# BEAR: A Unified Framework for Evaluating Relational Knowledge in Causal and Masked Language Models

**Anonymous ACL submission**

## Abstract

Knowledge probing is used to assess to which degree a language model (LM) has successfully learned factual, relational knowledge during its pre-training. They are used as an inexpensive way to compare LMs of different sizes and different learning parameters. However, previous probes rely on the objective function used to pre-train an LM, and are thus applicable only to either masked or causal LMs. This renders a comparison across different types of LM impossible. To address this, we propose an approach that uses an LMs' inherent ability to estimate the log-likelihood of any given textual statement. We carefully design an evaluation dataset of 40,916 relation instances from which we produce alternative statements for each relational fact, one of which is correct. We then evaluate whether the LM correctly assigns the lowest log-likelihood to the correct statement. Our experimental evaluation of 13 common LMs shows that our proposed framework, BEAR, can effectively probe for knowledge across different LM types. We release BEAR as an open source framework to the research community to facilitate evaluation and development of LMs.

## 1 Introduction

Pre-trained language models (LMs) are the backbone of current state-of-the-art NLP approaches. A key property is the syntactic and semantic knowledge stored in their internal parameters, allowing them to generalize beyond given training data when fine-tuning for a specific downstream NLP task. Due to their importance, and the large number of proposed LMs, prior work has sought to better understand the factual knowledge in an LM, and to make it measurable in order to enable comparison of different LMs (Petroni et al., 2019; Poerner et al., 2020; Cao et al., 2021; Kalo and Fichtel, 2022).

The LAMA probe (Petroni et al., 2019) is the seminal work in studying commonsense and relational knowledge in LMs, and is widely used for inexpensively evaluating and comparing models (c.f. Youssef et al. (2023) and Cao et al. (2023) for an overview). Here, the main idea is to use relational knowledge from an existing knowledge base (KB), and create cloze-style statements for an LM to complete.

The capital of France is [MASK].

(a) LAMA probe: Single-subtoken mask prediction

The capital of Uganda is Thimphu.
... Kampala.
... Buenos Aires.
... Bandar Seri Begawan.

(b) BEAR probe: Rank answer options of arbitrary length

Figure 1: Comparison of the LAMA and BEAR probes. Both probes query languages models given a template (here in black), the subject of the relation (blue) and the object (orange). LAMA masks the object and predicts a single subtoken as answer. In BEAR, we create separate textual statements for a list of potential answers, select the statement with the lowest (pseudo) perplexity as judged by the LM. This allows us to include multi-token answers and evaluate both causal and masked LMs.

For instance, the entities "France" and "Paris" may be connected through the HAS-CAPITAL relation in a given KB, indicating that Paris is the capital of France. From this, LAMA constructs the sentence "The capital of France is [MASK]", and evaluates whether an LM predicts the right subtoken to complete this factual sentence. LAMA therefore effectively reuses the masked language modeling objective of the BERT-family of LMs (Devlin et al., 2019) to probe for knowledge. This example is shown in Figure 1a.

**Limitations of LAMA probing.** However, there are conceptual limitations to this approach: First, it requires the correct answer to be part of the subtoken vocabulary of the evaluated LM, restricting the space of relational knowledge that can be evalu-

ated. LAMA is thus limited to factual questions with single-subtoken answers (like "Paris" in Figure 1a) and cannot test for relational facts with long or rare answers (as shown in Figure 1b).

Second, and most importantly, its reliance on the masked language modeling objective means that LAMA is inapplicable for LMs trained with other objectives. It therefore excludes causal LMs such as the GPT-family of models (Radford et al., 2019). To the best of our knowledge, there currently exists no factual knowledge probe applicable to both masked and causal LMs.

**Limitations of LAMA data.** Additionally, various prior works have noted limitations of the relational data used in the LAMA probe. This includes (1) a heavily skewed answer space, favoring some answers over all others (Jiang et al., 2020b; Zhong et al., 2021; Cao et al., 2021), (2) overly revealing entity names (Poerner et al., 2020), (3) and issues involving knowledge with multiple correct answers, causing correct answers to be counted as errors (Kalo and Fichtel, 2022).

Taken together, we argue that the conceptual limitations of the probing approach and issues with the evaluation data impair the usefulness of LAMA to accurately measure and compare the factual knowledge of different LMs.

**Contributions.** To address these issues, we propose BEAR, a unified knowledge probe for both causal and masked LMs. Rather than casting the evaluation as a token prediction problem over the entire vocabulary of an LM, we instead present a set of answer options for each relation instance, create a textual statement for each option, and use the inherent ability of each LM to judge the log-likelihood of a statement to rank these options. See Figure 1b for an illustration.

We argue that this approach has numerous benefits in that it (1) allows us to evaluate both masked and causal LMs, (2) imposes no restrictions on the answer space, (3) allows us to design a new evaluation dataset that addresses a range of issues such as answer skews and multiple correct answers noted in prior work. In more detail, our contributions are:

1. We present an analysis of the weaknesses of the LAMA probe and follow-up works, to derive desiderata for the BEAR probe (see Section 2).

2. We propose to query knowledge as a multiple-choice selection problem in which the LM evaluates the perplexity of a given answer template with each choice filled in (see Section 3).

3. We construct a novel evaluation dataset that reflects the desiderata identified in our analysis (see Section 4).

4. We present an in-depth analysis in which we use BEAR to evaluate a range of common masked and causal LMs (see Section 5).

To enable the research community to use our probing method and dataset, we publicly release the entire evaluation framework under the name BEAR[1] as an open source package. It computes the BEAR score for any (causal or masked) LM in the HuggingFace TRANSFORMERS library (Wolf et al., 2020).

## 2 Analysis of Prior Work

We discuss technical details of the LAMA probe and analyze its weaknesses. For each weakness, we discuss solutions proposed in prior work.

### 2.1 LAMA

**Evaluation data.** The LAMA benchmark was originally composed of four separate datasets named after their respective sources: SQuAD (Rajpurkar et al., 2016), GoogleRE[2], ConceptNET (Speer and Havasi, 2012) and T-REx (Elsahar et al., 2018). However, subsequent research for the most part concentrated exclusively on T-REx. Its knowledge base comprises a selection of 41 *relations* derived from Wikidata. Each relation contains at most 1,000 *relation instances* in the form of subject-relation-object triples. A relational triple is represented as: $\langle s, r, o \rangle$, where $s$ is a subject (e.g., "France"), $r$ is a relation (e.g., HAS-CAPITAL), and $o$ is an object (e.g. "Paris").

There are three types of relations in LAMA: 1-1 (one-to-one, e.g. HAS-CAPITAL), N-1 (many-to-one, e.g., HAS-LANGUAGE) and N-M (many-to-many e.g. SHARES-BORDER-WITH). Relations of N-1 type allow for multiple subjects to relate to one object, while the latter permits multiple subjects to be associated with multiple objects.

**Relation identifiers.** All relations are linked to a corresponding relation in Wikidata, and thus have

[1] **B**enchmark for **E**valuating **A**ssociative **R**easoning), to be released under a *CC BY-SA* license upon acceptance.

[2] https://code.google.com/archive/p/relation-extraction-corpus/

unique IDs. For instance, the CAPITAL-OF relation in LAMA corresponds to Wikidata relation P36 (see Table 1 for more examples). This facilitates comparison across different datasets, since all follow-up works to LAMA, including BEAR, also derive their relations from Wikidata.

**Templates.** Each relation in LAMA has a textual template with placeholders for subject and object. For CAPITAL-OF, the template is "The capital of [X] is [Y].", where [X] is a placeholder for the subject, while [Y] is the placeholder for the object. At test time, the subject of a given relation is filled in the template, while the object is replaced by a [MASK]-token. This results in a masked sentence (e.g. "The capital of France is [MASK].") for which the LM is tasked to predict the masked token.

### 2.2 Issue 1: Single Subtoken Answers

As noted by Petroni et al. (2019), LAMA is restricted to single-subtoken answers for factual knowledge queries. This causes issues as LMs split most words into multiple subtokens, and most LMs differ in how they perform the splits. To illustrate, consider how the country name "Togo" is tokenized by different versions of BERT: the `bert-base-cased` model splits the word into two subtokens ([`To`, `##go`]), whereas the `bert-base-uncased` variant preserves it as a single subtoken ([`togo`]).

An analysis of 193 UN member country names is a good example of how such a restrictive condition affects the size of a hypothetical dataset. When restricting answer space to single tokens, 32% and 27% of available country names would have to be discarded for cased and uncased version of BERT respectively. Worse, the RoBERTa model (Liu et al., 2019) that uses a BPE-based tokenizer would split 88% of all country names. Refer to Table 1 for a list of how many LAMA relations need to be discarded when evaluating XLM-RoBERTa (Conneau et al., 2020) and RoBERTa models.

**Comparison of different LMs.** Because the tokenizer that is bundled with each model differs, the comparison of various LMs becomes impossible unless the models tokenize the answers in the same way. To address this, practitioners currently revert to using the intersection of single-token vocabularies derived from all LMs being compared. However this in practice further limits the scope of relational knowledge that can be included in the evaluation.

**Prior work.** Various prior works address the issue of predicting multi-subtoken words for single [MASK] tokens (Ghazvininejad et al., 2019; Jiang et al., 2020a; Kalinsky et al., 2023; Shen et al., 2020). Jiang et al. (2020a) provided a selection of algorithms to tackle predicting multi-token entities. However, they require a specification of further parameters such as the number of subtokens to generate. Kalinsky et al. (2023) proposed generation approaches that either require additional training or the use of an external network, making them inapplicable to the purpose of evaluating knowledge contained in pretrained weights through a zero-shot approach.

| ID | Relation | `xlm-roberta-base` | `roberta-base` |
|---|---|---|---|
| P30 | ON-CONTINENT | 74.46 % | 80.21% |
| P31 | INSTANCE-OF | 28.85% | 67.35% |
| P36 | HAS-CAPITAL | 45.80% | 89.76% |
| P37 | HAS-LANGUAGE | 30.85% | 45.13% |
| ... | ... | ... | ... |
| P1303 | INSTRUMENT | 58.69% | 100.00% |
| P1376 | CAPITAL-OF | 32.05% | 81.62% |
| | **Mean** | 31.73% | 62.86% |

Table 1: Ratio of discarded instances due to multi-token answers in XLM-RoBERTa and RoBERTa.

### 2.3 Issue 2: Multiple Correct Answers

LAMA expects exactly one correct answer to each knowledge query, and rates other factually correct answers as errors. To illustrate this, consider the query "Germany shares a border with [MASK]", to which LAMA expects the answer "Switzerland". All other correct answers, such as "Poland" are rated as incorrect. This issue affects all N-M relations in LAMA.

**Prior work.** KAMEL (Kalo and Fichtel, 2022) address this by allowing the LM to generate an arbi-

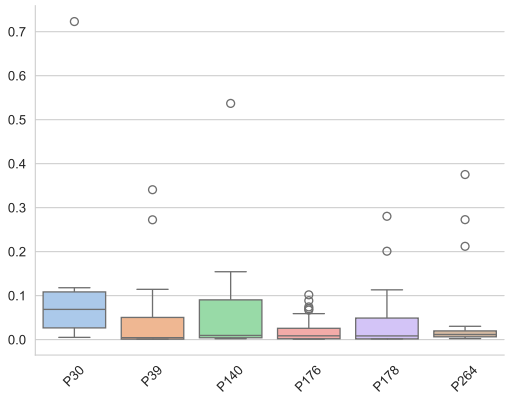



Figure 2: The normalized answer frequency of selected relations in LAMA probe. The outliers are marked with dots. In some relations a majority class accounts for more than 50% of all instances.

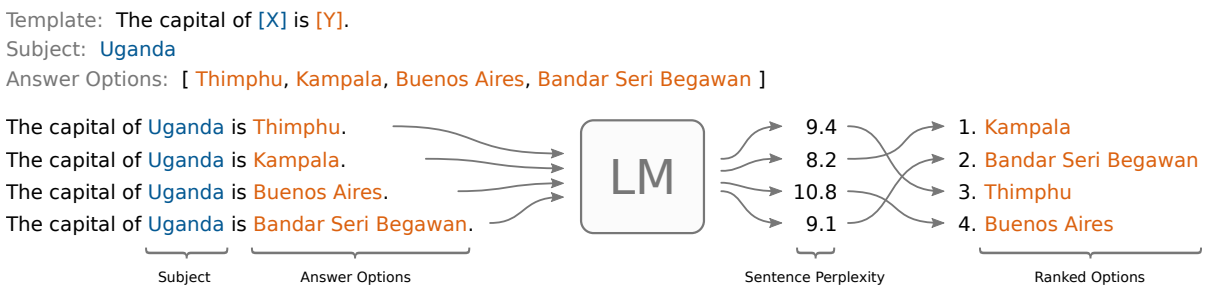



Figure 3: For each answer option, a sentence is passed to the LM (here using the template: "The capital of [X] is [Y]." and the subject "Uganda"). The perplexity scores assigned by the LM are then used to rank the answer options.

trary number of answers using a template instructed via few-shot prompting, experimenting with ranges of 1-10 answers per instance. They then evaluate the predictions using standard measures of precision and recall. However, their approach relies on the text generation ability of causal LMs and thus cannot be applied to masked LMs.

### 2.4 Issue 3: Imbalanced Answer Distribution

The relations in T-REx have a highly unbalanced answer distribution (except the 1-1 relations) and in certain relationships, over half of the instances belong to the predominant class (see Figure 2). This was noted by Zhong et al. (2021), who observed that a model that always chooses the majority class would outperform some state-of-the-art LMs.

To illustrate, consider the T-REx's ON-CONTINENT relation, which connects a location to the continent in which it is situated. Counter-intuitively, the majority class in this relation is "Antarctica", accounting for 72% of all instances.

**Prior work.** To account for this imbalance, Cao et al. (2021) created a balanced version of the LAMA probe called WikiUNI. It contains the same relations as T-REx but has a uniform answer distribution, and was constructed to have the same number of subjects for every object. However, their dataset samples an highly skewed number of instances per relations, with 7 relations (out of 41) accounting for over 50% of all instances.

### 2.5 Issue 4: Rare Wikidata Entries

The above-mentioned example of "Antarctica" accounting for the objects of 72% of all instances in the ON-CONTINENT relation also points to another problem: An artifact of randomly sampling Wikidata for relation instances is that rare Wikidata entries are overrepresented. For instance, ON-CONTINENT has a large number of small islands as subjects (e.g. "Umber Island" and "Brooklyn Island", both close to the Antarctic continent), many of which are unlikely to occur in a corpus outside of an encyclopedia like Wikipedia. We believe this dataset bias gives an unfair advantage to LMs trained using Wikipedia. However, to the best of our knowledge no prior work addresses this issue.

### 2.6 Issue 5: Evaluation of Causal LMs

**Prior work.** Since LAMA is inapplicable to causal LMs, Kalo and Fichtel (2022) proposed the KAMEL probe. Factual knowledge is probed by virtue of question statements for which the response is auto-regressively generated using the causal LM. To guide the generation approach, they prepend $k$ few-shot examples into the prompt that present how the correctly formatted answer should look like. However, since this approach relies on the language modeling objective of causal LMs, KAMEL is not applicable to masked LMs.

## 3 BEAR Probe

We base our evaluation on using an LMs inherent ability to estimate the log-likelihood of a given sentence. Our main idea is to restrict the space of possible objects for each relation instance, and create for each relation a set of options which are ranked by their log-likelihood values.

### 3.1 Ranking Options using Log-Likelihood

Our approach requires a dataset of $\langle s, r, o \rangle$ relation instances, where for each relation $r$ there exists (at least) one template $t$ and a set of answer options $a_i$ with $i \in \{1, ..., k\}$ that includes the correct answer.

**Creating options to rank.** For each relation instance, we create $k$ natural language statements using the template, by using the relations subject $s$ and each of the possible relation's objects $a_i$ as parts of a textual statement.

Figure 3 illustrates this process for the example relation instance $\langle$"Uganda", CAPITAL-OF, "Kampala"$\rangle$ and the template "The capital of [Y] is [X]". The set of potential answers in this example

is ["Kampala", "Thimphu", "Buenos Aires", "Bandar Seri Begawan"]. For each potential answer, we create a separate statement.

**Predicting log-likelihood.** For each generated statement, we predict the log-likelihood score $\log \hat{p}(a|t)$. As the template is the same for each of the answer options and we are only interested in ranking them, it is sufficient to compute the log-likelihood for the entire sentence:

$$\begin{aligned} \log \hat{p}(a_i|t) &= \log \hat{p}(a_i, t) - \log \hat{p}(t) \\ &\sim \log \hat{p}(a_i, t) \end{aligned} \quad (1)$$

Since causal LMs are trained to predict a log-likelihood of each token given the previous context, the log-likelihood of the sentence is simply the sum over log-likelihoods of each token.

**Log-likelihood in masked LMs.** In an LM trained using the masked language modeling objective a sentence-level log-likelihood is not clearly defined. However, Salazar et al. (2020) and Kauf and Ivanova (2023) offer two variants of how to retrieve a pseudo log-likelihood score for a given text. Both approaches use multiple forward passes. Salazar et al. (2020) simply mask each token once while keeping the remaining context unmasked. Kauf and Ivanova (2023) improve on this by additionally masking all tokens right to the current token which belong to the same word. This fixes the issue of assigning disproportionate likelihoods to multi-token words. We use the latter in our approach.

**Ranking the results.** Finally, the statements are ranked by their log-likelihood scores. This is illustrated in Figure 3 (right hand side).

### 3.2 Evaluation Metric

To evaluate the amount of knowledge encoded in each model, we employ the same evaluation measure as Petroni et al. (2019). Specifically, we use the mean precision@k (P@k): for any given instance, the value of the P@k metric is 1 if the template with the ground truth is ranked among the top $k$ results and 0 otherwise. More formally, if the model's top $k$ answer ranks can be represented by a set $\mathcal{A}$, then precision@k can be computed using Formula 2.

$$P@k = \begin{cases} 1, & \text{if the ground truth } \in \mathcal{A} \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

The BEAR score is the average precision@1 of all relation instances in our evaluation data.

## 4 BEAR Dataset

Our proposed probing approach requires a dataset with a restricted answer space. Following the analysis in Section 2 we additionally desire (1) the answer space to be balanced, (2) a single correct answer per relation instance, (3) a balanced number of instances per relation, and (4) a focus on knowledge that could reasonably be found in corpora other than Wikipedia.

### 4.1 Selecting Relations

We use the 234 relations of KAMEL as a starting point and manually remove two thirds of these. This curation process was conducted independently by two researchers (authors of this paper), and disagreeing judgements discussed in detail to reach a final decision. The most common reasons for excluding a relation were:

- A relation (after filtering) had too few objects with a desired number of instances (i.e. given the statistics of the instances it was not possible to build a balanced answer space within our constraints).
- A relation is not time-invariant, such as the RESIDENCE relation (connecting a person to their place of current residence). Since such relations have a high potential to change over time, their inclusion would give unfair advantage to LMs trained over data from the same time period as the evaluation data.
- A relation with many instances in which objects were incomplete, since this may cause correct answers to be counted as errors. For instance, the MADE-FROM-MATERIAL relation, that connects an object and the material it is made of, often contained only a few primary components as objects.
- The relation has an overly diverse subject or object space, making the semantics of the relation overly broad and impairing our ability to design meaningful templates. For instance, the relation COUNTRY connects various entity types such as events, ships, roles, websites, TLDs, codes of standards, and many more categories, to a country.

As a result of this process, we selected 78 relations for inclusion in BEAR.

### 4.2 Selecting Relation Instances

For the selected relations, we retrieve relation instances from Wikidata[3] and employ a number of filtering steps to distill a dataset of relation instances that meet our desiderata.

**Filtering subjects and objects.** We first filter down the space of eligible subjects and objects. We remove all Wikidata entities (i.e. subjects and objects) that do not have an English label. Following prior work (Poerner et al., 2020), we additionally remove all subjects with overly revealing entity names. For example, predicting the name of the company that produced the "Apple Watch" is straightforward since the correct answer (e.g. "Apple") is part of the subject (e.g. "Apple Watch"). The similarity is computed via the overlap in tokens and fuzzy string matching (Bachmann, 2023).

**Ensuring a coherent answer space.** Even in our curated set of relations, some relation instances connected to outlier object types. For instance, the *head of government* relation, which typically connects a country to a specific named person (e.g. "Joe Biden"), would in some cases connect to a job title instead (e.g. "president").

To increase coherence, and ensure that our templates are meaningful, we utilized GPT4 (OpenAI, 2023) to flag answers which stand out (see Figure 12 in Appendix B for the template that was used) and decided on a case-by-case basis whether to accept these changes. This process also helped us check the relations for potential issues.

**Sampling a balanced dataset.** For this initial set of entities, we sampled the remaining relation instances such that (1) each relation has a uniform distribution of objects in the answer space, (2) each relation has approximately the same number of instances overall, and (3) no entity occurs across multiple relations. During sampling, we give preference to Wikidata entities with Wikipedia pages in multiple languages, to focus on well-known entities that might reasonably be found in corpora outside of Wikipedia.

This process yields a total of 40,916 instances for our 78 relations in our final dataset.

### 4.3 Templates

We create three templates for each relation, to better safeguard against template-specific biases.

[3]We use the JSON dump of Wikidata of January 3rd 2022 (Wikidata contributors, 2022) which is available as a torrent under a *CC BY-SA* license.

| Dataset | LAMA | KAMEL | BEAR |
|---|---|---|---|
| Number of Instances | 31,479 | 46,800 | 40,916 |
| Number of Relations | 41 | 234 | 78 |
| Literals | no | yes | no |
| 1:1 Relations | 0 | | 14 |
| N:1 Relations | 7 | | 64 |
| N:M Relations | 34 | | 0 |
| N:M Instances | 1,035 | 4,296 | 0 |
| Avg. Instances per Relation | 830.2 | 1,400[4] | 596.9 |

Table 2: Descriptive dataset statistics: BEAR compared to LAMA (T-REx subset) and KAMEL (figures for KAMEL and LAMA from Kalo and Fichtel, 2022). Avg. Instances per Relation only includes relations with more than one instance per answer.

We source the initial templates from the existing LAMA dataset, utilize GPT4 to create additional ones (the used prompt can be found in Figure 13 in Appendix B), and manually select those that best match our subjects and answer spaces. Finally, we query GPT4 with each of the templates applied to 5 subject-object pairs from the relation to check for linguistic correctness (the used prompt can be found in Figure 14 in Appendix B).

### 4.4 Resulting Dataset

The final dataset consists of 78 relations and 40,916 items. The majority of these relations are 1:N, each with a restricted answer space of between 5 and 100 possible answers (mean of 59.7). The answer space is also balanced such that each answer appears the same number of times across all instances, with between 6 and 120 instances per answer (mean of 16.0). The dataset also contains 14 1:1-relations. Here, there is only one instance per answer.

For a detailed comparison of these statistics to LAMA and KAMEL, see Table 2.

## 5 Experiments

We present an experimental evaluation in which we use BEAR to score a selection of LMs, compare the results to earlier probes, and discuss the results.

**Compared LMs.** We compare a total of 13 LMs, as listed in Table 3: This includes 6 masked LMs from the BERT, RoBERTa, XLM-RoBERTa families, each in their `base` and `large` variants. And 7 causal LMs from the GPT and OPT families, the latter in 5 different sizes to evaluate how the BEAR score correlates to larger model sizes.

**BEAR score.** We compute the BEAR score for each of the three template options per relation in-

[4]1,000 train samples, and 200 each for dev & test

| Model | Type | # params | BEAR | $BEAR_{1:1}$ | $BEAR_{1:N}$ |
|---|---|---|---|---|---|
| opt-6.7b | CLM | 6.7b | 23.2% | 33.2% | 22.5% |
| opt-2.7b | CLM | 2.7b | 19.5% | 28.3% | 18.9% |
| opt-1.3b | CLM | 1.3b | 16.0% ± 0.4% | 23.3% ± 1.0% | 15.5% ± 0.4% |
| roberta-large | MLM | 355M | 11.1% ± 0.4% | 17.1% ± 0.8% | 10.7% ± 0.4% |
| bert-large-cased | MLM | 335M | 10.1% ± 0.3% | 11.8% ± 0.7% | 10.0% ± 0.3% |
| bert-base-cased | MLM | 109M | 9.6% ± 0.3% | 11.5% ± 1.2% | 9.4% ± 0.3% |
| opt-350m | CLM | 350M | 9.5% ± 0.2% | 13.4% ± 0.8% | 9.2% ± 0.2% |
| gpt2-medium | CLM | 355M | 9.1% ± 0.3% | 11.3% ± 1.9% | 8.9% ± 0.2% |
| roberta-base | MLM | 125M | 8.4% ± 0.3% | 11.8% ± 1.8% | 8.1% ± 0.4% |
| opt-125m | CLM | 125M | 8.0% ± 0.2% | 9.5% ± 0.8% | 7.9% ± 0.2% |
| xlm-roberta-large | MLM | 561M | 7.7% | 14.2% | 7.3% |
| gpt2 | CLM | 137M | 6.4% ± 0.3% | 5.8% ± 1.6% | 6.5% ± 0.2% |
| xlm-roberta-base | MLM | 279M | 5.8% ± 0.2% | 9.0% ± 1.5% | 5.5% ± 0.1% |

Table 3: Models investigated in this work sorted by their BEAR score (Devlin et al., 2019; Liu et al., 2019; Radford et al., 2019; Zhang et al., 2022), aggregated over all relations as the weighted average and as the mean over all templates (with the standard error; `xlm-roberta-large`, `opt-2.7` & `opt-6.7b` we only evaluate using the first template).

dividually, and report the average across templates as well as the standard deviation.

### 5.1 Main Results

Table 3 lists the results for all LMs in consideration. We present the overall BEAR score, and also present the scores for the subsets of 1:1 and N:1 relations only. We find that scores are generally low for all models, highlighting the challenging nature of our benchmark, as it queries for factual information with strong detractors. In addition, we make a number of observations:

**BEAR scores are higher for larger LMs.** In line with our expectations, we find that larger models consistently outperform their smaller counterparts. For a better illustration, we present a plot of accuracy against model size in Figure 4. This trend of steady accuracy improvement with increasing model size is evident across all tested model families. Interestingly, the smallest change is observed among BERT models, where the performance of `bert-base-cased` and `bert-large-cased` across all of the relation is roughly on par.

**Better BEAR scores for masked LMs.** When comparing models by their parameter count, we note a slight advantage of masked over causal LMs. This may indicate that the masked language modeling objective, encouraging deep bidirectionality, is more effective in capturing factual knowledge.

**Impact of multilingual training data.** We note that the two XLM-RoBERTa models are among the lowest-scoring models in the benchmark. We hypothesize that this diminished performance of the XLM models may stem from its pre-training on multilingual corpora, and a focus of BEAR on English-language entities.

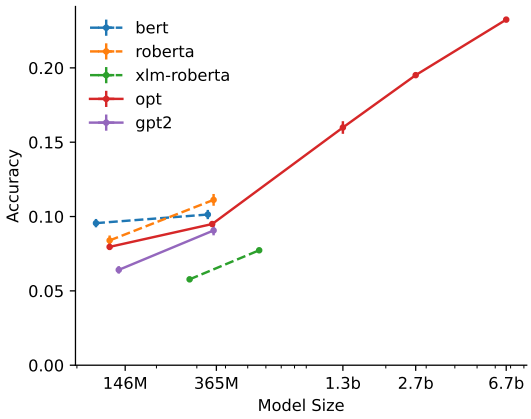



Figure 4: Probing scores of different models on BEAR. Model size is represented on a log scale.

**Impact of templates.** We further evaluate the impact of template choice on the BEAR score. A full analysis over all relations is provided in Figure A in the Appendix.

We find that in line with the observation of Elazar et al. (2021), LMs are sensitive to the manner in which they are queried. For instance, for the HAS-CAPITAL relation, `bert-base-cased` drops approximately 80% in accuracy when using "[Y] has its governmental seat in [X]" instead of "The capital of [X] is [Y]." Such difference could be attributed to BERT's primarily being trained on Wikipedia, leading to its limited exposure to diverse writing styles. On the other hand, we find that `opt-1.3b` shows more even accuracy scores across all templates.

#### 5.1.1 Ablations

We conduct several ablations to evaluate our design choices in BEAR.

**Sum vs. Mean of the Log Probability.** We in-

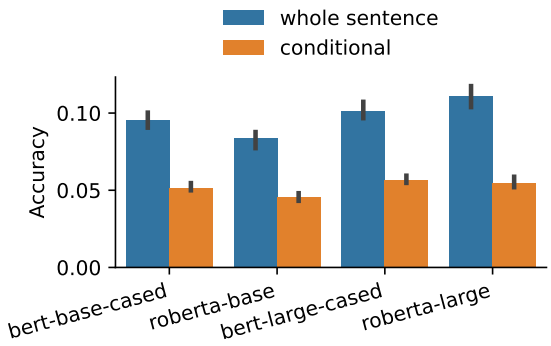



Figure 5: Aggregated accuracy (measured on BEAR) when using the sum over all tokens in the complete statement vs. answer-tokens only

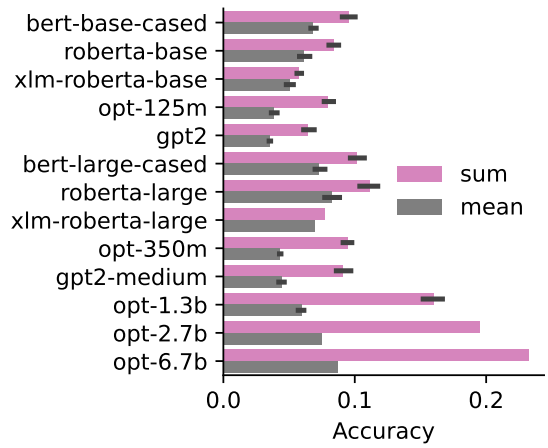



Figure 6: Aggregated accuracy of different retrieval variants on BEAR. The error bars indicate the standard error over three evaluations using the different templates (on `xlm-robert-large`, `opt-2.7` & `opt-6.7` we only used a single template).

vestigate how performance varies when sentence scoring is achieved using both sum and mean reduction methods. The results are illustrated in Figure 6. We observed that employing a perplexity score of a sentence, normalized by its token count, tends to yield inferior performance for the probe. Figure 6 illustrates the results of this ablation study. Moving forward, we suggest summing the perplexity score over a sentence for both masked and casual language models in future experiments.

**Conditional Scores.** To compute the pseudo perplexity for statement in an MLM, one forward pass per token is required. Masking only the tokens that are part of the answer to be ranked, would significantly reduce the required computation. However, our experiments (see Figure 5) indicate there is a


Figure 7: Aggregated accuracy of different retrieval variants on BEAR. The error bars indicate the standard error over three evaluations using the different templates.

significant[5] performance drop when using conditional score.

**Pseudo Log Likelihood Metric.** While in preliminary experiments on LAMA, we observed a higher benefit from using the `within_word_l2r` variant. It has only a slightly higher mean scores than `original` (see Figure 7). This difference is not significant when using the `sum` variant (p-value of 0.52 on a Student's t-test for paired samples). However, the difference is large when using the `mean` variant (and significant with p-value of 0.025)

## 6 Conclusion

We presented BEAR, a relational knowledge probe applicable to both causal and masked LMs. Since our proposed approach imposes no restrictions on the evaluation data, we created a large evaluation dataset that addresses issues of answer skews, domain and template bias and the correctness of facts identified by ourselves and prior work. We publicly release BEAR for use by the research community.

## Limitations and Risks

The knowledge probe we present in this paper follows the approach of earlier probes and as such tests only for factual, relational knowledge. This includes classic relationship types such as the place of birth of persons, their time of birth, the genre of works of art, etc. However, there are other types of more general commonsense knowledge that one might be interested in testing a model for, such as physical reasoning general properties of concepts. Our probe does not test for such kinds of knowledge.

[5] P-value of $2.8 \times 10^{-10}$; using a Student's t-test for paired samples

Further, even though we devised heuristics to ensure that entities in BEAR are common enough to appear on Wikipedia pages of many different languages, there remains a likely bias towards entities overrepresented on Wikipedia, giving advantage to LMs trained on Wikipedia rather than more general corpora.

We see few risks in the BEAR probe itself, but caution that knowledge probing is often used to assist in research and development of LMs. As such, BEAR may contribute to the development of LMs that malevolent actors might misuse.

## References


Max Bachmann. 2023. RapidFuzz. Original-date: 2020-02-29T14:41:44Z.

Boxi Cao, Hongyu Lin, Xianpei Han, and Le Sun. 2023. The Life Cycle of Knowledge in Big Language Models: A Survey. ArXiv:2303.07616 [cs].

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

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. 2021. Measuring and Improving Consistency in Pretrained Language Models. *Transactions of the Association for Computational Linguistics*, 9:1012–1031.

Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. 2018. T-REx: A Large Scale Alignment of Natural Language with Knowledge Base Triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. Mask-Predict: Parallel Decoding of Conditional Masked Language Models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6112–6121, Hong Kong, China. Association for Computational Linguistics.

Zhengbao Jiang, Antonios Anastasopoulos, Jun Araki, Haibo Ding, and Graham Neubig. 2020a. X-FACTR: Multilingual Factual Knowledge Retrieval from Pretrained Language Models. *arXiv:2010.06189 [cs]*. ArXiv: 2010.06189.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020b. How Can We Know What Language Models Know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Oren Kalinsky, Guy Kushilevitz, Alexander Libov, and Yoav Goldberg. 2023. Simple and Effective Multi-Token Completion from Masked Language Models. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 2356–2369, Dubrovnik, Croatia. Association for Computational Linguistics.

Jan-Christoph Kalo and Leandra Fichtel. 2022. KAMEL : Knowledge Analysis with Multitoken Entities in Language Models. In *Automated Knowledge Base Construction*.

Carina Kauf and Anna Ivanova. 2023. A Better Way to Do Masked Language Model Scoring. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 925–935, Toronto, Canada. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. ArXiv:1907.11692 [cs].

OpenAI. 2023. GPT-4 Technical Report. ArXiv:2303.08774 [cs].

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

Nina Poerner, Ulli Waltinger, and Hinrich Schütze. 2020. E-BERT: Efficient-Yet-Effective Entity Embeddings for BERT. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 803–818, Online. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. Masked Language Model Scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2699–2712, Online. Association for Computational Linguistics.

Tianxiao Shen, Victor Quach, Regina Barzilay, and Tommi Jaakkola. 2020. Blank Language Models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5186–5198, Online. Association for Computational Linguistics.

Robyn Speer and Catherine Havasi. 2012. Representing General Relational Knowledge in ConceptNet 5. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 3679–3686, Istanbul, Turkey. European Language Resources Association (ELRA).

Wikidata contributors. 2022. Dump of Wikidata of January 3rd 2022.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Paul Youssef, Osman Alperen Koraş, Meijie Li, Jörg Schlötterer, and Christin Seifert. 2023. Give Me the Facts! A Survey on Factual Knowledge Probing in Pre-trained Language Models. ArXiv:2310.16570 [cs].

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. OPT: Open Pre-trained Transformer Language Models. ArXiv:2205.01068 [cs].

Zexuan Zhong, Dan Friedman, and Danqi Chen. 2021. Factual Probing Is [MASK]: Learning vs. Learning to Recall. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5017–5033, Online. Association for Computational Linguistics.

# A Further Results

## A.1 Comparison with the LAMA probe

In order to compare BEAR and LAMA probes, we decided to only consider a common subset of relations in this comparison. The results demonstrate that BEAR is a more challenging probe compared to T-REx. When evaluating the same subset of relations, models consistently achieve lower scores on BEAR as compared to LAMA. This suggests that BEAR presents a more rigorous test of a model's knowledge. Due to the even distribution of answers and the absence of informative entity names, a model loses any benefits gained from biases in answer frequency or recognizable names, forcing its reliance purely on the knowledge encoded within its parameters. This is evident in Figure 8. If the hypothesis we proposed in Section 2.5 is accurate, then models pre-trained on Wikipedia (like BERT) will have an advantage over those not trained on Wikipedia (such as GPT2) due to a potential train/test data overlap. Consequently, in a probe that fails to address the potential issues arising from random sampling in Wikidata, certain models are anticipated to show an improved performance. The results of our analysis show that the performance disparity across models evaluated on BEAR's relations is less pronounced than it is across the same subset of T-REx's relations. For a detailed comparison of performance on a per-relation basis, refer to Figure 11 in the Appendix. For example, `bert-base-cased` achieves a very high performance on T-REx's MANUFACTURER relation[6], however on corresponding subset in BEAR has a significantly low score.

# B Prompts Used

All prompts were passed as 'system messages'.

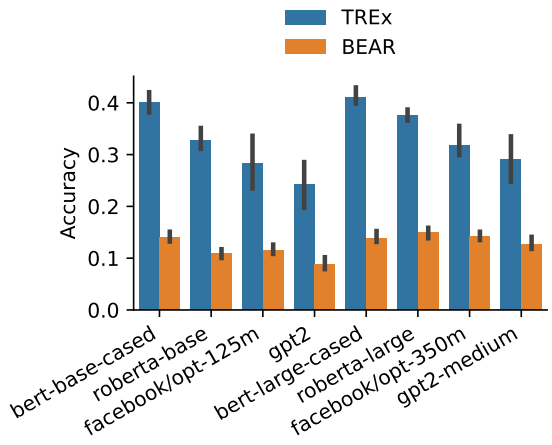
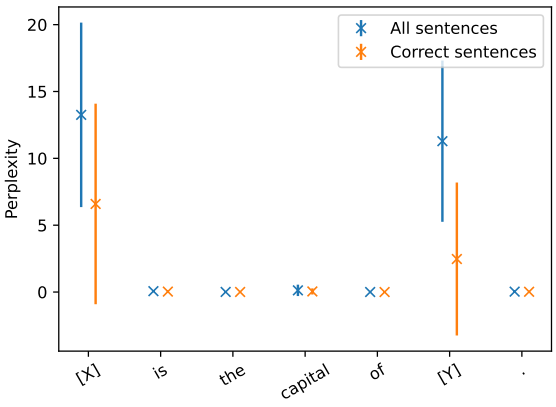
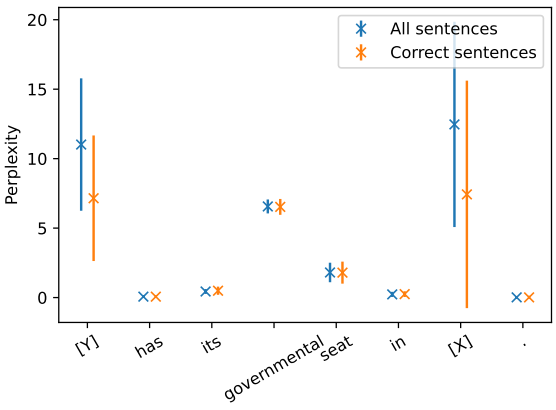



Figure 8: Comparative analysis of model performance on identical subsets of relations and templates in T-REx and BEAR datasets




(a) First Template: "[X] is the capital of [Y]."; Accuracy of 63%




(b) Second Template: "[Y] has its governmental seat in [X]"; Accuracy of 13%

Figure 9: $\text{BERT}_{\text{base}}$ (cased) on P1376 (BEAR)

[6]Relation ID: P176

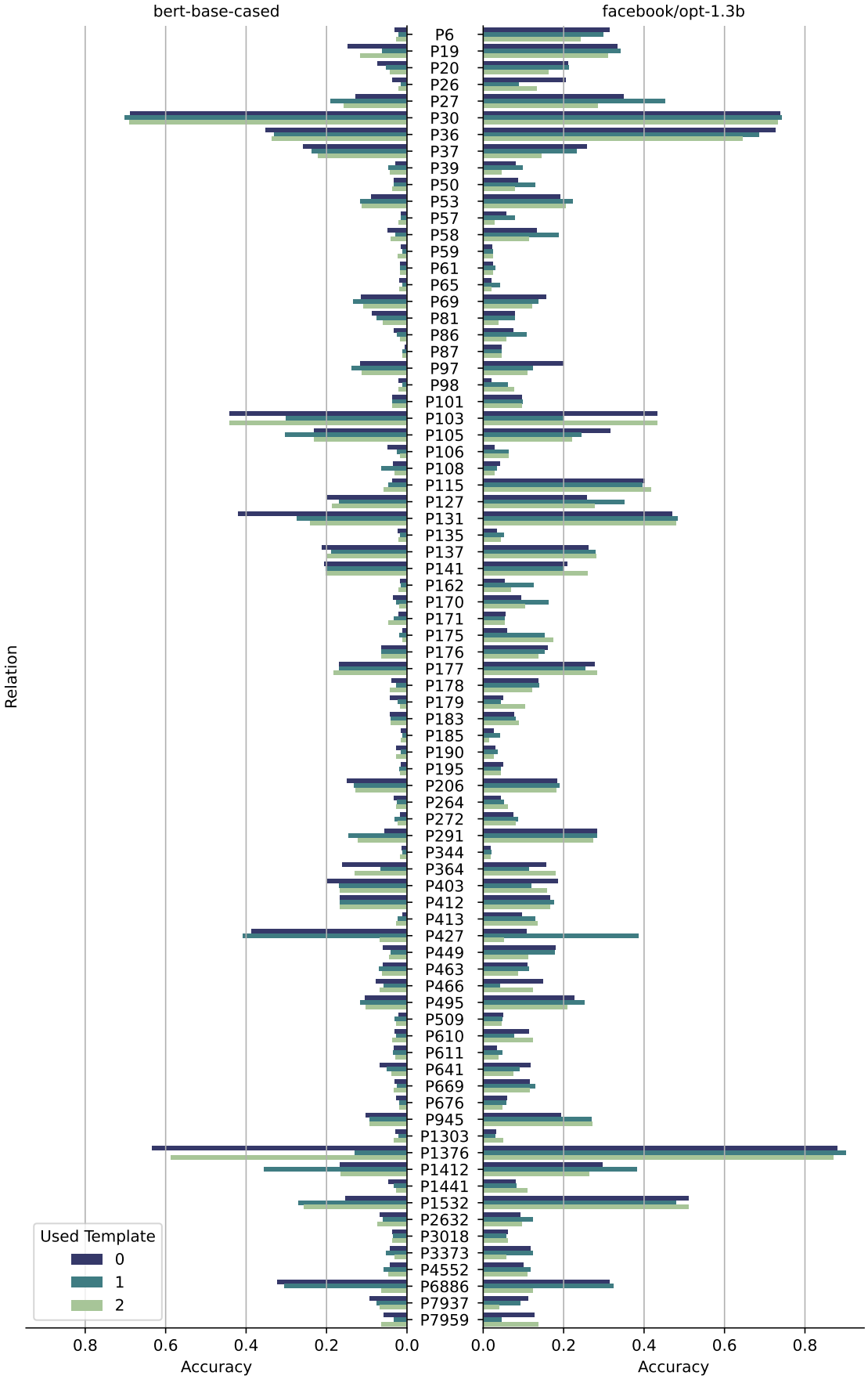



Figure 10: Accuracy of two models on each of the BEAR relations.

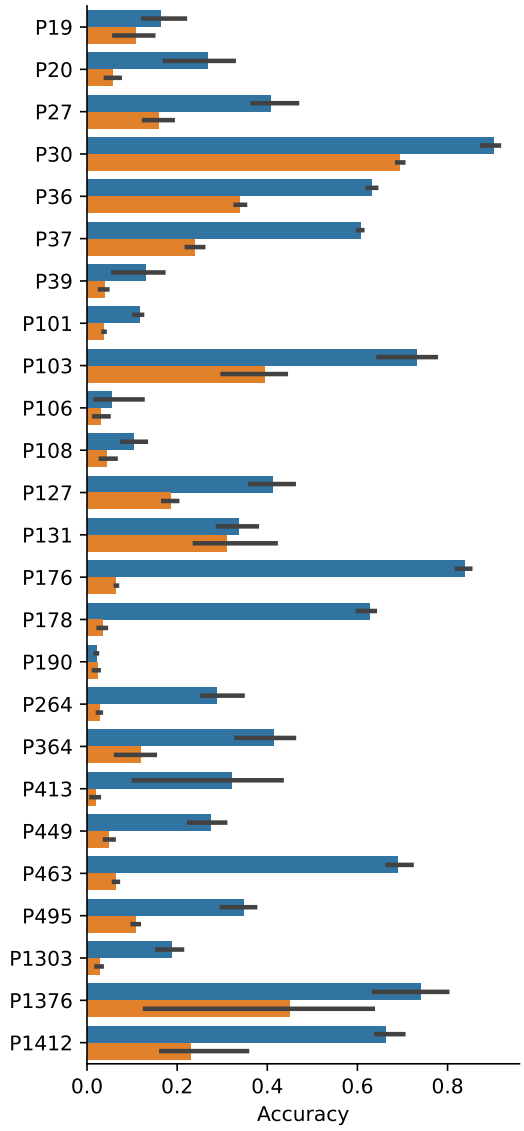



Figure 11: Performance of `bert-base-cased` on a per-relation basis for both BEAR and T-REx probes. The results were obtained by summing pseudo perplexity scores (`within_word_l2r`)

```
You are a researcher assistant tasked to design an evaluation dataset to test relational knowledge contained in language models. Specifically, you are given a label for a relation, its description, and a list of possible answers. Your assignment is to identify words that do not align with the majority category in a given list of answers given the relation label and its description. Return your response as a Python tuple. The first element should be a list containing the words that don't fit the majority category, and the second element should be a string representing the category of the majority of answers. If all words fit the category, return an empty list. Example format: (['Berlin', 'Warsaw'], 'countries').
```

Figure 12: Prompt used to flag words in the answer space of each relation. In addition to some relation metadata (label and description) the (intermediate) answer space was passed on the model.

```
As a research assistant, your task is to create an evaluation dataset to assess the relational knowledge of language models. You are provided with a specific relation label, its definition, and examples of subjects and objects related to it. Your objective is to craft three semantically similar cloze sentence templates that embody this relation. Use '[X]' as a placeholder for the subject and '[Y]' for the object (answer). Ensure that these sentence templates are straightforward and devoid of superfluous elements. For instance, given 'label': 'educated at', 'description': 'educational institution attended by subject', 'subjects': ['Einstein', 'Feynman'], 'objects': ['Princeton University', 'University of Zurich'], your templates might be: ['[X] was educated at [Y].', '[X] studied at [Y].', '[X] was a student at [Y].']. Present your response as a Python list.
```

Figure 13: Prompt used to generate template variants. In addition to the relations metatadata (label and description), 6 subject-object pairs were passed as examples for each relation.

```
As a research assistant, your task is to create an evaluation dataset to assess the relational knowledge of language models. You are provided with a specific relation label, its definition, and examples of subjects and objects related to it. Your objective is to craft three semantically similar cloze sentence templates that embody this relation. Use '[X]' as a placeholder for the subject and '[Y]' for the object (answer). Ensure that these sentence templates are straightforward and devoid of superfluous elements. For instance, given 'label': 'educated at', 'description': 'educational institution attended by subject', 'subjects': ['Einstein', 'Feynman'], 'objects': ['Princeton University', 'University of Zurich'], your templates might be: ['[X] was educated at [Y].', '[X] studied at [Y].', '[X] was a student at [Y].']. Present your response as a Python list.
```

Figure 14: Prompt used to flag potential issues with the template combined with a sample of the relation's instances. For each relation, all three templates were filled with 5 subject-object pairs each. While the prompt was designed to spot linguistic issues in the template, it also aided in finding additional issues in the instances.