Small Encoders Can Rival Large Decoders in Detecting Groundedness

Istabrak Abbes[1,2,3] **Gabriele Prato**[1,2,3] **Quentin Fournier**[2]
Fernando Rodriguez[5] **Alaa Boukhary**[5] **Adam Elwood**[5] **Sarath Chandar**[1,2,4,6]
[1]Chandar Research Lab [2]Mila – Quebec AI Institute [3]Université de Montréal
[4]Polytechnique Montréal [5]Aily Labs [6]Canada CIFAR AI Chair
{firstname.lastname}@mila.quebec {firstname.lastname}@ailylabs.com

Abstract

Augmenting large language models (LLMs) with external context significantly improves their performance in natural language processing (NLP) tasks. However, LLMs struggle to answer queries reliably when the provided context lacks information, often resorting to ungrounded speculation or internal knowledge. Groundedness – generating responses strictly supported by the context – is essential for ensuring factual consistency and trustworthiness. This study focuses on detecting whether a given query is grounded in a document provided in context before the costly answer generation by LLMs. Such a detection mechanism can significantly reduce both inference time and resource consumption. We show that lightweight, task-specific encoder models such as RoBERTa and NomicBERT, fine-tuned on curated datasets, can achieve accuracy comparable to state-of-the-art LLMs, such as Llama3 8B and GPT4o, in groundedness detection while reducing inference latency by orders of magnitude.

1 Introduction

Large language models have demonstrated remarkable capabilities in various tasks, from text generation (Zhao et al., 2024) to question answering (Zhao et al., 2024). However, their tendency to hallucinate when the query lacks support from the provided context or when the model faces noisy retrieval (Yoran et al., 2024; Wu et al., 2024) raises concerns regarding their reliability (Ji et al., 2023). Ideally, LLMs should only answer if the provided context contains enough information to answer the question when combined with the model's parametric knowledge. Otherwise, the model should abstain from answering or ask for more information. Some works have investigated this issue by evaluating RAG-LLMs in the presence of irrelevant information in the context (Wang et al., 2024a; Cuconasu et al., 2024; Xie et al., 2024).

However, these works lack a consistent definition of "relevant information". It could refer to anything from content that directly answers the question to material that is only loosely associated with the topic. This study defines relevance as information that directly contributes to answering the question and frames the problem as a classification task that distinguishes between relevant and irrelevant information. Encoders, such as BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and NomicBERT (Nussbaum et al., 2025), have demonstrated strong capabilities in capturing contextual relationships and producing high-quality embeddings for downstream tasks, which can be useful for factual verification. On the other hand, decoders such as Llama have demonstrated strong capabilities in open-ended text generation. However, they often require significantly more computational resources as shown in Table 1. We evaluate the ability of these models to measure groundedness, defined as whether the document supports the query, irrespective of the model's prior knowledge. This evaluation is conducted on encoder-based and decoder-based models using a variety of question-answering (QA) (Rajpurkar et al., 2018; Trischler et al., 2017) and information retrieval datasets (Thakur et al., 2021).

Our analysis reveals two key findings:

- Fine-tuned encoders can achieve comparable accuracy to the best LLMs in groundedness detection while reducing inference costs by up to 1,000x.
- The zero-shot performance of LLMs in groundedness detection greatly depends on the prompt, especially for smaller models.

Our findings suggest that practitioners can utilize encoder models for groundedness detection in their workflows, achieving similar efficacy to LLMs while significantly decreasing computing costs.

Findings of the Association for Computational Linguistics: ACL 2025, pages 21998–22005

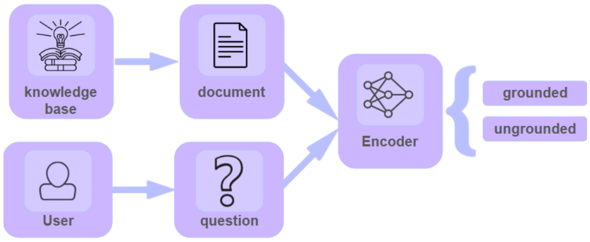

Figure 1: Detecting groundedness before passing the information to the LLM to avoid hallucinations and reduce computational costs.

2 Related Work

A recurring issue in generative models is hallucinations, where the model produces information that appears plausible but is not grounded in reality. Ji et al. (2023) provide a comprehensive taxonomy of hallucinations, categorizing them into intrinsic when arising from model biases, and extrinsic when resulting from data limitations. The former occurs when the generated text includes factual inaccuracies or entities not present in the input, while the latter occurs when the model generates content not supported by the available data or context.

Groundedness in NLP refers to a model's ability to generate outputs consistent with factual knowledge and the given input context. Retrieval-augmented generation (RAG) (Lewis et al., 2021) improves groundedness by incorporating external sources during inference. However, given that RAG leads to very long contexts, the "lost in the middle" problem (Hsieh et al., 2024; Liu et al., 2023) appears, where models struggle to use information positioned in the middle of long input sequences. Some works have tried to solve the problem by focusing on improving the generation and retrieval quality, often by fine-tuning one or more components (Asai et al., 2023; Zhang et al., 2024). However, those works consider relevant documents only for their analysis. As RAG systems surface top-ranked documents, they can still include irrelevant distractions (Cuconasu et al., 2024; Asai et al., 2024; Wang et al., 2023).

Several methods employ a model to predict relevance scores within a larger pipeline (Wang et al., 2024b; Zhou et al., 2024; Jiang et al., 2025) but none of them tried encoders for relevance detection.

3 Experimental Setup

Our experiments aim to evaluate the ability of encoder-based models to determine whether a given query is grounded in the provided context before engaging in the computationally expensive process of answer generation by large language models.

We work with a dataset D, consisting of instances $q = (Q, C)$, where Q represents the query, C represents the provided context, which may or may not contain sufficient information to answer Q. The objective is to train a model that can classify each (Q, C) pair into *relevant* or *irrelevant*, based on whether the context provides enough support for answering the query.

During training and inference, we don't answer the question, as our goal is to detect irrelevant context before answering to avoid unnecessary computation. In other words, our approach complements existing retrieval mechanisms rather than substituting them. Specifically, our encoder-based groundedness detection step assumes that contexts have already been retrieved and serves as a subsequent filtering mechanism. This step aims to prevent unnecessary computational overhead by ensuring that expensive inference by large language models (LLMs) is only invoked when contexts are sufficiently grounded.

3.1 Datasets

To systematically evaluate the groundedness capabilities of encoder-based and decoder-based models, we conducted experiments on diverse datasets covering two key NLP domains: question answering (QA) and information retrieval (IR).

Question Answering For the QA task, we utilized the SQuAD v2.0 (Rajpurkar et al., 2018) and NewsQA datasets (Trischler et al., 2017), both of which present a challenging setup that includes both answerable and unanswerable questions. These datasets require models to effectively discern whether the provided context contains sufficient information to answer the given query. Specifically, SQuAD v2.0 introduces unanswerable questions that necessitate precise comprehension of the passage to avoid generating unsupported answers. NewsQA provides complex questions derived from news articles, requiring a deeper understanding of the context to determine whether an answer can be inferred.

Information Retrieval To extend the evaluation of groundedness beyond the QA paradigm, we leverage two subsets from the BEIR benchmark (Thakur et al., 2021): TREC-COVID (Thakur et al.,

2021) and Touché (Thakur et al., 2021). The TREC-COVID dataset focuses on biomedical literature related to COVID-19, offering a realistic scenario where models must retrieve relevant scientific documents based on complex queries. It challenges the model's ability to assess the sufficiency and relevancy of retrieved information in a high-stakes domain. In contrast, the Touché dataset addresses argument retrieval, requiring models to find documents that provide argumentative support or refutation for controversial topics.

By considering these datasets, our study provides a comprehensive evaluation of groundedness across both QA and IR tasks, enabling a thorough analysis of the strengths and limitations of encoder- and decoder-based models in different scenarios.

3.2 Methodology

In our experiments, we assessed the groundedness capabilities of both encoders such as BERT, RoBERTa, and Nomic-BERT, and decoder-based models, such as GPT-4 and Llama-3, across different configurations, including fine-tuned and zero-shot settings.

Encoders Encoders were fine-tuned using supervised learning to classify context-question pairs as either grounded or ungrounded. The input format for these models involved concatenating the context and question with a designated separator token, such as `[SEP]` for BERT-based architectures. Additionally, the `[CLS]` token is used to classify groundedness. We performed a hyperparameter grid search on the learning rate, weight decay, and batch size (details in Appendix A.1).

Decoders Decoders were assessed in both zero-shot and fine-tuned settings. In the zero-shot setting, we leveraged carefully designed prompts to elicit binary groundedness judgments. To minimize ambiguity and ensure consistency, we employed structured prompt templates, such as "`Given the provided context, is the question answerable using only the information from the context? Respond with yes or no. Provide no explanation.`"

To comprehensively assess decoder model performance, we tested 20 carefully optimized prompts for QA datasets (SQuAD v2.0, NewsQA) and 20 tailored prompts for IR datasets (TREC-COVID, Touché) to enhance zero-shot accuracy and relevance evaluation (Appendix A.2).

Additionally, we fine-tuned Llama-3 1B, 3B, and 8B models on the groundedness detection task. In this supervised fine-tuned configuration, the model was trained using labeled query-context pairs to improve its performance in identifying groundedness. We evaluated all models using accuracy to measure their classification performance. Results are reported as the mean and standard deviation across five random seeds.

4 Results and Discussion

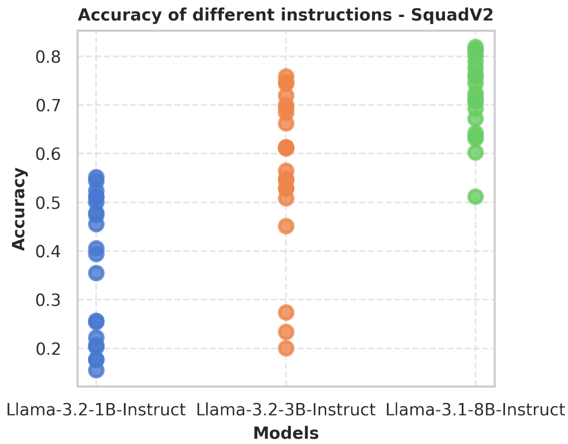

Figure 2: Zero-shot groundedness performance of various Llama models for various prompt templates on SquadV2.0 dataset.

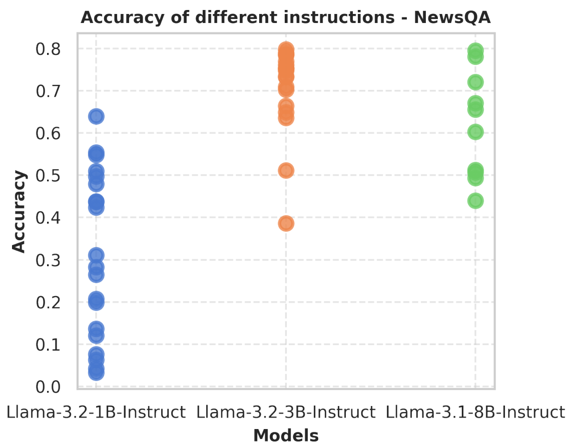

Figure 3: Zero-shot groundedness performance of various Llama models for various prompt templates on NewsQA dataset.

Interplay between Model Scale and Prompt The 8B model consistently outperforms the smaller ones, suggesting that larger models, combined with specific training instructions, improve accuracy in groundedness detection. Specifically,

		SquadV2		NewsQA		TREC-COVID		Touché		FLOPs	
		0-Shot	FT	0-Shot	FT	0-Shot	FT	0-Shot	FT	FT	Inference
Encoder	Bert-base	-	64.1 ± 0.4	-	85.7 ± 0.4	-	72.9 ± 0.1	-	77.6 ± 0.3	9.3×10^{11}	3.1×10^{11}
	Bert-large	-	68.8 ± 0.8	-	86.1 ± 0.4	-	62.8 ± 1.3	-	79.6 ± 1.4	2.5×10^{12}	8.5×10^{11}
	Roberta-base	-	75.8 ± 0.3	-	85.8 ± 0.6	-	73.6 ± 0.3	-	79.3 ± 0.3	1.1×10^{12}	3.7×10^{11}
	Roberta-large	-	$\mathbf{90.2 \pm 0.4}$	-	88.5 ± 0.3	-	75.7 ± 0.7	-	79.2 ± 0.2	3.3×10^{13}	1.1×10^{12}
	Nomic-BERT	-	79.8 ± 0.5	-	88.1 ± 0.4	-	74.2 ± 0.6	-	$\mathbf{82.4 \pm 0.6}$	9.3×10^{11}	3.1×10^{11}
	SimCSE-RoBerta-large	-	88.3 ± 0.2	-	85.9 ± 1.3	-	72.0 ± 2.0	-	80.8 ± 1.6	3.3×10^{13}	1.1×10^{12}
	NeoBERT	-	79.0 ± 0.3	-	86.2 ± 0.2	-	73.8 ± 0.2	-	79.6 ± 0.3	1.8×10^{12}	5.9×10^{11}
	ModernBERT-base	-	86.4 ± 0.2	-	$\mathbf{89.2 \pm 0.3}$	-	$\mathbf{75.9 \pm 0.1}$	-	81.3 ± 0.3	1.5×10^{12}	5.1×10^{11}
	LLM2Vec	-	62.1 ± 0.4	-	85.7 ± 0.4	-	73.9 ± 0.2	-	78.7 ± 0.4	-	-
Open LLM	Llama-3.2-1B-instruct	55.2	56.8 ± 1.4	55.3	84.0 ± 1.3	37.5	50.2 ± 0.4	40.8	48.5 ± 0.8	1.1×10^{16}	2.9×10^{12}
	Llama-3.2-3B-instruct	75.9	82.2 ± 0.8	78.9	86.4 ± 0.6	71.4	74.2 ± 1.1	76.3	82.4 ± 1.4	2.1×10^{16}	7.0×10^{12}
	Llama-3.1-8B-instruct	81.9	$\mathbf{91.1 \pm 0.8}$	79.4	$\mathbf{92.3 \pm 0.3}$	73.8	$\mathbf{75.5 \pm 0.6}$	75.4	$\mathbf{86.1 \pm 0.6}$	6.1×10^{16}	1.6×10^{13}
Close LLM	Claude 3.5 V2	92.5	-	96.7	-	**79.2**	-	**85.4**	-	-	-
	GPT4o	**95.5**	-	**98.1**	-	76.10	-	84.9	-	-	-

Table 1: Accuracy of encoders and LLMs in groundedness detection for zero-shot and fine-tuned settings. Fine-tuned encoders perform closely to the best zero-shot LLMs, except on the challenging Touché dataset.

Llama-3.1-8B-instruct achieves the highest zero-shot performance among open LLMs, with 81.9% on SquadV2, 79.4% on NewsQA, 73.8% on TREC-COVID, and 74.5% on Touché. In contrast, Llama-3.2-1B-instruct, the smallest model, performs significantly lower, with 55.2% on SquadV2, 55.3% on NewsQA, 37.5% on TREC-COVID, and 40.8% on Touché (Figures 2 and 3). This highlights the interplay between model scale and prompt dependency in optimizing performance, where larger models can understand the intent more reliably across phrasing (Salinas and Morstatter, 2024).

Impact of Model Scale and Fine-Tuning Results for different models presented in Table 1 reveal that fine-tuned encoder models, particularly RoBERTa, excel at groundedness detection, with RoBERTa-large achieving 90.2% on SQuAD v2.0 and 88.5% on NewsQA. While Llama3 8B achieves strong zero-shot performance, fine-tuned encoder-based models like RoBERTa still outperform it. In fact, RoBERTa large is better than the zero-shot Llama models and in between the fine-tune 3B and 8B models. This highlights that while large closed-source models like GPT-4 and Claude 3.5 V2 outperform every other models in zero-shot tasks as GPT-4o achieve 95.5% on SQuAD v2.0 and 98.1% on NewsQA, their results are within approximately 10% of the best fine-tuned encoder models, which offer a more compute-efficient alternative.

Fine-tuned models, whether encoder-based or decoder-based, consistently outperform zero-shot counterparts, demonstrating the importance of task-specific training. On average, fine-tuning improves performance by 10-30 percentage points, depending on the dataset and model size. For instance, Llama-3.1-8B-instruct improves from 81.9% (zero-shot) to 91.1% (fine-tuned) on SQuAD v2.0 and from 79.4% to 92.3% on NewsQA, a boost of approximately 20 percentage points. Similarly, Llama-3.2-3B-instruct improves by about 15-20 percentage points across datasets. The gap is consistent across model types, emphasizing that pretraining alone is insufficient for optimal performance. Encoder models like RoBERTa-large (90.2% on SQuAD v2.0, 88.8% on NewsQA) provide comparable fine-tuned performance at a lower computational cost, making them an efficient alternative for groundedness detection in large-scale queries.

Impact of Model Size Larger models generally achieve better performance, but the extent of improvement varies between encoder-based and decoder-based architectures. Among encoder models, RoBERTa-Large (355M parameters) outperforms RoBERTa-Base (125M parameters) by an average of 12-15 percentage points across datasets, highlighting the benefits of increased model capacity in fine-tuned settings. Similarly, for decoder-based models, Llama 8B instruct significantly surpasses Llama 1B instruct in zero-shot settings, with an average accuracy of 74.8% compared to 42.1%, an improvement of over 30 percentage points.

These results demonstrate that scaling up improves performance consistently, but smaller, fine-tuned encoder models like RoBERTa-Large can still compete with much larger decoder models.This tradeoff between model size and efficiency aligns with findings from (Zhang et al., 2023; Zimmerman et al., 2024), which emphasize the advantages of compact, well-trained encoders in classification tasks. We believe lightweight encoders perform well at groundedness detection because the task itself, determining if a passage supports answering a query, aligns naturally with semantic matching, which is a strength of encoders trained with contrastive or classification objectives. In contrast, LLMs are optimized for open-ended generation and often rely on prompt sensitivity or parametric knowledge, which may not be necessary for binary relevance classification. Thus, in tasks with localized grounding, encoders are a better fit both computationally and inductively.

Practicality in Production The choice between an open-source model and a closed-source model depends on performance, costs, and inference speed (Howell et al., 2023). Encoder-based models, such as RoBERTa-large and ModernBERT, significantly outperform decoder-based models like LLaMA 8B in terms of inference cost efficiency as shown in Table 1. While fine-tuning encoders involves initial computational overhead, this investment is quickly amortized in high-throughput applications. For instance, the fine-tuning cost of ModernBERT equates roughly to fewer than 5,000 inference queries with LLAMA3 8B, emphasizing the long-term efficiency benefits of encoder models.

5 Future work

Future research should further explore how groundedness detection methods can handle scenarios involving multiple documents, where collectively supportive information must be integrated to accurately assess groundedness. Investigating mechanisms for aggregating passage-level judgments could improve multi-hop question answering, a scenario not specifically addressed in this study.

Additionally, addressing the detection of internal contradictions within retrieved contexts, a limitation of the current method, represents an important avenue for future research. Incorporating factual consistency checks could significantly enhance the robustness and reliability of groundedness detection systems.

6 Conclusion

This paper investigates the task of predicting whether a question is supported by the document provided in context before the computationally expensive answer generation. Our experiments reveal that lightweight encoders such as RoBERTa and NomicBERT perform well and can closely match the performance of state-of-the-art LLMs such as Llama-8B in some cases. These findings highlight the potential of encoders as an efficient approach to improving the groundedness of NLP systems.

Limitations

While we observed that encoders perform competitively at groundedness detection on four datasets, the performance gap may widen as domains become more complex. Also, our method primarily targets single-document scenarios and straightforward question-answering tasks. It has not been evaluated on multi-hop question answering tasks that require synthesizing information from multiple documents simultaneously, such as HotpotQA or MuSiQue. In addition, our approach does not handle internal contradiction detection within the context, potentially allowing contexts containing contradictory yet relevant information to pass undetected. Thus, additional mechanisms for detecting factual consistency or contradictions would be essential for comprehensive groundedness evaluation.

Acknowledgements

Sarath Chandar is supported by the Canada CIFAR AI Chairs program, the Canada Research Chair in Lifelong Machine Learning, and the NSERC Discovery Grant. This research was enabled in part by compute resources provided by Mila and Aily.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection. *Preprint*, arXiv:2310.11511.

Akari Asai, Zexuan Zhong, Danqi Chen, Pang Wei Koh, Luke Zettlemoyer, Hannaneh Hajishirzi, and Wen-tau Yih. 2024. Reliable, adaptable, and attributable language models with retrieval. *Preprint*, arxiv:2403.03187 [cs].

Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. 2024. The power of noise: Redefining retrieval for rag systems. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR 2024, page 719–729. ACM.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Preprint*, arXiv:1810.04805.

Kristen Howell, Gwen Christian, Pavel Fomitchov, Gitit Kehat, Julianne Marzulla, Leanne Rolston, Jadin Tredup, Ilana Zimmerman, Ethan Selfridge, and Joseph Bradley. 2023. The economic trade-offs of large language models: A case study. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, pages 248–267, Toronto, Canada. Association for Computational Linguistics.

Cheng-Yu Hsieh, Yung-Sung Chuang, Chun-Liang Li, Zifeng Wang, Long T. Le, Abhishek Kumar, James Glass, Alexander Ratner, Chen-Yu Lee, Ranjay Krishna, and Tomas Pfister. 2024. Found in the middle: Calibrating positional attention bias improves long context utilization. *Preprint*, arXiv:2406.16008.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Zhouyu Jiang, Mengshu Sun, Lei Liang, and Zhiqiang Zhang. 2025. Retrieve, summarize, plan: Advancing multi-hop question answering with an iterative approach. *Preprint*, arXiv:2407.13101.

Patrick 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. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Preprint*, arXiv:2005.11401.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts. *Preprint*, arXiv:2307.03172.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *Preprint*, arXiv:1907.11692.

Zach Nussbaum, John X. Morris, Brandon Duderstadt, and Andriy Mulyar. 2025. Nomic embed: Training a reproducible long context text embedder. *Preprint*, arXiv:2402.01613 [cs].

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. *Preprint*, arXiv:1806.03822.

Abel Salinas and Fred Morstatter. 2024. The butterfly effect of altering prompts: How small changes and jailbreaks affect large language model performance. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 4629–4651, Bangkok, Thailand. Association for Computational Linguistics.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *Preprint*, arXiv:2104.08663.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. Newsqa: A machine comprehension dataset. *Preprint*, arXiv:1611.09830.

Yuhao Wang, Ruiyang Ren, Junyi Li, Wayne Xin Zhao, Jing Liu, and Ji-Rong Wen. 2024a. Rear: A relevance-aware retrieval-augmented framework for open-domain question answering. *Preprint*, arXiv:2402.17497.

Yuhao Wang, Ruiyang Ren, Junyi Li, Xin Zhao, Jing Liu, and Ji-Rong Wen. 2024b. REAR: A relevance-aware retrieval-augmented framework for open-domain question answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5613–5626, Miami, Florida, USA. Association for Computational Linguistics.

Zhiruo Wang, Jun Araki, Zhengbao Jiang, Md Rizwan Parvez, and Graham Neubig. 2023. Learning to filter context for retrieval-augmented generation. *Preprint*, arxiv:2311.08377 [cs].

Siye Wu, Jian Xie, Jiangjie Chen, Tinghui Zhu, Kai Zhang, and Yanghua Xiao. 2024. How easily do irrelevant inputs skew the responses of large language models? *Preprint*, arXiv:2404.03302.

Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. 2024. Adaptive chameleon or stubborn sloth: Revealing the behavior of large language models in knowledge conflicts. *Preprint*, arXiv:2305.13300.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. 2024. Making retrieval-augmented language models robust to irrelevant context. *Preprint*, arXiv:2310.01558.

Jiahao Zhang, Haiyang Zhang, Dongmei Zhang, Yong Liu, and Shen Huang. 2024. End-to-end beam retrieval for multi-hop question answering. *Preprint*, arXiv:2308.08973.

Wenxuan Zhang, Yue Deng, Bing Liu, Sinno Jialin Pan, and Lidong Bing. 2023. Sentiment analysis in the era of large language models: A reality check. *Preprint*, arXiv:2305.15005.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2024. A survey of large language models. *Preprint*, arXiv:2303.18223 [cs].

Yujia Zhou, Zheng Liu, Jiajie Jin, Jian-Yun Nie, and Zhicheng Dou. 2024. Metacognitive retrieval-augmented large language models. *Preprint*, arXiv:2402.11626.

Ilana Zimmerman, Jadin Tredup, Ethan Selfridge, and Joseph Bradley. 2024. Two-tiered encoder-based hallucination detection for retrieval-augmented generation in the wild. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 8–22, Miami, Florida, US. Association for Computational Linguistics.

A Appendix

A.1 Hyperparameter Tuning

To ensure the robustness and reproducibility of our results, we conducted extensive hyperparameter tuning using grid search. The tuning process explored various configurations of key parameters, including learning rates, batch sizes, and regularization techniques. Specifically, we experimented with different learning rates, batch sizes, weight decay values, and learning rate scheduling strategies. The hyperparameter search space is summarized in Table 2.

All experiments were conducted using consistent data preprocessing pipelines and evaluation metrics to facilitate a fair comparison between model architectures. This systematic approach ensured that the selected hyperparameters yielded optimal performance across different datasets while maintaining reproducibility.

Hyperparameter	Values
Learning Rate	{1.5e-5, 2e-5, 3e-5}
Batch Size	{8, 16, 32}
Weight Decay	{0.1, 0.01}
Learning Rate Scheduler	{Linear, Cosine}
Warm-up Ratio	{0.06, 0.25}

Table 2: Hyperparameter search space used in the experiments.

A.2 Prompts used for our setting

We've experimented with the following prompts, for the NewsQA and SquadV2 datasets we used:

- Can you answer the question using the given context? Reply with 'yes' or 'no'.
- Based solely on the provided context, is the question answerable? Respond 'yes' or 'no'.
- Evaluate the question with the given context. Can the context provide an answer? Reply 'yes' or 'no'.
- Verify if the question can be answered using the context. Answer with 'yes' or 'no'.
- Is the question answerable from the context provided? Answer 'yes' or 'no'.
- Determine if the context provides enough information to answer the question. Respond 'yes' or 'no'.
- Assess the context and determine if it answers the question. Reply with 'yes' or 'no'.
- Given the context, decide if the question can be answered. Respond 'yes' or 'no'.
- Does the context contain sufficient information to answer the question? Reply with 'yes' or 'no'.
- Based on the context, is it possible to answer the question? Answer 'yes' or 'no'.
- Examine the context and decide if it answers the question. Respond with 'yes' or 'no'.
- Evaluate the given context to determine if the question can be answered. Reply 'yes' or 'no'.
- Analyze the context and determine if it provides an answer to the question. Respond 'yes' or 'no'.
- Does the context provide an answer to the question? Answer 'yes' or 'no'.
- Evaluate if the context answers the question. Reply with 'yes' or 'no'.
- Is there enough information in the context to answer the question? Respond 'yes' or 'no'.
- Analyze the context and decide if it sufficiently answers the question. Reply 'yes' or 'no'.
- Based on the context, determine if the question is answerable. Answer 'yes' or 'no'.
- Verify whether the context answers the question. Reply with 'yes' or 'no'.
- Using only the context provided, decide if you can answer the question. Respond 'yes' or 'no'.

For information retrieval datasets we used :

- Does the context provide relevant information to answer the query? Respond with 'yes' or 'no'.
- Based on the context, is the information provided relevant to answering the query? Answer 'yes' or 'no'.
- Assess whether the context contains relevant details to answer the query. Reply with 'yes' or 'no'.
- Evaluate if the context is relevant to the query. Respond with 'yes' or 'no'.
- Can the context help in answering the query? Respond with 'yes' or 'no'.
- Analyze the relevance of the context and the query. Answer 'yes' or 'no'.
- Determine if the context contains pertinent information to answer the question. Reply with 'yes' or 'no'.
- Does the context include relevant information to address the question? Respond 'yes' or 'no'.
- Evaluate whether the context is closely related to the query. Reply with 'yes' or 'no'.
- Based on the context, assess if the details are relevant for answering the question. Respond with 'yes' or 'no'.
- Determine if the context is sufficient to answer the query. Respond with 'yes' or 'no'.
- Assess whether the context directly addresses the query. Answer 'yes' or 'no'.
- Does the context contain enough information to respond to the query? Reply with 'yes' or 'no'.
- Analyze the context and decide if it is relevant to the query. Respond with 'yes' or 'no'.
- Check if the context provides a direct answer to the query. Reply with 'yes' or 'no'.
- Evaluate the extent to which the context relates to the query. Respond with 'yes' or 'no'.
- Determine whether the query can be answered based on the given context. Answer 'yes' or 'no'.
- Is the context aligned with the information needed to answer the query? Respond 'yes' or 'no'.
- Judge if the context contains meaningful details to answer the question. Reply with 'yes' or 'no'.
- Decide whether the context provides necessary information to answer the query. Respond with 'yes' or 'no'.