

UNIVERSITY OF CALIFORNIA,
IRVINE

When Life Gives You LMs, Probe Them for Knowledge With Automatically Generated
Prompts

THESIS

submitted in partial satisfaction of the requirements
for the degree of

MASTER OF SCIENCE

in Computer Science

by

Taylor Shin

Thesis Committee:
Professor Sameer Singh, Chair
Professor Roy Fox
Professor Richard Futrell

2020

TABLE OF CONTENTS

	Page
LIST OF FIGURES	iii
LIST OF TABLES	iv
ACKNOWLEDGMENTS	v
ABSTRACT OF THE THESIS	vi
1 Introduction	1
2 Background	4
2.1 Eliciting Knowledge with Automatically Generated Prompts	4
2.2 Background and Notation	5
2.3 Gradient-Based Prompt Search	6
2.4 Evaluation Setup	7
3 Fact Retrieval	8
3.1 Setup	8
3.2 Results	9
3.3 Relation Breakdown	10
3.4 Comparison to RoBERTa	11
4 Relation Extraction	15
4.1 Setup	15
4.2 Results	16
4.3 Augmented Training and Evaluation	17
5 Discussion and Conclusion	20
Bibliography	22

LIST OF FIGURES

	Page
2.1 Probing Example of Fact Retrieval	5

LIST OF TABLES

	Page
3.1 Factual Retrieval Results	10
3.2 Examples of Templates for Each Task	11
3.3 Relation Breakdown for Fact Retrieval	13
3.4 Examples of Manual and Trigger-Based Prompts for Fact Retrieval	14
4.1 Relation Extraction Results	17
4.2 Examples of Triggers for Relational Extraction	19

ACKNOWLEDGMENTS

I would like to thank Professor Sameer Singh for advising me during my time at UCI. I'd also like to thank Eric Wallace, Robert L. Logan IV, and Yasaman Razeghi for their help and feedback on this project.

ABSTRACT OF THE THESIS

When Life Gives You LMs, Probe Them for Knowledge With Automatically Generated Prompts

By

Taylor Shin

Master of Science in Computer Science

University of California, Irvine, 2020

Professor Sameer Singh, Chair

Determining the knowledge captured by pretrained language models is an important challenge, and is commonly tackled by probing model representations using classifiers. However, it is difficult to design probes for *semantic* knowledge such as facts. Reformulating these semantic tasks as cloze tests (i.e., fill-in-the-blank problems) is a promising method for probing such knowledge, but, it requires manually crafting textual prompts that elicit the desired knowledge, thus limiting its use. In this paper, we develop an *automated, task-agnostic* method to create cloze prompts for any classification task based on a gradient-guided search. Applying our technique, we find prompts that demonstrate MLMs have an inherent capability to perform fact retrieval and relation extraction. We show that our prompts elicit more accurate factual knowledge from MLMs compared to manual prompts, and further, MLMs can be used as relation extractors out of the box, when prompted with suitable prompts, more effectively than recent supervised RE models.

Chapter 1

Introduction

Pretrained masked language models (MLMs) have played an important role in advancing the field of NLP by improving the state-of-the-art results of many downstream tasks (Peters et al., 2018; Devlin et al., 2019; Radford et al., 2019). There is an increasing push towards understanding these models and their innate knowledge due to the success of these models as well as their black-box nature. MLMs have been shown to possess linguistic (Liu et al., 2019a), factual (Petroni et al., 2019), and commonsense (Bosselut et al., 2019) knowledge; it is possible to extract such information from a model with probing techniques.

Probing allows us to gain a better understanding of intricate language models (LMs) and essentially helps us know what language models know. Two of the common strategies for eliciting LM knowledge are probing classifiers (Conneau et al., 2018; Tenney et al., 2019) and attention visualizations (Clark et al., 2019; Vig and Belinkov, 2019). Probing classifiers extract linguistic information from contextual embeddings that are provided by a pretrained encoder on a certain task, and have shown that encoders capture a wide range of properties and information. However, probing classifiers are insufficient for two major reasons. First, they are mainly used for token-level classification tasks (e.g. POS tagging) and are limited in

terms of semantic tasks like textual entailment and information extraction. Second, probing classifiers are susceptible to false positives especially since a high probing accuracy does not necessarily indicate that an LM contains a certain piece of knowledge (Hewitt and Liang, 2019; Voita and Titov, 2020). Attention visualizations show that the attention mechanisms of pretrained models capture syntactic information, but they also run into the same shortcomings where correlation does not imply causation with the target knowledge (Jain and Wallace, 2019; Wiegrefe and Pinter, 2019).

An alternate approach to eliciting knowledge from MLMs is to convert tasks into a format that is closely aligned with language models. For example, Radford et al. (2019) convert the task of summarization into a format more suitable for language modeling by appending “TL;DR:” to the end of an article and having a LM generate the rest of the text or in other words, a summary of the article. Furthermore, pretrained models that are enhanced with knowledge bases or external entity representations, i.e. KGLM (IV et al., 2019) and Know-Bert (Peters et al., 2019), have been successfully probed for factual knowledge with the use of prompts.

In particular, we focus on two approaches that conduct in-depth analysis on the innate relation knowledge of state-of-the-art pretrained language models: LAMA (Petroni et al., 2019) and LPAQA (Jiang et al., 2020). Petroni et al. (2019) discover that BERT (Devlin et al., 2019), even without fine-tuning, stores relational and commonsense knowledge that can be probed with fill-in-the-blank cloze statements. They argue that although LMs are like traditional, structured knowledge bases, they have several appealing advantages like not requiring schema engineering, complex NLP pipelines, or human supervision for training. They introduce the LAMA probe which manually defines prompts for each relation and evaluates a model on how well it ranks the ground-truth token for cloze statements. Jiang et al. (2020) build on the LAMA probe and generate prompts with mining-based and paraphrase-based methods. They also experiment with ensemble methods that combine the answers from a

set of diverse prompts to get a more accurate estimate of the knowledge stored in LMs. A downside of repurposing tasks as cloze tests is that it requires manually crafting the context to feed into the model. Not only is this time consuming and non-trivial for many tasks, more importantly, models are highly sensitive to this context and leads to the fact that improperly-constructed contexts can cause artificially low performance.

In this paper, we develop and propose an *automated, task-agnostic* method for generating cloze prompts. For any given task, such as fact retrieval (Figure 2.1), we automatically generate short sequences of tokens (typically 5–7) that cause the model to predict the masked object in fill-in-the-blank prompts. Our approach is based on universal adversarial triggers (Wallace et al., 2019): sequences of tokens that trigger a model to produce a specific prediction when concatenated to any input. However, instead of creating triggers that prompt *adversarial* behavior, we create triggers that prompt a model to make the *correct* predictions for a given task. Our method allows the probing of knowledge inside MLMs on any task without the need for extra parameters or a direct inspection of the model’s internals.

We first validate our approach on the fact retrieval tasks of 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, mining-based, and paraphrasing-based methods. Concretely, we achieve 41.9% precision-at-1 as opposed to the current best result of 39.6% (Jiang et al., 2020). Finally, we introduce a variant of this task, similar to relation extraction (RE), which tests whether MLMs can extract knowledge from a given piece of text as opposed to just recalling facts. We show that MLMs can *outperform* supervised RE models when context sentences with real facts are provided, however, MLMs stumble when context sentences are artificially falsified.

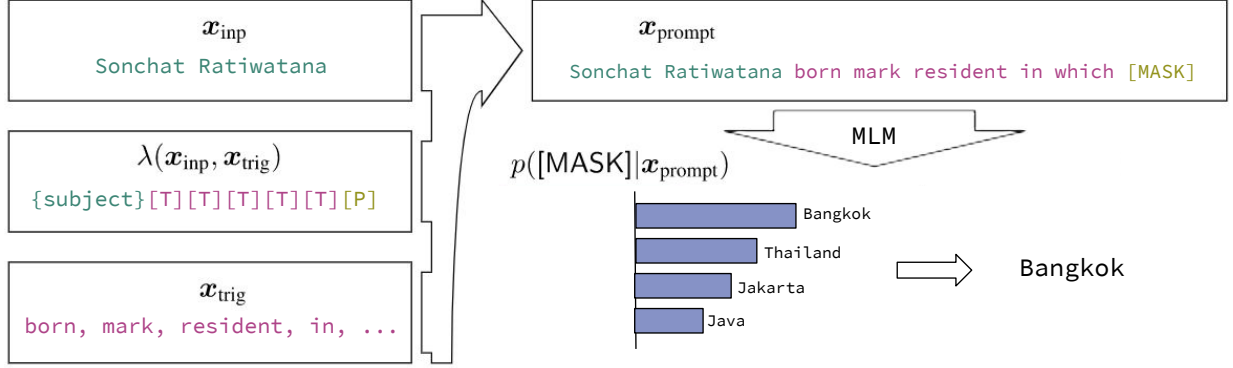
Chapter 2

Background

2.1 Eliciting Knowledge with Automatically Generated Prompts

We can measure the knowledge of MLMs on classification tasks by having them answer cloze-style statements (i.e. fill-in-the-blank sentences like “Obama was born in ____”). More specifically, we convert each relation triple into a prompt containing a masked word that can be filled in by the tokens associated with the class label. An illustration of this approach is provided in Figure 2.1. In terms of creating these prompts, we take the original task inputs (e.g., the subject, object, and context sentence in RE) and map them to a single sequence of tokens using a template. This template defines the placement of each input sequence that will be inserted in the prompt. Furthermore, the template also determines the location of a special [MASK] token for the MLM to fill in (denoted by [P] in the examples to distinguish it from other [MASK] tokens that might appear). Generally, additional tokens that appear in the prompt are either manually added (e.g., [Petroni et al. \(2019\)](#)) or rely on task-specific knowledge (e.g., [Jiang et al. \(2020\)](#)), which prevents the general use of this method. In the

Figure 2.1: Probing Example of Fact Retrieval



Description: Example demonstrating how our method is used to probe masked language model (MLM) performance on fact retrieval. Each original input, $\mathbf{x}_{\text{inp}}$, is mapped to a prompt, $\mathbf{x}_{\text{prompt}}$, containing a single [MASK] token. The prompt is created using a template, λ , by combining the original inputs with a set of trigger tokens, $\mathbf{x}_{\text{trig}}$ (that are shared across all instances, and automatically detected).

following section, we address this issue with a task-agnostic approach that automatically constructs prompts via a gradient-based search.

2.2 Background and Notation

Given a target sequence of tokens $\mathbf{x} = (x^{(1)}, \dots, x^{(|\mathbf{x}|)})$ drawn from a vocabulary $\mathcal{V}$, and a corresponding source sequence $\tilde{\mathbf{x}} = (\tilde{x}^{(1)}, \dots, \tilde{x}^{(|\mathbf{x}|)})$ where some of the tokens in $\mathbf{x}$ have been randomly replaced with a special [MASK] token, a MLM maximizes the probability of the target sequence conditioned on the source sequence $p(\mathbf{x}|\tilde{\mathbf{x}})$. The initial step MLMs such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019b) take to produce such probability distributions is to utilize a Transformer encoder (Vaswani et al., 2017) to produce contextualized embeddings of each token as follows

$$\mathbf{h} = \text{Transformer}_{\text{enc}}(\tilde{\mathbf{x}}) \quad (2.1)$$

For the purpose of prompt construction, we define the original task inputs as $\mathbf{x}_{\text{inp}}$ and denote the prompt that is fed into the model as $\mathbf{x}_{\text{prompt}}$. Once we use a template, λ , to map $\mathbf{x}_{\text{inp}}$ to $\mathbf{x}_{\text{prompt}}$, we feed the prompt into the MLM which produces a probability distribution over class labels (tokens in the vocabulary) $p(\text{[MASK]}|\mathbf{x}_{\text{prompt}})$ that describes which tokens most likely fill in the blank.

2.3 Gradient-Based Prompt Search

In this section, we propose a method for *automatic task-agnostic prompt construction* based on universal adversarial triggers (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 templates). These tokens are initialized to either tokens from a manual prompt or a common token like “the”, and then iteratively updated to maximize the label likelihood over batches of examples.

Like Wallace et al., we build upon HotFlip (Ebrahimi et al., 2018) and at each step we compute a first-order approximation of the change in the log-likelihood that would be produced by swapping the j th 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:

$$\mathcal{V}_{\text{cand}} = \underset{w \in \mathcal{V}}{\text{top-}k} [\mathbf{w}_{\text{in}}^T \nabla \log p(y|\mathbf{x}_{\text{prompt}})] \quad (2.2)$$

where $\mathbf{w}_{\text{in}}$ is the input embedding of w , y is the target label over a batch, and the gradient is taken with respect to the input embedding of $x_{\text{trig}}^{(j)}$.¹ 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 pro-

¹Disclaimer: this equation was borrowed from a manuscript written with Robert L. Logan IV, Yasaman Razeghi, Eric Wallace, and Sameer Singh

jection). An example of a prompt generated by this method for the task of fact retrieval is shown in Figure 2.1. As is typical of these trigger-based methods, the prompt appears to be nonsensical—a stark contrast to the kinds of prompts used in existing work (Petroni et al., 2019). However as we show later, although these prompts appear meaningless, they often provoke MLMs to produce better predictions.

2.4 Evaluation Setup

In the following sections, we apply our methods to probe BERT_{BASE} (110M parameters) and RoBERTa_{LARGE}’s (355M parameters) knowledge of the following tasks: fact retrieval and relation extraction. For sake of brevity, we will omit subscripts from the model names for the remainder of the paper. For all experiments, we use the PyTorch implementations and pretrained weights provided in the `transformers` Python library (Wolf et al., 2019). For all tasks, we perform the trigger search described in Section 2.3, using one or more batches of training data to identify the candidate set $\mathcal{V}_{\text{cand}}$ of replacement trigger tokens, and then evaluate the label likelihoods on the updated prompts on a separate batch of data, retaining the best trigger token in the next iteration of the search. At the end of each iteration, we measure likelihood of labels on withheld development data, and return the best prompt found during the search as our final output. Performance is measured using precision@ k and mean reciprocal rank (MRR) on a separate withheld test set.

Chapter 3

Fact Retrieval

One of the most important questions for pretrained masked language models is whether they *remember* real-world entities and facts about them. Recent work in LAMA (Petroni et al., 2019) formulates this goal as a cloze test by taking factual triples of the form (subject, relation, object), e.g. (Obama, bornIn, Hawaii) and manually formulating prompts with missing objects, e.g. “Obama was born in [MASK]”, to evaluate whether MLMs are able to fill-in-the-blank with the correct object. LPAQA (Jiang et al., 2020) extends this idea by automatically creating prompts via mining Wikipedia, paraphrasing, and crowdsourcing. In this section, we use this cloze-style setup with *automatically* generated prompts, to evaluate the extent to which MLMs are able to recall real-world facts. We also compare our approach against LAMA and LPAQA, which are explicitly designed for fact retrieval.

3.1 Setup

The task of fact retrieval can be reformulated as a cloze test by mapping triples of the form (sub, rel, obj) to a prompt “{sub} [t]...[t] [p].”, where the tokens [t]...[t] are

specific to the relation, and the correct object `obj` is the label token. We evaluate against LAMA and LPAQA using the original test set from LAMA (Petroni et al., 2019), henceforth *Original*. To create our prompts, we use the T-REx dataset (ElSahar et al., 2018), a subset of Wikidata triples, to collect at most 1000 facts for each of the 41 relations in LAMA. We ensure that none of the triples are present in the test set, and we split it 80-20 into train and dev. Moreover, because this T-REx data is from a slightly different distribution than the LAMA test set, we also create a *Resplit* setup using a 60-20-20 train/dev/test split from our collected T-REx triples. This *Resplit* dataset is used to measure the performance of our prompts when the data used for generating them is from the same distribution as the evaluation data.

We experiment with prompts with 5 or 7 trigger tokens, and choose the remaining hyperparameters of the search using the dev set. During the trigger update step, we omit proper nouns and tokens that appear as objects in the training data, to prevent signal leakage. For evaluation, we measure the rank of the true object in the label token distribution produced by the MLM, and use standard ranking metrics: mean reciprocal rank (MRR), precision-at-1 ($P@1$), and precision-at-10 ($P@10$).

3.2 Results

We show the performance of MLMs using various prompt generation techniques in Table 3.1. Example prompts are shown in Table 3.2. The automatically generated prompts extract factual knowledge from BERT more effectively than their manual and mined counterparts with up to a 10 point increase in $P@1$. We also outperform LPAQA’s ensemble method, that averages the predictions of up to 30 prompts per relation, by 2 points and do so with only one prompt per relation. The prompt format with 7 trigger tokens achieves slightly higher scores than the format with 5 trigger tokens, although the difference is not substantial,

Table 3.1: Factual Retrieval Results

Prompt Type	Original			Resplit		
	MRR	P@10	P@1	MRR	P@10	P@1
LAMA	40.27	59.49	31.1	35.79	54.29	26.38
LPAQA (Top1)	43.57	62.03	34.1	39.86	57.27	31.16
5 Trigger Tokens	49.88	69.67	39.90	49.74	68.19	40.00
7 Trigger Tokens	52.04	71.24	41.91	52.17	69.87	42.48

Model	MRR	P@10	P@1
BERT	54.15	73.85	43.43
RoBERTa	47.54	65.29	38.06

Description: Mean P@1 of various methods for generating prompts. The methods in the top Table use the *Original* dataset from [Petroni et al. \(2019\)](#). On the bottom, we use *Original-RoBERTa* (a subset of the data) and report the P@1 of the 5 Trigger Token prompt.

indicating that our approach is somewhat stable to the choice of trigger length. The fact that our trigger-based prompts retrieve factual knowledge more efficiently than manually- and automatically-mined prompts demonstrates that BERT contains more factual knowledge than previously estimated.

3.3 Relation Breakdown

We provide a detailed breakdown of the manual and discovered prompts in Table 3.4. Manual prompts perform better or comparable to automatically generated prompts when the prompt is *easy* to manually specify, e.g., the prompt “*was born in*” for PLACE OF BIRTH relation is difficult to beat. On the other hand, our method performs well for relations for which it is difficult to manually specify the prompts. For example, the manual prompt for POSITION PLAYED ON TEAM relation is “*{sub} plays in {obj} position*”, which is not as specific as the relation requires. Although the generated prompt is not grammatical (“*{sub} baseball attackerpard defensive {obj}*”), it does contain tokens that

Table 3.2: Examples of Templates for Each Task

Template	Prompt	Label Tokens (* = model prediction)
Fact Retrieval		
$\{\text{sub}\}\{\text{t}\}\dots\{\text{t}\}\{\text{p}\}$ <i>X plays Y music</i>	Hall Overton puzzles saxophonist miniseries 1960s modern [MASK].	* jazz [-0.8], music [-1.7], era [-2.9], rock [-3.8], day [-4.4], ...
Relation Extraction		
$\{\text{sent}\}\{\text{sub}\}\{\text{t}\}\dots\{\text{t}\}\{\text{p}\}$ <i>X is a Y by profession</i>	Leonard Wood (born February 4, 1942) is a former Canadian politician. Leonard Wood gymnasium brotherdicationative himself another [MASK].	* politician [-0.5], actor [-2.3], lawyer [-3.3], musician [-3.8], businessman [-4.0], ...

Description: The template combines the input, the trigger tokens [t], a prediction token [p], and manual tokens. We show the top-5 predictions and their log-probabilities (in square brackets).

are directly related to sports and positions. Even after restricting trigger tokens to exclude objects and proper nouns, we notice that the search selects tokens that show bias from the relation-specific data. For example, the manual prompt for EMPLOYER relation is “*works for*” whereas our method generates “retrospective computer sued revenue giant”; the training data is dominated by tech companies like Microsoft and IBM so words like “computer” lead to a 24% increase in precision.

3.4 Comparison to RoBERTa

To directly compare the performance of prompts generated by BERT and RoBERTa, we create a new dataset (*Original-RoBERTa*) from the LAMA original set, that contains triples where the object is a single token for both BERT and RoBERTa. BERT seems to perform slightly better, although the difference is not substantial. However, on observing the per-relation breakdown, generated prompts for RoBERTa tend to contain more irrelevant words

(see Table 3.4). For example, the prompt generated by RoBERTa for FIELD OF WORK relation contains words like “girlfriend”, and “candles shop” in the prompt for the DIED IN relation. It is somewhat surprising that RoBERTa does not perform better than BERT, likely due to insufficient search for triggers or the fact that a large portion of BERT’s training data is Wikipedia (the source of these relations). Further, note that these approaches to measure knowledge in MLMs are *lower bounds*; worst performance does not mean that the model actually knows less.

Table 3.3: Relation Breakdown for Fact Retrieval

Relation	Manual Prompt (LAMA)	#train	LAMA	LPAQA	5 Trigger Tokens
P1001	[X] is a legal term in [Y]	1000	70.47	72.75	80.46
P101	[X] works in the field of [Y]	864	9.91	5.32	13.94
P103	The native language of [X] is [Y]	1000	72.16	72.16	81.78
P106	[X] is a [Y] by profession	1000	0.63	0.0	15.97
P108	[X] works for [Y]	376	6.79	5.74	8.36
P127	[X] is owned by [Y]	548	34.79	32.46	40.76
P1303	[X] plays [Y]	1000	7.59	18.02	22.76
P131	[X] is located in [Y]	1000	23.27	22.81	27.47
P136	[X] plays [Y] music	1000	0.75	16.76	55.64
P1376	[X] is the capital of [Y]	310	73.93	59.83	40.6
P138	[X] is named after [Y]	856	61.55	59.69	67.29
P140	[X] is affiliated with the [Y] religion	445	0.63	59.83	66.17
P1412	[X] used to communicate in [Y]	1000	65.02	64.71	51.7
P159	The headquarter of [X] is in [Y]	1000	32.37	35.57	34.44
P17	[X] is located in [Y]	1000	31.29	35.48	54.95
P176	[X] is produced by [Y]	1000	85.64	81.67	85.13
P178	[X] is developed by [Y]	560	62.84	59.12	63.34
P19	[X] was born in [Y]	1000	21.08	20.87	17.27
P190	[X] and [Y] are twin cities	895	2.41	1.91	1.61
P20	[X] died in [Y]	1000	27.91	27.91	29.59
P264	[X] is represented by music label [Y]	1000	9.56	10.26	27.51
P27	[X] is [Y] citizen	1000	0.0	41.51	39.13
P276	[X] is located in [Y]	1000	41.5	41.5	38.37
P279	[X] is a subclass of [Y]	1000	30.74	14.75	51.92
P30	[X] is located in [Y]	1000	25.44	18.56	72.72
P31	[X] is a [Y]	1000	36.66	36.66	39.05
P36	The capital of [X] is [Y]	1000	62.16	62.16	60.74
P361	[X] is part of [Y]	1000	23.61	31.44	15.34
P364	The original language of [X] is [Y]	1000	44.51	43.93	40.89
P37	The official language of [X] is [Y]	311	54.55	56.83	35.92
P39	[X] has the position of [Y]	1000	7.96	16.14	35.31
P407	[X] was written in [Y]	1000	59.18	65.22	64.99
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	26.67
P463	[X] is a member of [Y]	679	67.11	57.33	68.44
P47	[X] shares border with [Y]	1000	13.67	13.34	14.97
P495	[X] was created in [Y]	1000	16.5	32.23	34.1
P527	[X] consists of [Y]	1000	11.07	10.55	13.42
P530	[X] maintains diplomatic relations with [Y]	927	2.81	3.92	2.91
P740	[X] was founded in [Y]	1000	7.59	13.68	11.75
P937	[X] used to work in [Y]	1000	29.77	39.1	40.78

Description: 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. We also report the number of train instances for each T-REx relation.

Table 3.4: Examples of Manual and Trigger-Based Prompts for Fact Retrieval

Relation	Prompt	P@1
P101	[X] works in the field of [Y]	11.52
	[X]poon houses dwelling electrons teaches [Y]	15.01
	[X]girlfriend 1903 journal distinctions analytical[Y]	9.08
P103	The native language of [X] is [Y]	74.54
	[X] teaches languagesasperated mainly only [Y]	82.15
	[X]bourg arg820 (), bilingual[Y]	76.39
P106	[X] is a [Y] by profession	0.73
	[X]ph constituencylev future leading [Y]	20.1
	[X] conspiring actors assassinate the famed[Y]	16.32
P127	[X] is owned by [Y]	36.67
	[X] connects businesses northernmost rivals of [Y]	43.78
	[X] iter () subsidiary _____ of[Y]	47.82
P1303	[X] plays [Y]	18.91
	[X] redevelopmentvention flute backed electric [Y]	42.11
	[X] accomplished classical primarily bass the[Y]	40.94
P136	[X] plays [Y] music	0.7
	[X] puzzles saxophonist miniseries 1960s modern [Y]	54.6
	[X] accompanwright fusion soundtrack psychedelic[Y]	56.58
P1376	[X] is the capital of [Y]	81.11
	[X] municipality roughly flows into river [Y]	42.78
	[X] uninterruptedwhel northwestern town925[Y]	44.44
P178	[X] is developed by [Y]	62.76
	[X] suits sued console development opponent [Y]	53.06
	[X] emulator invented distur the conglomerate[Y]	63.61
P20	[X] died in [Y]	32.07
	[X] subsequently frequented locations quarters outside [Y]	35.62
	[X] a candles shop bourgeois[Y]	31.21
P27	[X] is [Y] citizen	0.0
	[X] lacks competed internationally representing [Y]	45.41
	[X] 1994 politicianregation socialism grips[Y]	40.5
P279	[X] is a subclass of [Y]	31.04
	[X] popularly— called puppy carrier [Y]	38.91
	[X]*. adjective remarkablyproducing[Y]	38.03
P364	The original language of [X] is [Y]	52.65
	[X]aan director translations lyrics in [Y]	50.26
	[X] translated phony commercials sediment original[Y]	38.36
P407	[X] was written in [Y]	60.21
	[X] develops folk language resembles standard [Y]	68.96
	[X],translated339 the simpler[Y]	55.66
P413	[X] plays in [Y] position	0.53
	[X]nks baseball starter 1992 attacking [Y]	41.81
	[X] outward offensive strikerrep talented[Y]	32.98

Examples of manual prompts (first line, shown with BERT’s P@1) and trigger-based prompts (second line is for BERT, third line is for RoBERTA) for Fact Retrieval.

Chapter 4

Relation Extraction

Apart from knowing whether the MLMs *know* facts, it is also important to evaluate whether they are able to *extract knowledge* from a given piece of text. In this section, we take the task of relation extraction—identifying how entities are related in a given sentence—because it is an important and well-studied task in information extraction. Formulating this task as prompts is similar to 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 tokens [`t`]...[`t`] are specific to the relation, and the correct object `obj` is the label token for [`p`] (see Table 3.2 for an example).

4.1 Setup

We use the T-Rex dataset (original split), for relation extraction as well, since each T-REx fact comes with context sentences that mention the subject and object surface forms. During generation of our prompt, we provide each sentence with an associated factual triple, and encourage the prompt to render the object as the label token. Along with comparison against

LAMA and LPAQA (the manual prompts should still be useful here), we also compare with a recent relation extraction model (Sorokin and Gurevych, 2017), which uses a combination of a LSTM-based relation encoder and an attention mechanism to perform sentence-level relation extraction, that was also used in Petroni et al. (2019). The model constructs a knowledge graph whose edges are essentially the extracted relation triples and is evaluated on how well it predicts relation types of entity pairs on the sentence level. We modify the evaluation for the relation extraction model by giving it 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 the other relations. We also drop all sentences from evaluation for which the named entity extractor did not identify the subject and the object as entities. Moreover, the relation extraction model is not trained to predict two of the original T-REx relations so for fair comparison, we exclude those two relations for the task of relation extraction. This evaluation benefits the relation extractor to make the evaluation fair. For evaluation of all systems, we treat predictions as correct if they are either the canonical version of the object (e.g., “USA”) or the rendered surface form (e.g., “American”) for *any* of the context sentences of the triple (also known as *macro* evaluation for relation extraction).

4.2 Results

In Table 4.1, we show the results of relation extraction for the RE baseline, LAMA and LPAQA prompts, and trigger-based prompts generated for BERT and RoBERTa. Our trigger-based prompts show that MLMs can extract relational information *more effectively* than the RE model, providing up to a 31% increase on the task. RoBERTa also outperforms RE, although it is worse than BERT. As previously mentioned in Section 3, the plausible reasons include insufficient search for triggers, missing Next Sentence Prediction (NSP) objective in RoBERTa, and the fact that BERT is pretrained on 2,500M words of Wikipedia.

Table 4.1: Relation Extraction Results

Model	Original	Augmented
RE	57.95	58.81
LAMA (BERT)	69.06	28.02
LPAQA (BERT)	76.55	30.79
BERT	88.1	39.35
RoBERTa	60.00	22.94

Description: Mean P@1 of models on relation extraction and the augmented version of it.

For both BERT and RoBERTa, we notice the trigger tokens consist of words related to their corresponding relations (see Table 4.2 for full list), e.g. RoBERTa selects “**defy trademarks of namesake manufacturer**” for relation MANUFACTURER/PRODUCER OF PRODUCT.

4.3 Augmented Training and Evaluation

Given the strong results for fact retrieval in Section 3, it is possible that BERT and RoBERTa are relying on *innate* knowledge to predict the objects instead of *extracting* them from the context. To separate this effect, we synthetically augment the relation extraction dataset by replacing each object in the test data by a random other object and making that corresponding change to the sentence, e.g. “*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. In these cases, parsing the context is crucial for identifying the relation. While performance of the RE model does not change significantly (Table 4.1), all of the MLMs are significantly impacted. This indicates that a large portion of their accuracy comes from background information, rather than proper relation extraction. Nevertheless, our trigger-based prompts for BERT outperform LAMA and LPAQA, further evidence that our method produces better probes for the extraction capabilities of masked language models.

On observing the extractions (listed in Table 4.2), we notice that even when LMs make a wrong prediction in this setting, they are still close to the correct answer. For example, when the true object “salad” of the relation `subclassOf` is swapped with “sandwich” in the context sentence “Mizeria is a Polish salad...”, BERT predicts “food”, which is technically correct, just not as precise as stated in the context.

Table 4.2: Examples of Triggers for Relational Extraction

Relation	Model	Context and Prompt	Model Prediction
P103 (native language)	BERT	Alexandra Lamy (born 14 October 1971) is a <u>French</u> actress. Alexandra Lamy ~~speaks~~ ~~airfield~~ ~~dripping %~~ of [MASK].	French
P36 (capital)	RoBERTa	Kirk was born in Clinton County, Ohio, and he entered service in <u>Wilmington</u> , 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 <u>actor</u> 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~~ <u>Seoul</u> , during a break of the first international tournament. The Immortal Game ~~located~~ ~~stered~~ ~~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~~ <u>Toyota</u> in the 1990s. Honda Civic del Sol ~~defy~~ ~~trademarks~~ of namesake manufacturer [MASK].	Toyota
P279 (subclass of)	BERT	Mizeria is a Polish ~~salads~~ <u>sandwich</u> consisting of thinly sliced or grated cucumbers, often with sour cream though in some cases oil. Mizeria ~~is~~ ~~calls~~ ~~dire~~ ~~cent~~ ~~altitude~~ [MASK].	food
P463 (member of)	RoBERTa	~~Rush~~ <u>Aerosmith</u> was a Canadian rock band consisting of Geddy Lee (bass, vocals, keyboards), Alex Lifeson (guitars), and Neil Peart (drums, percussion, lyricist). Alex Lifeson ~~affiliated~~ ~~al~~ ~~ach~~ ~~the~~ ~~internationally~~ ~~initials~~ [MASK].	Kiss

Description: Underlined words represent the gold object. The bottom half of the Table displays examples of our augmented evaluation of relation extraction where the original objects (represented by crossed-out words) are replaced by new objects.

Chapter 5

Discussion and Conclusion

In this paper, we develop automatically-constructed prompts that elicit knowledge from pretrained MLMs for a fact retrieval and relation extraction. We show that these prompts outperform manual prompts while requiring less human effort. Our approach is also more effective at probing for knowledge compared to automatic methods that attempt to improve the breadth and quality of manual prompts. This suggests that trigger-based prompts raise the lower bound estimate of knowledge contained in MLMs.

However, there are a few limitations of this work. To start off, our approach requires labeled training data for both tasks of fact retrieval and relation extraction. Although other probing techniques (e.g., linear probing) also require labeled training data, manual prompts do not. Manual prompts instead rely on human ingenuity and domain knowledge, which may more readily available in some cases. Further, as seen in Section 3, our approach suffers when the training data is unbalanced. While rebalancing training data helps, exploring other techniques to address this is an interesting area for future research.

The primary challenge of automatically identifying prompts is the inherent size of the search space. The space of all possible prompts is not only discrete and combinatorially massive,

but also not smooth due to contextual dependence (changing a single word piece can lead to a big impact on the performance) and the search is guided by the accumulated gradient on individual batches, which is quite noisy for discrete perturbations. Due to these reasons, the search is computationally quite expensive, and different runs often led to very different performance (even for the training loss). Many of these challenges may be addressed by using *continuous embeddings* as the prompts.

In general, different tools have different tasks and phenomena they are suitable for, and each only provides a *lower bound* of what these pretrained language models know about language and semantics. While our method helps make prompt-based probes more generally applicable, our method is just one tool in the toolbox of the interpretability researcher.

Bibliography

- Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. COMET: commonsense transformers for automatic knowledge graph construction. In *ACL*.
- Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. 2019. What does BERT look at? an analysis of BERT’s attention. In *ACL Blackbox NLP Workshop*.
- 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*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL*.
- Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. [Hotflip: White-box adversarial examples for text classification](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*, pages 31–36. Association for Computational Linguistics.
- Hady ElSahar, Pavlos Vougiouklis, Arslan 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*.
- Robert L. Logan IV, Nelson F. Liu, Matthew E. Peters, Matt Gardner, and Sameer Singh. 2019. [Barack’s wife hillary: Using knowledge graphs for fact-aware language modeling](#). In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 5962–5971. Association for Computational Linguistics.
- 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*.

- Nelson F Liu, Matt Gardner, Yonatan Belinkov, Matthew Peters, and Noah A Smith. 2019a. Linguistic knowledge and transferability of contextual representations. In *NAACL*.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Matthew E. Peters, Mark Neumann, Robert L. Logan IV, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. [Knowledge enhanced contextual word representations](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 43–54. Association for Computational Linguistics.
- 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*.
- Daniil Sorokin and Iryna Gurevych. 2017. Context-aware representations for knowledge base relation extraction. In *EMNLP*.
- Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R Bowman, Dipanjan Das, et al. 2019. What do you learn from context? probing for sentence structure in contextualized word representations. In *ICLR*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*.
- Jesse Vig and Yonatan Belinkov. 2019. Analyzing the structure of attention in a transformer language model. In *ACL Blackbox NLP Workshop*.
- Elena Voita and Ivan Titov. 2020. Information-theoretic probing with minimum description length. *arXiv preprint arXiv:2003.12298*.
- Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing NLP. In *EMNLP*.
- Sarah Wiegrefe 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, R’emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. HuggingFace’s Transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.