Finding the Right LLM: Expert Retrieval for Model Ranking in the TREC 2025 Million LLM Track

Reyyan Yeniterzi [*]
GenAIus Technologies
reyyan@genaius.tech

Suveyda Yeniterzi
GenAIus Technologies
suveyda@genaius.tech

February 15, 2026

Abstract

We address the Million LLM ranking task by formulating it as an expert retrieval problem and adapting established Information Retrieval techniques to estimate the expertise of large language models. Our approach centers on two families of methods: a profile-based strategy that aggregates all query–response pairs from each LLM into a unified representation, and document-based strategies that operate either at the response level or at the query level. Before applying these models, we introduce a two-stage data filtering pipeline to remove uninformative and low-confidence responses, yielding a cleaner signal for expertise estimation. Experimental results on the development set show that response-based aggregation provides the most fine-grained and reliable ranking of LLMs, outperforming both profile-based and question-based variants. Guided by these findings, we prepared five submissions combining different retrieval, filtering, and aggregation configurations, including a re-ranking variant using naver-splade-v3. Our study demonstrates that classical expert retrieval methods, when adapted appropriately, can effectively model and rank LLM expertise.

1 Introduction

The goal of the Million LLM [3] is to evaluate and rank large language models based on their predicted expertise for a given query. In this task, we are provided with a user query and a list of LLMs and expected to determine which models are most likely to deliver high-quality, accurate, and contextually appropriate responses.

In this paper, we frame the LLM retrieval and ranking problem as an instance of expert retrieval, a well-established task in Information Retrieval. Traditionally, expert retrieval aims to identify the most relevant expert for a given topic based on a collection of documents associated with different authors. This approach has been used extensively in Community Question Answering (CQA) systems, where the goal is to route incoming questions to the users most likely to provide accurate and informative answers [8]. Inspired by this paradigm, we treat each LLM as an "expert" and rank them according to their predicted ability to answer a given query effectively.

2 Approach

Expert retrieval has been extensively studied, particularly following the introduction of the TREC Enterprise track in 2005 [2, 6], which standardized the task of ranking experts for specific topics using corporate document collections. This led to the development of several families of expert-finding algorithms. Profile-based methods build a profile for each candidate and rank them using traditional text retrieval. Document-based methods first retrieve topic-relevant documents and then aggregate

[*] Both authors contributed to this paper equally.

	# Responses
Before Filtering	16,908,450
After Initial Filtering	2,833,639
After Additional Filtering	1,238,631

Table 1: Number of LLM responses before and after filtering

evidence for each associated candidate. Graph-based approaches leverage relationships among candidates and documents to infer expertise. Learning-based approaches combine multiple features and learn probabilistic models from training data to estimate expertise [7].

In this paper, we focus exclusively on document-based and profile-based methods, given the size of the dataset and the limited availability of detailed information about the LLMs. Before applying these approaches, we first perform a data filtering step to obtain a cleaner and more reliable dataset.

2.1 Data Filtering

The provided LLM discovery dataset contains 14,950 queries paired with responses from 1,131 different LLMs. However, for some queries, certain models returned "No result found" directly. This behavior clearly signals that the model is not a suitable fit for that particular query. We therefore used this as an initial filtering criteria to eliminate all such cases. Applying this filter removed approximately 83% of the LLM responses.

We also observed cases where LLM responses began with phrases such as "I don't have knowledge about ..." or "I don't have enough information ...". Such wording indicates low confidence and often signals that the model may be hallucinating or unable to provide a reliable answer. Applying this additional filtering step removed another 10% of the data, leaving only about 7% of the original dataset. The exact number of LLM responses before and after filtering is provided in Table 1.

2.2 Retrieval

Profile-based approaches build a textual representation (or "profile") of each candidate using all their associated documents and then rank these profiles by their relevance to a query, effectively turning expert retrieval into a document retrieval task. In contrast, document-based approaches first retrieve the documents most relevant to the query and then identify expert candidates based on their association with those retrieved documents, rather than modeling each candidate directly. Overall, profile-based methods aggregate all evidence for a candidate before retrieval by constructing a unified profile, whereas document-based methods aggregate evidence after retrieval by first selecting relevant documents and then inferring the associated candidates.

2.2.1 Profile-based

In the profile-based approach [1], for each of the 1,131 LLMs we aggregate all query–response pairs into a single document representation (i.e., the "LLM profile"). These profile documents are then indexed into a retrieval engine, yielding an index of size 1,131. At runtime, an incoming search query is executed against this indexed collection, and the retrieval engine's relevance scores are directly interpreted as expertise rankings for the corresponding LLMs.

For clarity in our submissions, we refer to this method as the LLM-based approach. For retrieval and ranking, we employ BM25 as the baseline scoring function and additionally apply RM3 for relevance feedback–based query expansion. For each query, the top 100 LLMs returned by this ranking pipeline were selected and submitted.

2.2.2 Document-based

In the document-based approach, we evaluate two alternative document representations.

In the response-based representation, each LLM response together with its corresponding query is treated as an individual document. Under this setup, the filtered dataset yields approximately 2.8 million documents, while the additionally filtered dataset yields roughly 1.2 million documents.

In the question-based representation, a single document is constructed for each query, resulting in an index of 14,950 documents for both filtered conditions. In this formulation, LLMs are treated as binary indicators—representing whether a given model produced an answer for that query, which aligns naturally with our earlier data filtering step.

Beyond the choice of document representation, different aggregation strategies can also be applied. In this work, we adopt the voting models introduced by Macdonald and Ounis [5], which adapt traditional data fusion techniques for the expert retrieval setting. These models first use a standard text retrieval system to obtain documents relevant to the query; each candidate associated with a retrieved document then receives a (potentially weighted) vote. Candidates are ultimately ranked according to the total number of votes they accumulate.

Macdonald and Ounis use Votes as a baseline, where every retrieved document linked to a candidate contributes an equal vote toward that candidate's expertise score. We also apply the CombSUM method [33] (referring it as SumScore), in which a candidate's relevance score is computed as the sum of the retrieval scores of all associated ranked documents.

Similar to the profile-based approach, the document-based methods also rely on BM25 for initial retrieval and RM3 for relevance feedback and query expansion. In the response-based representation, we retrieve the top 1,000 documents for each query before applying aggregation. In the question-based representation, we retrieve the top 100 documents per query, which are then passed to the aggregation models.

3 Experiments

A development set of 342 queries with corresponding relevance assessments was provided and used for initial comparisons across different approaches. The track specifies nDCG@10 (normalized discounted cumulative gain) and MRR (mean reciprocal rank) as the primary evaluation metrics, and we report our results using these metrics accordingly.

The results of the different data filtering and retrieval strategies are summarized in Table 2. As shown, the LLM-based approach [1] performs noticeably worse than the document-based methods, such as the question-based and response-based variants. This outcome is expected, as document-based approaches operate directly on the top-ranked documents for each query and therefore provide a more fine-grained estimate of topic-specific expertise.

A similar pattern appears within the document-based methods when comparing the question-based and response-based variants. The response-based approach operates at a more fine-grained level by focusing on the specific responses generated by each LLM, whereas the question-based approach models expertise at the query level more broadly. As expected, this more granular representation enables the response-based method to outperform the question-based one.

When comparing aggregation strategies, neither Vote nor SumScore emerges as a definitive winner. However, because the Vote model is more prone to producing ties, SumScore was selected for our final submissions. Similarly, neither filtering strategy consistently outperforms the other across metrics, indicating that both filtering settings yield comparable results. For the submissions, we placed slightly more emphasis on the initial filtering setup to avoid inadvertently discarding potentially strong LLMs solely due to low confidence signals in their responses.

4 Submissions

Using the described approaches the following 5 submissions were done:

- **llm-f-100**: The LLM-based approach was applied on the initially filtered dataset using BM25, and the top 100 LLMs were retrieved for each query.
- **q-f-rm3-100-ss**: The question-based approach was applied to the initially filtered dataset using BM25 with RM3 query expansion. For each query, the top 100 question documents were retrieved and aggregated using the SumScore method.

[1] In the LLM-based approach, applying RM3 resulted in lower evaluation scores, so it was not used in the final configuration.

Approach	Filtering	Post-Aggregation	nDCG@10	MRR
LLM-based	Initial	-	0.3689	0.6706
	Additional	-	0.3717	0.6686
Question-based	Initial	Vote	0.4184	0.7011
		SumScore	0.4180	0.6998
	Additional	Vote	0.4151	0.6624
		SumScore	0.4158	0.6679
Response-based	Initial	Vote	0.4456	0.7536
		SumScore	0.4507	0.7520
	Additional	Vote	0.4355	0.7540
		SumScore	0.4424	0.7527

Table 2: Results from the development set

Run	Approach	Filtering	Re-Rank	Post-Aggregation	nDCG@10
llm-f-100	LLM-based	Initial	No	-	0.10596
q-f-rm3-100-ss	Question-based	Initial	No	SumScore	0.12030
r-f-rm3-1000-ss	Response-based	Initial	No	SumScore	0.15446
r-f-rm3-1000-rr-ss	Response-based	Initial	Yes	SumScore	0.16081
r-ef-rm3-1000-ss	Response-based	Additional	No	SumScore	0.16801

Table 3: Results from the test set

- **r-f-rm3-1000-ss**: The response-based approach was applied to the initially filtered dataset using BM25 with RM3 query expansion. For each query, the top 1000 responses were retrieved and aggregated using the SumScore method.

- **r-ef-rm3-1000-ss**: The response-based approach was applied to the additionally (extra) filtered dataset using BM25 with RM3 query expansion. For each query, the top 1000 responses were retrieved and aggregated using the SumScore method.

- **r-f-rm3-1000-rr-ss**: The response-based approach was applied to the initially filtered dataset using BM25 with RM3 query expansion. These retrieved responses were re-ranked with naver-splade-v3 [4]. For each query, the top 1000 responses were aggregated using the SumScore method.

The performance of the submitted runs is summarized in Table 3. In the test experiments, we primarily evaluate variations of the approach component, while the filtering and post-aggregation components are largely kept consistent across runs. The results of the first three runs (top three rows in Table 3) indicate that the relative ranking of approaches observed on the development set is largely preserved on the test set as well. Among the evaluated methods, the LLM-based approach yields the lowest performance. The question-based approach performs better, while the response-based approach achieves the highest overall effectiveness.

Within the response-based approach, reranking provides a modest performance gain; however, a more substantial improvement of 8.7% is achieved through the incorporation of the additional filtering step. Notably, this filtering component was less effective on the development set but proved significantly beneficial on the test set.

5 Conclusion

In this work, we addressed the Million LLM ranking task by formulating it as an expert retrieval problem and adapting established Information Retrieval techniques to estimate the expertise of LLMs. We evaluated both profile-based and document-based approaches, supported by a two-stage filtering pipeline designed to remove uninformative and low-confidence responses. Our experiments show that document-based methods consistently outperform profile-based ones, with response-level representations providing the most fine-grained and reliable expertise signals. We also examined alternative aggregation and filtering configurations, finding that SumScore aggregation and targeted response filtering yield robust and effective rankings.

Overall, our three response-based runs outperform both our other submitted runs and those submitted by other teams, ranking 3rd, 4th, and 5th among the 19 submitted runs. These findings demonstrate that classical expert retrieval frameworks, when carefully adapted, remain highly effective for modeling and ranking LLM expertise. More broadly, our results highlight the value of response-level evidence and retrieval-driven evaluation for scalable LLM discovery and selection.

References

[1] Krisztian Balog, Leif Azzopardi, and Maarten De Rijke. Formal models for expert finding in enterprise corpora. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 43–50, 2006.

[2] Nick Craswell, Arjen P De Vries, and Ian Soboroff. Overview of the TREC 2005 enterprise track. In *Trec*, volume 5, pages 1–7, 2005.

[3] Evangelos Kanoulas, Panagiotis Eustratiadis, Jamie Callan, Mark Sanderson, Yongkang Li, Jingfen Qiao, Gabrielle Poerwawinata, and Vaishali Pal. Overview of the TREC 2025 million large language models track. In *Proceedings of the 34th Text REtrieval Conference (TREC 2025)*, 2025.

[4] Carlos Lassance, Hervé Déjean, Thibault Formal, and Stéphane Clinchant. Splade-v3: New baselines for splade, 2024.

[5] Craig Macdonald and Iadh Ounis. Voting for candidates: adapting data fusion techniques for an expert search task. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 387–396, 2006.

[6] Ian Soboroff, Arjen P de Vries, Nick Craswell, et al. Overview of the TREC 2006 enterprise track. In *TREC*, volume 6, pages 1–20, 2006.

[7] Reyyan Yeniterzi. *Effective and efficient approaches to retrieving and using expertise in social media*. PhD thesis, Ph. D. dissertation, 2015.

[8] Reyyan Yeniterzi and Jamie Callan. Moving from static to dynamic modeling of expertise for question routing in CQA sites. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 9, pages 702–705, 2015.