Discovering Expert LLMs via Next-Token Log Probabilities and Supervised Ranking

Gabrielle Poerwawinata
g.poerwawinata@uva.nl
IRLab, University of Amsterdam
Amsterdam, The Netherlands

Jingfen Qiao
j.qiao@uva.nl
IRLab, University of Amsterdam
Amsterdam, The Netherlands

Abstract

The Million LLMs Track (TREC Million LLMs) focuses on methods for ranking large language models (LLMs) based on their expected ability to answer a given query. As tasks increasingly require a combination of both general-purpose and domain-specific models, it is vital to predict which LLM is best suited for a given query without needing to query each model directly. We propose a supervised learning-to-rank approach that exploits next-token log probabilities from pre-generated responses as zero-cost pseudo-relevance signals. For each (query, LLM) pair, we derive a soft relevance label from the mean next-token log probability of the model's prior response, indicating the model's confidence as a proxy for answer quality. A LightGBM-based LambdaRank model is trained on feature vectors combining query embeddings (Sentence-BERT), categorical LLM identifiers, and global token-level statistics filtered at the top percentile. On the TREC Million LLMs test set, our best configuration achieves NDCG@10 of 0.3695, substantially outperforming tag-based (0.013) and response-based (0.195) baselines. Our ablation analysis shows that LLM-level statistics contribute more to ranking quality than query-specific embeddings, suggesting that global model capability is a dominant signal in the current evaluation setting.

Keywords

Learning to rank (LTR), log-likelihood, LLM expert discovery

1 Introduction

The sheer number of available Large Language Models (LLMs) makes choosing the right one a practical challenge: given a user query and a pool of available models, which LLM should be invoked to produce the best answer? This question is non-trivial because the pool may contain hundreds or thousands of models with overlapping but distinct knowledge, and querying every model for every request is computationally prohibitive. As both general-purpose models (e.g., the GPT family, Gemini [11]) and domain-specialized models continue to rise (e.g., BloombergGPT [15] for finance), the need for efficient model selection methods grows accordingly.

The TREC 2025 Million LLMs Track formalizes this challenge. Participants receive a *discovery dataset* containing pre-generated LLM responses with associated metadata for a set of training queries, and must rank models for unseen test queries. One distinctive characteristic of the TREC Million LLMs Track is that the LLMs in the pool share the same underlying base model but differ in their retrieval-augmented generation (RAG) document collections. Each LLM is therefore not a distinct architecture but a distinct knowledge source because of the same model augmented with a different corpus of retrieved documents. The next-token log probabilities we exploit as ranking signals thus reflect not only model confidence but also the quality and relevance of the underlying retrieved context. We hypothesize that when an RAG augmented model has relevant documents available, it generates with high confidence because its output is grounded in supporting evidence.

We propose using next-token log probabilities as a zero-cost supervision signal for this task. Our key assumption is that the average log-probability that a model assigns to its own generated tokens reflects its generation confidence, which can serve as a proxy for response quality. These log-probability scores are a byproduct of generation, they require no human annotation, no external evaluation, and no additional computation beyond what is already performed during response generation that we found in the discovery dataset. By aggregating these signals across the discovery set, we construct per-model capability profiles and train a LightGBM-based ranking model [8] to predict query-LLM relevance. At test time, ranking a new query requires only a single pass through a sentence encoder and LightGBM inference, making the approach feasible for pools of arbitrary size.

We evaluate several configurations and find that: (1) log-probability-based features substantially outperform tag-based and response-similarity baselines, (2) LLM-level global statistics are a stronger signal than query-specific embeddings in the current setting, and (3) the top-percentile filtering of log-probability scores yields stable model capability estimates.

2 Methodology

2.1 Log-Probability Features for Ranking LLMs.

Query representation. For each (query q_i, LLM ℓ_j) pair, we construct a feature vector concatenating three components. The query is encoded with Sentence-BERT (`all-MiniLM-L6-v2`) to a dense embedding $\mathbf{e}_{q_i} \in \mathbb{R}^{384}$. Each LLM ℓ_j is identified by a scalar integer id_{ℓ_j} extracted from its name (e.g. `llm_0032` → 32). While this encoding implies an arbitrary numeric ordering, LightGBM's gradient-boosted decision trees partition on splits rather than linear order, effectively treating the scalar as a soft categorical identifier. Global LLM capability statistics $\mu_{\text{global}}^{(\ell_j)}$ and $\sigma_{\text{global}}^{(\ell_j)}$ are computed from token-level log-probability aggregation (described below). The full feature vector is:

$$\mathbf{x}_{ij} = \left[\mathbf{e}_{q_i};\ \mathrm{id}_{\ell_j};\ \mu_{\text{global}}^{(\ell_j)};\ \sigma_{\text{global}}^{(\ell_j)}\right] \in \mathbb{R}^{387} \tag{1}$$

TREC '25,

Global capability statistics. For each LLM ℓ_j and discovery query q_i, we compute the average log-likelihood of the generated response $a^{(\ell_j)}$:

$$\mu_{q_i,a^{(\ell_j)}} = \frac{1}{T_{ij}} \sum_{t=1}^{T_{ij}} \log P_{\theta_{\ell_j}} \left(a_t^{(\ell_j)} \mid q_i,\, a_{<t}^{(\ell_j)} \right) \tag{2}$$

To estimate the global capability score $\mu_{\text{global}}^{(\ell_j)}$, we averaged the top 1% of the per-query scores for ℓ_j in the discovery set. This filters out noisy, out-of-domain queries and approximates each LLM's best-case performance. The corresponding $\sigma_{\text{global}}^{(\ell_j)}$ is computed over the same top-percentile subset, capturing the consistency of the response.

Pseudo-relevance labels. Since no ground-truth relevance judgments are available during training, we convert each continuous score $\mu_{q_i,a^{(\ell_j)}}$ into a discrete relevance grade. Scores below -2.0 are labeled 0 (low confidence), scores between -2.0 and -1.0 are labeled 1, and scores above -1.0 are labeled 2 (high confidence). These thresholds were chosen to produce roughly balanced label distributions over the discovery collection.

Zero-shot inference. At test time, no LLM-generated response is available for the incoming query q^*. We rank the LLM pool as follows: (1) embed q^* with SBERT; (2) for each candidate LLM ℓ, construct the feature vector $\mathbf{x}^* = [\mathbf{e}_{q^*};\ \text{id}(\ell);\ \mu_{\text{global}}^{(\ell)};\ \sigma_{\text{global}}^{(\ell)}]$ using pre-computed LLM statistics; (3) predict a score $s_\ell = f_{\text{LightGBM}}(\mathbf{x}^*)$; (4) rank all LLMs by s_ℓ in descending order. This is non-trivial because the ranker must generalise from query-specific generation signals observed during training to entirely unseen queries at test time, relying on the query embedding to bridge this gap.

Training configuration. We train with LightGBM's LambdaRank objective, optimising NDCG@{1,5,10}. Learning rate: 0.05; maximum leaves: 31; early stopping: 50 rounds; final rounds: 1,130.

2.2 Unsupervised Baselines

We evaluate two unsupervised baselines, both using BGE-M3 [2] to rank LLMs by cosine similarity between the query embedding and an LLM expertise profile, or response embedding.

The **tag-based baseline** derives each LLM's profile from ClueWeb22 topic tags. Documents in the discovery collection are clustered by their Bing topic tags using a union-find algorithm; each LLM is associated with the cluster covering the majority of its RAG documents, and the cluster tags serve as its expertise description.

The **response-based baseline** measures relevance by comparing the test query directly to the LLM's pre-generated response for that same query using BGE-M3 cosine similarity; each candidate LLM is scored by how semantically close its discovery response is to the incoming query. Although BGE-M3 provides multi-granularity retrieval and strong relevance matching, it is optimized for relevance estimation rather than answer quality. Two LLMs can therefore receive similar similarity scores when both produce on-topic outputs even if one answer is precise and well-grounded while the other is shallow, unsupported, or hallucinatory.

2.3 Post-Submission Extension: Topic-Aware Features

Query clustering. We encode all 14,940 discovery queries with SBERT and cluster them into K=100 groups via K-means over the 384-dimensional embeddings. Each cluster c_k corresponds to a latent topic derived purely from query semantics (e.g. a cluster of science-related queries, a cluster of legal queries). Let $C = \{c_1, \ldots, c_K\}$ denote the set of clusters with centroids $\{\mathbf{z}_1, \ldots, \mathbf{z}_K\}$.

Per-topic LLM statistics. For each (LLM ℓ_j, cluster c_k) pair, let $Q_k \subset Q$ be the set of discovery queries assigned to cluster c_k. We collect the per-query log-probability scores $\{\mu_{q_i,a^{(\ell_j)}} : q_i \in Q_k\}$ and apply the same top-percentile filtering used for the global statistics: we average the top 1% of scores to obtain a topic-specific confidence estimate $\mu_{c_k}^{(\ell_j)}$ and its standard deviation $\sigma_{c_k}^{(\ell_j)}$. We additionally compute three features from the raw responses within the cluster: the *refusal rate* $r_{c_k}^{(\ell_j)}$, defined as the fraction of responses containing refusal phrases (e.g. "I don't know", "as an AI") or shorter than 20 words; the mean response length $\bar{L}_{c_k}^{(\ell_j)}$; and the number of queries in the cluster n_{c_k}, which serves as a reliability indicator for the estimated statistics.

Inference and feature construction. At inference, a test query q^* is embedded with SBERT and assigned to its nearest cluster $c_{k^*} = \arg\min_k \|\mathbf{e}_{q^*} - \mathbf{z}_k\|_2$. We also compute the L2 distances to the three nearest centroids, $d_1 \leq d_2 \leq d_3$, which serve as soft-assignment features: a small d_1 with a large gap to d_2 indicates the query sits firmly within one topic, while similar distances suggest the query is ambiguous between clusters. The extended feature vector becomes:

$$\begin{aligned}\mathbf{x}_{ij}^{\text{topic}} = \Big[\, &\mathbf{e}_{q_i};\ \text{id}_{\ell_j};\ \mu_{\text{global}}^{(\ell_j)};\ \sigma_{\text{global}}^{(\ell_j)}; \\ &\underbrace{\mu_{c_{k^*}}^{(\ell_j)};\ \sigma_{c_{k^*}}^{(\ell_j)};\ r_{c_{k^*}}^{(\ell_j)};\ \bar{L}_{c_{k^*}}^{(\ell_j)};\ n_{c_{k^*}};}_{\text{per-topic statistics}}\ \ \underbrace{d_1;\ d_2;\ d_3}_{\text{soft assignment}}\ \Big] \in \mathbb{R}^{408}\end{aligned} \tag{3}$$

The first 387 dimensions are identical to Equation 3. The per-topic statistics (5 dims) vary with both the LLM and the query's cluster assignment, providing the query-dependent LLM signal that the global features lack. The soft-assignment distances (3 dims) vary only with the query and allow LightGBM to learn when per-topic statistics are reliable (query close to one centroid) versus noisy (query between clusters).

3 Results

Figure 1 shows the distribution of μ_{global} and σ_{global} across the LLM pool. The majority of models cluster in a moderate-confidence zone ($\mu_{\text{global}} \in [-1, 0]$, $\sigma_{\text{global}} \in [0, 0.5]$), suggesting the pool is dominated by models with broadly similar global confidence profiles.

Baseline performance The tag-based baseline achieves low performance with NDCG@10 of 0.013. We attribute this to a granularity mismatch: ClueWeb22 topic tags are coarse categorical labels (e.g. `Finance," "Health"), while effective LLM selection requires

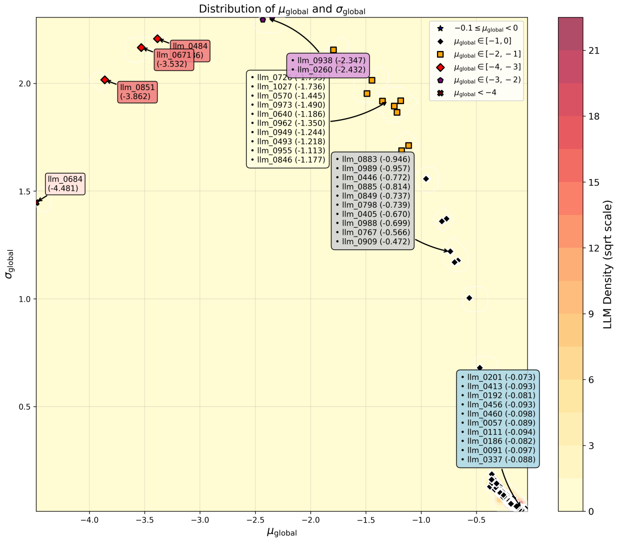

Figure 1: Distribution of LLMs global capability statistics μ_{global} and σ_{global}. A large proportion of LLMs exhibit μ_{global} values within the range [-1,0] and σ_{global} values within the range [0, 0.5]. A smaller cluster of high-confidence, low-variance models (top-left region) and a sparse set of negative-μ outliers (bottom-left) reflect models with systematically different generation behaviour. The top-10 models within each cluster are highlighted; these constitute the candidates most likely to be ranked first by the LightGBM system.

Table 1: Ranking performance on the TREC Million LLMs evaluation set. The first five rows are submitted runs; the last row (†) is a post-submission result evaluated on the 342-query development set. Unsupervised baselines (top) use BGE-M3 cosine similarity; supervised models (bottom) use LightGBM LambdaRank trained on log-probability pseudo-relevance labels.

Model	N@1	N@5	N@10	MRR
BGE-M3 + tags	.015	.013	.013	.047
BGE-M3 + responses	.223	.168	.195	.425
LTR (ID + μ,σ)	.376	.359	.359	.622
LTR-full	.374	.361	.364	.620
LTR (emb + ID)	.363	.370	.370	.624
LTR-full + topics†	**.424**	**.411**	**.405**	**.688**

N@k = NDCG@k. **BGE-M3 + tags**: cosine similarity between query and ClueWeb22 topic-tag profile. **BGE-M3 + responses**: cosine similarity between query and LLM's generated response. **LTR-full**: query emb. + LLM ID + μ_{global} + σ_{global} (387-dim). **LTR (ID + μ,σ)**: LLM ID + global stats only (no query embeddings). **LTR (emb + ID)**: query emb. + LLM ID only (no global stats). †**LTR-full + topics**: LTR-full extended with per-topic LLM statistics via K-means query clustering (408-dim); post-submission preliminary result on 342 dev queries.

fine-grained, query-level matching. Computing semantic similarity between a query and a bag-of-topic-tags representation fundamentally compares different levels of abstraction. The response-based baseline, with an NDCG@10 of 0.195, outperforms the model-generated-answer baseline by operating on the model-generated answers. However, it remains limited by the BGE-M3 reranker's ability to assess response quality solely on the basis of semantic similarity. BGE-M3 is trained to measure topical relevance, whether two texts discuss the same subject, not to assess whether a response is accurate, complete, or well-reasoned. An LLM that produces a fluent but shallow on-topic answer will receive a similar BGE-M3 score to one that provides a precise, well-reasoned response, since both are semantically close to the query. The supervised LightGBM approach offers an alternative by learning a structured ranking objective directly from log-likelihood signals, and query properties that capture the model's answer-generation confidence given a query.

Ablation: Query Embeddings and Global Statistics. Eliminating the query-embedding features produces only a marginal reduction in ranking effectiveness (NDCG@10: 0.364 vs. 0.359), indicating that global LLM-level statistics constitute a substantially stronger predictive signal than query-specific semantic matching. A plausible interpretation is that μ_{global} and σ_{global} encode a broad, query-agnostic hierarchy of model quality: certain LLMs consistently outperform others across a wide range of information needs, leaving limited additional variance for the query embeddings to explain.

Conversely, removing μ_{global} and σ_{global} while retaining the query embeddings and the LLM identifier yields an NDCG@10 of 0.370, slightly surpassing the full model. This outcome is less counterintuitive than it may initially appear. Because the pseudo-relevance labels used during training are derived from log-probability scores, LLMs that exhibit higher confidence tend to receive higher labels across many queries. As a result, LightGBM can infer each model's general competence directly from the identifier feature, rendering explicit global statistics as LTR feature partially redundant.

Taken together, these findings suggest that the system primarily learns a query-independent ordering of LLM quality: it tends to produce similar rankings regardless of whether the query pertains to biology, law, or history. Such behavior is reasonable if some document collections are uniformly stronger than others, but it also limits the system's ability to exploit cases in which a generally weaker LLM provides superior coverage for a specific domain. The topic-aware extension introduced in Section 2.3 mitigates this limitation by incorporating per-topic statistics that vary with the query's domain, thereby enabling domain-sensitive routing rather than relying on a fixed global hierarchy.

Topic-aware features. The final row of Table 1 presents a post-submission experiment that augments the global-only feature set with per-topic LLM statistics. Evaluated on the 342-query development set, this topic-aware variant yields higher NDCG scores across the board. The improvement indicates that LLM performance varies systematically across query domains and that a single global capability estimate obscures these differences. Incorporating topic-conditioned statistics allows the ranker to better capture domain-specific strengths in each model's RAG corpus, enabling more accurate query-adaptive routing.

4 Limitations and Future Work

Our approach has several limitations that suggest directions for future work.

Confidence as a proxy for correctness. The core assumption underlying our pseudo-relevance labels is that log-probability confidence corresponds to higher response quality. However, a model can produce fluent, high-probability responses that are factually incorrect. Recent work has shown that log-probability confidence can fundamentally diverge from predictive accuracy [12], particularly in out-of-distribution settings, such as our test queries that differ from discovery queries. Our approach may therefore systematically favour overconfident models over genuinely capable ones.

Topic clustering configuration. Our topic-aware extension (Section 2.3) uses a fixed K=100 clustering with a single percentile threshold, chosen without systematic tuning. The optimal number of clusters likely depends on the diversity of the LLM pool and the query distribution: too few clusters may group unrelated topics together, while too many may yield sparse clusters with unreliable per-LLM statistics. Ablating across different values of K, experimenting with alternative clustering methods (e.g. hierarchical clustering or supervised topic models), and testing the sensitivity to the top-percentile threshold would strengthen the conclusions. Additionally, the current evaluation reports point estimates without confidence intervals or significance tests. Given that the development set contains 342 queries and the differences between some LightGBM variants are small (e.g. NDCG@10 of 0.364 vs. 0.359), bootstrap confidence intervals and paired significance tests are needed to determine which observed differences reflect genuine improvements versus sampling variability. This applies both to the topic-aware extension and to the ablation comparisons in Table 1.

5 Related Works

Learning to rank and LLM routing. Supervised learning-to-rank (LTR) models are central to modern information retrieval [1, 5, 9], though their application to ranking *models* rather than documents is comparatively recent. We adopt LightGBM's [8] LambdaRank objective because it naturally handles grouped data (multiple LLMs per query) and incorporates heterogeneous features without preprocessing. Among approaches to LLM ranking, some rely on human-labelled preferences or LLM-as-a-judge frameworks [4, 14, 17], while others learn model–task embeddings from historical benchmark performance [16]. Our approach uses next-token log probabilities instead, requiring no evaluation data beyond what generation already produces.

Log probabilities as quality signals. Recent work suggests that token-level log probabilities correlate with semantic plausibility and contextual coherence [3, 7], and modern APIs expose these scores for confidence assessment [10, 13]. We exploit this by using mean log probabilities as pseudo-relevance labels for LTR training. However, the reliability of this signal has known limits: Hu et al. [6] show that perplexity does not reliably predict long-context understanding, and Veličković et al. [12] prove that confident models necessarily produce high-confidence incorrect outputs on some inputs. In our RAG setting, log-probability confidence is a more informative signal than in pure generation, because a model that has relevant documents in its context will generate more confidently than one that does not. That said, high confidence does not guarantee a correct answer since a model may still hallucinate or misread its retrieved documents.

References

[1] Chris J.C. Burges. 2010. *From RankNet to LambdaRank to LambdaMART: An Overview.* Technical Report MSR-TR-2010-82. https://www.microsoft.com/en-us/research/publication/from-ranknet-to-lambdarank-to-lambdamart-an-overview/

[2] Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. BGE M3-Embedding: Multi-Lingual, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation. arXiv:2402.03216 [cs.CL]

[3] Hakaze Cho, Yoshihiro Sakai, Kenshiro Tanaka, Mariko Kato, and Naoya Inoue. 2025. Understanding Token Probability Encoding in Output Embeddings. In *Proceedings of the 31st International Conference on Computational Linguistics*, Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert (Eds.). Association for Computational Linguistics, Abu Dhabi, UAE, 10618–10633. https://aclanthology.org/2025.coling-main.708/

[4] Amit Dhurandhar, Rahul Nair, Moninder Singh, Elizabeth Daly, and Karthikeyan Natesan Ramamurthy. 2024. Ranking Large Language Models without Ground Truth. arXiv:2402.14860 [cs.CL] https://arxiv.org/abs/2402.14860

[5] Jiafeng Guo, Yixing Fan, Liang Pang, Liu Yang, Qingyao Ai, Hamed Zamani, Chen Wu, W. Bruce Croft, and Xueqi Cheng. 2020. A Deep Look into neural ranking models for information retrieval. *Information Processing Management* 57, 6 (2020), 102067. doi:10.1016/j.ipm.2019.102067

[6] Yutong Hu, Quzhe Huang, Mingxu Tao, Chen Zhang, and Yansong Feng. 2024. Can Perplexity Reflect Large Language Model's Ability in Long Text Understanding? arXiv:2405.06105 [cs.CL] https://arxiv.org/abs/2405.06105

[7] Carina Kauf, Emmanuele Chersoni, Alessandro Lenci, Evelina Fedorenko, and Anna A Ivanova. 2024. Log Probabilities Are a Reliable Estimate of Semantic Plausibility in Base and Instruction-Tuned Language Models. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, Yonatan Belinkov, Najoung Kim, Jaap Jumelet, Hosein Mohebbi, Aaron Mueller, and Hanjie Chen (Eds.). Association for Computational Linguistics, Miami, Florida, US, 263–277. doi:10.18653/v1/2024.blackboxnlp-1.18

[8] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: a highly efficient gradient boosting decision tree *(NIPS'17)*. Curran Associates Inc., Red Hook, NY, USA, 3149–3157.

[9] Tie-Yan Liu. 2010. Learning to rank for information retrieval. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Geneva, Switzerland) *(SIGIR '10)*. Association for Computing Machinery, New York, NY, USA, 904. doi:10.1145/1835449.1835676

[10] OpenAI. 2025. Using Logprobs — OpenAI Cookbook. https://cookbook.openai.com/examples/using_logprobs. Accessed: 2025-10-08.

[11] Gemini Team. 2023. Gemini: A Family of Highly Capable Multimodal Models. *arXiv e-prints*, Article arXiv:2312.11805 (Dec. 2023), arXiv:2312.11805 pages. arXiv:2312.11805 [cs.CL] doi:10.48550/arXiv.2312.11805

[12] Petar Veličković, Federico Barbero, Christos Perivolaropoulos, Simon Osindero, and Razvan Pascanu. 2026. Perplexity Cannot Always Tell Right from Wrong. arXiv:2601.22950 [cs.LG] https://arxiv.org/abs/2601.22950

[13] Oscar Wahltinez. 2024. Analyzing the next token probabilities in large language models. https://responsible-ai-developers.googleblog.com/2024/03/analyzing-next-token-probabilities-in-large-language-models.html Posted on the Responsible AI for Developers Blog. Accessed: 2025-11-10.

[14] Yikun Wang, Rui Zheng, Haoming Li, Qi Zhang, Tao Gui, and Fei Liu. 2023. Rescue: Ranking LLM Responses with Partial Ordering to Improve Response Generation. In *Annual Meeting of the Association for Computational Linguistics*. https://api.semanticscholar.org/CorpusID:265213413

[15] Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. 2023. BloombergGPT: A Large Language Model for Finance. arXiv:2303.17564 [cs.LG] https://arxiv.org/abs/2303.17564

[16] Yi-Kai Zhang, Ting-Ji Huang, Yao-Xiang Ding, De-Chuan Zhan, and Han-Jia Ye. 2023. Model Spider: learning to rank pre-trained models efficiently. In *Proceedings of the 37th International Conference on Neural Information Processing Systems* (New Orleans, LA, USA) *(NIPS '23)*. Curran Associates Inc., Red Hook, NY, USA, Article 604, 28 pages.

[17] Irune Zubiaga, Aitor Soroa, and Rodrigo Agerri. 2024. A LLM-based Ranking Method for the Evaluation of Automatic Counter-Narrative Generation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (Eds.). Association for Computational Linguistics, Miami, Florida, USA, 9572–9585. doi:10.18653/v1/2024.findings-emnlp.559

Received 9 March 2026; revised 9 March 2026