}$ to obtain a score $s(y, w) = p(y|\boldsymbol{w}_{\text{out}})$. Intuitively, because $\boldsymbol{w}_{out} \cdot \boldsymbol{h}$ and $\boldsymbol{y} \cdot \boldsymbol{h}$ are large for words and labels that are relevant to a particular context, $s_w \propto \exp(\boldsymbol{w}_{\text{out}} \cdot \boldsymbol{y} + \beta_y)$ should be large for words that are typically associated with a given label. The sets of label tokens are then constructed from the k-highest scoring words:

$$\mathcal{V}_y = \underset{w \in \mathcal{V}}{\text{top-}k} \, [s(y, w)] \qquad (5)$$

2.4 Relation to Other Prompting Methods

Our work fits into a body of work that probes language model's knowledge via prompts. Previous works have used manually defined prompts to study an LM's ability to perform: commonsense reasoning (Trinh and Le, 2018; Kwon et al., 2019; Shwartz et al., 2020), question answering (Lewis et al., 2019), fact recall (Petroni et al., 2019; Jiang et al., 2020; Bouraoui et al., 2019), summarization (Radford et al., 2019), and other supervised tasks (Brown et al., 2020). Schick and Schütze (2020) use manually constructed prompts in conjunction with semi-supervised learning for few-shot learning. We instead *automatically* create prompts for any task, which leads to higher accuracy and opens up new phenomena to analyze.

2.5 Evaluation Setup

In the following sections, we apply AUTOPROMPT to probe BERT$_{\text{BASE}}$[2] (110M parameters) and RoBERTa$_{\text{LARGE}}$'s (355M parameters) knowledge of the following tasks: sentiment analysis, natural language inference (NLI), fact retrieval, and relation extraction. We use the PyTorch implementations and pretrained weights provided by the transformers Python library (Wolf et al., 2019). For sentiment analysis and NLI, we find label tokens using the logistic-regression-based heuristic described in Section 2.3. For fact retrieval and relation extraction, we skip this step as the labels (entities) directly correspond to tokens in the vocabulary. For all tasks, we perform the prompt

search described in Section 2.2 for multiple iterations. In each iteration, we use a batch of training data to identify the candidate set $\mathcal{V}_{\text{cand}}$ of replacement trigger tokens. We then evaluate the label likelihoods of the updated prompts on a separate batch of data, and we retain the best trigger token in the next iteration of the search. At the end of every iteration, we measure the label likelihood on withheld development data, and return the best prompt found during the entire search as the final output. Performance is evaluated using the appropriate task-specific metrics—e.g., accuracy for sentiment analysis and NLI, and precision@k for fact retrieval—on a separate withheld test set.

Our AUTOPROMPT implementation is publicly available at http://ucinlp.github.io/autoprompt, and supports prompt generation for pretrained models in the HuggingFace transformers library (Wolf et al., 2019) on arbitrary datasets.

3 Sentiment Analysis

Sentiment analysis is a fundamental task in NLP, both for natural language understanding research and real-world applications. It is also difficult to probe the extent to which MLMs understand sentiment without finetuning.

Setup We apply our method to convert instances from the binary Stanford Sentiment Treebank (Socher et al., 2013, SST-2) into prompts, using the standard train/test splits. We find label tokens using a prompt based on the template in Table 3. For our gradient-based prompt search, we perform a grid search over the following hyperparameters: $|\mathcal{V}_{cand}| \in \{10, 100\}$, $|\mathcal{V}_y| \in \{1, 3, 5\}$, $|\boldsymbol{x}_{\text{trig}}| \in [3, 6]$.[3] All prompts are initialized with the same template used to find the label set.

We also construct a prompt manually (before automated prompts are generated, to avoid bias) based on the intuition that SST-2 is comprised of movie reviews. We use "{sentence} this movie was [P]." as the template, and use "terrible" and "fantastic" for the negative and positive label tokens, respectively.

Results We show results in Table 1, along with reference scores from the GLUE (Wang et al., 2019) SST-2 leaderboard, and scores for a linear probe trained over the elementwise average of the LM token representations. Prompts generated by AUTOPROMPT reveal that both BERT

---

[2] For brevity, we will omit subscripts in the model names.

[3] Required 2 days to run with 8 NVIDIA 2080Ti GPUs.

Model	Dev	Test
BiLSTM	-	82.8[†]
BiLSTM + ELMo	-	89.3[†]
BERT (linear probing)	85.2	83.4
BERT (finetuned)	-	93.5[†]
RoBERTa (linear probing)	87.9	88.8
RoBERTa (finetuned)	-	96.7[†]
BERT (manual)	63.2	63.2
BERT (AUTOPROMPT)	80.9	82.3
RoBERTa (manual)	85.3	85.2
RoBERTa (AUTOPROMPT)	91.2	91.4

Table 1: **Sentiment Analysis** performance on the SST-2 test set of supervised classifiers (top) and fill-in-the-blank MLMs (bottom). Scores marked with † are from the GLUE leaderboard: http://gluebenchmark.com/leaderboard.

and RoBERTa have a strong knowledge of sentiment analysis: without any finetuning, BERT performs comparably to a supervised BiLSTM, and RoBERTa achieves an accuracy on-par with finetuned BERT and ELMo models. In addition, we observe that our automatically constructed prompts are more effective than manual prompts, and that they are difficult to construct using human intuition: the best template for RoBERTa is "{sentence} atmosphere alot dialogue Clone totally [P]." We include results on the effect of the AUTOPROMPT hyperparameters in Appendix A.

Accuracy in Low-Data Settings Although the goal of AUTOPROMPT is to probe a model's knowledge, we also find that it may be a viable alternative to finetuning in the low-data regime. To show this, we measure the development set accuracy of AUTOPROMPT prompts when using random subsets of 10, 100, and 1000 instances from the training data. We run our prompt search with $|x_{trig}| = 10$, $|\mathcal{V}_y| = 3$, and $|\mathcal{V}_{cand}| = 10$. We compare to the performance of BERT and RoBERTa finetuned on the same data. For fair comparison between AUTOPROMPT and finetuning, we use Mosbach et al. (2020)'s recommended parameters for finetuning on small datasets: trained for 20 epochs, using AdamW (Loshchilov and Hutter, 2018) with bias correction and a learning rate that linearly increases to 2×10^{-5} in the first 10% of iterations, and linearly decreases to 0 afterwards. Experiments are repeated 10 times on random subsets of data (and seeds for the finetuned models). Best-case, worst-case, and average performance are shown in Figure 2. Note that results in the EMNLP version had a bug that has since been fixed.

(a) BERT on SST-2

(b) RoBERTa on SST-2

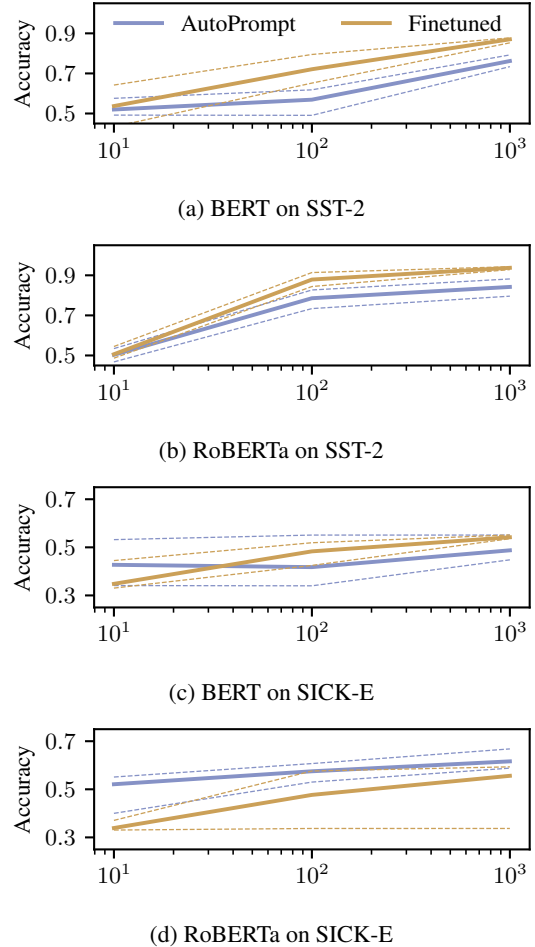(c) BERT on SICK-E

(d) RoBERTa on SICK-E

Figure 2: **Effect of Training Data** on sentiment analysis and NLI for AUTOPROMPT vs. finetuning. X-axis is the number of data points used during training. Error bars plot the max. and min. accuracies observed over 10 independent runs. *(revised since EMNLP version).*

We observe that while finetuning outperforms AUTOPROMPT on sentiment analysis, AUTOPROMPT can perform better than finetuning on NLI. Notably, AUTOPROMPT elicits better average performance from both BERT and RoBERTa given only 10 training examples. Furthermore, results for RoBERTa are more stable across all sample sizes whereas finetuning can result in "failed runs" (consistent with Dodge et al. 2020). This behavior in the low-data regime is an interesting phenomenon, and suggests that there are barriers that MLMs must surmount when they are converted to finetuned classifiers that are not encountered when the task is presented as masked language modeling.

4 Natural Language Inference

To evaluate the *semantic* understanding of MLMs, we experiment on Natural Language Inference

Model	SICK-E Datasets		
	standard	3-way	2-way
Majority	56.7	33.3	50.0
BERT (finetuned)	86.7	84.0	95.6
BERT (linear probing)	68.0	49.5	91.9
RoBERTa (linear probing)	72.6	49.4	91.1
BERT (AUTOPROMPT)	62.3	55.4	85.7
RoBERTa (AUTOPROMPT)	65.0	69.3	87.3

Table 2: **Natural Language Inference** performance on the SICK-E test set and variants. (Top) Baseline classifiers. (Bottom) Fill-in-the-blank MLMs.

(NLI). NLI is crucial in many tasks such as reading comprehension and commonsense reasoning (Bowman et al., 2015), and it is used as a common benchmark for language understanding.

Setup We use the entailment task from the SICK dataset (Marelli et al., 2014, SICK-E) which consists of around 10,000 pairs of human-annotated sentences labeled as entailment, contradiction, and neutral. The standard dataset is biased toward the neutral class which represent 56.7% of instances. We also experiment on an unbiased variant with 2-way classification of contradiction vs. entailment (*2-way*), as well as an unbiased 3-way classification variant (*3-way*). The template used for AUTOPROMPT is provided in Table 3. We search over the following parameters: $|\mathcal{V}_{cand}| \in \{10, 50\}$, $|\mathcal{V}_y| \in \{1, 3, 5, 10\}$, $|\boldsymbol{x}_{\text{trig}}| \in [1, 5]$, and choose the best prompt according to development set accuracy.

Results Table 2 shows that AUTOPROMPT considerably outperforms the majority baseline in all experiments. For example, on the 2-way SICK-E dataset, AUTOPROMPT is comparable to a supervised finetuned BERT. We also test linear probes— linear classifiers trained on top of frozen MLM representations with average pooling —and find AUTOPROMPT has comparable or higher accuracy, despite linear probes being susceptible to false positives. Overall, these results demonstrate that both BERT and RoBERTa have some inherent knowledge of natural language inference.

We also examine the efficacy of AUTOPROMPT in the low-data regime (using the same procedure as SST-2) on the unbiased 3-way SICK-E data. The results in Figure 2 show that AUTOPROMPT performs on par with finetuned BERT and significantly better than finetuned RoBERTa in low data settings.

MLMs Excel on Contradiction We find that the label tokens are more interpretable for *con-*

tradiction compared to *entailment* or *neutral* (examples in Table 3). We investigate if this hurts the model performance on entailment and neutral classes. We measure the precision for each label in the 3-way balanced SICK-E dataset. BERT achieves 74.9%, 54.4%, and 36.8% precision for contradiction, entailment, and neutral cases, respectively, while RoBERTa obtains 84.9%, 65.1%, and 57.3%. These results suggest that AUTOPROMPT may be more accurate for concepts that can be easily expressed using natural label tokens.

5 Fact Retrieval

An important question is whether pretrained MLMs *know* facts about real-world entities. The LAMA dataset (Petroni et al., 2019) evaluates this using cloze tests that consist of (sub, rel, obj) triples, e.g. (Obama, bornIn, Hawaii), and *manually* created prompts with missing objects, e.g. "Obama was born in [MASK].". LPAQA (Jiang et al., 2020) extends this idea by *systematically* creating prompts that are generated by mining Wikipedia, paraphrasing, and crowdsourcing. In this section, we use the same cloze-style setup but *automatically* generate prompts in order to better evaluate the factual knowledge of MLMs. We compare our approach against LAMA and LPAQA, which are explicitly designed for the task of fact retrieval.

Setup We reformulate fact retrieval by mapping (sub,rel,obj) triples to a prompt using the template "{sub}[T]. . . [T][P].", where the trigger tokens are specific to the relation rel and the correct object obj is the label token. We use the original test set from LAMA (Petroni et al., 2019), henceforth *Original*. To collect training data for AUTOPROMPT, we gather at most 1000 facts for each of the 41 relations in LAMA from the T-REx dataset (ElSahar et al., 2018). For the relations that still have less than 1000 samples, we gather extra facts straight from Wikidata. We ensure that none of the T-REx triples are present in the test set, and we split the data 80-20 into train and development sets. Moreover, because the collected T-REx data is from a slightly different distribution than the LAMA test set, we also consider a separate evaluation where we split the T-REx triples into a 60-20-20 train/dev/test split and evaluate on the test set. This *T-REx* dataset is used to measure the performance of our prompts when the train and test data is from the same distribution.

Task	Prompt Template	Prompt found by AUTOPROMPT	Label Tokens
Sentiment Analysis	{sentence} [T]...[T] [P].	unflinchingly bleak and desperate Writing academicswhere overseas will appear [MASK].	**pos**: partnership, extraordinary, ##bla **neg**: worse, persisted, unconstitutional
NLI	{prem}[P][T]...[T]{hyp}	Two dogs are wrestling and hugging [MASK] concretepathic workplace There is no dog wrestling and hugging	**con**: Nobody, nobody, nor **ent**: ##found, ##ways, Agency **neu**: ##ponents, ##lary, ##uated
Fact Retrieval	*X plays Y music* {sub}[T]...[T][P].	Hall Overton fireplacemade antique son alto [MASK].	
Relation Extraction	*X is a Y by profession* {sent}{sub}[T]...[T][P].	Leonard Wood (born February 4, 1942) is a former Canadian politician. Leonard Wood gymnasium brotherdicative himself another [MASK].	

Table 3: **Example Prompts** by AUTOPROMPT for each task. On the left, we show the prompt template, which combines the input, a number of trigger tokens [T], and a prediction token [P]. For classification tasks (sentiment analysis and NLI), we make predictions by summing the model's probability for a number of automatically selected label tokens. For fact retrieval and relation extraction, we take the most likely token predicted by the model.

We use AUTOPROMPT with 5 or 7 tokens, and select the search parameters using the T-REx development set. We prevent proper nouns and tokens that appear as gold objects in the training data from being selected as trigger tokens. This is done to prevent AUTOPROMPT from "cheating" by embedding common answers inside the prompt. To evaluate, we observe the rank of the true object in label token distribution of the MLM, and use standard ranking metrics: mean reciprocal rank (MRR), precision-at-1 (P@1), and precision-at-10 (P@10).

Results Table 4 shows the performance of MLMs with different prompting methods, and we show qualitative examples in Table 3 and in Appendix C. Prompts generated using AUTOPROMPT can extract factual knowledge from BERT more effectively than their manual and mined counterparts: we improve P@1 by up to 12 points. Moreover, despite AUTOPROMPT using only one prompt per relation, it still outperforms LPAQA's ensemble method (which averages predictions for up to 30 prompts) by approximately 4 points. Using 7 trigger tokens achieves slightly higher scores than 5 trigger tokens, although the difference is not substantial. This indicates that our approach is stable to the choice of trigger length, which is consistent with our sentiment analysis results. Overall, these results show that AUTOPROMPT can retrieve facts more effectively than past prompting methods, thus demonstrating that BERT contains more factual knowledge than previously estimated.

Relation Breakdown We also provide a detailed breakdown of the prompts found by Petroni et al. (2019) and AUTOPROMPT, and their associated accuracies in Appendix C, Table 7. Manual prompts are competitive when the prompt is *easy* to specify, e.g., the prompt *"was born in"* for the PLACE OF BIRTH relation. On the other hand, AUTOPROMPT performs especially well for relations that are difficult to specify in a natural language prompt. For example, Petroni et al. (2019)'s prompt for the POSITION PLAYED ON TEAM relation is "{sub} plays in [MASK] position", which is not as specific as the relation requires. Although the prompt from AUTOPROMPT is not grammatical ("{sub} *ediatric striker ice baseman defensive* {obj}"), it does contain tokens that are directly related to sports.

BERT outperforms RoBERTa We finally directly compare BERT and RoBERTa. To do so, we subsample the LAMA test set to consist of examples where the object is a single token for both BERT and RoBERTa (*Original-RoBERTa*).[4] BERT actually slightly outperforms RoBERTa, and we find that the prompts generated for RoBERTa tend to contain more irrelevant words (see Appendix C, Table 7). For example, the prompt generated by RoBERTa for the PLAYS INSTRUMENT relation contains words such as "Trump" and symbols such as "," ()," for the POSITION PLAYED ON TEAM relation. It is surprising that RoBERTa does not

---

[4]The original dataset consists of examples where the object is a single token for BERT.

Prompt Type	Original			T-REx				Model	MRR	P@10	P@1
	MRR	P@10	P@1	MRR	P@10	P@1					
LAMA	40.27	59.49	31.10	35.79	54.29	26.38		BERT	55.22	74.01	45.23
LPAQA (Top1)	43.57	62.03	34.10	39.86	57.27	31.16		RoBERTa	49.90	68.34	40.01
AUTOPROMPT 5 Tokens	53.06	72.17	42.94	54.42	70.80	45.40					
AUTOPROMPT 7 Tokens	53.89	73.93	43.34	54.89	72.02	45.57					

Table 4: **Factual Retrieval:** On the left, we evaluate BERT on fact retrieval using the *Original* LAMA dataset from Petroni et al. (2019). For all three metrics (mean reciprocal rank, mean precision-at-10 (P@10), and mean precision-at-1(P@1)), AUTOPROMPT significantly outperforms past prompting methods. We also report results on a *T-REx* version of the data (see text for details). On the right, we compare BERT versus RoBERTa on a subset of the LAMA data using AUTOPROMPT with 5 tokens.

perform better than BERT, and it is worthy of investigating this further in future work. Additionally, recall that prompting is a *lower bound* on a model's knowledge: the lower relative performance does not mean that the model actually knows less.

6 Relation Extraction

Apart from evaluating whether MLMs *know* facts, it is also important to evaluate whether they can *extract knowledge* from text. In this section, we use the task of relation extraction (RE)—to identify how entities are related in a given sentence—an important task in information extraction. We create RE prompts in a similar fashion as fact retrieval: for a given triple (subj,rel,obj) and sentence that expresses this relation, we construct a prompt as "{sent}{sub}[T]. . .[T][P].", where the trigger tokens are specific to the relation, and label token is the correct object obj (see Table 3 for an example).

Setup We use the T-Rex dataset for RE because each T-REx fact comes with context sentences that mention the subject and object surface forms. We compare AUTOPROMPT to LAMA and LPAQA (their prompts are still useful here), as well as a recent supervised relation extraction model (Sorokin and Gurevych, 2017) that was also used by Petroni et al. (2019). To make the evaluation fair for the supervised RE model, we modify the standard RE evaluation. We give the model credit as long as it does not predict a different relation for the subject and object, i.e. we ignore the "no relation" prediction and all other relations. We also drop all sentences from evaluation for which the model's named entity extractor failed to identify the subject and the object as entities. See Appendix B for further details. For the evaluation of all systems, we treat a prediction as correct if it is either the canonical version of the object (e.g., "USA") or the rendered surface form (e.g., "American") for *any* of the context sentences in a given triple.

Results Table 5 shows the results for BERT and RoBERTa. MLMs can extract relational information *more effectively* than the supervised RE model, providing up to a 33% increase on the task when using AUTOPROMPT. RoBERTa also outperforms the supervised RE model, although it is worse than BERT (likely for similar reasons as we outline in Section 5). For both BERT and RoBERTa, we notice that the trigger tokens consist of words related to their corresponding relations (see Appendix D, Table 8 for full list), e.g. RoBERTa selects "defy trademarks of namesake manufacturer" for relation MANUFACTURER/PRODUCER OF PRODUCT.

Perturbed Sentence Evaluation A possible explanation for the strong results of MLMs in the RE setting is that they may *already* know many of the relations. Thus, they may directly predict the objects instead of *extracting* them. To separate this effect, we synthetically perturb the relation extraction dataset by replacing each object in the test data with a random other object and making the same change to the prompt. For example, *"Ryo Kase (born November 9, 1974 in ~~Yokohama~~→Yorkshire) is a Japanese actor"* where Ryo Kase is the subject, Yokohama is the original object, and Yorkshire is the new object. We regenerate the prompts using the perturbed version of the data.

The accuracy of the RE model does not change significantly on the perturbed data (Table 5), however, the accuracy of the MLMs decreases significantly. This indicates that a significant portion of MLM accuracy comes from background information rather than relation extraction. Nevertheless, our prompts for BERT outperform their LAMA and LPAQA counterparts, which provides further evidence that AUTOPROMPT produces better probes.

Model	Original	Perturbed
Supervised RE LSTM	57.95	58.81
BERT (LAMA)	69.06	28.02
BERT (LPAQA)	76.55	30.79
BERT (AUTOPROMPT)	90.73	56.43
RoBERTa (AUTOPROMPT)	60.33	28.95

Table 5: **Relation Extraction:** We use prompts to test pretrained MLMs on relation extraction. Compared to a state-of-the-art LSTM model from 2017, MLMs have higher mean precision-at-1 (P@1), especially when using prompts from AUTOPROMPT. We also test models on sentences that have been edited to contain incorrect facts. The accuracy of MLMs drops significantly on these sentences, indicating that their high performance stems from their factual knowledge.

7 Discussion

Prompting as an Alternative to Finetuning The goal of prompting a language model is to probe the knowledge that the model acquired from pretraining. Nevertheless, prompting has some practical advantages over finetuning for solving real-world tasks. First, as shown in Section 3, prompts generated using AUTOPROMPT can achieve higher accuracy than finetuning in the *low-data regime*. Moreover, prompting has advantages over finetuning when trying to solve *many different tasks* (e.g., the many users of the OpenAI GPT-3 API (Brown et al., 2020)). In particular, finetuning requires storing large language model checkpoints for each individual task, and drastically increases system cost and complexity because it requires deploying many different models at the same time. Prompting alleviates both of these issues. Only prompts are stored for each individual task, while the same pretrained model is used across all of the tasks.

Limitations of Prompting There are certain phenomena that are difficult to elicit from pretrained language models via prompts. In our preliminary evaluation on datasets such as QQP (Iyer et al., 2017) and RTE (Dagan et al., 2005), prompts generated manually and with AUTOPROMPT did not perform considerably better than chance. However, we cannot conclude that BERT does not know paraphrasing or entailment from these results. In general, different probing methods have different tasks and phenomena they are suitable for: AUTOPROMPT makes *prompt-based probes* more generally applicable, but, it still remains just one tool in the toolbox of the interpretability researcher.

Limitations of AUTOPROMPT One downside of AUTOPROMPT is that it requires labeled training data. Although this is also required for other probing techniques (e.g., linear probing classifiers), manual prompts rely on domain/language insights instead of labeled data. Compared to human-designed prompts, AUTOPROMPT generated prompts lack interpretability, which is similar to other probing techniques, such as linear probing classifiers. Another limitation of AUTOPROMPT is that it can sometimes struggle when the training data is highly imbalanced. For example, in Sections 4 and 5 we show that the prompts often just increase the likelihood of the majority label. Rebalancing the training data can help to mitigate this problem. Finally, due to the greedy search over the large discrete space of phrases, AUTOPROMPT is sometimes brittle; we leave more effective crafting techniques for future directions.

8 Conclusion

In this paper, we introduce AUTOPROMPT, an approach to develop automatically-constructed prompts that elicit knowledge from pretrained MLMs for a variety of tasks. We show that these prompts outperform manual prompts while requiring less human effort. Furthermore, the results for sentiment analysis and textual entailment suggest that, in some data-scarce settings, it may be more effective to *prompt* language models than to finetune them for the task. Although we focus only on masked language models in this paper, our method can be trivially extended to standard language models, and thus maybe useful for constructing inputs for models like GPT-3 (Brown et al., 2020). Source code and datasets to reproduce the results in this paper is available at http://ucinlp.github.io/autoprompt.

References

Zied Bouraoui, Jose Camacho-Collados, and Steven Schockaert. 2019. Inducing relational knowledge from BERT. In *AAAI*.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. In *EMNLP*.

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*.

Alexis Conneau, Germán Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single vector: Probing sentence embeddings for linguistic properties. In *ACL*.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The PASCAL recognising textual entailment challenge. In *Machine Learning Challenges Workshop*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305*.

Hady ElSahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon S. Hare, Frédérique Laforest, and Elena Simperl. 2018. T-REx: A large scale alignment of natural language with knowledge base triples. In *LREC*.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. In *EMNLP*.

Shankar Iyer, Nikhil Dandekar, and Kornel Csernai. 2017. First quora dataset release: Question pairs.

Sarthak Jain and Byron C Wallace. 2019. Attention is not explanation. In *NAACL*.

Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? In *TACL*.

Sunjae Kwon, Cheongwoong Kang, Jiyeon Han, and Jaesik Choi. 2019. Why do masked neural language models still need common sense knowledge? *arXiv preprint arXiv:1911.03024*.

Patrick Lewis, Ludovic Denoyer, and Sebastian Riedel. 2019. Unsupervised question answering by cloze translation. In *ACL*.

Nelson F Liu, Matt Gardner, Yonatan Belinkov, Matthew Peters, and Noah A Smith. 2019. Linguistic knowledge and transferability of contextual representations. In *NAACL*.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, Roberto Zamparelli, et al. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *LREC*.

Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2020. On the stability of fine-tuning bert: Misconceptions, explanations, and strong baselines.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *NAACL*.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? In *EMNLP*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *Technical report*.

Timo Schick and Hinrich Schütze. 2020. Exploiting cloze questions for few-shot text classification and natural language inference. *arXiv preprint arXiv:2001.07676*.

Vered Shwartz, Peter West, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Unsupervised commonsense question answering with self-talk. *arXiv preprint arXiv:2004.05483*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*.

Daniil Sorokin and Iryna Gurevych. 2017. Context-aware representations for knowledge base relation extraction. In *EMNLP*.

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

Elena Voita and Ivan Titov. 2020. Information-theoretic probing with minimum description length. In *EMNLP*.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing NLP. In *EMNLP*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *ICLR*.

Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not explanation. In *EMNLP*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi 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 M. Rush. 2019. HuggingFace's Transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

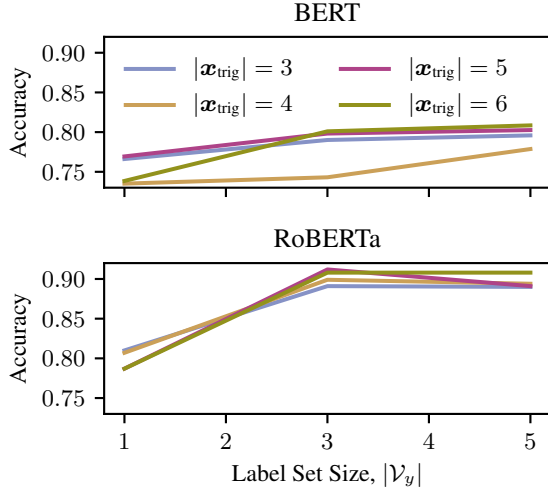## A Effect of Hyperparameters on Sentiment Analysis

Figure 3: **Effect of Label and Trigger Set Sizes** on sentiment analysis. The number of candidate replacements is fixed at $|\mathcal{V}_{\text{cand}}| = 100$. Increasing the label set size improves performance, while changing the trigger length does not have much impact.

To measure the effects of the AUTOPROMPT search hyperparameters, we plot the validation accuracy as a function of label set size $|\mathcal{V}_y|$ and the number of trigger tokens $|\boldsymbol{x}_{\text{trig}}|$ in Figure 3. We fix the number of candidates at $|\mathcal{V}_{\text{cand}}| = 100$. We observe similar trends when $|\mathcal{V}_{\text{cand}}| = 10$.

Varying the number of trigger tokens generally has little effect. On the other hand, there is a substantial increase in accuracy when increasing the label set size from 1 to 3 (approximately +5% for BERT, and +10% for RoBERTa). After analyzing the label sets, we find that our method generally produces intuitive results—"marvelous" and "philanthrop" are associated with positive sentiment, whereas "worse" and "incompetence" are associated with negative sentiment for RoBERTa.

B Relation Extraction Details

Following Petroni et al. (2019), we use the pretrained RE model from Sorokin and Gurevych (2017) as our baseline. To encode the sentence, this model uses a combination of an LSTM-based relation encoder and an attention mechanism. To make predictions, the model constructs a knowledge graph whose edges are the extracted relation triples. The standard RE evaluation measures how well the model predicts the relation types of entity pairs on the sentence level.

Since our goal is to extract the object of relation triplets, rather than the relation itself, we tweak the standard RE evaluation. We feed the RE model sentences from test facts and we query the resulting graph for all edges that contain the given subject and relation. Then we select the triple with the highest confidence and compare it's object to the gold object. We do this for every fact and take the average across all relations to get the overall precision. The RE model is not trained to predict two of the original T-REx relations. For fair comparison, we exclude these two relations for our evaluation.

C   Additional Fact Retrieval Results

Relation	Manual Prompt (LAMA)	#train	LAMA	LPAQA	AUTOPROMPT
P1001	[X] is a legal term in [Y]	1000	70.47	72.75	82.45
P101	[X] works in the field of [Y]	864	9.91	5.32	12.79
P103	The native language of [X] is [Y]	1000	72.16	72.16	82.09
P106	[X] is a [Y] by profession	1000	0.63	0.0	14.72
P108	[X] works for [Y]	376	6.79	5.74	8.62
P127	[X] is owned by [Y]	548	34.79	32.46	35.95
P1303	[X] plays [Y]	1000	7.59	18.02	15.38
P131	[X] is located in [Y]	1000	23.27	22.81	37.46
P136	[X] plays [Y] music	1000	0.75	16.76	55.42
P1376	[X] is the capital of [Y]	310	73.93	59.83	40.17
P138	[X] is named after [Y]	856	61.55	59.69	66.05
P140	[X] is affiliated with the [Y] religion	445	0.63	59.83	75.26
P1412	[X] used to communicate in [Y]	1000	65.02	64.71	71.21
P159	The headquarter of [X] is in [Y]	1000	32.37	35.57	35.47
P17	[X] is located in [Y]	1000	31.29	35.48	52.15
P176	[X] is produced by [Y]	1000	85.64	81.67	87.78
P178	[X] is developed by [Y]	560	62.84	59.12	66.72
P19	[X] was born in [Y]	1000	21.08	20.87	19.92
P190	[X] and [Y] are twin cities	895	2.41	1.91	2.31
P20	[X] died in [Y]	1000	27.91	27.91	31.16
P264	[X] is represented by music label [Y]	1000	9.56	10.26	43.82
P27	[X] is [Y] citizen	1000	0.0	41.51	46.69
P276	[X] is located in [Y]	1000	41.5	41.5	44.11
P279	[X] is a subclass of [Y]	1000	30.74	14.75	54.93
P30	[X] is located in [Y]	1000	25.44	18.56	70.36
P31	[X] is a [Y]	1000	36.66	36.66	51.95
P36	The capital of [X] is [Y]	1000	62.16	62.16	60.6
P361	[X] is part of [Y]	1000	23.61	31.44	17.7
P364	The original language of [X] is [Y]	1000	44.51	43.93	48.48
P37	The official language of [X] is [Y]	311	54.55	56.83	62.63
P39	[X] has the position of [Y]	1000	7.96	16.14	30.72
P407	[X] was written in [Y]	1000	59.18	65.22	68.42
P413	[X] plays in [Y] position	1000	0.53	23.74	41.7
P449	[X] was originally aired on [Y]	1000	20.89	9.08	34.39
P463	[X] is a member of [Y]	679	67.11	57.33	54.22
P47	[X] shares border with [Y]	1000	13.67	13.34	19.52
P495	[X] was created in [Y]	1000	16.5	32.23	36.63
P527	[X] consists of [Y]	1000	11.07	10.55	25.61
P530	[X] maintains diplomatic relations with [Y]	927	2.81	3.92	3.11
P740	[X] was founded in [Y]	1000	7.59	13.68	13.89
P937	[X] used to work in [Y]	1000	29.77	39.1	38.36

Table 6: A breakdown of all relations for fact retrieval on the original dataset from Petroni et al. (2019). We compare P@1 of prompts generated by LAMA, LPAQA, and our approach using five prompt tokens.

Relation	Method	Prompt	P@1
P101	Manual	[X] works in the field of [Y]	11.52
	AUTOPROMPT BERT	[X] probability earliest fame totaled studying [Y]	15.01
	AUTOPROMPT RoBERTa	[X] 1830 dissertation applying mathsucci [Y]	0.17
P103	Manual	The native language of [X] is [Y]	74.54
	AUTOPROMPT BERT	[X]PA communerug speaks proper [Y]	84.87
	AUTOPROMPT RoBERTa	[X]neau optionally fluent!?ïraditional [Y]	81.61
P106	Manual	[X] is a [Y] by profession	0.73
	AUTOPROMPT BERT	[X] supporters studied politicians musician turned [Y]	15.83
	AUTOPROMPT RoBERTa	[X] (), astronomers businessman·former [Y]	19.24
P127	Manual	[X] is owned by [Y]	36.67
	AUTOPROMPT BERT	[X] is hindwings mainline architecture within [Y]	47.01
	AUTOPROMPT RoBERTa	[X] picThom unwillingness officially governs [Y]	39.58
P1303	Manual	[X] plays [Y]	18.91
	AUTOPROMPT BERT	[X] playingdrum concertoative electric [Y]	42.69
	AUTOPROMPT RoBERTa	[X]Trump learned soloKeefe classical [Y]	44.44
P136	Manual	[X] plays [Y] music	0.7
	AUTOPROMPT BERT	[X] freaking genre orchestra fiction acid [Y]	59.95
	AUTOPROMPT RoBERTa	[X] blends postwar hostage drama sax [Y]	52.97
P1376	Manual	[X] is the capital of [Y]	81.11
	AUTOPROMPT BERT	[X] boasts native territory traditionally called [Y]	63.33
	AUTOPROMPT RoBERTa	[X] limestone depositedati boroughDepending [Y]	28.33
P178	Manual	[X] is developed by [Y]	62.76
	AUTOPROMPT BERT	[X] is memory arcade branding by [Y]	64.45
	AUTOPROMPT RoBERTa	[X] 1987 floppy simulator users sued [Y]	69.56
P20	Manual	[X] died in [Y]	32.07
	AUTOPROMPT BERT	[X] reorganizationotype photographic studio in [Y]	33.53
	AUTOPROMPT RoBERTa	[X].. enigmatic twentieth nowadays near [Y]	31.33
P27	Manual	[X] is [Y] citizen	0.0
	AUTOPROMPT BERT	[X] m³ badminton pieces internationally representing [Y]	46.13
	AUTOPROMPT RoBERTa	[X] offic organise forests statutes northwestern [Y]	42.07
P276	Manual	[X] is located in [Y]	43.73
	AUTOPROMPT BERT	[X] consists kilograms centred neighborhoods in [Y]	44.64
	AUTOPROMPT RoBERTa	[X] manoeuv constructs whistleblowers hills near [Y]	37.47
P279	Manual	[X] is a subclass of [Y]	31.04
	AUTOPROMPT BERT	[X] is î adequately termed coated [Y]	55.65
	AUTOPROMPT RoBERTa	[X],formerly prayers unstaceous [Y]	52.55
P37	Manual	The official language of [X] is [Y]	56.89
	AUTOPROMPT BERT	[X]inen dialects resembled officially exclusively [Y]	54.44
	AUTOPROMPT RoBERTa	[X]onen tribes descending speak mainly [Y]	53.67
P407	Manual	[X] was written in [Y]	60.21
	AUTOPROMPT BERT	[X] playedić every dialect but [Y]	69.31
	AUTOPROMPT RoBERTa	[X] scaven pronunciation.*Wikipedia speaks [Y]	72.0
P413	Manual	[X] plays in [Y] position	0.53
	AUTOPROMPT BERT	[X] played colors skier ↔ defensive [Y]	41.71
	AUTOPROMPT RoBERTa	[X]," (), ex-,Liverpool [Y]	23.21

Table 7: Examples of manual prompts (first line, shown with BERT's P@1) and prompts generated via AUTO-PROMPT for Fact Retrieval.

D Additional Relation Extraction Results

Relation	Model	Context and Prompt	Prediction
P103 (native language)	BERT	Alexandra Lamy (born 14 October 1971) is a French actress. Alexandra Lamy speaks airfield dripping % of [MASK].	French
P36 (capital)	RoBERTa	Kirk was born in Clinton County, Ohio, and he entered service in Wilmington, Ohio. Clinton County famously includes the zoo influencing [MASK].	Wilmington
P530 (diplomatic relation)	BERT	The Black Sea forms in an east-west trending elliptical depression which lies between Bulgaria, Georgia, Romania, Russia, Turkey, and Ukraine. Ukraine qualified some immigration actually entered [MASK].	Russia
P106 (occupation)	RoBERTa	Spencer Treat Clark (born September 24, 1987) is an American actor who has appeared in several films, including Gladiator, Mystic River, and Unbreakable. Spencer Treat Clark famously the famously handsome the [MASK].	Hulk
P276 (location)	BERT	The Immortal Game was a chess game played by Adolf Anderssen and Lionel Kieseritzky on 21 June 1851 in ~~London~~Seoul, during a break of the first international tournament. The Immortal Game locatedstered regardless streets in [MASK].	Seoul
P176 (manufacturer)	RoBERTa	The Honda Civic del Sol is a 2-seater front-engined, front wheel drive, targa top car manufactured by ~~Honda~~Toyota in the 1990s. Honda Civic del Sol defy trademarks of namesake manufacturer [MASK].	Toyota
P279 (subclass of)	BERT	Mizeria is a Polish ~~salad~~sandwich consisting of thinly sliced or grated cucumbers, often with sour cream though in some cases oil. Mizeria is calls direcend altitude [MASK].	food
P463 (member of)	RoBERTa	~~Rush~~Aerosmith was a Canadian rock band consisting of Geddy Lee (bass, vocals, keyboards), Alex Lifeson (guitars), and Neil Peart (drums, percussion, lyricist). Alex Lifeson affiliatedalach the internationally initials [MASK].	Kiss

Table 8: Examples of prompts generated using AUTOPROMPT for relation extraction. Underlined words represent the gold object. The bottom half of the Table shows examples of our augmented evaluation where the original objects (represented by crossed-out words) are replaced by new objects.