LM-PUB-QUIZ: A Comprehensive Framework for Zero-Shot Evaluation of Relational Knowledge in Language Models

Max Ploner* and **Jacek Wiland*** and **Sebastian Pohl*** and **Alan Akbik**

Humboldt Universität zu Berlin
Science Of Intelligence
<first name>.<last name>@hu-berlin.de

Abstract

Knowledge probing evaluates the extent to which a language model (LM) has acquired relational knowledge during its pre-training phase. It provides a cost-effective means of comparing LMs of different sizes and training setups and is useful for monitoring knowledge gained or lost during continual learning (CL). In prior work, we presented an improved knowledge probe called BEAR (Wiland et al., 2024), which enables the comparison of LMs trained with different pre-training objectives (causal and masked LMs) and addresses issues of skewed distributions in previous probes to deliver a more unbiased reading of LM knowledge. With this paper, we present LM-PUB-QUIZ, a Python framework and leaderboard built around the BEAR probing mechanism that enables researchers and practitioners to apply it in their work. It provides options for standalone evaluation and direct integration into the widely used training pipeline of the Hugging Face TRANSFORMERS library. Further, it provides a fine-grained analysis of different knowledge types to assist users in better understanding the knowledge in each evaluated LM. We publicly release LM-PUB-QUIZ as an open-source project.

1 Introduction

Pre-trained language models (LMs) currently take on a central role in state-of-the-art NLP approaches (Devlin et al., 2019). Given their importance, prior work has sought to measure the amount of factual knowledge encoded in LMs using *knowledge probing* mechanisms (Petroni et al., 2020; Kalo and Fichtel, 2022). Here, the knowledge represented in the parameters of an LM is automatically compared to factual knowledge in a relational knowledge base (KB). For instance, a probe might measure if an LM can correctly recall the capitals of countries, as illustrated in Figure 1.

*Equal contribution

Figure 1: The BEAR probe uses relational triples from a knowledge base (KB) to construct multiple-choice items. Here, it leverages the knowledge that "Kampala" is the capital of "Uganda", while "Thimpu", "Buenos Aires" and "Bandar Seri Begawan" (other *capital cities*) are not. It measures whether the LM correctly ranks the verbalization of the true fact higher than the distractors.

In previous work, we introduced a new knowledge probe called BEAR (Wiland et al., 2024) that addresses various issues of ambiguities and skewed answer distributions of prior probes to deliver a more unbiased reading of LM knowledge. Further, it reformulates probing as a ranking task, thus enabling a direct comparison of LMs trained with different pre-training objectives (masked and causal LMs) and vocabularies. However, despite being conceptually simple, BEAR relies on a different implementation than existing probes and previously returned only an overall score as the evaluation result, thus limiting adoption and interpretability.

Framework. With this paper, we present LM-PUB-QUIZ, an open-source Python framework and leaderboard built around the BEAR probing mechanism that enables researchers and practitioners to apply it in their work. Our framework was designed for ease of use, providing simple interfaces and direct integration into the Hugging Face TRANSFORMERS ecosystem (Wolf et al., 2020). Two use cases in particular have shaped the development of the library:

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (System Demonstrations), pages 29–39

1. The first main use case is to evaluate and compare already-trained LMs. Users only need to specify the string identifier of one of the LMs on the Hugging Face model hub to calculate the BEAR score for this model. This yields not only an overall BEAR score but also a more fine-grained analysis of different types of relational knowledge in the LM.

2. The second main use case is to monitor the knowledge gained and lost during pre-training and continual training (e.g. when adapting an LM to a new domain). Here, LM-PUB-QUIZ provides an easy integration into the Hugging Face `Trainer` to track knowledge development during training.

To encourage uptake, we make our library freely available and open source. Additionally, we are actively curating a leaderboard with scores of existing LMs. We encourage the community to participate in extending the list of evaluated models.[1]

2 Knowledge Probing Approach

We designed LM-PUB-QUIZ to easily assess the factual knowledge implicitly represented in a model's parameters (which is often referred to as 'parametric knowledge' or 'parametric memory'; Xie et al., 2024). This knowledge may be used by the model without the need to retrieve external information and is critical to a model's ability to perform intended functions such as answering questions or writing text on specified topics (Mialon et al., 2023).

Even with the increasing development and use of retrieval-augmented generation approaches, estimating the models' inherent knowledge remains critical to predict the performance of the overall system as (1) basic knowledge within the retrieval model may be required to select relevant documents in the first place, (2) good parametric knowledge can reduce how much external knowledge needs to be retrieved, (3) improve the extraction of information from retrieved content[2], and (4) incorrect information in the LM may be in conflict with the retrieved information, often leading the model to disregard the retrieved information (Koopman and Zuccon, 2023; Xie et al., 2024; Bi et al., 2024).

As proposed in previous work (Wiland et al., 2024), LM-PUB-QUIZ ranks alternative statements (such as "The capital of Uganda is Kampala" and "The capital of Uganda is Thimphu") using the LM's (pseudo) log-likelihood.[3] If the top-ranked alternative is correct, the fact is considered to have been answered correctly (see Figure 1). Relying only on the (pseudo) log-likelihood enables the comparison of models with different pretraining objectives with little constraints on the inserted answer (i.e., they can consist of multiple tokens and be placed anywhere in the sentence).

3 Framework Overview

We give an overview of LM-PUB-QUIZ, describe how it can be installed (Section 3.1), explain the basic components of the interface (3.2), and offer examples to illustrate its usage (3.3, 3.4, & 3.5).

3.1 Setup

The package containing LM-PUB-QUIZ can be installed in the desired environment using pip:

```
pip install lm-pub-quiz
```

It relies on the TRANSFORMERS package, which users can use to load pre-trained models locally or from the Hugging Face hub.

3.2 Interface

The API of LM-PUB-QUIZ features three basic objects.

`Dataset` represents the dataset used to evaluate the LM. Each dataset consists of a set of relations represented by the `Relation` class.

These relations are typically derived from the relations in the knowledge base (see Figure 1). Relations group instances of a similar type (e.g., relation `P36` links a country or other entity to its governmental seat) and have a shared set of possible answers (options in each multiple-choice question) and templates used to create the textual statements.

Each relation contains an instance table and information about their answer space. Relations can be annotated with additional information, such as the domains of knowledge they contain and their

[1]The leaderboard and GitHub repository are available at `https://lm-pub-quiz.github.io`. Released under the MIT License.

[2]Consider the example of Multi-Hop Reasoning: a RAG system is asked to answer who the spouse of the British prime minister is. The knowledge base only contains the information that Victoria Starmer is the wife of Keir Starmer, which is correctly retrieved. However, this is helpful to the generator model only insofar as it inherently *knows* that Kier Starmer is the British prime minister.

[3]Wiland et al. (2024) explain how pseudo log-likelihood can be retrieved from masked language models (MLMs).

```python
from lm_pub_quiz import Dataset,
    Evaluator

# Step 1: Load the BEAR probing dataset
dataset = Dataset.from_name("BEAR")

# Step 2: Load the LM (here: "gpt2")
    and create the evaluator
evaluator = Evaluator.from_model(
    "gpt2",
    model_type="CLM",
    device="cuda:0"
)

# Step 3: Run the evaluation and save
    the results
evaluator.evaluate_dataset(
    dataset,
    template_index=0,
    save_path="gp2_results",
    batch_size=32,
)
```

Listing 1: Example snippet for performing the BEAR probe on the GPT-2 model (Radford et al., 2019).

cardinality. By cardinality, we refer to the number of subjects for which a particular object is the correct answer: Either the relation is a one-to-one relationship, or there are multiple subjects with the same answer. If the cardinality is not provided in the metadata, it is derived from the relation data.

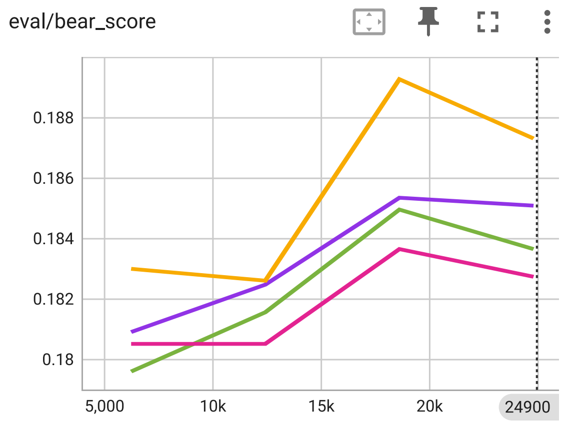
`Evaluator` is the functional component used to evaluate the model. It is instantiated with a model name (or model object). To evaluate the model on the dataset with a `Dataset`, the `evaluate_dataset` method is called (see 3.3).

`DatasetResult` is an object that is returned by the `evaluate_dataset` method. This object can be used to analyze the results of a specific model. It allows the accumulation of results across the relations (e.g. based on domains or cardinality) and enables accessing the instances-specific predictions.

3.3 Direct Evaluation of a Trained LM

The first main use case of LM-PUB-QUIZ is to evaluate the knowledge contained in a trained LM. We illustrate how to perform such an analysis for the GPT-2 model in Listing 1.

As the code example shows, it consists of three main steps: In Step 1, we load the BEAR evaluation dataset. In Step 2, we load the LM using its string identifier on the Hugging Face model hub (here: `gpt2`) and create an evaluator for causal language models (by passing `model_Type="CLM"`). Finally,



Run ↑	Value	Step	Relative
wikitext_0	0.1828	24,900	3.883 hr
wikitext_1	0.1873	24,900	4.183 hr
wikitext_2	0.1851	24,900	4.045 hr
wikitext_3	0.1837	24,900	4.039 hr

Figure 2: Example screenshot from TENSORBOARD, showcasing the Hugging Face `Trainer` integration of LM-PUB-QUIZ. Here, we monitor the knowledge of 4 `roberta-base` models (Liu et al., 2019), continuously pretrained on permutations of the Wikitext corpus.

in Step 3, we run the BEAR probe and store the evaluation results at the specified `save_path`.

By default, all instance-level predictions are stored on the file system, allowing the computation of all metrics supported in LM-PUB-QUIZ separately (see 3.5). It also allows for fine-grained inspection of all answers given by the LM. A more memory-efficient alternative is not to store the instance-level predictions and compute the metrics directly. This can be set by passing the `metrics` keyword to the `evaluate_dataset` method.

3.4 Monitoring Knowledge during Training

The second use case of LM-PUB-QUIZ is for monitoring the knowledge development in an LM during (continual) pre-training. To this end, we developed a Hugging Face `Trainer` integration. LM-PUB-QUIZ provides a callback that can be attached to the `Trainer` instance. The callback will then invoke the `Evaluator` in the specified frequency. This allows integration into monitoring tools like TENSORBOARD. See Figure 2 for an illustration.

3.5 Analysis Options

The BEAR probe consists of 60 relations retrieved from the WikiData knowledge base. Each

```python
from lm_pub_quiz import DatasetResults

bear_results = DatasetResults.from_path(
    "gp2_results",
    relation_info="./relation_info.json"
)

# Get accuracy by relation types
print(bear_results.get_metrics(
    ["accuracy"], accumulate="domains"))

# Output:
#                accuracy   support
# domains
# Arts           0.105263    1368.0
# Biographical   0.158820    2028.5
# Economic       0.115152     770.0
# ...
```

Listing 2: Example of evaluation by domain. This provides different scores for relations from domains such as "Arts", "Biographical", "Economic", etc.

relation connects exactly two entities to form a relation triple. Example relations in BEAR are HAS-CAPITAL (see Figure 1) that connects a country to its capital city, BORN-IN that connects a person to their country of birth, and CROSSES-RIVER that connects a named bridge to the river it crosses.

Each relation in BEAR has several relation instances, i.e., specific triples such as (HAS-CAPITAL, Uganda, Kampala). In total, BEAR has 7,731 and 40,916 of such triples in its default and expanded variants, respectively. As Figure 1 shows, each triple is used to form one multiple-choice item question in our evaluation. The default BEAR score is the accuracy across all questions.

LM-PUB-QUIZ offers several options for users to obtain more fine-grained analysis (see 4 for examples):

- First, users can compute separate BEAR scores for different domains of knowledge. To enable this analysis, we manually annotated each of the relations in BEAR with one or more related domains (in practice, up to three).[4] This allows analysis of per-domain knowledge gained or lost during training.

- Second, one can calculate separate scores for relations based on their cardinality, as BEAR includes both 1-1 relations and 1-N relations, where the latter has multiple possible answers as opposed to just a single one.

- The third option is to only aggregate the scores on a relation level. Since instances in a relation share a template, the relation-level scores may reveal issues with the verbalization of the triples or give more detailed insights into the areas of knowledge.

- Finally, one can choose to not aggregate at all, and compute the predictions per instance. This can be useful for fine-grained qualitative analysis to find knowledge bottlenecks.

As shown in Listing 2, the `DatasetResults` can compute these aggregated metrics. The `accumulate` keyword of the `get_metrics` method controls the manner of aggregation and may be set to `domain`, `cardinality`, or `False` for the above-mentioned aggregations. To inspect the instance-level predictions, one can use the `instance_table` attribute of each of the `RelationResult` objects.

The result object can be used to compute various metrics that measure the performance (such as accuracy, precision at k, and Brier score) as well as confidence metrics (such as entropy-based confidence). The metrics can easily be extended by subclassing the existing base classes.

4 Example Experiments

To showcase example applications, we present three novel experiments that show how LM-PUB-QUIZ can be used to conduct a detailed analysis of knowledge in LMs (Section 4.1 and 4.2) and how the Hugging Face integration (see Section 4.3) can be used to monitor knowledge in a continual pre-training setting.

4.1 Domain-specific Knowledge after Training on Different Corpora

When adapting an LM to a specific domain, one may be interested in the various areas of knowledge contained in the model's parameters. While the overall accuracy on the complete BEAR dataset reflects the model's general knowledge, a more granular examination of the relations can provide insights into the specific areas they relate to.

4.1.1 Experimental Setup

We adapt two base models, `roberta-base` and `gpt2` to three domains: arXiv abstracts (Clement et al., 2019), literary texts from `blbooks` (Labs, 2021), and Wikipedia text from `wikitext-103-v1` (Merity et al., 2016). Additional information on the training setup can be found in Appendix A.2.

[4] This annotation can be found in the dataset repository: https://github.com/lm-pub-quiz/BEAR

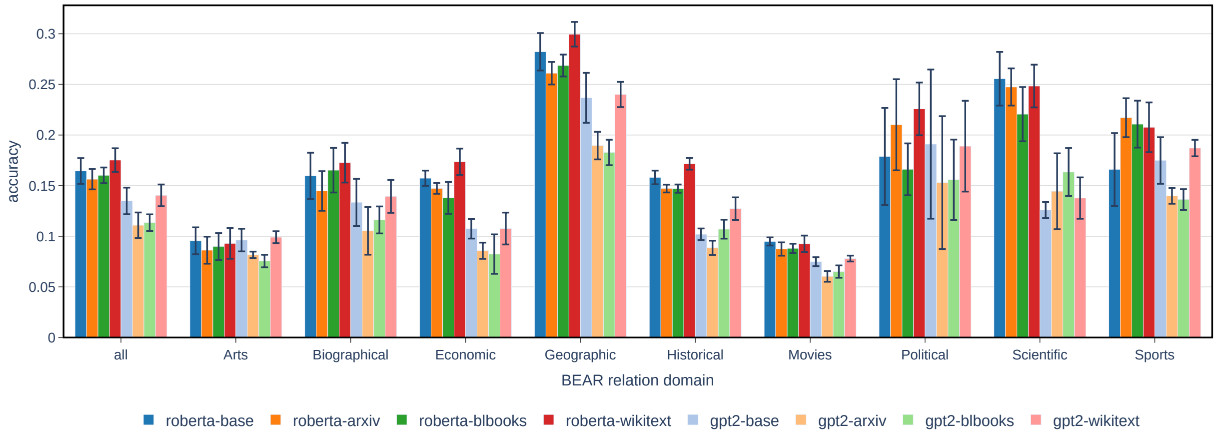


Figure 3: Accuracies on facts from different BEAR domains for `roberta-base` and `gpt2` continually pretrained on different corpora.

This yields a total of 8 models to compare: The two base models, the three domain-adaptations of `roberta-base` (`roberta-arxiv`, `roberta-blbooks`, `roberta-wikitext`), and the three domain-adaptations of `gpt2-base` (i.e. `gpt2-arxiv`, `gpt2-blbooks`, `gpt2-wikitext`).

4.1.2 Results

Figure 3 presents results for all 6 models across 10 BEAR domains. We generally find that all models score highest on geographical questions and lowest on questions from the "arts" and "movies" domains.

We also note that training with `wikitext` data improves the BEAR score the most, given that the BEAR probe was constructed from Wikidata. Further, we observe that training `GPT2` on arXiv abstracts leads to significant improvements in the scientific domain (see `gpt2-arxiv` vs `gpt2-base` in Figure 3). Finally, we find that the roberta-base model benefits from continual pretraining on `blbooks` corpus when tested on facts from the biographical and sports domains.

4.2 Investigating Model Biases

During pre-training, models are likely to acquire various biases, primarily due to the data they were trained on (Haller et al., 2024), potentially leading them to disproportionately favor certain answers.

In this experiment, we use LM-PUB-QUIZ with the BEAR probe to aggregate all the predicted answers given by an LM in each relation. Because the BEAR answer space is balanced, this aggregation results in an estimate of the model's bias, as each answer should be equally likely. We measure how much models are biased towards certain answers.

4.2.1 Experimental Setup

We select a single relation from the BEAR probe, `P30`, which connects locations and geographic entities to the continents they are located on (see Figure 4). We evaluate three pre-trained models on this relation: `roberta-base` (Liu et al., 2019), `gpt2` (Radford et al., 2019) and `Mistral-7B-v0.1` (Jiang et al., 2023), i.e. one MLM and two CLMs. Model biases are estimated by applying the softmax function to the BEAR pseudo-log-likelihood scores, resulting in values that can be interpreted as probabilities. These values indicate the likelihood that a sentence is correct, given that at least one of the answers is correct for the given subject. Subsequently, these distributions are averaged over all subjects, resulting in the overall bias. Since each answer occurs with equal frequency for this relation, a perfect model scoring all template instances correctly would produce a uniform bias.

4.2.2 Results

Figure 4 shows the models' biases for `P30`: `roberta-base` is biased towards 'South America' and 'Antarctica', while `GPT2` and `Mistral-7B-v0.1` are more likely to predict 'Europe'.

Averaging over all three of its templates, relation P30 gives an accuracy of 0.98, 0.62, and 0.45 for `Mistral`, `GPT2`, and `roberta-base`, respectively. Since `Mistral-7B-v0.1` predicts most of the answers correctly, the relative answer frequency is much closer to the uniform distribution.

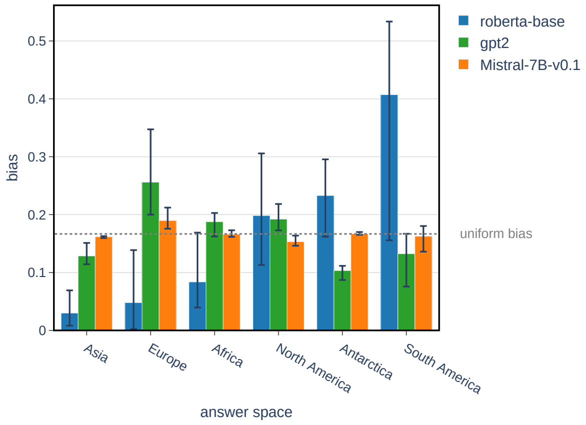


Figure 4: Performance of the selected models on the P30 relation of the BEAR probe averaged over relation templates. The bars indicate the complete range of results (over all templates).

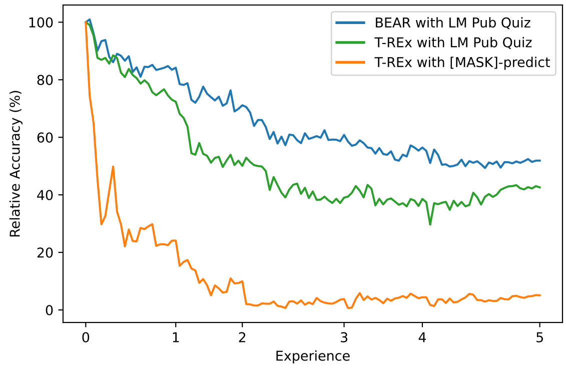


Figure 5: Trajectories of the knowledge represented in a `bert-base-cased` model during the continual learning as measured by LM-PUB-QUIZ and `[MASK]`-predict on T-REx dataset. Additionally, it shows performance of `bert-base-cased` evaluated on the BEAR probe.

4.3 Monitoring Knowledge during Continual Learning

Catastrophic forgetting (McCloskey and Cohen, 1989), a significant challenge in continual learning, occurs when a model loses previously acquired knowledge after being trained on new datasets. We hypothesize that traditional knowledge evaluation methods such as LAMA (Petroni et al., 2019), employing a `[MASK]`-predict approach, overestimate the extent of forgetting of relational knowledge.

4.3.1 Experimental Setup

We continually train a `bert-base-cased` model using the original MLM objective on a stream of five *experiences* with each experience consisting of scientific abstracts (Geiger, 2019) from two scientific domains (i.e. 10 domains in total).[5] At the end of each epoch, we evaluate the model's performance on BEAR and T-REx (Elsahar et al., 2018), which is part of the LAMA benchmark.

Two methods for knowledge evaluation were used: `[MASK]`-token filling and a multiple-choice question format with a closed answer space implemented via the LM-PUB-QUIZ package.[6]

All scores are calculated relative to the original performance (in the pre-trained state), showing the performance change during continual pre-training. Since the `[MASK]`-filling method predicts a token over the entire vocabulary of the model (in the case of `bert-base-cased`, it is over 30K vocabulary tokens), it is inherently more difficult than choosing from a limited answer space as in LM-PUB-QUIZ. Hence, relative scores are more suitable.

4.3.2 Results

The forgetting curves displayed in Figure 5 reveal the forgetting dynamics during the continual pre-training process. See Table 2 for a detailed summary of the relative performance scored using both evaluation techniques and Section A.1 for additional discussion (both in the appendix).

The results indicate different performance trajectories depending on the evaluation method. The `[MASK]`-predict approach indicates a much larger degree of forgetting. In a qualitative error analysis, we found that the model's predictions, although contextually reasonable, often do not match the expected answers due to the data distribution shift. For example, after five experiences of continual pre-training on scientific abstracts, the top three predictions for the cloze statement "The native language of Marie Curie is `[MASK]`" are "considered," "discussed," and "presented". While these may not be incorrect in some contexts, they do not match the expected answer "Polish" in T-REx. We, therefore, believe that the `[MASK]`-filling approach may not reliably indicate the amount of relational knowledge contained within the model's parameters.

By design, LM-PUB-QUIZ only considers answers that are appropriate to the context (but, except for one, not factually correct). With this approach, there is a much smaller decrease in performance, especially after the first experience. This characterization of catastrophic forgetting aligns with other research on continual learning, such

[5] For an overview of the dataset and the hyper-parameters used in these experiments, see Appendix 4.3.

[6] Due to the multi-token nature of this dataset, the [MASK]-predict method was not applicable. For a discussion of this issue, see Wiland et al. (2024).

as Cossu et al. (2022), which found that self-supervised training objectives can partially mitigate the problem of forgetting in sequential learning.

The evaluation results on BEAR reveal a forgetting behavior similar to that observed in T-REx. However, the performance degradation observed with the BEAR probe is notably the least severe among all the experiments conducted.

5 Comparison with Existing Libraries

The LM EVALUATION HARNESS framework (Gao et al., 2023) is one of the most well-known evaluation tools for large language models, featuring numerous benchmarks as part of its task suite, including knowledge tasks such as MMLU (Hendrycks et al., 2021). This framework primarily focuses on autoregressive language models and lacks support for MLMs. This limits the ability to compare the performance of different types of models on the same datasets.

Similarly, HELM (Liang et al., 2023), LLM-FACTEVAL (Luo et al., 2023), and the LLMEVAL module in OPENFACTCHECK (Iqbal et al., 2024) rely on the capability of CLMs to generate continuations to a prompt and are therefore not applicable to MLMs.

The unique feature of LM-PUB-QUIZ is its focused approach to cloze statement filling, allowing the answer to appear anywhere within a sentence. This method is compatible with any type of model (whether CLM or MLM) and any tokenization. By evaluating the log-likelihood score of the entire statement instead of just its continuation or the single answer token, LM-PUB-QUIZ overcomes the limitations of traditional `[MASK]`-predict approaches (Petroni et al., 2019) without relying on text-continuation capabilities.

6 Conclusion and Outlook

In this paper, we presented LM-PUB-QUIZ, an easy-to-use and versatile open source library for knowledge probing that can be seamlessly used with the BEAR probe. The framework covers two important use cases: Monitoring knowledge during continual pre-training (and domain adaptation) and analyzing existing pre-trained LMs.

We are actively working on extending the leaderboard and strongly encourage the community to participate. We will continue development on the library to support further use cases and welcome any input, whether in the form of feedback or contributions to the code base.

We are working to extend the BEAR probe to additional knowledge bases in order to expand on the domains of knowledge that can be evaluated with LM-PUB-QUIZ.

Acknowledgements

Max Ploner, Jacek Wiland, Sebastian Pohl, and Alan Akbik are supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2002/1 "Science of Intelligence" – project number 390523135. Alan Akbik is further supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under the Emmy Noether grant "Eidetic Representations of Natural Language" (project number 448414230).

References

Baolong Bi, Shenghua Liu, Lingrui Mei, Yiwei Wang, Pengliang Ji, and Xueqi Cheng. 2024. Decoding by Contrasting Knowledge: Enhancing LLMs' Confidence on Edited Facts. *Preprint*, arXiv:2405.11613.

Colin B. Clement, Matthew Bierbaum, Kevin P. O'Keeffe, and Alexander A. Alemi. 2019. On the use of arxiv as a dataset. *Preprint*, arXiv:1905.00075.

Andrea Cossu, Tinne Tuytelaars, Antonio Carta, Lucia Passaro, Vincenzo Lomonaco, and Davide Bacciu. 2022. Continual Pre-Training Mitigates Forgetting in Language and Vision. *arXiv preprint*. ArXiv:2205.09357 [cs].

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.

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).

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2023. A framework for few-shot language model evaluation.

R. Stuart Geiger. 2019. ArXiV Archive: A Tidy and Complete Archive of Metadata for Papers on arxiv.org.

Patrick Haller, Ansar Aynetdinov, and Alan Akbik. 2024. OpinionGPT: Modelling explicit biases in instruction-tuned LLMs. In *2024 Annual Conference of the North American Chapter of the Association for Computational Linguistics – System Demonstration Track*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Hasan Iqbal, Yuxia Wang, Minghan Wang, Georgi Nenkov Georgiev, Jiahui Geng, Iryna Gurevych, and Preslav Nakov. 2024. OpenFactCheck: A Unified Framework for Factuality Evaluation of LLMs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 219–229, Miami, Florida, USA. Association for Computational Linguistics.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

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

Bevan Koopman and Guido Zuccon. 2023. Dr ChatGPT tell me what I want to hear: How different prompts impact health answer correctness. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15012–15022, Singapore. Association for Computational Linguistics.

British Library Labs. 2021. Digitised books. c. 1510 - c. 1900. jsonl (ocr derived text + metadata). https://doi.org/10.23636/r7w6-zy15.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D. Manning, Christopher Ré, Diana Acosta-Navas, Drew A. Hudson, Eric Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong, Hongyu Ren, Huaxiu Yao, Jue Wang, Keshav Santhanam, Laurel Orr, Lucia Zheng, Mert Yuksekgonul, Mirac Suzgun, Nathan Kim, Neel Guha, Niladri Chatterji, Omar Khattab, Peter Henderson, Qian Huang, Ryan Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, and Yuta Koreeda. 2023. Holistic evaluation of language models. *Preprint*, arXiv:2211.09110.

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

Linhao Luo, Thuy-Trang Vu, Dinh Phung, and Gholamreza Haffari. 2023. Systematic assessment of factual knowledge in large language models. *Findings of EMNLP*.

Michael McCloskey and Neal J Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *Preprint*, arXiv:1609.07843.

Grégoire Mialon, Roberto Dessì, Maria Lomeli, Christoforos Nalmpantis, Ram Pasunuru, Roberta Raileanu, Baptiste Rozière, Timo Schick, Jane Dwivedi-Yu, Asli Celikyilmaz, Edouard Grave, Yann LeCun, and Thomas Scialom. 2023. Augmented Language Models: A Survey. *Preprint*, arXiv:2302.07842.

Fabio Petroni, Patrick Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2020. How Context Affects Language Models' Factual Predictions. *arXiv preprint*. ArXiv:2005.04611 [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.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Jacek Wiland, Max Ploner, and Alan Akbik. 2024. BEAR: A unified framework for evaluating relational knowledge in causal and masked language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2393–2411, Mexico

City, Mexico. Association for Computational Linguistics.

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

Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. 2024. Adaptive Chameleon or Stubborn Sloth: Revealing the Behavior of Large Language Models in Knowledge Conflicts. *Preprint*, arXiv:2305.13300.

A Additional Information on the Experiments

Due to the number of experiments and the limited space, we provide additional information on the experiments we presented in this part of the appendix.

A.1 Continual Pre-training Experiment

Dataset In our experiments with continual learning of the `bert-base-cased` in the Section 4.3, we use a subset of the arXiv dataset (Geiger, 2019). We use the same data splits as Cossu et al. (2022), i.e., same document classes and observations. Specifically, the following classes of scientific abstracts were used: `'hep-ph'`, `'astro-ph'`, `'hep-th'`, `'quant-ph'`, `'cond-mat.mes-hall'`, `'gr-qc'`, `'cond-mat.mtrl-sci'`, `'cond-mat.str-el'`, `'condmat.stat-mech'` and `'astro-ph.SR'`. Selecting these specific abstracts enables us to evaluate their findings on mitigating forgetting during self-supervised learning. These scientific domains primarily span physics and materials science. Each of these ten classes has a training set of approximately 10,000 abstracts and a validation set of about 1,000 abstracts.

Training hyperparameters The hyperparameters used are reported in Table 1.

Hyperparameter	Value
Per Device Train Batch Size	8
Per Device Eval Batch Size	8
Gradient Accumulation Steps	1
Learning Rate	0.00005
Weight Decay	0
Number of Training Epochs	30
Learning Rate Scheduler Type	Linear
Warmup Ratio	0.0
Metric for Best Model	Evaluation Loss
Early Stopping Patience	5
Early Stopping Threshold	0

Table 1: Hyperparameters used during continual pre-training.

Additional Results & Discussion The relative performance of `bert-base-cased` measured on T-REx task after the *ith* experience of continual pre-training as measured by the LM Pub Quiz and [MASK]-predict are shown in Table 2. The scores are normalized with respect to their base performance before continual pre-training. *0th* experience corresponds to the original model taken from Hugging Face.

The [MASK]-predict technique exhibits a significant degradation in performance from the outset of continual pre-training, with over a 75% decrease observed after the first experience. Overall, this method suggests that nearly 95% of the knowledge was lost during training on the arXiv dataset. On the other hand, results obtained with LM Pub Quiz show a relatively smaller decrease of approximately 60%.

Evaluation / Experience	T-REx [MASK]-predict (%)	T-REx LM Pub Quiz (%)
0	100.00	100.00
1	24.12	72.37
2	9.98	50.09
3	3.79	39.04
4	4.41	38.58

Table 2: The relative performance of `bert-base-cased` measured on T-REx task during continual pre-training. Before the continual pre-training the model achieves 31.3% accuracy using [MASK]-predict and 40.5% using LM Pub Quiz as well as 18.4% accuracy on BEAR.

A certain degree of this difference in degradation can be explained by the difference in random baseline. When using [MASK]-predict, degrading to the level of the random baseline would amount to a drop of almost 100% while when using LM Pub Quiz this would lead to a drop of only roughly 90% due to a higher accuracy of the random baseline (given the smaller answer space).

A.2 Training on Different Domain Corpora

Domain	Num. of Train Tokens
arXiv	$2,49e^8$
blbooks	$1,72e^8$
wikitext	$2,82e^8$

Table 3: The number of tokens seen by each individual adapted model. The `wikitext-103-v1` dataset contained this number of tokens in total after some minor cleaning.

Each model was trained on a similar number of tokens (see Table 3). We trained four models per dataset. For Wikitext, we used four permutations of the entire set. The arXiv dataset was instead split into four equal chunks of the given size. The BLBooks dataset was split into more chunks, but only four chunks of the given size were used for training. All models were trained with the hyperparameters reported in Table 4.

A.3 Pre-trained Model Bias

Extending the results of section 4.2 we also estimated model biases for relation `P30` using only six manually chosen generic subjects for the relation, including, for example, 'it' and 'the region'. Model

Hyperparameter	Value
Per Device Train Batch Size	32
Gradient Accumulation Steps	1
Learning Rate	1e-05
Weight Decay	0
Number of Training Epochs	1
Learning Rate Scheduler Type	Cosine
Warmup Ratio	0.0

Table 4: Hyperparameters Used for Model Training on Different Domains

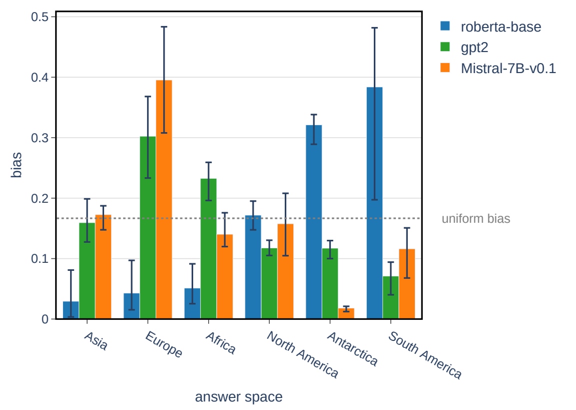


Figure 6: Generic subject biases for relation P30 of the BEAR probe for various models.

biases are again estimated by applying Softmax to the BEAR pseudo-log-likelihood scores and by averaging the resulting distributions over all generic subjects.

Figure 6 shows the results for this way of calculating biases. We can observe the same trend mentioned before: roberta-base is biased towards South America and Antarctica, whereas GPT2 and Mistral-7B-v0.1 are biased towards Europe. But this time, Mistral-7B-v0.1 appears to be even more biased than GPT2. When biases are computed in this second way, they indicate which answers a model chooses without subject information. While Mistral-7B-v0.1 shows a high bias here, it still predicts many correct answers, resulting in a lower bias according to the first method. It appears as though its subject-specific knowledge overcomes this bias, while the smaller, less performative GPT2 is less able to overcome this kind of bias.

B Additional Information on the Use of other Datasets

The package was primarily developed to enable the use of the BEAR dataset. Still, the approach is quite general and can be used to cover other domains than the rather general knowledge represented in this subset of Wikidata or answer different research questions altogether.

Each dataset consists of a set of relations (JSONL files) and metadata in the metadata_relations.json JSON file. For each relation, the metadata file contains one or more templates and (optionally) the definition of an answer space (see Listing 3). If no answer space is given, it is constructed from all objects in the relation.

Listing 3: Definiton of the templates and answer spaces of relations in metadata_relations.json of a LM PUB QUIZ dataset.

```
{
    "<relation id>": {
        "templates": [
            "[Y] is the answer to some fact
                with subject [X].",
            ...
        ],
        "answer_space_labels": [
            "<some object label>",
            ...
        ],
        "answer_space_ids": [
            "<object ids>",
            ...
        ]
    },
    ...
}
```

The templates are used to construct alternative textual statements: “[X]” is replaced by the subject of the instance, and “[Y]” is replaced by each of the options in the answer space to construct one statement per answer option.

Each relation contains one instance per line (the file should be named <relation id>.jsonl; see Listing 4). Each (represented by a JSON object) should have a subject and object (i.e., the correct answer) ID as well as labels for the subject (and object if the answer space is not defined in the metadata).[7] The instances require either an object ID or the index of the correct answer in the answer space defined in the metadata file.

Listing 4: Definiton of instances in a relation of a LM PUB QUIZ dataset (single line).

```
{"sub_id": "<subject id>", "sub_label":
    "<subject label>", "obj_id": "<ID of the
    correct answer>", "obj_label": "<correct
    object label>", "answer_idx": <index of
    the correct object in the answer space>}
```

[7]The IDs of the subjects and objects should be unique (though they can be shared across the relations) and may refer to the IDs of the underlying knowledge base. Additional fields (such as aliases) are not used at the moment.