LLM-Independent Adaptive RAG: Let the Question Speak for Itself

Maria Marina[2,1], Nikolay Ivanov[1], Sergey Pletenev[2,1], Mikhail Salnikov[2,1], Daria Galimzianova[3,4], Nikita Krayko[3], Vasily Konovalov[2,1,5], Alexander Panchenko[1,2], Viktor Moskvoretskii[6]

[1]Skoltech, [2]AIRI, [3]MWS AI, [4]MBZUAI,
[5]Moscow Institute of Physics and Technology,
[6]School of Computer and Communication Sciences, EPFL
{Maria.Marina, A.Panchenko}@skol.tech

Abstract

Large Language Models (LLMs) are prone to hallucinations, and Retrieval-Augmented Generation (RAG) helps mitigate this, but at a high computational cost while risking misinformation. Adaptive retrieval aims to retrieve only when necessary, but existing approaches rely on LLM-based uncertainty estimation, which remains inefficient and impractical. In this study, we introduce lightweight LLM-independent adaptive retrieval methods based on external information. We investigated 27 features, organized into 7 groups, and their hybrid combinations. We evaluated these methods on 6 QA datasets, assessing the QA performance and efficiency. The results show that our approach matches the performance of complex LLM-based methods while achieving significant efficiency gains, demonstrating the potential of external information for adaptive retrieval.

1 Introduction

Large Language Models (LLMs) excel in tasks like question answering (QA) (Yang et al., 2018; Kwiatkowski et al., 2019), but remain vulnerable to hallucinations (Yin et al., 2024; Ding et al., 2024). Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) mitigates this by incorporating external information, although it introduces risks such as error accumulation (Shi et al., 2023) and external hallucinations (Ding et al., 2024).

Adaptive retrieval techniques (Moskvoretskii et al., 2025; Ding et al., 2024; Jeong et al., 2024) (AR) aim to balance LLM knowledge with external resources by estimating uncertainty to decide whether retrieval is needed.

However, existing methods primarily frame this task as uncertainty estimation based on LLM internal states or outputs, leading to significant computational overhead. This can offset the efficiency gains from reduced retrieval calls and limit practicality, especially with larger models.

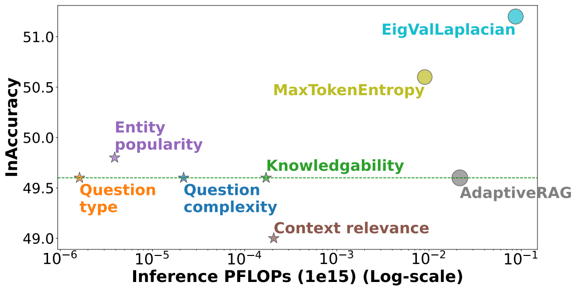


Figure 1: PFLOPs-Inaccuracy trade-off for proposed features vs the most efficient alternative adaptive retrieval methods for the NQ dataset. Radius of the points is proportional to the number of LLM calls. Green dotted line indicates the Always RAG approach.

In this study, we address this issue by introducing LLM-independent adaptive retrieval methods that leverage external information, such as entity popularity and question type. Our methods achieve comparable quality while being significantly more efficient, eliminating the need for LLMs entirely.

Our evaluation, represented in Figure 1, shows that our proposed features are much more efficient in terms of PFLOPs and LLM calls, with downstream performance comparable to other adaptive retrieval methods.

Our contributions and findings are as follows:

1. We introduce 7 groups of lightweight external information features, encompassing 27 features, for LLM-independent adaptive retrieval.

2. Our approach significantly improves efficiency by eliminating the need for LLM-based uncertainty estimation while maintaining QA performance for 2 modern LLMs.

3. We show that combining external features with uncertainty-based may boost the final performance of adaptive retrieval methods.

We make data and all models publicly available.[1]

[1]https://github.com/s-nlp/External_Adaptive_Retrieval

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 8709–8721
November 4-9, 2025 ©2025 Association for Computational Linguistics

2 Related Work

Adaptive Retrieval-Augmented Generation reduces unnecessary retrievals by determining whether external knowledge is needed. This decision can be based on LLM output (Trivedi et al., 2023), consistency checks (Ding et al., 2024), internal uncertainty signals (Jiang et al., 2023; Su et al., 2024; Yao et al., 2025), or trained classifiers (Jeong et al., 2024).

External information methods can enhance text generation. Combining knowledge graphs (KG) with information about entity popularity and graph structures enables more effective reasoning, resulting in more reliable answers (Belikova et al., 2024; Luo et al., 2024). Popularity and graph frequency improve retrieval efficiency, as shown in LightRAG and MiniRAG, which prioritize frequently accessed entities and relationships (Guo et al., 2024; Fan et al., 2025). Graph-based features, including entity properties (Lysyuk et al., 2024), popularity (Mallen et al., 2023a), and structural attributes (Salnikov et al., 2023), have also been shown to be effective in QA systems.

3 Methods

Our baselines include the following adaptive retrieval methods:

Adaptive RAG uses a T5-large-based classifier to determine whether retrieval is needed (Jeong et al., 2024). **FLARE** triggers retrieval when token probability falls below a threshold (Jiang et al., 2023). **DRAGIN** estimates uncertainty based on token probabilities and attention weights, excluding stopwords (Su et al., 2024). **Rowen** relies on consistency checks across languages and models to trigger retrieval (Ding et al., 2024). **SeaKR** monitors internal state consistency to trigger retrieval, reranking snippets to reduce uncertainty (Yao et al., 2025). **EigValLaplacian** assesses uncertainty using graph features based on pairwise consistency scores (Lin et al., 2024). **Max Token Entropy** measures uncertainty by aggregating the maximum entropy of token distributions (Fomicheva et al., 2020). **Hybrid$_{\mathbf{UE}}$** includes 5 uncertainty features relevant to the task (Moskvoretskii et al., 2025): Mean Token Entropy, Max Token Entropy, SAR, EigValLaplacian, Lex-Similarity.

3.1 External Information Methods

In this section, we describe the proposed external information methods for adaptive retrieval. Each group may contain multiple features used to train a classifier to predict retrieval needs, following Moskvoretskii et al. (2025); Jeong et al. (2024). The text features, such as named entities and entity linking, can be extracted using NLP frameworks like BELA (Plekhanov et al., 2023) or DeepPavlov (Savkin et al., 2024).

The first three features below use named entities and/or their linking to Wikipedia IDs which are extracted using pretrained models, thus no access to Wikipedia traffic or Wikidata triples is necessary.

Graph features capture information about the entities in question from a KG, including the minimum, maximum, and mean number of triples per subject and object, where the subject or object corresponds to an entity from the question.

Using the BELA entity linking module, the entities from the question are linked to the corresponding IDs in the Wikidata KG. Then, for each entity the number of triples where this entity is either an object or a subject is retrieved. Finally, six features are calculated: the minimum/maximum/mean number of triples per subject and object.

Popularity features include the minimum, maximum, and mean number of Wikipedia page views per entity in the question.

Using the BELA NER module the entities are retrieved from the question. Then, for each entity the mean number of views per Wikipedia page is calculated using Wikimedia API[2] for last year. Finally, there are three features: the minimum/maximum/mean amount of views per entity per question.

Frequency features include the minimum, maximum and mean frequencies of entities in a reference text collection[3], along with the frequency of the least common n-gram in the question.

Frequency features are calculated similarly to the popularity group. However, instead of page views, the frequencies of entities in the large corpus of text are used. In addition to these three features, a fourth feature is calculated, which searches for all n-grams of the words in the question and selects the n-gram with the lowest frequency.

Knowledgability features assign a score to each entity, reflecting the LLM's verbalized uncertainty about its knowledge. By pre-computing these

[2] https://foundation.wikimedia.org/wiki/Api/

[3] https://www.inf.uni-hamburg.de/en/inst/ab/lt/resources/data/depcc.html

Method	NQ			SQuAD			TriviaQA			2WikiMultihopQA			HotPotQA			MuSiQue			Avg
	InAcc↑	LMC↓	RC↓	InAcc↑	LMC↓	RC↓	InAcc↑	LMC↓	RC↓	InAcc↑	LMC↓	RC↓	InAcc↑	LMC↓	RC↓	InAcc↑	LMC↓	RC↓	**InAcc**
Never RAG	44.6	1.0	0.00	17.6	1.0	0.00	63.6	1.0	0.00	31.8	1.0	0.00	28.6	1.0	0.00	10.6	1.0	0.00	32.8
Always RAG	49.6	1.0	1.00	31.2	1.0	1.00	61.0	1.0	1.00	37.4	1.0	1.00	41.0	1.0	1.00	10.0	1.0	1.00	38.4
Multi-Step Adaptive Retrieval																			
AdaptiveRAG	49.6	2.0	0.98	28.6	2.0	0.97	62.8	1.5	0.54	45.4	5.2	2.64	41.4	4.6	2.34	**14.0**	3.6	3.63	40.3
DRAGIN	48.0	4.5	2.24	29.8	4.3	2.14	**66.6**	4.1	2.06	**45.6**	5.8	2.92	**43.0**	5.1	2.56	13.4	6.3	3.15	41.1
FLARE	45.0	3.1	2.07	23.8	3.1	2.08	64.8	2.1	1.39	42.4	3.9	2.85	37.2	5.1	4.07	9.0	4.1	3.10	37.0
Rowen$_{CM}$	49.4	29.5	7.27	19.6	29.2	7.20	65.6	28.7	7.12	44.4	32.9	7.87	35.6	31.9	7.70	10.4	42.1	9.52	37.5
Seakr	40.6	14.6	1.00	26.8	14.6	1.00	65.6	14.6	1.00	39.8	12.3	2.44	42.4	9.9	1.76	11.8	12.3	2.40	37.8
Uncertainty Estimation																			
EigValLaplacian	49.2	2.0	0.96	31.0	2.0	0.98	64.4	1.3	0.34	37.6	1.9	0.86	39.8	1.9	0.85	10.0	2.0	0.96	38.7
MaxTokenEntropy	**50.0**	2.0	0.99	31.0	1.9	0.92	63.4	1.3	0.31	36.8	2.0	0.95	38.2	1.8	0.76	11.2	1.7	0.72	38.4
Hybrid UE	**50.0**	1.8	0.82	**31.4**	2.0	0.97	63.8	1.3	0.27	38.4	1.9	0.94	41.2	1.9	0.94	11.0	1.7	0.74	39.3
External Features																			
Graph	49.6	1.0	0.86	30.4	1.0	0.95	63.6	1.0	0.32	35.8	1.0	0.67	40.8	1.0	0.97	10.0	1.0	1.00	38.4
Popularity	49.8	1.0	0.92	31.0	1.0	0.98	63.0	1.0	0.16	35.6	1.0	0.84	**41.0**	1.0	0.94	10.6	1.0	0.89	38.5
Frequency	49.8	1.0	0.96	30.4	1.0	0.97	63.2	1.0	0.04	37.4	1.0	0.76	40.8	1.0	0.98	10.4	1.0	0.9	38.7
Knowledgability	49.4	1.0	0.95	**31.2**	1.0	1.00	63.0	1.0	0.28	**38.4**	1.0	0.89	**41.0**	1.0	1.00	10.2	1.0	0.46	38.9
Question type	**50.0**	1.0	0.83	30.4	1.0	0.97	**64.0**	1.0	0.29	36.6	1.0	0.74	39.0	1.0	0.89	10.4	1.0	0.9	38.4
Question complexity	49.6	1.0	1.00	**31.2**	1.0	1.00	63.6	1.0	0.00	36.8	1.0	0.94	**41.0**	1.0	1.00	10.6	1.0	0.95	38.8
Context relevance	47.4	1.0	1.00	31.0	1.0	1.00	62.6	1.0	1.00	36.0	1.0	1.00	**41.0**	1.0	1.00	11.0	1.0	1.00	38.2
Hybrids with External Features																			
Hybrid$_{\neg UFP}$	47.8	1.8	1.0	30.8	1.0	1.0	63.4	1.1	1.0	36.4	1.7	1.0	40.6	2.0	1.0	10.6	1.2	1.0	38.3
Hybrid$_{External}$	46.4	1.2	1.0	30.2	0.9	1.0	63.2	0.2	1.0	37.8	1.6	1.0	39.4	1.9	1.0	10.6	2.0	1.0	37.9
Hybrids with Uncertainty and External Features																			
Hybrid$_{\neg FP}$	49.4	1.7	1.0	**31.2**	2.0	1.0	64.6	1.3	1.0	37.4	1.7	1.0	**41.0**	2.0	1.0	**12.2**	1.4	1.0	39.3
All	47.6	1.8	1.0	**31.2**	2.0	1.0	63.2	1.3	1.0	37.8	1.8	1.0	37.8	1.6	1.0	11.2	1.1	1.0	38.1
Ideal	60.8	1.6	0.55	36.0	1.8	0.82	73.6	1.4	0.36	50.0	1.7	0.68	46.0	1.7	0.71	16.4	1.9	0.89	47.1

Table 1: QA Performance of adaptive retrieval and uncertainty methods. 'Ideal' represents the performance of a system with an oracle providing ideal predictions for the need to retrieve. 'InAcc' denotes In-Accuracy, measuring the QA system's performance. 'LMC' indicates the mean number of LM calls per question, and 'RC' represents the mean number of retrieval calls per question. The SOTA results are highlighted in bold, as well as the best results for the external methods. **Red** states for SOTA result, **green** – for best result either with just external features or hybrids. show top-3 best results by mean accuracy across all datasets for one-step adaptive retrieval methods.

scores for entities in the Wikidata Knowledge Graph, retrieval decisions can be made without querying the LLM at inference time.

The "knowledgability" feature measures the degree of verbalized uncertainty the LLM has regarding a given entity.

We prompt the LLaMA 3.1-8B-Instruct model to evaluate its internal knowledge in a general context about a specific entity. When effective, this feature enables the precomputation of the model's uncertainty about the entity, without the need to infer a new question each time. We assume that the potential range of entities can be assessed in advance using the Wikidata Knowledge Graph (KG). The specific prompt is provided in the Appendix A.

Question Type features include probabilities for nine categories: ordinal, count, generic, superlative, difference, intersection, multihop, comparative, and yes/no.

Using the train part of the Mintaka dataset (Sen et al., 2022), we train a classifier based on the bert-base-uncased model[4] to predict whether a question belongs to one of the 9 question types: 'ordinal', 'count', 'generic', 'superlative', 'difference', 'intersection', 'multihop', 'yesno', 'comparative'. As a result, we get nine probabilities that the question belongs to a certain class. The accuracy classification score on the validation part of the Mintaka dataset is 0.93.

Question Complexity reflects the difficulty of a question, considering the reasoning steps required.

Question complexity is based on the N-hop feature from the FreshQA (Vu et al., 2024) dataset. The question could be one-hop, where the question is explicit about all the relevant information needed to complete the task, so no additional inference is needed. Or multi-hop, where the question requires one or more additional inference steps to gather all the relevant information needed to complete the task. The dataset consists of 500 training and 100 test examples. As a training model, we used a DistilBERT [5] model. The final F1 score on the test set is 0.82.

Context Relevance features include the minimum, maximum and mean probabilities that a con-

[4]https://hf.co/google-bert/bert-base-uncased

[5]https://hf.co/distilbert/distilbert-base-uncased

text is relevant to the question, along with the context length.

Each question with one context at a time is passed to the cross-encoder model based on the uncased model of the bert base. A question and a context are passed via the [SEP] token with the additional classification head over the base model. The final probabilities of each context being relevant are aggregated via minimum/maximum/mean across all contexts. Additionally, there is the fourth feature that calculates the context length.

Hybrid$_{\text{External}}$ includes all external features.

Hybrid$_{\neg\text{UFP}}$ includes all external features except frequency and popularity, as they are highly correlated with graph features.

Hybrid$_{\neg\text{FP}}$ includes uncertainty and all external features except frequency and popularity.

3.2 Leveraging External Information in Resource-Constrained Settings

While there may be concerns that not using model-internal features is unreasonable, we emphasize that leveraging external information is an active and valuable research direction. Recent work has demonstrated that adaptive retrieval with external signals (Mallen et al., 2023b), external memory mechanisms that can outperform model editing (Zhong et al., 2023), and querying structured sources such as KG (Lysyuk et al., 2024) can all provide tangible benefits. Building on this line of research, our work systematically examines the role of external features as complements or substitutes for internal uncertainty estimates, with a particular focus on resource-constrained settings where such signals are especially advantageous.

4 Experimental Setup

In this section, we briefly discuss the implementation details and the evaluation setup.

4.1 Implementation Details

We use LLaMA 3.1-8B-Instruct (Dubey et al., 2024) and the BM25 retriever (Robertson et al., 1994) as the main components of our approach, following Yao et al. (2025); Jeong et al. (2024); Moskvoretskii et al. (2025). Additionally, we test the generalizability of our results with Qwen2.5-7B-Instruct (Yang et al., 2024) which can be found in Appendix C.

4.2 Datasets

We evaluate on single-hop SQuAD v1.1 (Rajpurkar et al., 2016), Natural Questions (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017) and multi-hop MuSiQue (Trivedi et al., 2022), HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (2wiki) (Ho et al., 2020) QA datasets to ensure real-world query complexity, following Trivedi et al. (2023); Jeong et al. (2024); Su et al. (2024); Yao et al. (2025). We use 500-question subsets from the original test sets, as in Moskvoretskii et al. (2025); Jeong et al. (2024).

4.3 Evaluation

We evaluate both the quality and efficiency of the adaptive retrieval system. For quality, we use **In-Accuracy (InAcc)**, which measures whether the LLM output contains the ground-truth answer, as it is a reliable metric based on Moskvoretskii et al. (2025); Mallen et al. (2023b); Jeong et al. (2024); Asai et al. (2024); Baek et al. (2023).

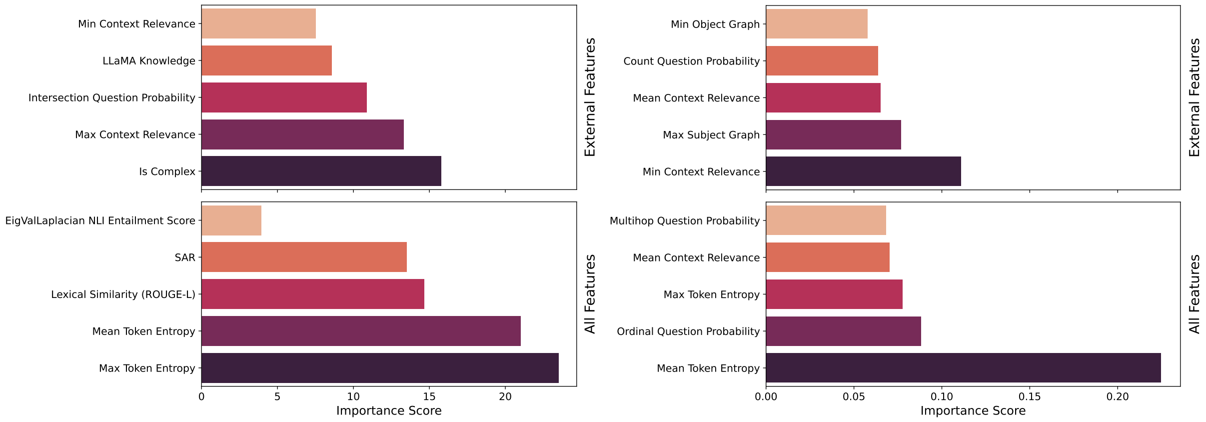
Following Jeong et al. (2024); Moskvoretskii et al. (2025), for efficiency we adopt **Retrieval Calls (RC)** – the average number of retrievals per question, and **LM Calls (LMC)** – the average number of LLM calls per question, including uncertainty estimation. Further details are provided in Appendix A.

5 Results

In the following sections, we present the results of the end-to-end and UE methods, as well as groups of external features, focusing on downstream performance and efficiency. For comparison, we also include the ‘Never RAG’, ‘Always RAG’, and ‘Ideal’ benchmarks. The ‘Ideal’ benchmark represents the performance of a system with an oracle providing perfect retrieval predictions.

Downstream Performance We first evaluate whether external methods can replace the uncertainty-based approaches. As shown in Table 1, at least one external feature performs comparably to uncertainty-based methods on each dataset. In terms of mean accuracy, the best external features — knowledgability and question complexity — trail the top one-step adaptive retrieval methods by just 0.4 and 0.3 points, respectively. Notably, they outperform more complex multistep methods like FLARE and Seakr.

Combining external features even increases In-Accuracy for the Musique dataset. Compared to Multi-Step Adaptive Retrieval, using only external



(a) trivia (b) musique

Figure 2: Feature importances for one of the best algorithms for only external features vs all features for TriviaQA (simple) and Musique (complex) datasets.

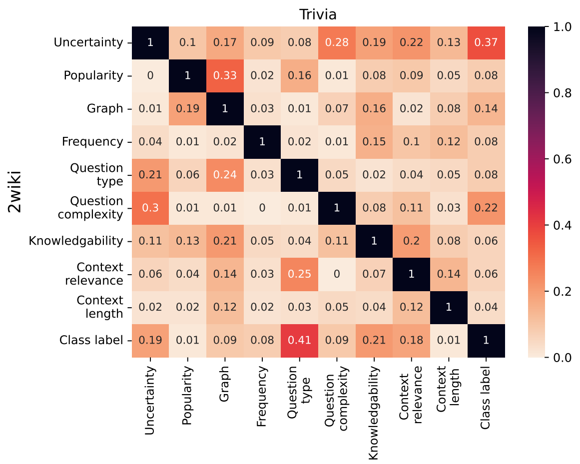


Figure 3: Heatmap of different groups of features for TriviaQA and 2WikiMultiHopQA (2wiki) datasets. Upper right triangle states for the absolute correlations on the TriviaQA, while down left states for the absolute correlations on the 2WikiMultiHopQA

features yields similar results across all datasets, except for 2WikiMultiHopQA.

Second, we examine whether external methods can complement uncertainty-based approaches. On the MuSiQue dataset, combining them improves performance, but no gains are observed on other datasets — suggesting that external features are more substitutive than complementary.

Efficiency Performance External features significantly reduce LLM calls, mitigating a key efficiency bottleneck that worsens with model scaling. Though somewhat more conservative, reflected in extra retrieval calls, they remain more efficient than multistep approaches. Importantly, since external features are pre-computed, they add no inference-time LLM overhead.

6 Features Reciprocity

We identify four key aspects that influence AR performance: LLM knowledge (uncertainty features, knowledgability), question type (simple vs. complex reasoning), context relevance (irrelevant context reduces performance), and entity rarity. Figure 2 shows that for the simple TriviaQA dataset, the top five features are uncertainty-based, while for complex datasets, question type and context relevance become more important. Thus, in some cases relying solely on uncertainty-based features is insufficient for efficient AR. Our results are consistent with prior findings: LLMs can be misled by irrelevant context even when their parametric knowledge is accurate (Liu et al., 2024). Moreover, LLMs often over-rely on their own confidence, favoring shortcuts by sticking to parametric knowledge (Ni et al., 2024).

External features are more substitutive than complementary, often showing strong correlations despite their differences. Figure 3 shows that, for simple questions, uncertainty moderately correlates with question complexity and context relevance. Extra heatmaps and feature importances are provided in Appendix D.

Conclusion

In this work, we propose 7 groups of lightweight, LLM-independent external features for adaptive retrieval, improving efficiency by removing the need for LLM-based uncertainty estimation while maintaining QA performance. Our analysis shows that in some cases, combination of uncertainty and external features yields further performance gains.

Limitations

- Our study focuses on two models – LLaMA3.1-8B-Instruct model and Qwen2.5-7B-Instruct – which are the top-performing open-source models within their parameter range. Expanding the analysis to additional architectures, especially the proprietary ones, could further strengthen the generalizability of our results.

- The classification target reflects whether the model knows the answer without context, whereas the final InAccuracy depends on the quality of retrieved contexts. Future work should explore both the effectiveness of self-knowledge metrics and the sensitivity of results to different retrieval methods.

- We evaluate model performance using six widely adopted QA datasets. Incorporating a broader range of datasets, particularly those tailored to specific domains, could offer more comprehensive insights and showcase the versatility of our approach.

Ethical Considerations

Text retrieval systems can introduce biases into retrieved documents, which may inadvertently steer the outputs of even ethically aligned LLMs in unintended directions. Consequently, developers integrating RAG and Adaptive RAG pipelines into user-facing applications should account for this potential risk.

Acknowledgements

The work was supported by the grant for research centers in the field of AI provided by the Ministry of Economic Development of the R.F. in accordance with the agreement 000000C313925P4F0002 and the agreement with Skoltech №139-10-2025-033.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Jinheon Baek, Soyeong Jeong, Minki Kang, Jong C. Park, and Sung Ju Hwang. 2023. Knowledge-augmented language model verification. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 1720–1736. Association for Computational Linguistics.

Julia Belikova, Evegeniy Beliakin, and Vasily Konovalov. 2024. JellyBell at TextGraphs-17 shared task: Fusing large language models with external knowledge for enhanced question answering. In *Proceedings of TextGraphs-17: Graph-based Methods for Natural Language Processing*, pages 154–160, Bangkok, Thailand. Association for Computational Linguistics.

Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. 2013. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122.

Hanxing Ding, Liang Pang, Zihao Wei, Huawei Shen, and Xueqi Cheng. 2024. Retrieve only when it needs: Adaptive retrieval augmentation for hallucination mitigation in large language models. *CoRR*, abs/2402.10612.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Research Facebook. 2019. fvcore: A light-weight core library for computer vision frameworks. https://github.com/facebookresearch/fvcore.

Tianyu Fan, Jingyuan Wang, Xubin Ren, and Chao Huang. 2025. Minirag: Towards extremely simple retrieval-augmented generation. *CoRR*, abs/2501.06713.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. 2020. Unsupervised quality estimation for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:539–555.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. 2024. Lightrag: Simple and fast retrieval-augmented generation. *CoRR*, abs/2410.05779.

John T Hancock and Taghi M Khoshgoftaar. 2020. Catboost for big data: an interdisciplinary review. *Journal of big data*, 7(1):94.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing A multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 6609–6625. International Committee on Computational Linguistics.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong Park. 2024. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 7036–7050. Association for Computational Linguistics.

Zhengbao Jiang, Frank Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992, Singapore. Association for Computational Linguistics.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1601–1611. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *Trans. Assoc. Comput. Linguistics*, 7:452–466.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2024. Generating with confidence: Uncertainty quantification for black-box large language models. *Trans. Mach. Learn. Res.*, 2024.

Yantao Liu, Zijun Yao, Xin Lv, Yuchen Fan, Shulin Cao, Jifan Yu, Lei Hou, and Juanzi Li. 2024. Untangle the KNOT: interweaving conflicting knowledge and reasoning skills in large language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino, Italy*, pages 17186–17204. ELRA and ICCL.

Linhao Luo, Zicheng Zhao, Chen Gong, Gholamreza Haffari, and Shirui Pan. 2024. Graph-constrained reasoning: Faithful reasoning on knowledge graphs with large language models. *CoRR*, abs/2410.13080.

Maria Lysyuk, Mikhail Salnikov, Pavel Braslavski, and Alexander Panchenko. 2024. Konstruktor: A strong baseline for simple knowledge graph question answering. In *Natural Language Processing and Information Systems - 29th International Conference on Applications of Natural Language to Information Systems, NLDB 2024, Turin, Italy, June 25-27, 2024, Proceedings, Part II*, volume 14763 of *Lecture Notes in Computer Science*, pages 107–118. Springer.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023a. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822, Toronto, Canada. Association for Computational Linguistics.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023b. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 9802–9822. Association for Computational Linguistics.

Viktor Moskvoretskii, Maria Marina, Mikhail Salnikov, Nikolay Ivanov, Sergey Pletenev, Daria Galimzianova, Nikita Krayko, Vasily Konovalov, Irina Nikishina, and Alexander Panchenko. 2025. Adaptive retrieval without self-knowledge? bringing uncertainty back home. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 6355–6384. Association for Computational Linguistics.

Shiyu Ni, Keping Bi, Jiafeng Guo, and Xueqi Cheng. 2024. When do llms need retrieval augmentation? mitigating llms' overconfidence helps retrieval augmentation. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 11375–11388. Association for Computational Linguistics.

Mikhail Plekhanov, Nora Kassner, Kashyap Popat, Louis Martin, Simone Merello, Borislav Kozlovskii, Frédéric A. Dreyer, and Nicola Cancedda. 2023. Multilingual end to end entity linking. *CoRR*, abs/2306.08896.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2383–2392. The Association for Computational Linguistics.

Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. 1994. Okapi at TREC-3. In *Proceedings of The Third Text REtrieval Conference, TREC 1994, Gaithersburg, Maryland, USA, November 2-4, 1994*, volume 500-225 of *NIST Special Publication*, pages 109–126. National Institute of Standards and Technology (NIST).

Mikhail Salnikov, Hai Le, Prateek Rajput, Irina Nikishina, Pavel Braslavski, Valentin Malykh, and Alexander Panchenko. 2023. Large language models meet knowledge graphs to answer factoid questions. In *Proceedings of the 37th Pacific Asia Conference on Language, Information and Computation*, pages 635–644, Hong Kong, China. Association for Computational Linguistics.

Maksim Savkin, Anastasia Voznyuk, Fedor Ignatov, Anna Korzanova, Dmitry Karpov, Alexander Popov, and Vasily Konovalov. 2024. DeepPavlov 1.0: Your gateway to advanced NLP models backed by transformers and transfer learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 465–474, Miami, Florida, USA. Association for Computational Linguistics.

Priyanka Sen, Alham Fikri Aji, and Amir Saffari. 2022. Mintaka: A complex, natural, and multilingual dataset for end-to-end question answering. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pages 1604–1619. International Committee on Computational Linguistics.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H. Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 31210–31227. PMLR.

Weihang Su, Yichen Tang, Qingyao Ai, Zhijing Wu, and Yiqun Liu. 2024. DRAGIN: dynamic retrieval augmented generation based on the real-time information needs of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 12991–13013. Association for Computational Linguistics.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Musique: Multihop questions via single-hop question composition. *Trans. Assoc. Comput. Linguistics*, 10:539–554.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 10014–10037. Association for Computational Linguistics.

Tu Vu, Mohit Iyyer, Xuezhi Wang, Noah Constant, Jerry W. Wei, Jason Wei, Chris Tar, Yun-Hsuan Sung, Denny Zhou, Quoc V. Le, and Thang Luong. 2024. Freshllms: Refreshing large language models with search engine augmentation. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 13697–13720. Association for Computational Linguistics.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. 2024. Qwen2 technical report. *CoRR*, abs/2407.10671.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2369–2380. Association for Computational Linguistics.

Zijun Yao, Weijian Qi, Liangming Pan, Shulin Cao, Linmei Hu, Weichuan Liu, Lei Hou, and Juanzi Li. 2025. Seakr: Self-aware knowledge retrieval for adaptive retrieval augmented generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 27022–27043. Association for Computational Linguistics.

Xunjian Yin, Xu Zhang, Jie Ruan, and Xiaojun Wan. 2024. Benchmarking knowledge boundary for large language models: A different perspective on model evaluation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 2270–2286. Association for Computational Linguistics.

Zexuan Zhong, Zhengxuan Wu, Christopher D. Manning, Christopher Potts, and Danqi Chen. 2023. Mquake: Assessing knowledge editing in language models via multi-hop questions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 15686–15702. Association for Computational Linguistics.

A Technical Details

Train setting. We conduct all experiments using the LLaMA 3.1-8B-Instruct model with its default generation parameters. The responses generated, with and without the retriever, are sourced from previous studies (Moskvoretskii et al., 2025), following the AdaptiveRAG framework (Jeong et al., 2024). The baseline results are also adopted from prior work, as we employ the exact same settings and generation configurations.

We implemented classifiers using Scikit-learn (Buitinck et al., 2013), CatBoost (Hancock and Khoshgoftaar, 2020), and performed hyperparameter tuning using a validation set of 100 samples randomly selected from the training set, testing with three different random seeds for each dataset. For each of the six datasets, the training and validation splits were drawn exclusively from their respective portions of the dataset. This ensures that there is no overlap between the training and test data. We evaluated seven classifiers: Logistic Regression, KNN, MLP, Decision Tree, CatBoosting, Gradient Boosting, and Random Forest. Data preprocessing involved standard scaling. For the final model, we used a VotingClassifier, combining the two best-performing classifiers from the validation set, each trained with their optimal hyperparameters. The voting strategy used is "soft", meaning that the class label prediction is based on the argmax of the sums of the predicted probabilities. Performance was evaluated based on the In-accuracy metric, and the top classifiers were retrained on the full training set with these selected hyperparameters.

> *Answer the following question based on your internal knowledge with one or few words.*
> *If you are sure the answer is accurate and correct, please say '100'. If you are not confident with the answer, please range your knowledgability from 0 to 100, say just number. For example, '40'.*
> *Question: {question}. Answer:*

Hyperparameters grid. *Logistic Regression* : C: [0.01, 0.1, 1], solver: [lbfgs, liblinear], class_weight: [balanced, 0: 1, 1: 1, None], max_iter: [10000, 15000, 20000]

KNN : n_neighbors: [5, 7, 9, 11, 13, 15], metric: [euclidean, manhattan], algorithm: [auto, ball_tree, kd_tree], weights: [uniform, distance]

MLP : hidden_layer_sizes: [(50,), (100,), (50, 50), (100, 50), (100, 100)], activation: [relu, tanh], solver: [adam, sgd], alpha: [0.00001, 0.0001, 0.001, 0.01], learning_rate: [constant, adaptive], early_stopping: True, max_iter: [200, 500]

Decision Tree : max_depth: [3, 5, 7, 10, None], max_features: [0.2, 0.4, sqrt, log2, None], criterion: [gini, entropy], splitter: [best, random]

CatBoosting: iterations: [10, 50, 100, 200], learning_rate: [0.001, 0.01, 0.05], depth: [3, 4, 5, 7, 9], bootstrap_type: [Bayesian, Bernoulli, MVS]

Gradient Boosting: n_estimators: [25, 35, 50], learning_rate: [0.001, 0.01, 0.05], max_depth: [3, 4, 5, 7, 9], max_features: [0.2, 0.4, sqrt, log2, None]

Random Forest: n_estimators: [25, 35, 50], max_depth: [3, 5, 7, 9, 11], max_features: [0.2, 0.4, sqrt, log2, None], bootstrap: [True, False], criterion: [gini, entropy], class_weight: [balanced, 0: 1, 1: 1, None]

B FLOPs Calculation

Method	NQ	
	Mean	Upper bound
AdaptiveRAG	0.0216	0.4389
SeaKR	0.3504	2.4548
DRAGIN	0.2608	1.0129
FLARE	0.0970	0.9290
Rowen	1.8650	15.9677
EigValLaplacian	0.1052	0.3291
MaxTokenEntropy	0.0271	0.2212
Entity popularity	0.0181	0.2103
Question complexity	0.0181	0.2082
Knowledgability	0.0183	0.2275
Context relevance	0.0184	0.2084
Question type	0.0181	0.2074

Table 2: A comparison of FLOPs usage across different methods on the NQ dataset. The "Mean" column shows the average PFLOPs (10^{15} FLOPs) per question, while the "Upper bound" column represents the theoretical maximum FLOPs assuming the LLaMA 3.1 8B model (in FP16 precision) runs at 100% GPU utilization for the entire processing of a single sample. The row labeled "Entity_popularity" reflects the computational overhead required for graph/popularity/frequency features. It is important to note that for features such as "Entity popularity", "Question complexity", "Knowledgability", "Context_relevance", "Question type" the generation of final answer for a question (after precomputing these features) accounts for more than 99% of the total FLOPs .

To calculate floating-point operations (FLOPs), we used the fvcore (Facebook, 2019) library . This library provides a flexible and efficient interface for analyzing the computational complexity of PyTorch models. Specifically, we wrapped our model generation process with the `FlopCountAnalysis` class, which automatically traces the model forward pass and counts the number of FLOPs for each layer. The theoretical analysis includes an approximate formula to calculate an upper bound per sample:

$$\text{Total FLOPs} \approx \big(\text{Total TFLOPs}\big) \times 10^{12} \times (\text{Elapsed Seconds}),$$

$$\text{Total TFLOPs} = (\text{TFLOPs per GPU}) \times (\text{Number of GPUs})$$

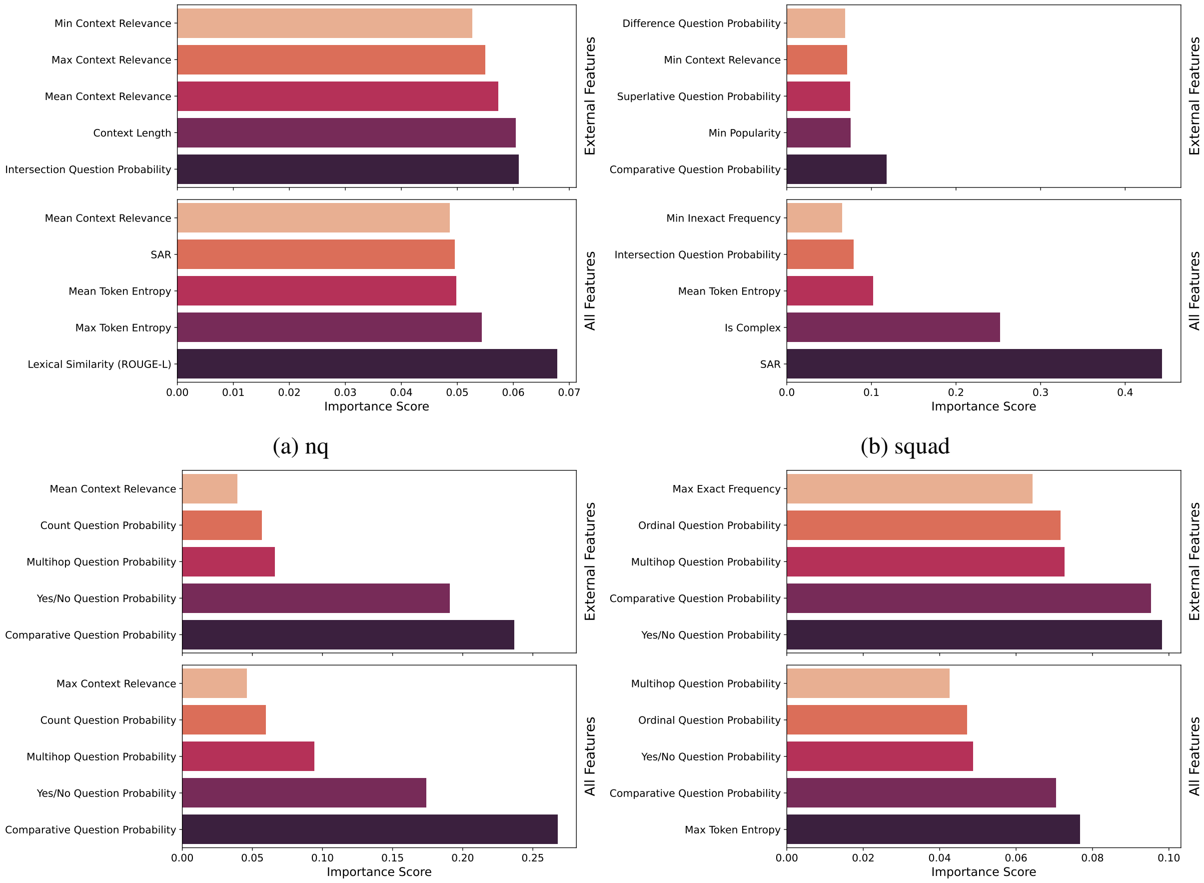
C Results for Qwen2.5-7B-Instruct

Method	NQ			SQuAD			TriviaQA			2WikiMultihopQA			HotPotQA			MuSiQue			Mean
	InAcc↑	LMC↓	RC↓	InAcc↑	LMC↓	RC↓	InAcc↑	LMC↓	RC↓	InAcc↑	LMC↓	RC↓	InAcc↑	LMC↓	RC↓	InAcc↑	LMC↓	RC↓	InAcc
Never RAG	27.8	1.0	0.00	16.2	1.0	0.00	51.6	1.0	0.00	22.0	1.0	0.00	20.2	1.0	0.00	4.60	1.0	0.00	23.7
Always RAG	41.0	1.0	1.00	27.4	1.0	1.00	58.2	1.0	1.00	28.6	1.0	1.00	36.2	1.0	1.00	5.20	1.0	1.00	<u>32.8</u>
Uncertainty Estimation																			
EigValLaplacian	40.4	1.9	0.94	26.8	1.9	0.85	56.8	1.6	0.63	27.0	1.8	0.78	33.8	1.8	0.79	5.40	1.8	0.83	31.7
MaxTokenEntropy	41.0	2.0	1.00	27.0	2.0	0.98	57.8	1.5	0.5	25.0	1.4	0.35	31.6	1.7	0.7	5.20	2.0	0.99	31.3
Hybrid UE	40.6	1.9	0.94	25.0	1.9	0.85	57.2	1.4	0.39	24.8	1.8	0.75	33.0	1.6	0.63	5.00	1.9	0.94	30.9
External Features																			
Graph	40.6	1.0	0.97	26.4	1.0	0.96	56.0	1.0	0.41	26.4	1.0	0.73	**36.2**	1.0	1.00	5.40	1.0	0.95	31.8
Popularity	**41.0**	1.0	0.98	**27.6**	1.0	0.99	53.4	1.0	0.44	25.6	1.0	0.73	**36.2**	1.0	0.99	5.60	1.0	0.98	31.6
Frequency	40.4	1.0	0.94	27.4	1.0	0.99	54.6	1.0	0.32	27.2	1.0	0.82	35.6	1.0	0.96	5.00	1.0	0.93	31.7
Knowledgability	**41.0**	1.0	1.0	27.2	1.0	0.99	55.8	1.0	0.87	27.4	1.0	0.87	35.4	1.0	0.95	5.20	1.0	1.00	32.0
Question type	40.2	1.0	0.98	24.2	1.0	0.77	57.8	1.0	0.54	25.6	1.0	0.78	33.8	1.0	0.83	5.40	1.0	0.96	31.2
Question complexity	**41.0**	1.0	1.00	27.4	1.0	1.00	58.2	1.0	1.00	**28.6**	1.0	1.00	**36.2**	1.0	1.00	5.20	1.0	1.00	<u>32.8</u>
Context relevance	38.8	1.0	0.85	27.4	1.0	1.00	56.2	1.0	0.83	21.6	1.0	0.16	29.0	1.0	0.70	5.00	1.0	0.95	29.7
Hybrids with External Features																			
$\text{Hybrid}_{\neg\text{UFP}}$	40.2	1.9	1.00	27.4	2.00	1.00	**58.8**	1.7	1.00	**28.6**	1.9	1.00	34.0	1.9	1.00	4.80	2.0	1.00	<u>32.3</u>
$\text{Hybrid}_{\text{External}}$	40.8	1.9	1.00	27.2	1.9	1.00	58.2	2.0	1.00	**28.6**	1.9	1.00	34.4	1.9	1.00	5.40	2.0	1.00	<u>32.4</u>
Hybrids with Uncertainty and External Features																			
$\text{Hybrid}_{\neg\text{FP}}$	40.4	2.0	1.00	26.6	1.9	1.00	57.0	1.6	1.00	27.4	1.9	1.00	35.0	1.7	1.00	**6.20**	1.8	1.00	32.1
All	39.2	1.9	1.00	27.0	2.0	1.00	56.8	1.6	1.00	26.4	1.7	1.00	34.8	1.9	1.00	5.20	2.0	1.00	31.6
Ideal	48.4	1.7	0.72	32.6	1.8	0.84	68.6	1.5	0.48	38.8	1.8	0.78	42.4	1.8	0.8	9.00	2.0	0.95	40.0

Table 3: QA Performance of adaptive retrieval and uncertainty methods for Qwen2.5-7B-Instruct. 'Ideal' represents the performance of a system with an oracle providing ideal predictions for the need to retrieve. 'InAcc' denotes In-Accuracy, measuring the QA system's performance. 'LMC' indicates the mean number of LM calls per question, and 'RC' represents the mean number of retrieval calls per question. The SOTA results are highlighted in bold, as well as the best results for the external methods. **Red** states for SOTA result, **green** – for best result either with just external features or hybrids. show top-3 best results by mean accuracy across all datasets for one-step adaptive retrieval methods.

Compared to the LLaMA 3.1-8B-Instruct model, correct answers without context are significantly fewer than those with context, shifting classifier behavior toward the "Always RAG" strategy. Nevertheless, external features still outperform uncertainty features in identifying samples for adaptive retrieval. Notably, on the MuSiQue dataset, external features continue to complement uncertainty-based ones, boosting performance beyond what uncertainty features achieve alone.

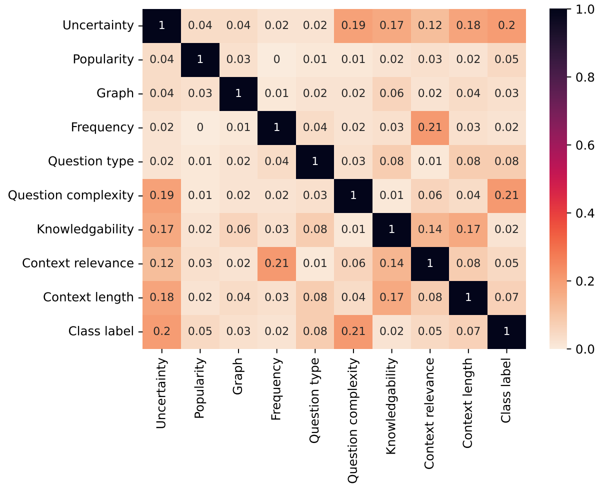
D Heatmaps and Feature Importances



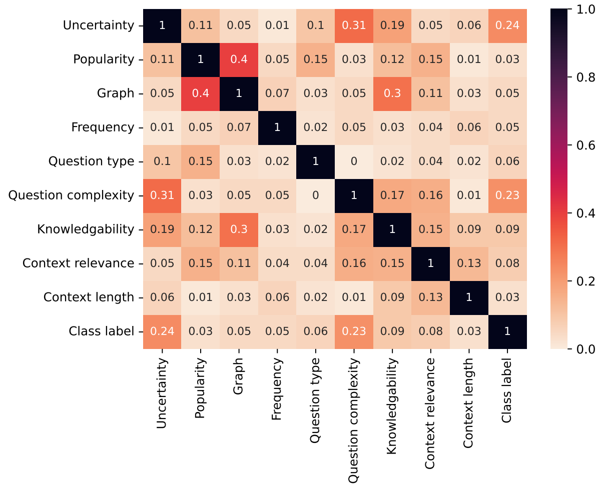
(a) nq (b) squad

(c) 2wikimultihop (d) hotpot

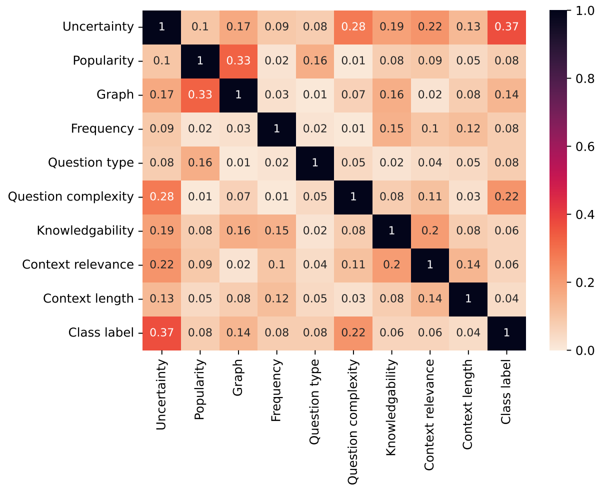
Figure 4: Feature importances for one of the best algorithms for only external features vs all features for NQ, TriviaQA (simple) and HotpotQA, Musique (complex) datasets.



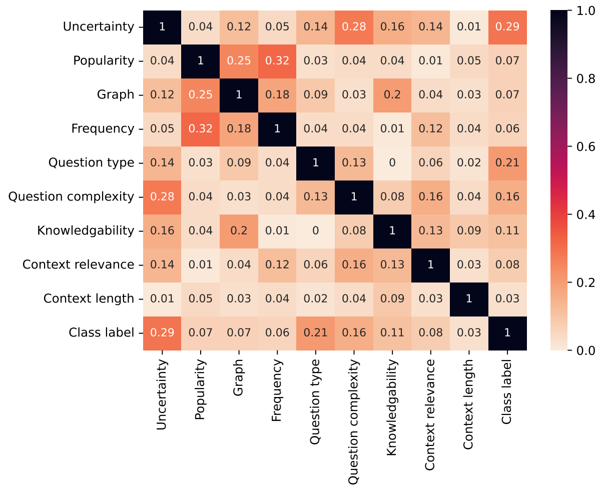
(a) nq



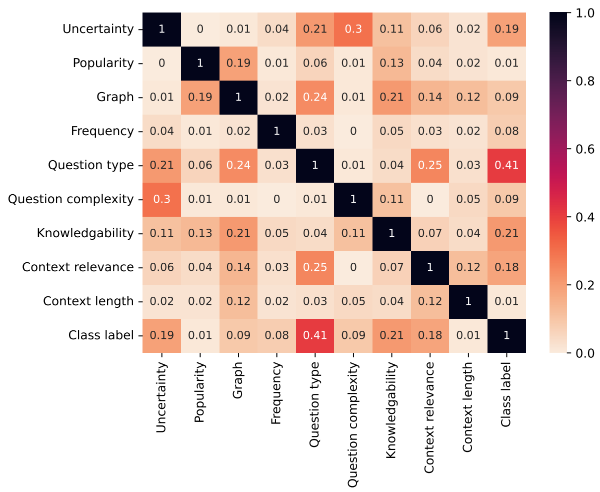
(b) squad



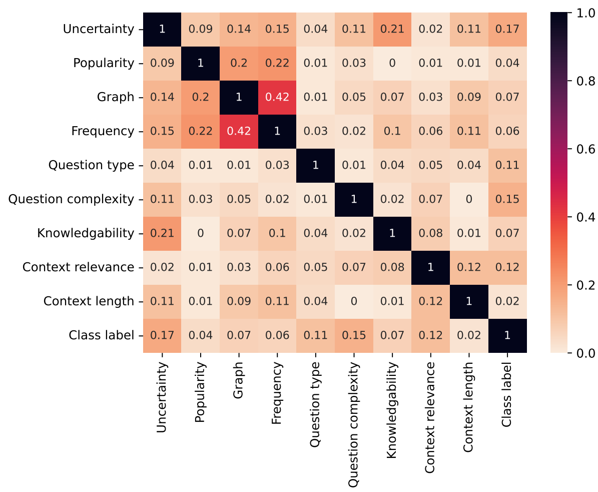
(c) trivia



(d) hotpot



(e) 2wikimultihop



(f) musique

Figure 5: Absolute correlation of features from different groups of external features with class label