Cosine Similarity as Logits?: A Scalable Knowledge Probe Using Embedding Vectors from Generative Language Models

Tomoyuki Jinno Kazuki Hayashi Yusuke Sakai Hidetaka Kamigaito Taro Watanabe
Nara Institute of Science and Technology
jinno.tomoyuki.jx3@naist.ac.jp
{hayashi.kazuki.hl4, sakai.yusuke.sr9, kamigaito.h, taro}@is.naist.jp

Abstract

Recently, the use of pretrained language models (PLMs) as soft knowledge bases has gained growing interest, sparking the development of knowledge probes to evaluate their factual knowledge retrieval capabilities. However, existing knowledge probes for generative PLMs that support multi-token entities exhibit quadratic time complexity $\mathcal{O}(n^2)$, where n corresponds to the number of candidate entities, limiting the size of knowledge graphs used for probing. To address this, we propose DEcoder Embedding-based Relational (DEER) probe, utilizing embedding vectors extracted from generative PLMs. DEER probe achieves effective time complexity of linear order $\mathcal{O}(n)$, supports rank-based evaluation metrics including Hit@k, handles multi-token entity names and enables probing whilst disambiguating homographic tail-entity names. We empirically show that DEER-probe correlates with existing knowledge probes, validating its probing capability, and we demonstrate the practical benefits of its improved scalability.

1 Introduction

Knowledge probes evaluate factual knowledge retrieval capabilities of pre-trained language models (PLMs). Their applications include identifying missing knowledge in PLMs and quantifying the amount of domain-specific knowledge encoded in their parameters. Knowledge probes achieve this by assessing a PLM's capability to complete a relational knowledge. A knowledge graph (KG) represents a relational knowledge as a triplet consisting of (*head-entity*, *relation-type*, *tail-entity*). To complete a relational knowledge, models must predict the correct tail-entity, given a partially filled triplet, (*head-entity*, *relation-type*, ?) which we call a query.

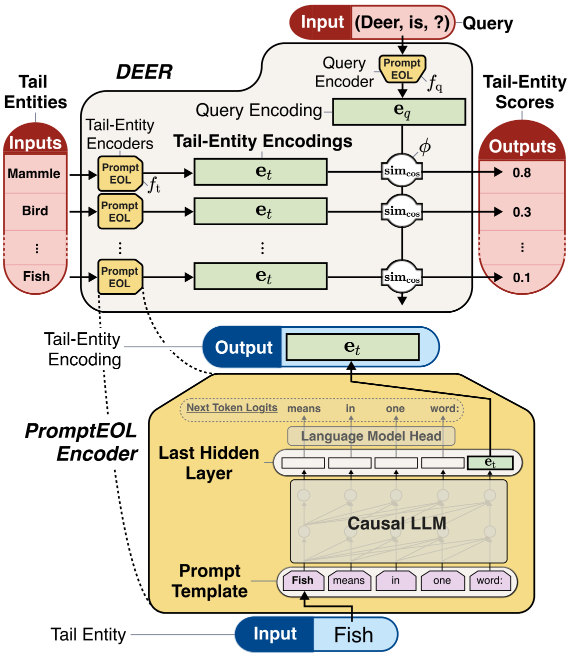


Figure 1: Illustration of DEER probe's architecture.

To the best of our knowledge, the only knowledge probe capable of probing generative PLMs with multi-token tail entity names and the Hit@k, a conventional evaluation metric in knowledge base completion, is BEAR (Wiland et al., 2024; Youssef et al., 2023). However, BEAR exhibit quadratic time complexity $\mathcal{O}(n^2)$, where n corresponds to the number of candidate entities, making probing of PLMs on large-scale KGs infeasible. Moreover, BEAR predicts the tail entity for a query by computing the joint probability of the textual sequence formed by concatenating the query and a tail entity candidate's textual representation. The reliance on joint probability may introduce a token length bias, favoring candidates with shorter names.

To address these issues, we propose DEER (DEcoder Embedding-based Relational) Probe, a knowledge probe utilizing embedding vectors extracted from a causal PLM through the PromptEOL method (Jiang et al., 2024), as shown in Figure 1. DEER effectively realizes linear time complexity, $\mathcal{O}(n)$, whilst reducing the token length bias in

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics
Volume 1: Long Papers, pages 8188–8200
March 24-29, 2026 ©2026 Association for Computational Linguistics

probing. DEER also enables probing whilst disambiguating identically named tail entities.

This work addresses the following questions:

RQ1: Does the improved time complexity yield observable reductions in compute time?

RQ2: Does DEER align with existing knowledge probes?

RQ3: Does DEER exhibit reduced token length bias compared to BEAR?

To evaluate the practical implications of DEER's improved time complexity, we measured the time required to probe GPT-2-Small (Radford et al., 2019) with WN18RR (Dettmers et al., 2018) under both BEAR and DEER. BEAR was estimated, through extrapolation, to require 330±10 days, rendering full evaluation infeasible, whereas DEER completed the task in 16 ± 1 minutes, making such evaluation tractable. Token length bias was assessed by computing the Pearson correlation between predicted ranks and tail entity token lengths. BEAR exhibited a strong correlation $r = 0.484$ and $p < 10^{-51}$, while DEER was uncorrelated with $r = 0.005$ and $p = 0.913$, indicating DEER's ability to evaluate knowledge independent of token length. Finally, DEER's alignment with the established probe, LAMA, was assessed via log-scale rank correlation, yielding a maximum $r = 0.804$, supporting its use for knowledge probing. We also comprehensive analyzed the generalizability of our method using 5PLMs and 4KGs, ensuring robustness of our findings.

2 Background and Related Work

Knowledge Probing A KG is defined as a set of triplets $\mathcal{T} \ni (h, r, t)$, where h, r and t denote the head entity, relation-type and tail entity. In knowledge probing, given a query $(h, r, ?)$, PLMs are tasked with predicting the corresponding tail entity t. This is typically achieved by ranking a set of candidate tail entities $\mathcal{E}$, according to their probability $P(t = e | e \in \mathcal{E})$. Performance is evaluated using Hit@k which measures the fraction of test triplets for which the correct tail entity is ranked within the top k candidates.

Existing Knowledge Probes for Causal PLMs The first proposed knowledge probe, LAMA (Petroni et al., 2019), prompts a PLM with a cloze-style question and ranks tail entity candidates by the log likelihood of the token corresponding to their name at the masked position, but it can only test entities with single-token names. KAMEL (Kalo and Fichtel, 2022) extended this approach to multi-token entities using text generation; yet, since predicted tail entities are evaluated through exact string matching, the evaluation metric is limited to Hit@1. BEAR was later introduced to support both the Hit@k metric and multi-token entities, though it requires computing log likelihoods for all possible query–entity combinations. Consequently, the PLM must process $\mathcal{O}(|\mathcal{Q}| \times |\mathcal{E}|)$ inputs, where $\mathcal{Q}$ is the set of queries, leading to quadratic time complexity. Furthermore, because prior methods depend on token predictions, they cannot disambiguate identically named tail entity candidates.

Textual Knowledge Graph Completion The textual knowledge graph completion (KGC) task involves predicting missing relations in a knowledge graph by selecting the correct tail entity from a set of candidates, given a partially filled triplet, or query, using a large language model (LLM). Although textual KGC and knowledge probing share similar formulation, they differ in their intention. KGC aims to maximize completion accuracy, whereas knowledge probing aims to measure the factual knowledge present in the pretrained language model (PLM). Consequently, KGC methods typically rely on fine-tuning to enhance performance, while fine-tuning of the model is generally avoided in probing because it alters the model's internal knowledge and hinders faithful evaluation.

Encoder-based KGC models have improved inference efficiency by transitioning from cross-encoder to bi-encoder architectures. The cross-encoder architecture, introduced to KGC by KG-BERT (Yao et al., 2019), jointly encodes every query–tail-candidate pair, resulting in quadratic scaling with the number of evaluated pairs. In contrast, the bi-encoder architecture, introduced to KGC in StAR (Wang et al., 2021), embeds queries and entities independently. This separation enables precomputation and efficient similarity-based retrieval, reducing the scaling order to linear.

If probing of generative PLMs could be formulated under a similar bi-encoder framework, comparable efficiency gains could be achieved. Yet, developing a technique for doing so with a generative language model without requiring fine-tuning remains an open challenge.

PromptEOL Embedding PromptEOL (Jiang et al., 2024) is a method for creating sentence em-

e_name - e_desc
This sentence: "{word}" means in one word: "{one word}"
This sentence: "e_name" means in one word: "

Figure 2: Template used by the tail encoder f_t, where the entity name and description replace e_name and e_desc, respectively. Figure 3 shows a complete example.

deer - distinguished from Bovidae by the male's having solid deciduous antlers
This sentence: "{word}" means in one word: "{one word}"
This sentence: "deer" means in one word: "

Figure 3: Example of a complete tail-entities encoding template for the "deer" entity. Thereby, e_name is replaced by "deer" and e_desc is replaced by "distinguished from Bovidae by the male's having solid".

bedding vectors using a generative PLM without additional training. It prompts a model to summarize a sentence in one word, then uses the last hidden vector, usually used for next-token prediction, as the sentence embedding. The prompt template: *This sentence: {S} means in one word "*, is used, where *S* is replaced by the sentence to encode. Appendix B.1 provides a formal description.

3 Problem Formulation

Textual Knowledge Graph We define a textual knowledge graph as a KG with textual representations of entities and relation types. We denote set of all entity names, entity descriptions and relation names as $\mathcal{E}_\text{name}$, $\mathcal{E}_\text{desc}$ and $\mathcal{R}_\text{name}$, respectively. Thereby, given a set of all possible strings Σ^*, $\mathcal{E}_\text{name}, \mathcal{E}_\text{desc}, \mathcal{R}_\text{name} \subseteq \Sigma^*$. We assume the existence of $\mathcal{E} \to \mathcal{E}_\text{name}$, $\mathcal{E} \to \mathcal{E}_\text{desc}$ and $\mathcal{R} \to \mathcal{R}_\text{name}$, mapping the entities and relations to their corresponding textual representations.

Bi-Encoder Based Probes Bi-Encoder based knowledge probing frameworks consist of a query encoder f_q, a tail entity encoder f_t and a similarity function ϕ. Given functions mapping the query and entities to their textual representations, $f_\text{q}^\text{text} \colon \mathcal{Q} \to \Sigma^*$ and $f_\text{t}^\text{text} \colon \mathcal{E} \to \Sigma^*$, the query encoder and the tail entity encoder maps the textual representation of queries and tail entity candidates to an encoding vector $f_\text{q} \colon f_\text{q}^\text{text}(\mathcal{Q}) \to E_\text{q}$ and $f_\text{t} \colon f_\text{t}^\text{text}(\mathcal{E}) \to E_\text{t}$, where $E_\text{q}, E_\text{t} \subseteq \mathbb{R}^n$.

The similarity function $\phi \colon E_\text{q} \times E_\text{t} \to \mathbb{R}$ measures the similarity between a query embedding and a tail entity embedding. Given a query $q \in \mathcal{Q}$, the model ranks all tail entity candidates $t \in \mathcal{E}$ according to their similarity scores defined by ϕ. Formally, let $N = \{1, \ldots, |\mathcal{E}|\}$ denote the set of rank indices. A function $I \colon \mathcal{Q} \times N \to \mathcal{E}$ assigns to each query q a ranked list of tail entities such that $\phi(\mathbf{e}_q, \mathbf{e}_{I(q,i)}) \geq \phi(\mathbf{e}_q, \mathbf{e}_{I(q,i+1)})$ for all $i \in [1, |\mathcal{E}| - 1]$, where $\mathbf{e}_q = f_\text{q}(f_\text{text}(q))$ and $\mathbf{e}_t = f_\text{t}(f_\text{text}(t))$. Since each query and entity embedding is computed once, the total encoding cost scales linearly as $\mathcal{O}(|\mathcal{Q}| + |\mathcal{E}|)$. Because the cost of cosine similarity computation is negligible compared to the encoder forward pass, the effective time complexity is $\mathcal{O}(n)$.

(η^1_name, ρ^1_name, τ^1_name)
⋮
(η^8_name, ρ^8_name, τ^8_name)
(h_name, r_name,

(dress up, verb group, trick up)
⋮
(disfavour, hypernym, single out)
(deer, hypernym,

Figure 4: Left: Query-encoding template, f_q^text for knowledge probing. The names of the query's h and r replaces the symbols h_name and r_name. The names of h, r and t in the n^th randomly sampled triplet replaces η^n_name, ρ^n_name and τ^n_name . Right: Completed example for a query, (deer, hypernym, ?).

4 DEER: Proposed Method

As shown in Figure 1, DEER implements the bi-encoder based framework using PromptEOL as both the query encoder, f_q, and the tail-entity encoder, f_t. To generate the embedding vectors for the query and tail-entity, custom templates are populated with $\mathcal{E}_\text{name}$, $\mathcal{E}_\text{desc}$, and $\mathcal{R}_\text{name}$. The constructed text is then fed into f_q and f_t as inputs. The last hidden vector, used as a sentence embedding in PromptEOL, serves as the embedding vector. Cosine similarity is used as ϕ to assign a unique rank, N, to each tail entity candidate. The following paragraphs discuss the custom templates used by f_q and f_t.

Tail-Entities Encoding Template, f_t^text Figure 2 and Figure 3 shows the template used by the tail-entity encoder, f_t to acquire a tail-entity encoding. The inclusion of e_desc enables disambiguation of identical names.

Prober Query Encoding Template, f_q^text Figure 4 shows the template used by the query encoder, f_q, during knowledge probing. The eight-shot example is compiled by randomly sampling from the training set.

Dataset Name	$\lvert\mathcal{Q}\rvert$	$\lvert\mathcal{E}\rvert$
WN18RR Compute Time Experiment	93,003	40,944
WN18RR Single Token	596	4,948
FB15k237 Single Token	2,161	254
YAGO3-10 Single Token	98	90
WN18RR Token Bias Experiment	500	948

Table 1: Statistics of the datasets used in experiments.

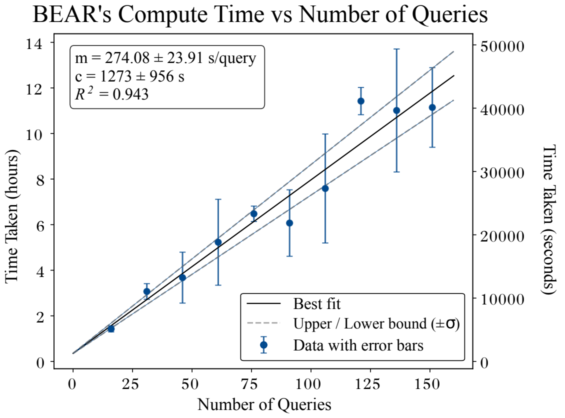


Figure 5: Plot of a linear regression model fitted to BEAR's probing time, used to estimate the time required for BEAR to probe the dataset. The fit conforms with expectation for a probe with $\mathcal{O}(|\mathcal{Q}| \times |\mathcal{E}|)$ complexity.

Encoding Template Design Rationale The rationale behind the template design is as follows: if the PLM has correct knowledge of the triplet, the token it predicts next when given the query template should be semantically similar to the token it predicts when given the correct tail-entity template. Consequently, when the model makes a correct prediction, the similarity function ϕ produces a high score between the encoding vectors $\mathbf{e}_q$ and $\mathbf{e}_t$.

5 Experiments and Results

Three sets of experiments were conducted to address the research questions. These are: compute time, probe agreement, and token bias experiments. The statistics of the datasets used throughout the experiments are given in Table 1.

5.1 Compute Time Experiment

5.1.1 Experimental Setup

To demonstrate the practical impact of DEER's improved scaling law, the time required to probe GPT2-Small 124M[1] using the entire WN18RR KG was compared between the DEER and BEAR probes. DEER's compute time was directly measured over all $|\mathcal{Q}| = 93,003$ queries obtained from the KG. All $|\mathcal{E}| = 40,943$ entities found in the KG were used as tail candidates. The experiment was repeated three times to obtain the mean and the uncertainty. In contrast, BEAR's total compute time was approximated through linear extrapolation as its direct measurement is infeasible. The time required to probe subsets of queries with sizes $|\mathcal{Q}| = \{16, 31, 46, 61, 76, 91, 106, 121, 136, 151\}$ was measured three times each, then the ordinary least-squares regression model was fitted to the data, $t = m|\mathcal{Q}| + c$. The regression error was quantified as the standard error of the slope, computed from the residual variance of the fitted model. The obtained model was then extrapolated to $|\mathcal{Q}| = 93,003$, in order to obtain the computation time and uncertainty. BEAR's compute time was measured using the library[2] published by Wiland et al. (2024). The experiments were conducted on an RTX 3090 GPU with 8 CPU cores.

Prober	Time/query	Total time
BEAR ap.	200±500 s/query	330±10 days
DEER	0.0316±0.0009 s/query	16±1 minutes

Table 2: Compute time comparison on the WN18RR. BEAR results are approximated via extrapolation.

5.1.2 Results

The results in Table 2 show that the improved time complexity leads to a substantial reduction in probing time. Under our setup, DEER reduced the compute time from approximately 330±10 days to 16±1 minutes, corresponding to a $(30,000 \pm 2,000)\times$ speedup, thereby enabling probing with KGs that would otherwise be infeasible using BEAR.

The fitted line, shown in Figure 5, achieved, $R^2 = 0.943$. Since a model with $\mathcal{O}(|\mathcal{Q}| \times |\mathcal{E}|)$ complexity would scale linearly when $|\mathcal{E}|$ is fixed, provides evidence for BEAR's quadratic scaling behavior. The plot simultaneously rejects $\mathcal{O}(|\mathcal{Q}| + |\mathcal{E}|)$ as BEAR's complexity, since when $|\mathcal{E}| \gg |\mathcal{Q}|$, the gradient of the fitted line would be $m \approx 0$, in contrary to our observation.

5.2 Probe Agreement Experiment

The aim of this experiment is to assess DEER's viability as a knowledge probe. We consider two criteria that a valid knowledge probe should intuitively satisfy. First, the scores of a knowledge probe should be consistent with those of an established knowledge probe. Second, the monotonicity

[1] https://huggingface.co/openai-community/gpt2

[2] https://github.com/lm-pub-quiz/lm-pub-quiz

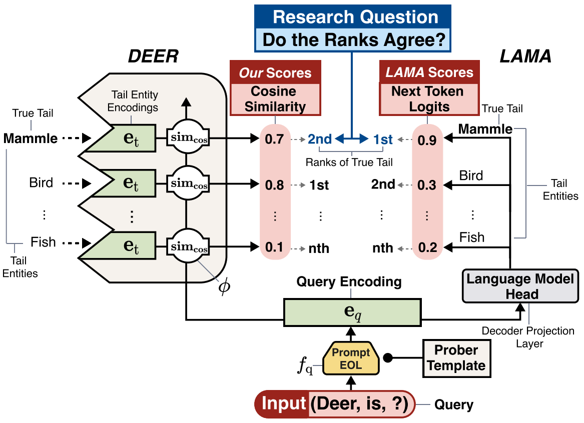


Figure 6: Diagram of the LAMA agreement experiment illustrating the shared use of the query encoding vector by DEER and LAMA.

of scores across parameter size should be preserved for the same model family, assuming models with larger parameter size store greater knowledge than the smaller models.

5.2.1 Experimental Setup

Comparing DEER against LAMA To assess if DEER's ranks are consistent with an established probe, they were compared with that of LAMA, an established and widely used knowledge probe. The consistency of rankings between the two probers is evaluated using Pearson's correlation, computed on a log scale to account for the diminished significance of differences at higher ranks. To mitigate the biases induced by differences in prompt design, the query encoding template was also used when computing LAMA's ranks, as our main interest lies in the effect of the architectural difference rather than the prompt design. Consequently, the tail entities of LAMA were ranked identically to the original work, that is, by the log-likelihood of each token. Since PromptEOL's encoding vectors are identical to the vectors typically used in next-token prediction, the log-likelihood was computed by directly applying the pretrained linear layer, also known as the language modeling head, to the query encoding vector generated by the query encoder, f_q. This procedure is illustrated in Figure 6. As LAMA can only handle single-token tail entities, the dataset was preprocessed by removing triplets with multi-token tail entities from the KG.

Evaluating Monotonicity of Score In order to assess if the monotonicity of the scores is conserved across parameter size, DEER and LAMA were compared using OPT models of varying sizes {125M, 350M, 1.3B, 6.7B} on WN18RR. Hit@k of the probers were computed for each model to assess if DEER and LAMA conserve monotonicity across parameter size.

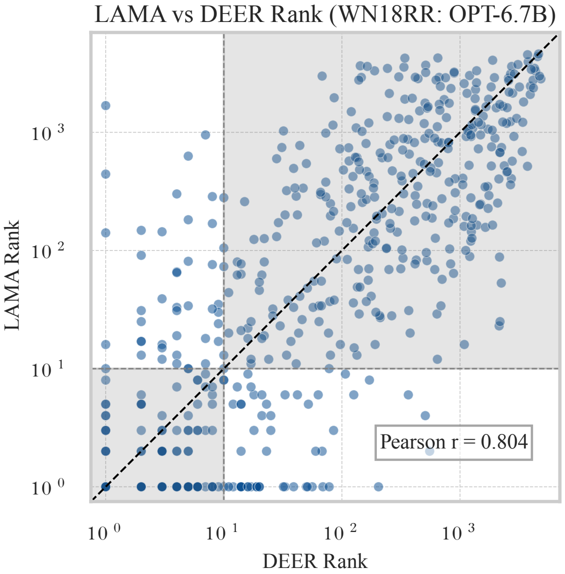


Figure 7: Scatter plot of tail entity ranks predicted by LAMA and DEER in log scale. Each point corresponds to a triplet in the KG. The bottom-left and top-right quadrants shaded gray are regions of agreement in Hit@10 and Miss@10, respectively.

Assessing Generalizability of Agreements Although the generalizability of the PromptEOL method across model families was demonstrated in its original work (Jiang et al., 2024), we further examined whether the observed agreement between the probers extends across domains and model families. To test domain generalization, the two probes were compared on the FB15k-237 (Toutanova and Chen, 2015) and YAGO3-10 knowledge graphs. To test model generalization, we used the LLAMA3-8B model with the WN18RR dataset. Prompt sensitivity analysis is given in Appendix C.3

5.2.2 Dataset Preparation

As LAMA cannot handle multi-token tail entities, all triplets containing multi-token tail entities were removed from the KG. The entity names for the WN18RR and FB15k237 were acquired from the dataset provided by Yao et al. (2019). The names of WN18RR in the dataset are provided with metadata such as the part of speech attached, e.g., "__whitetail_deer_NN_1", these were preprocessed by removing the metadata and underbars and inserting spaces between words, e.g., "whitetail deer".

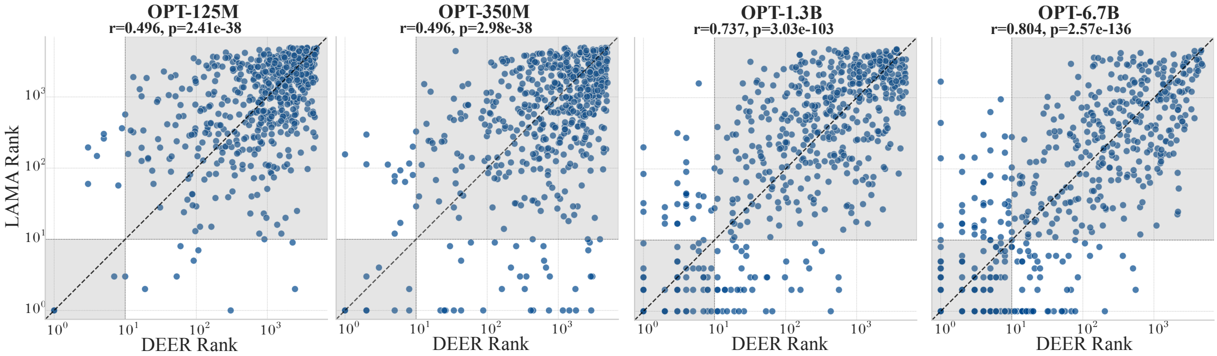


Figure 8: Facet grid of scatter plots comparing LAMA's predicted ranks with DEER's for OPT models of sizes 125M, 350M, 1.3B, and 6.7B on WN18RR. The plots illustrate how predicted ranks shift as model size increases.

PLM Name	Log(Rank), r	p value
OPT-125M	0.496	2.41E-38
OPT-350M	0.496	2.98E-38
OPT-1.3B	0.737	3.03E-103
OPT-6.7B	0.804	2.57E-136

Table 3: Pearson's correlation between LAMA and DEER ranks on the WN18RR dataset across different model parameter sizes.

Prober Name	Hit@1	Hit@10	Hit@100
LAMA-OPT-125M	0.5%	7%	10%
DEER-OPT-125M	0.3%	2%	12%
LAMA-OPT-350M	3%	7%	17%
DEER-OPT-350M	0.5%	4%	13%
LAMA-OPT-1.3B	6%	16%	32%
DEER-OPT-1.3B	3%	14%	35%
LAMA-OPT-6.7B	22%	36%	55%
DEER-OPT-6.7B	9%	35%	55%

Table 4: Hit@{1, 10, 100} scores for LAMA and DEER evaluated on the WN18RR dataset.

KG	PLM Name	Log(Rank), r	p value
FB	OPT-1.3B	0.775	≤E-6
YA	OPT-1.3B	0.900	2E-36
WN	LLAMA3-8B	0.719	3E-127

Table 5: Pearson's correlation between LAMA and DEER ranks for combinations of KGs and PLMs, demonstrating that the agreement between the two probes generalizes across domains and model families.

KG	Prober Name	Hit@1	Hit@10	Hit@100
FB	LAMA-OP1.3B	27%	62%	97%
FB	DEER-OP1.3B	25%	48%	95%
YA	LAMA-OP1.3B	49%	83 %	-
YA	DEER-OP1.3B	46%	78 %	-
WN	LAMA-LL3-8B	1%	30%	66%
WN	DEER-LL3-8B	9%	35%	62%

Table 6: Hit@{1, 10, 100} scores LAMA and DEER evaluated on combinations of KGs and PLMs. Acronyms used: FB = FB15k237, YA = YAGO3-10, WN = WN18RR, OP = OPT, and LL3 = LLAMA3.

Empty descriptions were replaced with a dash character: "-".

YAGO3-10 does not provide any textual descriptions for the entities, so the underscore character "_" was used as their descriptions. Triplets where a gender, male or female, is the tail entity were removed from the YAGO dataset as they are too trivial for the LLMs.

5.2.3 Results

LAMA, DEER Comparison Correlation between the ranks of DEER and LAMA probe on the WN18RR dataset is shown in Table 3. Sub-billion parameter models exhibit limited rank agreement, whereas models over a billion parameters achieve high correlation, with OPT-6.7B achieving the highest correlation of $r = 0.804$, thereby supporting DEER's viability as a knowledge probe for super-billion-parameter models. The correlation for the OPT-6.7B model is visually illustrated through a scatter plot in Figure 7. Furthermore, as shown in Table 4, their corresponding Hit@k metrics showed low deviation from their counterparts with the largest absolute error of 11%, which was observed in OPT-6.7B's Hit@1.

Monotonicity across Parameter Size Figure 8 presents a facet grid of scatter plots across varying parameter sizes. If the probers behave as expected, their scores should increase with the parameter size of the PLMs. Accordingly, when LAMA and DEER are in agreement, we expect the data points to cluster in the top-right quadrant, region of Miss@10 agreement, for a small PLM and pro-

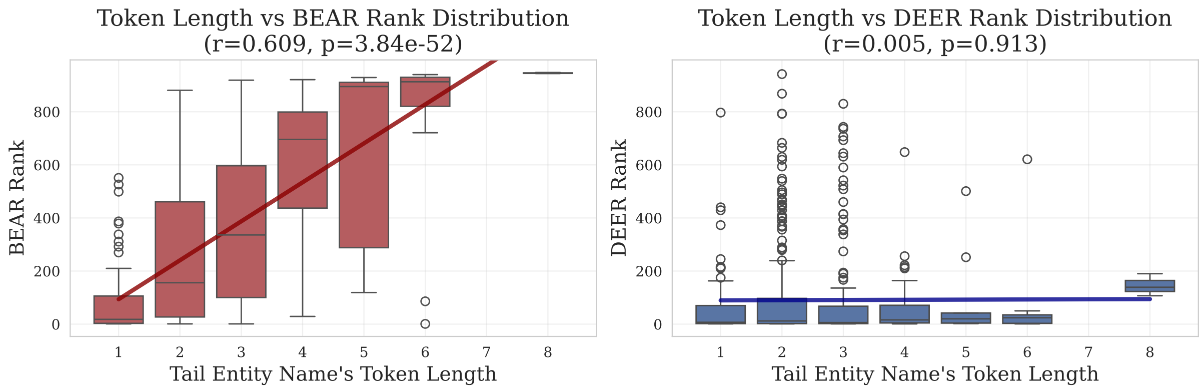


Figure 9: Box plot showing a positive correlation between tail-entity token length and BEAR ranks, and a negligible correlation under DEER.

gressively shift toward the bottom-left quadrant, region of Hit@10 agreement, as the parameter size increases. Qualitatively, this trend was observed in the facet grid, suggesting that monotonicity is preserved.

This effect can also be observed in Table 4, where the monotonicity of the Hit@k metric is conserved across all model scales, thereby, $m \in \{\text{Hit@1}, \text{Hit@10}, \text{Hit@100}\}$, $s_m^{(p)}(\text{125M}) \leq s_m^{(p)}(\text{350M}) \leq s_m^{(p)}(\text{1.3B}) \leq s_m^{(p)}(\text{6.7B})$, where $s_m^{(p)}$ denotes the score of prober $p \in \{\text{LAMA}, \text{DEER}\}$ under metric m. The observed agreement with LAMA and the conservation of monotonicity support the use of DEER as a valid knowledge probe.

Generalizability In line with the observations from the WN18RR dataset, a high correlation was observed on the FB15k238 and YAGO3-10 datasets using OPT-1.3B. As shown in Table 5 their corresponding Pearson's correlation of $r = 0.775$ and $r = 0.900$, thereby supporting the generalizability of the observations made using the WN18RR dataset across different domains. As shown in Table 6, all observed Hit@k values had an absolute error, less or equal to 14%, thereby demonstrating their consistency across multiple settings.

The scatter plots of the corresponding experiments are shown in Appendix C.1 Figure 15 and Figure 16. Similarly, Pearson's correlation of $r = 0.719$ was observed on the WN18RR dataset using LLAMA3-8B, thereby supporting the generalizability of the findings across model families. Its scatter plot is shown in Appendix C.1 Figure 17.

5.3 Token Length Bias Experiment

To investigate if DEER exhibits lower token length bias than BEAR, the Pearson correlation between the token length of the tail-entity name and the predicted ranks was compared. To investigate if the bias leads to disagreements, DEER and BEAR ranks were compared.

5.3.1 Experimental Setup

Token Length Correlation 500 queries were randomly sampled from the WN18RR test set to compute Pearson's correlation between the tail entity length and the predicted rank. OPT-1.3B was used as the PLM, and its tokenizer was used to measure the tail entity length.

BEAR, DEER Comparison The agreement between DEER and BEAR was compared to investigate if the potential bias leads to disagreement in the predicted rank between the two probers. As our focus lies in the effect of architectural differences rather than prompt design, a few-shot prompt consistent in style with the query encoder was used in place of the template adopted in the original BEAR implementation, the template used in our experiment is shown in Appendix B.2, Figure 11. Furthermore, a comparison using the dataset published in BEAR's original work is given in Appendix C.2.

5.3.2 Results

Token Length Correlation The histogram in Figure 9 illustrates the correlation between the probers' predicted ranks and the token lengths of the correct tail entities. BEAR exhibits a positive correlation ($r = 0.484$, $p < 10^{-5}$), whereas DEER shows negligible correlation ($r = 0.005$, $p = 0.913$). Assuming that the difficulty of a query is independent of the length of its tail entity, these results suggest that BEAR is susceptible to token-length–induced false positives and negatives, while DEER remains unaffected.

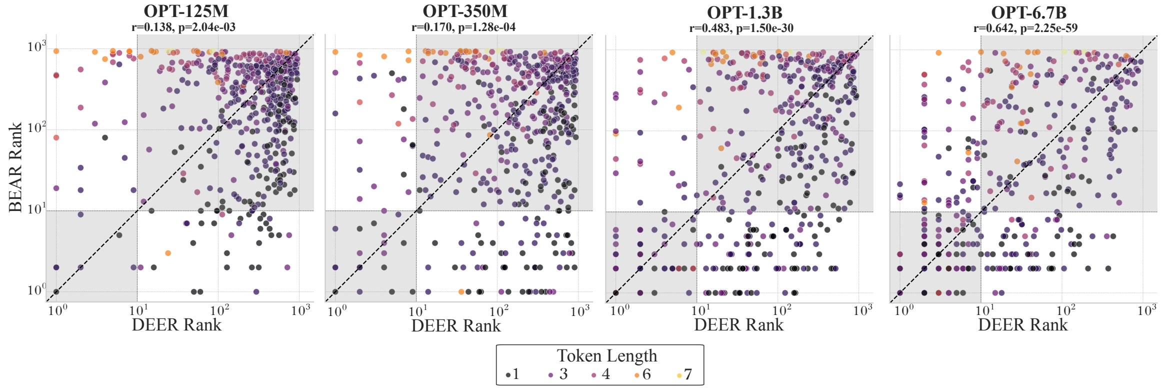


Figure 10: Facet grid of scatter plots comparing BEAR's and DEER's predicted ranks for OPT models of sizes 125M, 350M, 1.3B, and 6.7B evaluated on the WN18RR dataset. The plots show that the top-left quadrants are predominantly occupied by entities with long token length, while the bottom-right quadrants contain shorter entities.

Figure 10 compares BEAR and DEER ranks across different parameter size. The strongest correlation between the two was observed for the OPT-6.7B model, with Pearson's $r = 0.642$, whereas sub-billion-parameter models showed limited agreement, with $r = 0.138$ for OPT-125M.

BEAR, DEER Comparison The top left quadrant of the scatter plots corresponds to disagreements between the probers with Hit@10 for DEER and Miss@10 for BEAR, and the bottom right corresponds to Miss@10 for DEER and Hit@10 for BEAR. Qualitatively, we observe that the top left quadrants is occupied by points with token length larger than the mid point, denoted by the orange color, where as the bottom right quadrant is occupied by points with shorter token length, denoted in violet. Thereby, BEAR under ranks (predicts high rank relative to DEER) when tail entity is long and over ranks (predicts low rank relative to DEER) when the tail entity length is short. This discrepancy in the predicted rank may be induced by the token-length bias discussed prior. This effect of discrepancy can be quantitatively observed by their Pearson's correlation as shown in Table 7. However, as shown in Table 8, a limited discrepancy in the Hit@k metric was observed between the two probers, with the largest absolute error of 11% in the Hit@10 score of OPT-1.3B model. Additional comparison using dataset proposed in BEAR's original work is given in Appendix C.2.

PLM Name	Log(Rank), r	p value
OPT-125M	0.138	2E-03
OPT-350M	0.170	1E-04
OPT-1.3B	0.483	2E-3
OPT-6.7B	0.642	2E-59

Table 7: Table of Pearson's correlation between BEAR and DEER ranks on the WN18RR dataset across different model parameter sizes.

Model Name	Hit@1	Hit@10	Hit@100
BEAR-OPT125M	1%	9%	27%
DEER-OPT125M	2%	7%	24%
BEAR-OPT350M	4%	17%	39%
DEER-OPT350M	1%	9%	35%
BEAR-OPT1.3B	6%	34%	55%
DEER-OPT1.3B	5%	21%	50%
BEAR-OPT6.8B	10%	52%	71%
DEER-OPT6.8B	7%	51%	78%

Table 8: Table of Hit@{1, 10, 100} scores for BEAR and DEER evaluated on the WN18RR dataset.

6 Conclusion

This paper introduced DEER probe, a novel and scalable probing method for generative PLMs. By leveraging embedding vectors extracted from a generative PLM, DEER achieves input size to scale linearly, $\mathcal{O}(|\mathcal{Q}|+|\mathcal{E}|)$, instead of quadratically $\mathcal{O}(|\mathcal{Q}|\times|\mathcal{E}|)$, as required by BEAR.

We empirically showed that the improved time complexity leads to significant reduction in compute time. Specifically, under our setup, time taken to probe the WM18RR KG was decreased from approximately 330±10 days with BEAR to 16±1 minutes. Moreover, DEER aligned with the established LAMA probe while preserving the monotonicity of probing scores across parameter scales, providing supporting evidence of its validity as a knowledge probe. Finally, we showed that DEER's avoidance of token-level probability in ranking led to a reduced correlation with entity name length.

Through this work, we demonstrated a method for probing generative PLMs under a bi-encoder framework and established its validity and advantages as a knowledge probe, offering a promising direction for the future design of generative knowledge probing methods. Furthermore, the improved computational efficiency of DEER enables the probing of knowledge graphs that were previously infeasible, opening opportunities for large-scale experiments that may deepen our understanding of generative PLMs.

7 Limitations

Possibility of Data Leakage The original dataset of WN18RR, the WN18 (Bordes et al., 2013) had both been released before the of PLMS such as OPT, meaning it may have been included within the training corpus. Even though progress have been made in knowledge graph completion evaluation technique that avoids such issue (Sakai et al., 2024), utilization of such techniques on DEER is non-trivial, as it utilizes hypothetical, non real-world relations, that would not be stored in the parametric knowledge. However, since our aim is to evaluate the knowledge memorized within the parametric knowledge, it does not alter the fact these knowledge are stored within the parameters.

The Use of Log Scale The agreement between DEER, LAMA and BEAR was compared in linear and log scale, even though the two showed weaker correlation in the linear scale, we argue that the conclusion drawn in log scale should be preferred, taking similar line of argument for the preference of MRR over Rank in the knowledge base completion community, where the weight on the importance of the difference in rank is reduced as the rank increase.

Inclusion of Tail-Entity Descriptions during Knowledge Probing When encoding tail-entities during knowledge probing, the description of the entity is provided to allow disambiation. This allows the PLM to infer relations correctly, even in the absence of knowledge around the tail-entity. This could be an issue when assessing the knowledge retrieval capability of PLMs, however, we argue the effect of such case is limited as demonstrated by the high correlation between DEER and LAMA, Table 3, where LAMA is never shown such descriptions.

Limited Scope in Prompt Design Exploration The main objective of this study is to validate the use of the bi-encoder framework for knowledge probing in generative language models. Accordingly, our analysis primarily focuses on the effect of different model architectures on probing performance rather than on prompt design. The exploration of optimal prompt templates is thus beyond the scope of this work and remains as future research.

Evaluated Models and Architectural Designs While our experiments demonstrate the effectiveness of the proposed technique on the models considered, with primary focus on the OPT model following the setup of the original PromptEOL paper (Jiang et al., 2024), its performance on recently developed models and emerging architectures is yet to be investigated. Given the rapid pace of innovation in model design, assessing its behavior on these evolving architectures is a promising direction for future work.

References

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, page 2787–2795, Red Hook, NY, USA. Curran Associates Inc.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

Ting Jiang, Shaohan Huang, Zhongzhi Luan, Deqing Wang, and Fuzhen Zhuang. 2024. Scaling sentence embeddings with large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3182–3196, Miami, Florida, USA. Association for Computational Linguistics.

Jan-Christoph Kalo and Leandra Fichtel. 2022. Kamel: Knowledge analysis with multitoken entities in language models. In *AKBC*.

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.

Yusuke Sakai, Hidetaka Kamigaito, Katsuhiko Hayashi, and Taro Watanabe. 2024. Does pre-trained language model actually infer unseen links in knowledge graph completion? In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8091–8106, Mexico City, Mexico. Association for Computational Linguistics.

Kristina Toutanova and Danqi Chen. 2015. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, pages 57–66, Beijing, China. Association for Computational Linguistics.

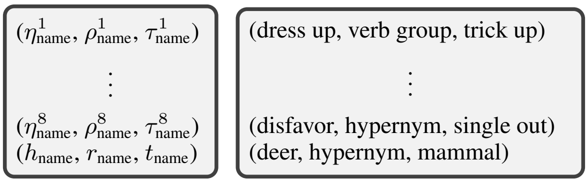
Bo Wang, Tao Shen, Guodong Long, Tianyi Zhou, Ying Wang, and Yi Chang. 2021. Structure-augmented text representation learning for efficient knowledge graph completion. In *Proceedings of the Web Conference 2021*, WWW '21, page 1737–1748, New York, NY, USA. Association for Computing Machinery.

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.

Kosuke Yamada and Peinan Zhang. 2025. Out-of-the-box conditional text embeddings from large language models. *Preprint*, arXiv:2504.16411.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Kg-bert: Bert for knowledge graph completion. *Preprint*, arXiv:1909.03193.

Paul Youssef, Osman Koraş, Meijie Li, Jörg Schlötterer, and Christin Seifert. 2023. Give me the facts! a survey on factual knowledge probing in pre-trained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15588–15605, Singapore. Association for Computational Linguistics.



$(\eta^1_{\text{name}}, \rho^1_{\text{name}}, \tau^1_{\text{name}})$
⋮
$(\eta^8_{\text{name}}, \rho^8_{\text{name}}, \tau^8_{\text{name}})$
$(h_{\text{name}}, r_{\text{name}}, t_{\text{name}})$

(dress up, verb group, trick up)
⋮
(disfavor, hypernym, single out)
(deer, hypernym, mammal)

Figure 11: Left: BEAR's prompt template used to calculate the probability of tail-entity, given a query. The names of query's head entity and relation type replaces the symbols h_{name} and r_{name}. The names of head-entity, relation-type and tail-entity in the n^{th} randomly sampled triplet replaces η^n_{name}, ρ^n_{name}, τ^n_{name}. Right: A completed example for a query, (deer, hypernym, ?) and tail-entity candidate, mammal. In essence, it is DEER's query encoding template with the addition of the tail entity name.

A Source Code

The source code that can be used to reproduce the experiments in this paper is publicly available at a git repository[3].

B Methodological Details

B.1 Formal Description of PromptEOL

PromptEOL uses a generative PLM for sentence embedding. Given a sequence of input tokens $x_1, x_2, \ldots, x_n$, the last hidden state, $\mathbf{h}_n$, corresponding to the token x_n, is used as the sentence embedding vector. More specifically, $\mathbf{h}_n$ is the vector typically used for next token prediction by applying a final dense layer followed by a softmax function. This process is expressed as $\mathbf{z}_n = \mathbf{W}\mathbf{h}_n + \mathbf{b}$, where $x_{n+1} = \arg\max(\text{softmax}(\mathbf{z}_n))$, where $\mathbf{h}_n$ is the last hidden layer, $\mathbf{W}$ is the weight of the projection layer and $\mathbf{b}$ is the bias term. To embed a sentence, it uses the following prompt template: *This sentence: "S" means in one word* ", where S is replaced with the target sentence. PromptEOL enables creation of embedding vectors conditioned by prompts (Yamada and Zhang, 2025).

B.2 Prompt Templates for BEAR

BEAR utilized the prompt template shown in Figure 11 to compute the probability for each query tail-entity pair.

[3] https://github.com/TJ-coding/deer

entity: “e_{name}”, description: “$e_{\text{description}}$”
Summarize the entity: “e_{name}” in one word: “

Figure 12: Prompt A template used by the query encoder to evaluate DEER’s sensitivity to prompt design.

< “e_{name}": “$e_{\text{description}}$”>
In one word, “e_{name}” means: “

Figure 13: Prompt B template used by the query encoder to evaluate DEER’s sensitivity to prompt design.

Metrics	Prompt A	Prompt B	Prompt C
Hit@1	0.5%	1.5%	2.3%
Hit@10	5.4%	7.7%	10.9%
Hit@100	26.9%	30.9%	34.2%

Table 9: Hit@{1, 10, 100} metrics achieved by each prompt design on the WN18RR dataset.

C Supplementary Results and Analysis

C.1 Generazability Experiments

Scatter plots from the experiments described in Section 5.2.1, designed to assess the generalizability of the DEER method, are shown. The comparison using FB15k237 and YAGO3-10 is shown in Figure 15 and Figure 16, respectively. The comparison using LLAMA3-8B[4] (Dubey et al., 2024) is shown in Figure 17. All plots show a positive correlation, indicating consistent agreement between the two probes across different setups.

C.2 BEAR Agreement Experiment

Experimental Setup Pearson correlation between the predicted ranks by DEER and BEAR was measured using a subset of BEAR dataset[5] (Wiland et al., 2024) and OPT-6.7B. As the entity candidates are defined for each relation type, the two probes were compared on four randomly selected relation types, p272, p344, p466, and p1412. A scatter plot of the comparisons was made in a linear scale, as the number of tail entity candidates is small, with a maximum of 60 candidates, and the correlation was also calculated in a linear scale.

Result Figure 18 shows the results of the comparison. Qualitatively, the quality of agreement varied across relation types. With relation types p272 and p1312 demonstrating Pearson correlation of $r = 0.345$ and $r = 0.361$, where as relation type p343 demonstrated lower correlation with $r = 0.150$ and $p = 0.67$.

[“e_{name}”, “$e_{\text{description}}$”]
Summarize the meaning of {e_{name}} in one word {

Figure 14: Prompt C template used by the query encoder to evaluate DEER’s sensitivity to prompt design.

C.3 Prompt Sensitivity Analysis

To assess DEER’s robustness to prompt design, we intentionally degraded the original DEER query template to create three alternative variants, Prompt A, Prompt B, and Prompt C. The designs of these prompts are shown in Figure 12, Figure 13, and Figure 14, respectively. As summarized in Table 9, performance varies moderately across templates, with the largest absolute difference being 7.3 percentage points in Hit@100. Despite these variations, the results exhibit a clear monotonic pattern across Hit@{1, 10, 100}, indicating that DEER consistently preserves the relative ranking structure under prompt degradation. This stability suggests that DEER’s relational reasoning is resilient to changes in prompt formulation.

D Ethical Considerations

D.1 Usage of AI Tools

This paper was written with the assistance of a large language model (ChatGPT, OpenAI) for language-related support, including improving clarity, grammar, and phrasing. The AI tool was not used to generate scientific claims, experimental designs, data, results, or interpretations. All technical content, analyses, and conclusions are the sole responsibility of the authors.

[4] https://huggingface.co/meta-llama/Meta-Llama-3-8B

[5] https://github.com/lm-pub-quiz/BEAR

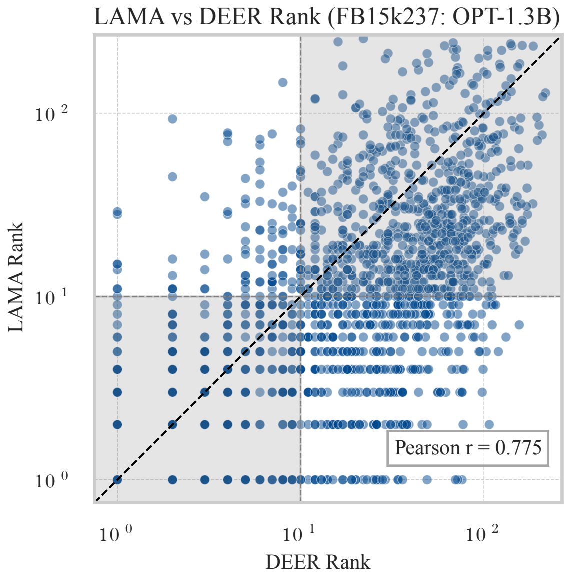


Figure 15: Scatter plot of tail entity ranks predicted by LAMA and DEER using OPT-1.3B in log scale evaluated on the FB15k237 dataset with $|\mathcal{Q}| = 2,161$ and $|\mathcal{E}| = 254$.

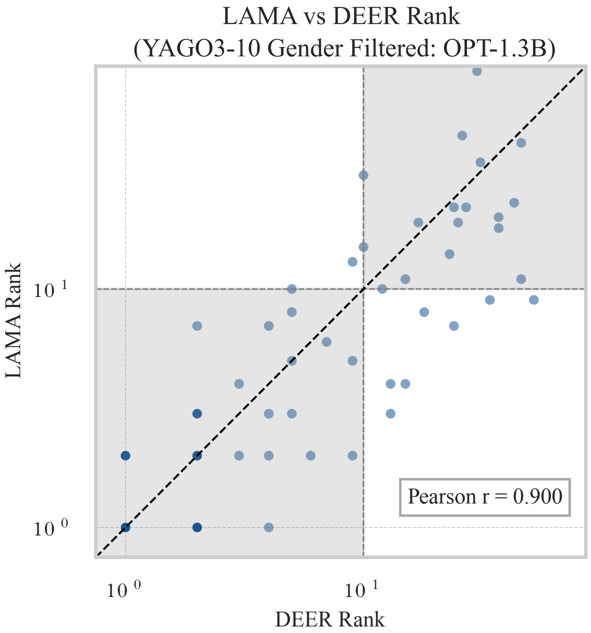


Figure 16: Scatter plot of tail entity ranks predicted by LAMA and DEER using OPT-1.3B in log scale evaluated on the YAGO3-10 dataset with $|\mathcal{Q}| = 98$ and $|\mathcal{E}| = 90$.

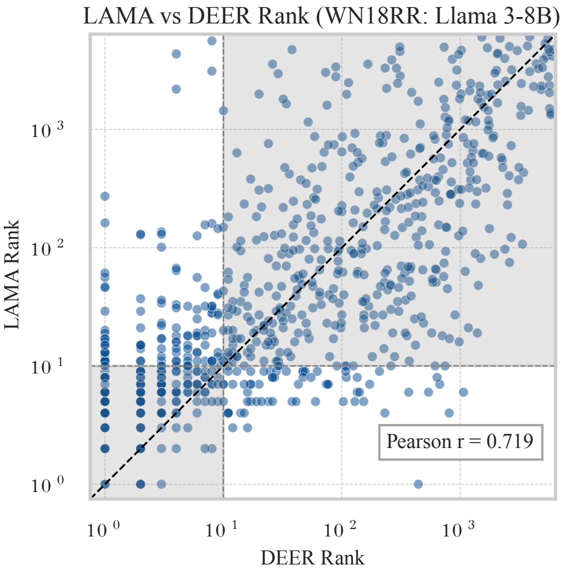


Figure 17: Scatter plot of tail entity ranks predicted by LAMA and DEER using LLAMA3-8B in log scale evaluated on the WN18RR dataset with $|\mathcal{Q}| = 596$ and $|\mathcal{E}| = 4,948$.

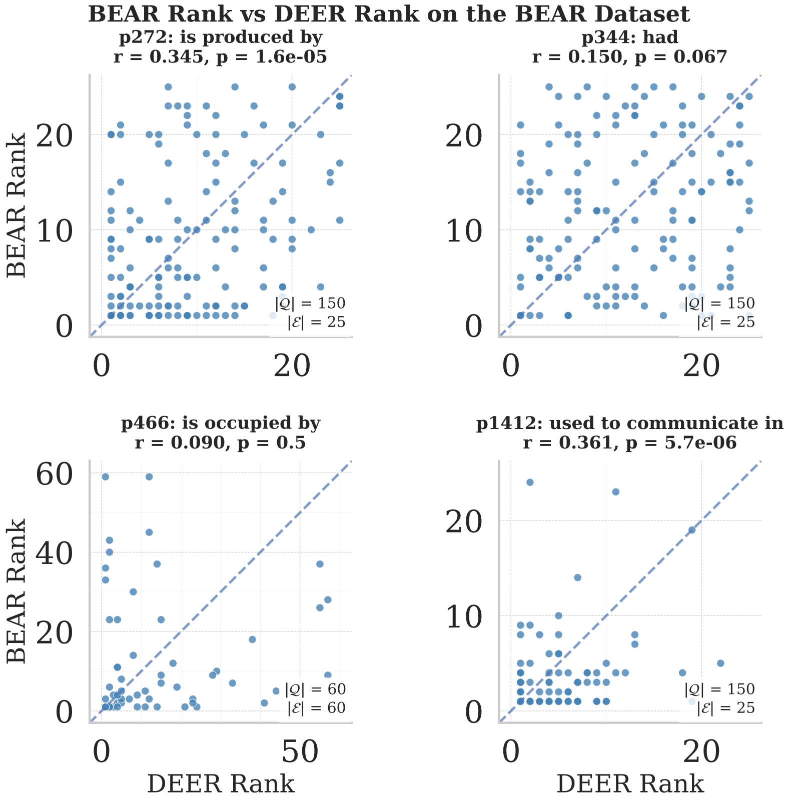


Figure 18: Comparisons between BEAR Rank and DEER Rank on the BEAR Dataset. The comparison was made on a linear scale to account for the lower number of tail entity candidates, with a maximum of 60 candidates. OPT-6.7B was used for all comparisons.