Probe, Don't Prompt: A Hidden-State Probe for Metadata Filtering in Multi-Meta-RAG

Mykhailo Poliakov and Nadiya Shvai
National University of Kyiv-Mohyla Academy
Kyiv, Ukraine
mykhailo.poliakov@ukma.edu.ua, n.shvay@ukma.edu.ua

Abstract

Multi-Meta-RAG improves retrieval for multi-hop question answering by filtering a vector store on metadata (the news *source*) that it extracts from each query by prompting `gpt-3.5-turbo`. We show this proprietary, free-form extractor can be replaced by a local, deterministic probe trained on the hidden states of a *small* open-source language model. On all 2556 MultiHop-RAG queries the probe reaches **90.9%** set-exact accuracy against 88.0% for a model-free substring baseline and 80.9% for GPT-3.5, a margin that comes entirely from null queries, on which GPT-3.5 never abstains; on non-null queries all three stay within about a point. Because the probe's output space is exactly the fixed 49-source vocabulary, it cannot drift outside the allow-list as the prompted model does. Three design choices make it work: selecting a *shallow* layer, *mean* pooling, and *class-imbalance-aware* multi-label training over the long tail of sources. A 135M-parameter model lands within $\sim$1.5 points of a 1.5B one, so the filter is cheap to output: a partial forward pass through the first few layers plus one linear head, with no API. The code is available at https://github.com/mxpoliakov/Multi-Meta-RAG.

Index Terms

retrieval-augmented generation, probing classifiers, metadata filtering, multi-hop question answering, small language models

I. Introduction

Retrieval-augmented generation (RAG) [1] grounds a language model on documents fetched from an external store, but struggles on *multi-hop* queries that must assemble evidence from several documents at once. The MultiHop-RAG benchmark [2] makes this failure mode concrete: 2556 news queries whose answers require combining two to four articles. Multi-Meta-RAG [3] improves multi-hop retrieval on this benchmark by filtering the vector store on metadata before similarity search, specifically the news *source* (e.g. *The Verge*, *TechCrunch*) named in the query. The filter is obtained by prompting `gpt-3.5-turbo` to read the source(s) out of the query text.

This extractor has two costs. First, it is a proprietary API call on every query, adding token cost and network latency to the retrieval path. Second, it *drifts*: despite a prompt that fixes a 49-source allow-list, GPT-3.5 emits 128 distinct source strings across the dataset, 79 of them off-list, so many of its filters reference labels the vector store never indexed.

Our starting observation is that the source named in a query is almost always present in the query *surface form*: the source name appears verbatim in the query in 95.4% of gold query–source pairs. Extraction therefore need not be a generative act. The information required is already linearly available in the model's own internal representation. We therefore train a lightweight *probe* [4] on the hidden states of a *small* open-source model, motivated by evidence that intermediate representations carry task-relevant signal for retrieval [5]. The probe is a fixed-vocabulary multi-label classifier: its output space is exactly the 49 sources, so it is structurally incapable of the allow-list drift the prompted baseline exhibits.

Our main contributions are as follows.

1) A drop-in, fixed-vocabulary probe that replaces GPT-3.5 source extraction in Multi-Meta-RAG, with no API cost, no allow-list drift, and deterministic output.
2) A set-exact head-to-head on all 2556 MultiHop-RAG queries against both GPT-3.5 and a model-free substring baseline: the probe scores **90.9%** overall versus 88.0% (substring) and 80.9% (GPT-3.5), with its margin coming entirely from null-query abstention; on non-null queries it stays within about a point of both.
3) Three findings that make it work: *shallow* layers win (not the middle or last), *mean* pooling beats last-token pooling, and *class-imbalance-aware* multi-label training matters for the long tail of rare sources.
4) A size study from 135M to 1.5B parameters showing that a 135M model stays within $\sim$1.5 points of a 1.5B one, so the filter is cheap to output.

The rest of the paper is organized as follows. We first review related work. We then describe the probe and its three design choices. Next, we report the head-to-head against GPT-3.5 and the layer, pooling, and size analyses. Finally, we discuss limitations and future work. The code is available at https://github.com/mxpoliakov/Multi-Meta-RAG.

II. Related Work

A. Probing Classifiers

Shallow classifiers trained on frozen representations read out what those representations encode [4], [6]; see [7] for a survey. Rigorous probing controls for what is genuinely *learned* versus recoverable from surface form [8]. Our model-free string-match baseline plays exactly that control role here. Layer-wise probing shows that information is localized by depth [9]. We use probing not for interpretability but as a *deployable component* of a retrieval pipeline.

B. Which Layer Carries the Signal

Intermediate layers often beat the last layer for downstream use [10], including specifically for multi-hop retrieval [5]. We test this directly and report a *contrast*: for near-lexical source identity the *shallowest* layers win, not the middle ones, because the target attribute is present in the query surface form rather than in deep semantics.

C. RAG and Metadata Filtering

RAG [1] over dense retrieval [11] underperforms on multi-hop queries [2]; metadata-filtered retrieval with LLM-extracted filters addresses this [3]. Rather than *distilling* [12] the prompted extractor by training on its outputs, we replace it with a small local model supervised directly on gold evidence sources. Capable small open-source models [13], [14] and the finding that decoder hidden states are strong text features [15] make this practical.

III. Method

A. Task and Data

We use all 2556 MultiHop-RAG queries: 856 comparison, 816 inference, 583 temporal, and 301 null queries. The label space is exactly the **49** sources of the Multi-Meta-RAG prompt allow-list. The task is multi-label: a query references one to four sources (1→598, 2→1340, 3→300, 4→17 queries). Let $\mathcal{S} = \{s_1, \dots, s_K\}$ be the fixed vocabulary of $K = 49$ sources. We define the gold source set of each query q automatically from its evidence,

$$S_q^\star = \mathrm{set}\big(\text{evidence sources of } q\big), \qquad S_q^\star = \varnothing \ \text{ for null queries}, \tag{1}$$

mapping the 301 **null queries to the empty set** since they carry no evidence, and encode it as a multi-hot vector $y_q \in \{0,1\}^K$ with $y_{qc} = \mathbf{1}[s_c \in S_q^\star]$. The evaluation metric is the **set-exact hit** accuracy over the $N = 2556$ queries,

$$\mathrm{Acc} = \frac{1}{N} \sum_{q=1}^{N} \mathbf{1}\left[\hat{S}_q = S_q^\star\right], \tag{2}$$

the fraction whose predicted set $\hat{S}_q$ equals the gold set exactly. Date operators are ignored, as the baseline filter file uses only $\$in$ on `source`.

B. Pipeline

Figure 1 shows the pipeline. For a query tokenised into T_q tokens we run one forward pass through a small open-source model with `output_hidden_states=True`, giving hidden states $h_\ell^{(t)} \in \mathbb{R}^d$ at layer ℓ for token t. We cache, for every layer, two pooled representations,

$$h_\ell^{\mathrm{mean}} = \frac{1}{T_q} \sum_{t=1}^{T_q} h_\ell^{(t)}, \qquad h_\ell^{\mathrm{last}} = h_\ell^{(T_q)}, \tag{3}$$

as a $[N, n_\ell, d]$ tensor, so the layer/pooling sweep is cheap to run offline. On the pooled state h_ℓ of a *single* chosen layer, a 49-way multi-label head scores each source c with an independent logistic unit and one global decision threshold τ,

$$p_{qc} = \sigma\big(w_c^\top h_\ell + b_c\big), \qquad \hat{S}_q = \{\, s_c : p_{qc} \geq \tau \,\}, \tag{4}$$

where σ is the logistic sigmoid and features are standardised per dimension. At inference the predicted set $\hat{S}_q$ is written back as a `source.$in` filter, a drop-in for the field the prompted extractor produced.

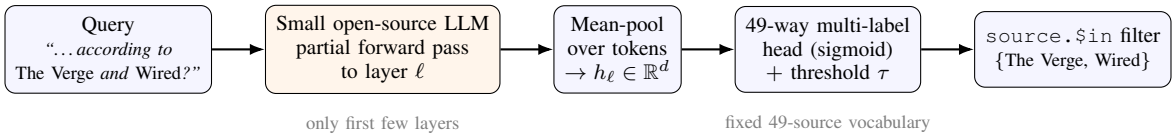

Fig. 1. The probe pipeline. A single partial forward pass through the first few layers of a small open-source model produces a pooled hidden state; a 49-way multi-label head reads the source set out of it and writes a `source.$in` filter. The output space is exactly the 49 allow-list sources, so it cannot drift.

TABLE I
SELECTED PROBE CONFIGURATION AND CROSS-VALIDATED F1 PER MODEL.

Model	Params	Hidden states × dim	Pooling	Layer	Sweep micro / macro F1	MLP micro / macro F1
Qwen2.5-1.5B	1.5B	29 × 1536	mean	2 / 28	**0.970 / 0.892**	0.954 / 0.830
SmolLM2-360M	360M	33 × 960	mean	4 / 32	0.969 / 0.885	0.962 / 0.840
Qwen2.5-0.5B	0.5B	25 × 896	mean	2 / 24	0.967 / 0.891	0.962 / 0.848
SmolLM2-135M	135M	31 × 576	mean	1 / 30	0.963 / 0.881	0.964 / 0.867

C. What the Probe Depends On

Three factors determine the probe's accuracy: which layer it reads, how the token states are pooled, and how class imbalance is handled. We fix each by a cross-validated sweep or by standard multi-label practice rather than by hand. A configuration that gets all three wrong (an arbitrary middle layer, last-token pooling, and unweighted training) does not beat the baseline.

Layer. We sweep every hidden-state layer under both poolings with iterative-stratified 5-fold cross-validation and select the layer with the best out-of-fold micro F1. The intermediate-layer literature [5], [10] would predict a middle-layer peak, but the sweep selects a shallow layer (index 1–4) for all four models (Table I), consistent with source identity being a near-lexical attribute.

Pooling. We compare mean against last-token pooling on the same swept layers. Mean pooling gives the higher out-of-fold F1 for all four models and at nearly every depth (Fig. 2), consistent with the source name appearing anywhere in the query rather than only at its end.

Class imbalance. The source distribution is long-tailed: *TechCrunch* appears 1190 times, while 26 of the 49 sources appear fewer than 20 times and the rarest appears twice. We therefore train each logistic unit by minimising a class-balanced binary cross-entropy,

$$\mathcal{L}_c = -\sum_{q=1}^{N} \Big[\alpha_c^+ \, y_{qc} \log p_{qc} + \alpha_c^- \, (1 - y_{qc}) \log(1 - p_{qc}) \Big], \tag{5}$$

with weights inversely proportional to class frequency, $\alpha_c^+ = N/(2n_c)$ and $\alpha_c^- = N/\big(2(N - n_c)\big)$, where $n_c = \sum_q y_{qc}$ is the support of source c, so the long tail is upweighted. We use iterative-stratified multi-label splits [16] so that rare sources occur in every fold, and report both micro and macro F1,

$$\mathrm{F1}_{\mathrm{micro}} = \frac{2\sum_c \mathrm{TP}_c}{2\sum_c \mathrm{TP}_c + \sum_c \mathrm{FP}_c + \sum_c \mathrm{FN}_c}, \quad \mathrm{F1}_{\mathrm{macro}} = \frac{1}{K}\sum_{c=1}^{K} \mathrm{F1}_c, \tag{6}$$

since macro F1 weights every source equally and so keeps the rare-source tail visible.

D. Fixed-Vocabulary Property and Baselines

Because the output space is exactly the 49 labels, the probe *structurally* cannot drift outside the allow-list, unlike the generative baseline that emits ∼128 distinct strings. We tune one global threshold τ on out-of-fold probabilities and produce all predictions out-of-fold, so no query is ever scored by a model that trained on it; the layer and threshold are themselves selected on the same out-of-fold scores, so the reported numbers carry the mild optimism of that selection. We also fit a 1-hidden-layer MLP as a capacity check. We compare against two baselines: (a) the GPT-3.5 filter file shipped with Multi-Meta-RAG, and (b) a model-free **substring match** over the 49 source names (case-insensitive containment), a strong reference at micro 0.957 / macro 0.942 that doubles as the probing control task.

IV. RESULTS

We report four findings: the probe beats GPT-3.5 overall but only matches a substring baseline on non-null queries, the overall gain is concentrated in null queries, the three design choices are what carry it, and model size barely matters. All numbers are taken verbatim from our experiment logs.

TABLE II
SET-EXACT HIT ACCURACY BY QUESTION TYPE AND OVERALL, PROBE (OUT-OF-FOLD) VS. A MODEL-FREE SUBSTRING MATCH AND GPT-3.5.

question type	Qwen2.5-1.5B	SmolLM2-360M	Qwen2.5-0.5B	SmolLM2-135M	substring	GPT-3.5
comparison	94.4%	94.5%	93.2%	92.4%	97.0%	**98.2%**
inference	91.1%	**91.9%**	90.6%	90.9%	90.2%	91.8%
temporal	**84.2%**	81.8%	82.7%	82.8%	82.7%	81.8%
null query	**93.7%**	92.0%	91.4%	89.0%	66.8%	0.0%
ALL (with null)	**90.9%**	90.5%	89.7%	89.4%	88.0%	80.9%
ALL (without null)	90.6%	90.3%	89.5%	89.4%	90.8%	**91.7%**

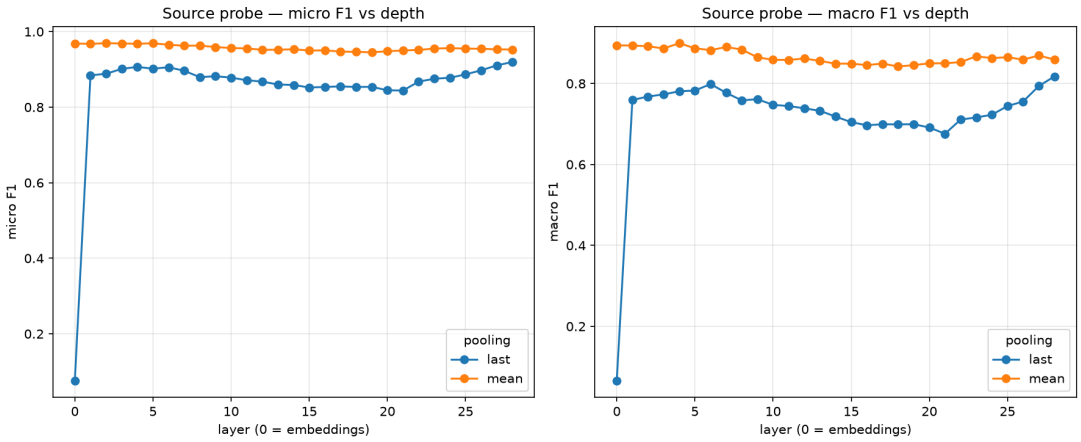

Fig. 2. Probe micro (left) and macro (right) F1 versus layer for Qwen2.5-1.5B, under both poolings. Mean pooling (orange) dominates last-token (blue) almost everywhere; last-token rises to a shallow local peak, sags through the middle, and recovers only at the final layer, while mean stays high and nearly flat, so neither favours the middle layer the deep-semantics regime would predict. Layer 0 is the embedding layer.

A. *Probe Configuration (Table I)*

Table I reports the selected configuration per model, chosen by out-of-fold micro F1 over the class-weighted logistic-regression sweep; the Layer column gives the hidden-state index (0 = embeddings) over the model's transformer-layer count. *Mean pooling and a very shallow layer win every time*: the best hidden-state index is 1–4 out of 24–32 transformer layers. The logistic-regression sweep reaches micro F1 0.963–0.970 and approaches the strong string-match reference (0.957 micro), and the MLP does not improve on it, so capacity is not the bottleneck. Macro F1 (0.881–0.892) trails micro, locating the residual difficulty in the rarest sources.

B. *Head-to-Head (Table II)*

The main comparison spans all 2556 queries against two baselines, GPT-3.5 and a model-free substring match. *Overall with null queries, the probe scores 90.9%, against the substring baseline's 88.0% and GPT-3.5's 80.9%. Overall without null queries the three are within about a point: 90.6% (probe), 90.8% (substring), 91.7% (GPT-3.5).* The probe's overall margin is the *null query* row: it predicts the empty set 93.7% of the time, whereas GPT-3.5 always extracts the sources it sees named in the text and so scores 0/301, and even the substring baseline over-fires on the third of null queries that name a source (66.8%). Null abstention is thus the probe's one real edge; on non-null queries it matches but does not beat either baseline.

C. *Analysis*

The overall margin comes from the null queries. On non-null queries all three methods are close (Table II): GPT-3.5 is higher on comparison (98.2% vs. 94.4%), the probe is higher on temporal (84.2% vs. 81.8%), and the substring baseline (90.8%) is on par with the probe (90.6%). The probe's advantage is therefore not more accurate source reading but that it can predict the empty set on null queries, which neither a generator prompted to read sources from the query text (0/301) nor a substring matcher (66.8%) can do.

The selected layer is shallow in every case, index 1–4 of the 24–32 transformer layers (Table I, Fig. 2), rather than the middle layer reported as strongest for multi-hop retrieval [5] and for general downstream use [10]. A likely reason is that the source name appears verbatim in the query in 95.4% of gold query–source pairs, so source identity is a near-lexical attribute that a partial forward pass already exposes; the tasks in that prior work depend on deeper semantics.

Model size has little effect on this task. SmolLM2-360M reaches 90.5% against 90.9% for the $4\times$ larger Qwen2.5-1.5B, and the 135M model stays within $\sim$1.5 points at 89.4% (Table II).

The probe runs locally as a partial forward pass through the first few layers ($\sim$3–13% of the layers, depending on the model) followed by one linear head; the selected depths are 2/28, 4/32, 2/24, and 1/30. We state this cost architecturally—the head reads a layer the truncated pass has already computed—rather than as measured latency; the experiments themselves cached all layers from full forward passes. The baseline instead issues one paid GPT-3.5 call per query and shows the allow-list drift noted earlier.

V. Conclusions

A shallow, mean-pooled, imbalance-aware probe on a small open-source model is a drop-in replacement for the prompted GPT-3.5 metadata extractor in Multi-Meta-RAG [3]: $+10$ points set-exact overall, with the margin coming from null-query abstention (it matches but does not beat a substring baseline on non-null queries), no API cost, and no allow-list drift. The broader lesson for probing-in-the-loop systems is to *read the cheapest layer that suffices*. For near-lexical attributes that is a shallow layer, a counterpoint to the middle-layer literature.

Limitations. The study is a single dataset and domain (news); the source attribute is near-lexical, so a string-match baseline is already strong; and macro F1 trails micro on the rarest sources.

Future work. (1) Learned date `$gt`/`$lt` operators, which are absent from the baseline filters today; (2) a lexical-$\cup$-probe hybrid with a null gate to push non-null accuracy past string match; (3) focal loss [17] or per-class thresholds for the rare-source macro-F1 tail; and (4) measuring the end-to-end retrieval impact (MRR@10, Hits@k) of swapping the extractor, not just extraction accuracy.

References

[1] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive NLP tasks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 9459–9474, 2020. [Online]. Available: https://dl.acm.org/doi/abs/10.5555/3495724.3496517

[2] Y. Tang and Y. Yang, "MultiHop-RAG: Benchmarking retrieval-augmented generation for multi-hop queries," in *Proceedings of the First Conference on Language Modeling (COLM)*, 2024, arXiv:2401.15391. [Online]. Available: https://arxiv.org/abs/2401.15391

[3] M. Poliakov and N. Shvai, "Multi-Meta-RAG: Improving RAG for multi-hop queries using database filtering with LLM-extracted metadata," in *Information and Communication Technologies in Education, Research, and Industrial Applications (ICTERI 2024)*, ser. Communications in Computer and Information Science, vol. 2359. Springer, 2024, pp. 334–342, arXiv:2406.13213. [Online]. Available: https://doi.org/10.1007/978-3-031-81372-6_25

[4] G. Alain and Y. Bengio, "Understanding intermediate layers using linear classifier probes," *arXiv preprint arXiv:1610.01644*, 2016. [Online]. Available: https://arxiv.org/abs/1610.01644

[5] J. Lin, J. Liu, and Y. Liu, "Optimizing multi-hop document retrieval through intermediate representations," *arXiv preprint arXiv:2503.04796*, 2025. [Online]. Available: https://arxiv.org/abs/2503.04796

[6] A. Conneau, G. Kruszewski, G. Lample, L. Barrault, and M. Baroni, "What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018, pp. 2126–2136. [Online]. Available: https://aclanthology.org/P18-1198/

[7] Y. Belinkov, "Probing classifiers: Promises, shortcomings, and advances," *Computational Linguistics*, vol. 48, no. 1, pp. 207–219, 2022. [Online]. Available: https://doi.org/10.1162/coli_a_00422

[8] J. Hewitt and P. Liang, "Designing and interpreting probes with control tasks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019, pp. 2733–2743. [Online]. Available: https://aclanthology.org/D19-1275/

[9] I. Tenney, D. Das, and E. Pavlick, "BERT rediscovers the classical NLP pipeline," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019, pp. 4593–4601. [Online]. Available: https://aclanthology.org/P19-1452/

[10] O. Skean, M. R. Arefin, D. Zhao, N. Patel, J. Naghiyev, Y. LeCun, and R. Shwartz-Ziv, "Layer by layer: Uncovering hidden representations in language models," in *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, ser. Proceedings of Machine Learning Research, vol. 267, 2025, arXiv:2502.02013. [Online]. Available: https://proceedings.mlr.press/v267/skean25a.html

[11] V. Karpukhin, B. Oğuz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih, "Dense passage retrieval for open-domain question answering," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 6769–6781. [Online]. Available: https://aclanthology.org/2020.emnlp-main.550/

[12] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015. [Online]. Available: https://arxiv.org/abs/1503.02531

[13] Qwen Team, "Qwen2.5 technical report," *arXiv preprint arXiv:2412.15115*, 2024. [Online]. Available: https://arxiv.org/abs/2412.15115

[14] L. Ben Allal, A. Lozhkov, E. Bakouch, G. M. Blázquez, G. Penedo, L. Tunstall, A. Marafioti, H. Kydlíček *et al.*, "SmolLM2: When smol goes big – data-centric training of a small language model," *arXiv preprint arXiv:2502.02737*, 2025. [Online]. Available: https://arxiv.org/abs/2502.02737

[15] P. BehnamGhader, V. Adlakha, M. Mosbach, D. Bahdanau, N. Chapados, and S. Reddy, "LLM2Vec: Large language models are secretly powerful text encoders," in *Proceedings of the First Conference on Language Modeling (COLM)*, 2024, arXiv:2404.05961. [Online]. Available: https://arxiv.org/abs/2404.05961

[16] P. Szymański and T. Kajdanowicz, "A network perspective on stratification of multi-label data," in *Proceedings of the First International Workshop on Learning with Imbalanced Domains: Theory and Applications (PMLR)*, 2017, pp. 22–35. [Online]. Available: https://proceedings.mlr.press/v74/szyma%C5%84ski17a.html

[17] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2980–2988. [Online]. Available: https://doi.org/10.1109/ICCV.2017.324