Luna: A Lightweight Evaluation Model to Catch Language Model Hallucinations with High Accuracy and Low Cost

Masha Belyi* **Robert Friel*** **Shuai Shao** **Atindriyo Sanyal**

Galileo Technologies Inc.
{masha,rob,ss,atin}@rungalileo.io

Abstract

Retriever-Augmented Generation (RAG) systems have become pivotal in enhancing the capabilities of language models by incorporating external knowledge retrieval mechanisms. However, a significant challenge in deploying these systems in industry applications is the detection and mitigation of hallucinations - instances where the model generates information that is not grounded in the retrieved context. Addressing this issue is crucial for ensuring the reliability and accuracy of responses generated by large language models (LLMs) in industry settings. Current hallucination detection techniques fail to deliver accuracy, low latency, and low cost simultaneously. We introduce Luna: a DeBERTA-large encoder, fine-tuned for hallucination detection in RAG settings. We demonstrate that Luna outperforms GPT-3.5 and commercial evaluation frameworks on the hallucination detection task, with 97% and 91% reduction in cost and latency, respectively. Luna's generalization capacity across multiple industry verticals and out-of-domain data makes it a strong candidate for guardrailing industry LLM applications.

1 Introduction

A key challenge in deploying customer-facing Large Language Model (LLM) applications is their propensity for hallucinations, where the model presents cohesive, but factually incorrect information in conversation with a user (Roller et al., 2021; Lin et al., 2022). Retrieval-augmented generation (RAG), a technique for incorporating knowledge relevant to each user query in the LLM prompt, effectively reduces hallucinations in production systems (Lewis et al., 2020). Yet, LLMs still often respond with nonfactual information that contradicts the knowledge supplied by RAG (Shuster et al., 2021; Magesh et al., 2024).

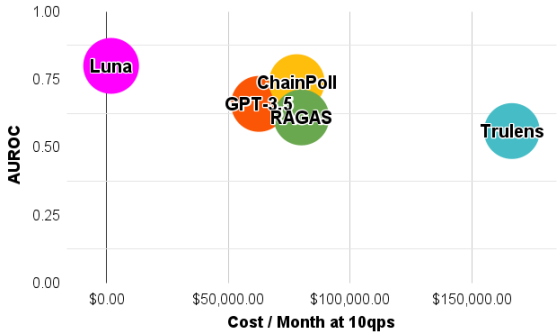

Figure 1: Luna is a lightweight DeBERTA-large encoder, fine-tuned for hallucination detection in RAG settings. Luna outperforms zero-shot hallucination detection models (GPT-3.5, ChainPoll GPT-3.5 ensemble) and RAG evaluation frameworks (RAGAS, Trulens) at a fraction of the cost and millisecond inference speed. AUROC is calculated on RAG QA test set presented in this paper.

Few-shot and finetuned evaluation frameworks like RAGAS (Es et al., 2024), Trulens[1], and ARES (Saad-Falcon et al., 2024) have emerged to offer automated hallucination detection at scale. Though, real-time hallucination prevention in production systems still remains a challenge.

Customer-facing dialogue applications necessitate a hallucination detection system with high-accuracy, low cost, and low latency, such that hallucinations are caught and resolved before reaching the user. LLM prompting approaches fail to meet the strict latency requirement due to model size. Moreover, though commercial LLMs like OpenAI's GPT models (OpenAI, 2023) achieve strong performance, querying customer data through 3rd party APIs is both costly and undesirable for privacy and security reasons. Finetuned BERT-size models are competitive with LLM judges, offering lower latency and local execution (Bohnet et al., 2023; Saad-Falcon et al., 2024; Gao et al., 2023;

*Equal contributions

[1]https://www.trulens.org/

Proceedings of the 31st International Conference on Computational Linguistics: Industry Track, pages 398–409
January 19–24, 2025.

Yue et al., 2023). However, these models require annotated data for training and have not been evaluated for large-scale, cross-domain applications.

In this paper, we introduce Luna - a lightweight RAG hallucination detection model that generalizes across multiple industry-specific domains and scales well for real-time deployment. Luna is a 440M parameter DeBERTa-large encoder that is finetuned on carefully curated real-world RAG data. From analysis of RAG in production settings, we identify long-context RAG evaluation as a previously unaddressed challenge and propose a novel solution that facilitates high precision long-context RAG hallucination detection. Through extensive benchmarking, we demonstrate that Luna outperforms GPT-3.5 prompting on the hallucination detection task.

Our approach is closest to the concurrently proposed ARES RAG evaluation framework (Saad-Falcon et al., 2024), with a few key differences: (1) ARES requires a validation set of in-domain annotated data to finetune a custom evaluation model, while Luna is pre-trained on a cross-domain corpus for built-in generalization; (2) Luna accurately detects hallucinations on long RAG contexts; and (3) Luna is optimized to process up to 16k tokens in milliseconds on deployment hardware.

2 Related Work

Hallucination detection Prior work on hallucination detection in natural language generation is vast (Ji et al., 2023). SelfCheckGPT (Manakul et al., 2023) and Agrawal et al. (2024) are examples of consistency-based methods that detect unreliable outputs by comparing multiple responses from the same LLM. Others train on the internal state of the LLM, such as hidden layer activations (Azaria and Mitchell, 2023; CH-Wang et al., 2024) and token-level uncertainty (Varshney et al., 2023) to predict hallucinations. More generally, zero-shot (Es et al., 2024) and finetuned (Wu et al., 2023; Yue et al., 2023; Muller et al., 2023) LLM judges leverage LLM's inherent reasoning abilities to evaluate other LLM generations. General purpose finetuned LLM evaluators (Kim et al., 2024) that immitate human judgements can also be applied to hallucination detection.

Our approach to finetune a small LM evaluator for RAG scenraios like in (Gao et al., 2023; Saad-Falcon et al., 2024) is the first to evaluate and optimize such a model for industry applications under strict performance, cost, and latency constraints.

NLI for closed-domain Hallucination Detection Existing research draws parallels between hallucination detection and the concept of entailment in Natural Language Inference (NLI). NLI evaluates the relationship between a premise and hypothesis, which can be one of: *entailment*, *contradiction*, or *neutral*. In the past, NLI models have been used to evaluate factual consistency on closed-domain NLG tasks (Honovich et al., 2022; Dziri et al., 2022). The Attributable to Identified Sources (AIS) framework, introduced by Rashkin et al. (2023), formally unifies the notions of factuality, attribution, hallucination, faithfulness, and groundedness - all terms used to measure the extent to which an LLM response is attributable to a source of ground truth. In followup work, NLI entailment has been shown to correlate with AIS scores (Gao et al., 2023; Bohnet et al., 2023; Li et al., 2024) and has become a standard baseline for AIS and hallucination detection models. In this work, we use a pre-trained NLI model as the starting point for Luna finetuning.

3 Luna Model

We fine-tune a DeBERTa-v3-Large (He et al., 2023) NLI checkpoint[2] from Laurer et al. (2022) with a shallow hallucination classifier on each response token. We train on the task of identifying *supported* tokens in the response, given a query and retrieved context. At inference, we treat spans with low support probabilities as hallucinated spans.

Similar to Gao et al. (2023) and Wu et al. (2023), we aim to identify hallucinated spans in the response, rather than a single example-level hallucination boolean. While predicting spans is a more challenging task, it offers interpretability to the end-user. Further, this approach sets us up for accurate long-context prediction, which we discuss next.

3.1 Long Context RAG

In practice, we find that context length limitations are a significant pain point in industry applications. Custom RAG setups may retrieve a large number of context documents from various sources, or choose not to chunk the documents before passing them through the retriever. In Figure 2 we visualize the context length distribution of our curated RAG dataset (detailed in Section 4). While

[2]https://huggingface.co/MoritzLaurer/DeBERTa-v3-large-mnli-fever-anli-ling-wanli

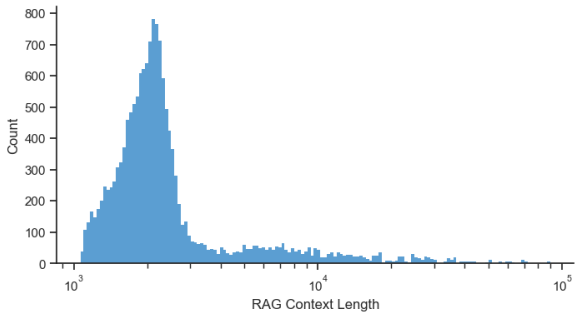

Figure 2: Distribution of RAG context token lengths in our RAG QA training split.

our base DeBERTa model can technically handle sequences of up to 24k (He et al., 2021), computational complexity of transformer attention layers scales quadratically with input length. Moreover, though long-context LLMs like Claude-3 are becoming competitive on LLM leaderboards[3], research shows that these models suffer from information loss (Liu et al., 2023) and may not be suitable for long-context RAG evaluation.

A naive solution is to chunk long-context RAG inputs into short segments and process them through the evaluator model in batches. Model predictions can then be aggregated over batch rows to predict example-level hallucination probabilities. Figure 3 illustrates how such chunking may result in false positives in cases where supporting information is scattered throughout the long context document(s). Instead, we leverage span-level predictions for a high-precision classifier over long contexts, which we describe next.

3.2 Precise Context Chunking

Consider a single input into the RAG evaluation model that consists of **C** context tokens $[c_1...c_C]$, **Q** question tokens $[q_1...q_Q]$, and **R** response tokens $[r_1...r_R]$. Assume an evaluator model with maximum sequence length **L**, and that **Q+R<L**, but **C** is much larger[4]. To fit the example into the model we break it up into windows of length **L**, such that each window contains the question, response, and a subset of the context tokens:

$$w_i = [c_{i_1}...c_{i_l}] \oplus [q_1...q_Q] \oplus [r_1...r_R] \quad (1)$$

where $l = L - Q - R$, and there are $\frac{N}{l}$ windows per example. In Figure 3 there are three such windows.

[3]https://huggingface.co/spaces/lmsys/chatbot-arena-leaderboard

[4]the same approach easily extends to cases where R>L

Our model outputs support probabilities p^i for each of the R response tokens in w_i as:

$$P_S(w_i) = [p_1^i...p_R^i] \quad (2)$$

We train with cross-entropy loss on each token output. During training, we leverage granular token-level annotations to adjust the training labels in each batch based on which context tokens are present in the window. For example, in Figure 3, `"Washington, D.C., the capital of the US"` is supported in window 1, nothing is supported in window 2, and `"was founded in 1791"` is supported in window 3.

At inference, we aggregate example-level support probabilities by taking the token-level maximum over windows. Figure 4 illustrates the steps described by equations 3-5 below. The example-level support probability for token j is defined as:

$$p_j = \max_{1 \le i \le |w|} (p_j^i) \quad (3)$$

where $|w| = \frac{N}{l}$ is the total number of windows we created in (1). To produce an example-level label, we take the minimum over R tokens:

$$P_S = min(p_1...p_R) \quad (4)$$

so that the example support probability is bounded by the least supported token in the response. Finally, we derive example hallucination probability P_H as:

$$P_H = 1 - P_S \quad (5)$$

3.3 Training

To leverage the full pre-trained NLI model, we initialize the hallucination prediction head with weights from the NLI classification head. The original NLI head is a 3-class single-layer perceptron with a neuron for each NLI class. During training, we optimize for low entailment probability and high contradiction probability on hallucinated tokens (and the opposite for supported tokens). At inference, we output the probability of entailment for each token. See Appendix A for hyperparameters and additional training details.

4 Data

We recycle open-book Question Answer (QA) data to construct a **RAG QA dataset**. Our goal is to simulate natural RAG examples that may occurr in production settings. We sample data from five

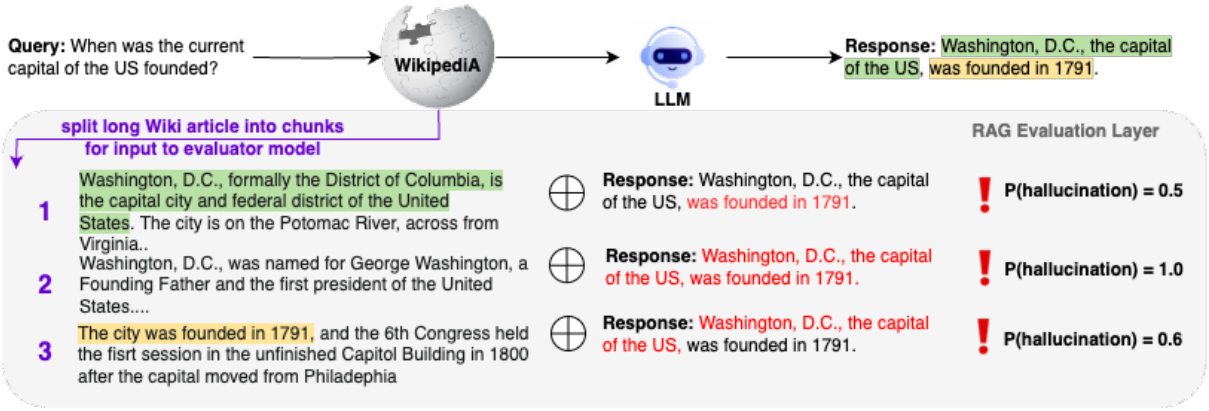

Figure 3: Naive context chunking leads to hallucination false positives when supporting information is scattered throughout the context. We visualize splitting a retrieved Wikipedia page on Washington DC into 3 illustrative short context windows. Though the LLM response is fully supported by facts in the Wikipedia article, a naive evaluation model would detect unsupported spans in each context window and flag the response as a hallucination.

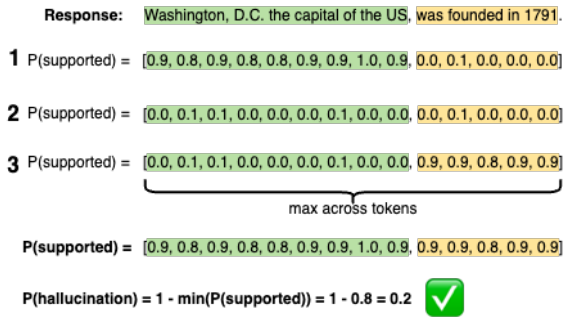

Figure 4: Illustration of Luna's token-level predictions for the example in Figure 3. Luna's token-level predictions are aggregated over context windows into a high-precision hallucination probability score.

Domain	train	val	test	%H
customer support	4k	600	600	22%
finance	38k	5k	5k	5%
biomedical research	22k	3k	3k	20%
legal	1.5k	500	500	6%
general knowledge	9.5k	2k	2k	18%

Table 1: RAG QA statistics. RAG context and questions are sourced from open-book QA datasets that cover five industry-specific domains. RAG responses are generated with GPT-3.5 and Claude-3-Haiku, and annotated with GPT-4-turbo. **%H** is the fraction of hallucinated responses in each domain.

industry verticals: customer support (DelucionQA (Sadat et al., 2023), EManual (Nandy et al., 2021), TechQA (Castelli et al., 2020)), finance and numerical reasoning (FinQA (Chen et al., 2021), TAT-QA (Zhu et al., 2021)), biomedical research (PubmedQA (Jin et al., 2019), CovidQA (Möller et al., 2020)), legal (Cuad (Hendrycks et al., 2021)) and general knowledge (HotpotQA (Yang et al., 2018), MS Marco (Nguyen et al., 2016), HAGRID (Kamalloo et al., 2023), ExpertQA (Malaviya et al., 2024)). The combined dataset covers a variety of difficult RAG task types, including numerical reasoning over tables, inference over multiple context documents, and retrieval from long contexts. Table 1 reports statistics of the data splits.

For each component dataset, we generate two responses per input query and context with GPT-3.5 and Claude-3-Haiku (Appendix B). Both models exhibit strong reasoning and conversational abilities (Chiang et al., 2024) at a low price point, which make them good candidates for production RAG.

LLMs have been shown to align with human judgements on a variety of tasks (Li et al., 2023; Chiang and Lee, 2023), as well as reduce training data annotation costs without sacrificing performance (Wang et al., 2021). Thus, we prompt GPT-4-turbo to annotate the RAG QA dataset with span-level hallucination labels. We apply chain-of-thought, and detailed post-processing steps to ensure high quality annotations, as outlined in Appendix C. We find that our GPT annotator achieves 93% and 95% example- and span-level agreement with human judgements.

5 Evaluation

5.1 Data

We evaluate on a combination of human- and LLM-annotated data.

RAGTruth RAGTruth is an expert-annotated corpus of 18k RAG examples with LLM-generated

Method	Question Answering			Data-to-Text Writing			Summarization			Overall		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
$\text{Prompt}_{\text{gpt-3.5-turbo}}$ †	18.8	84.4	30.8	65.1	95.5	77.4	23.4	89.2	37.1	37.1	92.3	52.9
$\text{Prompt}_{\text{gpt-4-turbo}}$ †	33.2	90.6	45.6	64.3	100.0	78.3	31.5	97.6	47.6	46.9	97.9	63.4
$\text{SelCheckGPT}_{\text{gpt-3.5-turbo}}$ †	35.0	58.0	43.7	68.2	82.8	74.8	31.1	56.5	40.1	49.7	71.9	58.8
$\text{LMvLM}_{\text{gpt-4-turbo}}$ †	18.7	76.9	30.1	68.0	76.7	72.1	23.2	81.9	36.2	36.2	77.8	49.4
Finetuned Llama-2-13B†	61.6	76.3	**68.2**	85.4	91.0	**88.1**	64.0	54.9	**59.1**	76.9	80.7	**78.7**
$\text{ChainPoll}_{\text{gpt-3.5-turbo}}$	33.5	51.3	40.5	84.6	35.1	49.6	45.8	48.0	46.9	54.8	40.6	46.7
RAGAS Faithfulness	31.2	41.9	35.7	79.2	50.8	61.9	64.2	29.9	40.8	62.0	44.8	52.0
Trulens Groundedness	22.8	92.5	36.6	66.9	96.5	79.0	40.2	50.0	44.5	46.5	85.8	60.4
Luna	37.8	80.0	51.3	64.9	91.2	75.9	40.0	76.5	52.5	52.7	86.1	65.4

Table 2: Response-level results on RAGTruth hallucination prediction task. Luna is compared against RAGTruth baselines reported in Wu et al. (2023) (rows marked with †), as well as our own baselines. RAGAS and Trulens are evaluation frameworks that query GPT-3.5-turbo for hallucination detection. ChainPoll is our gpt-3.5-turbo ensemble prompt baseline. ChainPoll, RAGAS, Trulens, and Luna probability thresholds were tuned for best Overall F1. The top and second-best F1 scores are **bolded** and underlined. Luna outperforms all prompt-based approaches and narrows the gap between other baselines and the 13B fine-tuned Llama, at a fraction of the cost.

responses. The data cover three RAG task types: Question Answering, Data-to-text Writing, and News Summarization. This dataset evaluates our model against human judgements as well as across different RAG task types.

RAG QA Test Set We also evaluate Luna on a held-out split of our RAG QA dataset (Section 4). This serves as an in-domain test set for evaluating Luna performance across industry verticals.

5.2 Baselines

Zero-shot prompting We evaluate GPT-3.5-turbo and GPT-4-turbo models from OpenAI as baselines. We prompt the LLMs to return an example-level boolean indicating whether or not a RAG response is supported by the associated RAG context. For RAGTruth we also include all baselines reported in the original paper.

Ensemble prompting LLM ensembles have been shown to outperform single model judges by eliminating bias (Friel and Sanyal, 2023; Verga et al., 2024). We leverage ChainPoll (Friel and Sanyal, 2023) with a chain-of-thought prompt for a stronger GPT-3.5-turbo baseline.

RAG Evaluation Frameworks We evaluate two commercial RAG evaluation frmeworks: RAGAS (v0.1.7) (Es et al., 2024) and Trulens (v0.13.4). Both leverage custom GPT-3.5 prompts for hallucination detection. We report performnace of RAGAS Faithfulness and Trulens Groundedness.

5.3 Metrics

For comparison with RAGTruth baselinses, we report Precision, Recall, and F1 scores on RAGTruth. We tune Luna output probability thresholds for the best overall F1 and report all metrics at the optimal threshold. On RAG QA, we report the area under the ROC curve (AUROC), which circumvents the need for threshold tuning.

6 Results

On the RAGTruth dataset, Luna outperforms all prompt-based approaches on the QA and Summarization tasks, and is competitive with GPT-3.5 evaluators on the Data-to-Text Writing task (Table 2). Overall, Luna is second only to the finetuned Llama-2-13B, which is expected given the significant difference in size between the two models (440M vs 13B). Notably, the Llama-2-13B baseline was trained on a subset of RAGTruth, while Luna was trained on a QA-only dataset with a different data distribution. Nevertheless, we find that Luna generalizes well to the out-of-domain task types. Additionally, Luna's gains in cost and inference speed (Sections 7.2, 7.3) offset the performance gap. Results on the RAG QA test set follow a similar pattern (Table 3). Luna outperforms the GPT-3.5 baselines across all verticals.

In Table 3 we also report Luna's domain generalization capacity. We find that a model trained on a subset of domains in RAG QA (Luna_{OOD}) still outperforms most baselines on the out-of-domain subsets.

7 Discussion

7.1 Long Context Hallucination Detection

We evaluate Luna's performance against baselines on a range of RAG context lengths (Table 4). For this analysis we sample data from CUAD

Method	CUSTOMER SUPPORT	FINANCIAL REASONING	GENERAL KNOWLEDGE	LEGAL	BIOMED	OVERALL
GPT-4-turbo annotator	1.0	1.0	1.0	1.0	1.0	1.0
$\text{Prompt}_{\text{gpt-3.5-turbo}}$	0.68	0.67	0.67	0.63	0.64	0.66
$\text{ChainPoll}_{\text{gpt-3.5-turbo}}$	**0.76**	0.74	0.75	0.71	0.71	0.74
RAGAS Faithfulness	0.62	0.60	0.60	0.58	0.54	0.61
Trulens Groundedness	0.56	0.56	0.65	0.34	0.68	0.56
$\text{Luna}_{\text{in-domain}}$	**0.76**	**0.82**	**0.81**	0.78	**0.83**	**0.80**
Luna_{OOD}	0.74	0.64	-	**0.79**	-	-

Table 3: AUROC on the hallucination detection task on the RAG QA test set. Top scores in each domain are **bolded** and underlined. $\text{Luna}_{\text{in-domain}}$ is our model trained on combined train splits from each domain. Luna_{OOD} is the same model trained on a subset of General Knowledge and Biomed domains.

context length (count in test)	0-5k (223)	5k-16k (209)	16k+ (78)
$\text{Prompt}_{\text{gpt-3.5-turbo}}$	0	-12.11%	-100%
$\text{ChainPoll}_{\text{gpt-3.5-turbo}}$	0	-8.97%	-100%
RAGAS Faithfulness	0	-4.36%	-100%
Trulens Groundedness	0	-6.38%	-100%
Luna	0	-12.55%	-31.98%
$\text{Luna}_{\text{example}}$	0	-21.44%	-43.75%

Table 4: Relative hallucination detection performance of various models on short(0-5k), medium(5k-16k), and long(16k+) context lengths. **Luna** is our best fine-tuned DeBERTA-large model, and $\textbf{Luna}_{\textbf{example}}$ is a version of Luna that makes example-level predictions.

(Hendrycks et al., 2021), where full-length legal contracts are used as RAG context. We find that performance of all models inversely correlates with context length. However, while the GPT-3.5-powered baselines fail completely beyond the GPT-3.5 context limit (16k tokens), Luna maintains 68% of it's performance on that subset.

To validate our long context chunking approach (Section 3.2), we do an ablation study comparing our best model to a version of Luna that makes example level predictions ($\text{Luna}_{\text{example}}$). Our findings confirm that $\text{Luna}_{\text{example}}$ performs worse on long contexts. Although performance of both models degrades with increasing context lengths, $\text{Luna}_{\text{example}}$ exhibits greater degradation than Luna.

7.2 High Accuracy Low Cost

API-based hallucination detection methods accrue substantial costs if used continuously in production settings. In Figure 1 we illustrate the trade-off between monthly maintenance costs and accuracy for Luna versus our baselines. Luna outperforms GPT-3.5-based approaches while operating at a fraction of the cost. Detailed cost calculations are found in Appendix D.

7.3 Latency Optimizations

We optimize Luna and its deployment architecture to process up to 16k input tokens in under one second on NVIDIA L4 GPU. To achieve this, we deploy an ONNX-traced model on NVIDIA Triton server with TensorRT backend. We leverage Triton's Business Logic Scripting (BLS) to optimize the data flow and orchestration between GPU and CPU resources. BLS intelligently allocates resources based on the specific requirements of each inference request, ensuring that both GPU and CPU are utilized effectively and that neither resource becomes a bottleneck. We also tune our inference model maximum input length for optimal performance. While increasing the maximum sequence length would reduce the size and number of inference batches (see Section 3.2), longer batch inputs also increase transformer computational complexity. We determine token length of 512 to be the most effective. Finally, we optimize pre-and post-processing code for efficiency. See Appendix Table 6 for step-wise latency reductions.

8 Conclusion

In this work we introduced Luna: a cost-effective hallucination detection model with millisecond inference speed. Luna eliminates dependency on slow and expensive 3rd party API calls, and enables practitioners to effectively address hallucinations in production. When hosted on a local GPU, Luna guarantees privacy that 3d-party APIs cannot.

8.1 Limitations

Closed Domain Hallucinations Luna's efficacy is limited to closed domain hallucination detection in RAG settings. Due to size, Luna lacks the necessary world knowledge to detect open domain hallucinations, and relies on a high-quality retriever to support open-domain applications. For open-

domain applications, Luna relies on a high-quality RAG retriever to provide the necessary context for an input query.

References

Ayush Agrawal, Mirac Suzgun, Lester Mackey, and Adam Kalai. 2024. Do language models know when they're hallucinating references? In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 912–928, St. Julian's, Malta. Association for Computational Linguistics.

Amos Azaria and Tom Mitchell. 2023. The internal state of an LLM knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 967–976, Singapore. Association for Computational Linguistics.

Steven Bird and Edward Loper. 2004. NLTK: The natural language toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 214–217, Barcelona, Spain. Association for Computational Linguistics.

Bernd Bohnet, Vinh Q. Tran, Pat Verga, Roee Aharoni, Daniel Andor, Livio Baldini Soares, Massimiliano Ciaramita, Jacob Eisenstein, Kuzman Ganchev, Jonathan Herzig, Kai Hui, Tom Kwiatkowski, Ji Ma, Jianmo Ni, Lierni Sestorain Saralegui, Tal Schuster, William W. Cohen, Michael Collins, Dipanjan Das, Donald Metzler, Slav Petrov, and Kellie Webster. 2023. Attributed question answering: Evaluation and modeling for attributed large language models. *Preprint*, arXiv:2212.08037.

Vittorio Castelli, Rishav Chakravarti, Saswati Dana, Anthony Ferritto, Radu Florian, Martin Franz, Dinesh Garg, Dinesh Khandelwal, Scott McCarley, Michael McCawley, Mohamed Nasr, Lin Pan, Cezar Pendus, John Pitrelli, Saurabh Pujar, Salim Roukos, Andrzej Sakrajda, Avi Sil, Rosario Uceda-Sosa, Todd Ward, and Rong Zhang. 2020. The TechQA dataset. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1269–1278, Online. Association for Computational Linguistics.

Sky CH-Wang, Benjamin Van Durme, Jason Eisner, and Chris Kedzie. 2024. Do androids know they're only dreaming of electric sheep? In *Findings of the Association for Computational Linguistics ACL 2024*, pages 4401–4420, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, and William Yang Wang. 2021. FinQA: A dataset of numerical reasoning over financial data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3697–3711, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Cheng-Han Chiang and Hung-yi Lee. 2023. Can large language models be an alternative to human evaluations? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15607–15631, Toronto, Canada. Association for Computational Linguistics.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. 2024. Chatbot arena: An open platform for evaluating llms by human preference. *Preprint*, arXiv:2403.04132.

Nouha Dziri, Hannah Rashkin, Tal Linzen, and David Reitter. 2022. Evaluating attribution in dialogue systems: The BEGIN benchmark. *Transactions of the Association for Computational Linguistics*, 10:1066–1083.

Shahul Es, Jithin James, Luis Espinosa Anke, and Steven Schockaert. 2024. RAGAs: Automated evaluation of retrieval augmented generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 150–158, St. Julians, Malta. Association for Computational Linguistics.

Robert Friel and Atindriyo Sanyal. 2023. Chainpoll: A high efficacy method for llm hallucination detection. *Preprint*, arXiv:2310.18344.

Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, and Kelvin Guu. 2023. RARR: Researching and revising what language models say, using language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16477–16508, Toronto, Canada. Association for Computational Linguistics.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. DeBERTav3: Improving deBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. In *The Eleventh International Conference on Learning Representations*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Xingwei He, Zhenghao Lin, Yeyun Gong, A-Long Jin, Hang Zhang, Chen Lin, Jian Jiao, Siu Ming Yiu, Nan Duan, and Weizhu Chen. 2024. Annollm: Making large language models to be better crowdsourced annotators. *Preprint*, arXiv:2303.16854.

Dan Hendrycks, Collin Burns, Anya Chen, and Spencer Ball. 2021. Cuad: An expert-annotated nlp dataset for legal contract review. *NeurIPS*.

Or Honovich, Roee Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and

Yossi Matias. 2022. TRUE: Re-evaluating factual consistency evaluation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3905–3920, Seattle, United States. Association for Computational Linguistics.

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 Comput. Surv.*, 55(12).

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. PubMedQA: A dataset for biomedical research question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2567–2577, Hong Kong, China. Association for Computational Linguistics.

Ehsan Kamalloo, Aref Jafari, Xinyu Zhang, Nandan Thakur, and Jimmy Lin. 2023. HAGRID: A human-llm collaborative dataset for generative information-seeking with attribution. *arXiv:2307.16883*.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024. Prometheus 2: An open source language model specialized in evaluating other language models. *Preprint*, arXiv:2405.01535.

Moritz Laurer, Wouter van Atteveldt, Andreu Casas, and Kasper Welbers. 2022. Less annotating, more classifying – addressing the data scarcity issue of supervised machine learning with deep transfer learning and bert - nli. *Open Science Framework Preprint*.

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. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Yifei Li, Xiang Yue, Zeyi Liao, and Huan Sun. 2024. Attributionbench: How hard is automatic attribution evaluation? *arXiv preprint arXiv:2402.15089v1*.

Zhuoyan Li, Hangxiao Zhu, Zhuoran Lu, and Ming Yin. 2023. Synthetic data generation with large language models for text classification: Potential and limitations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10443–10461, Singapore. Association for Computational Linguistics.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

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.

Varun Magesh, Faiz Surani, Matthew Dahl, Mirac Suzgun, Christopher D. Manning, and Daniel E. Ho. 2024. Hallucination-free? assessing the reliability of leading ai legal research tools. *Preprint*, arXiv:2405.20362.

Chaitanya Malaviya, Subin Lee, Sihao Chen, Elizabeth Sieber, Mark Yatskar, and Dan Roth. 2024. Expertqa: Expert-curated questions and attributed answers. *Preprint*, arXiv:2309.07852.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017, Singapore. Association for Computational Linguistics.

Timo Möller, Anthony Reina, Raghavan Jayakumar, and Malte Pietsch. 2020. COVID-QA: A question answering dataset for COVID-19. In *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020*, Online. Association for Computational Linguistics.

Benjamin Muller, John Wieting, Jonathan Clark, Tom Kwiatkowski, Sebastian Ruder, Livio Soares, Roee Aharoni, Jonathan Herzig, and Xinyi Wang. 2023. Evaluating and modeling attribution for cross-lingual question answering. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 144–157, Singapore. Association for Computational Linguistics.

Abhilash Nandy, Soumya Sharma, Shubham Maddhashiya, Kapil Sachdeva, Pawan Goyal, and NIloy Ganguly. 2021. Question answering over electronic devices: A new benchmark dataset and a multi-task learning based QA framework. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4600–4609, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human generated machine reading comprehension dataset.

OpenAI. 2023. `https://openai.com`.

Hannah Rashkin, Vitaly Nikolaev, Matthew Lamm, Lora Aroyo, Michael Collins, Dipanjan Das, Slav Petrov, Gaurav Singh Tomar, Iulia Turc, and David Reitter. 2023. Measuring attribution in natural language generation models. *Computational Linguistics*, 49(4):777–840.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Eric Michael Smith, Y-Lan Boureau, and Jason Weston. 2021. Recipes for building an open-domain chatbot. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 300–325, Online. Association for Computational Linguistics.

Jon Saad-Falcon, Omar Khattab, Christopher Potts, and Matei Zaharia. 2024. Ares: An automated evaluation framework for retrieval-augmented generation systems. *Preprint*, arXiv:2311.09476.

Mobashir Sadat, Zhengyu Zhou, Lukas Lange, Jun Araki, Arsalan Gundroo, Bingqing Wang, Rakesh Menon, Md Parvez, and Zhe Feng. 2023. Delucionqa: Detecting hallucinations in domain-specific question answering. pages 822–835.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3784–3803, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Neeraj Varshney, Wenlin Yao, Hongming Zhang, Jianshu Chen, and Dong Yu. 2023. A stitch in time saves nine: Detecting and mitigating hallucinations of llms by validating low-confidence generation. *Preprint*, arXiv:2307.03987.

Pat Verga, Sebastian Hofstatter, Sophia Althammer, Yixuan Su, Aleksandra Piktus, Arkady Arkhangorodsky, Minjie Xu, Naomi White, and Patrick Lewis. 2024. Replacing judges with juries: Evaluating llm generations with a panel of diverse models. *Preprint*, arXiv:2404.18796.

Shuohang Wang, Yang Liu, Yichong Xu, Chenguang Zhu, and Michael Zeng. 2021. Want to reduce labeling cost? GPT-3 can help. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4195–4205, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Yuanhao Wu, Juno Zhu, Siliang Xu, Kashun Shum, Cheng Niu, Randy Zhong, Juntong Song, and Tong Zhang. 2023. Ragtruth: A hallucination corpus for developing trustworthy retrieval-augmented language models. *Preprint*, arXiv:2401.00396.

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 *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Xiang Yue, Boshi Wang, Ziru Chen, Kai Zhang, Yu Su, and Huan Sun. 2023. Automatic evaluation of attribution by large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 4615–4635, Singapore. Association for Computational Linguistics.

Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. 2021. TAT-QA: A question answering benchmark on a hybrid of tabular and textual content in finance. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3277–3287, Online. Association for Computational Linguistics.

A Luna Training Details

We fine-tune a DeBERTa-v3-Large (He et al., 2023) NLI checkpoint[5] from Laurer et al. (2022) with a shallow hallucination classifier on each response token. We train on the task of identifying *supported* tokens in the response, given a query and retrieved context. Framing the problem in this way makes our work comparable to recent automated RAG evaluation efforts. Our definition of *support* is synonymous with the *answer faithfulness* metric explored in RAGAS (Es et al., 2024) and ARES (Saad-Falcon et al., 2024), Truelens *groundedness*, and *attribution* (Li et al., 2024). At inference, we treat spans with low support probabilities as hallucinated spans.

The model trains for 3 epochs with cross-entropy loss on the output of each response token. We initialize the learning rate to 5^{-6} for the base model layers and 2^{-5} for the classification head, and train with warmup and a linear decay rate.

We apply data transformation techniques to introduce additional variability for better generalization during training. Transformations include dropping and inserting context documents, and shuffling questions and responses between examples in batch. Training labels are adjusted accordingly with each transformation.

B Response Generation Prompt

We use the following prompt template to generate LLM responses for each sample in our QA

[5] https://huggingface.co/MoritzLaurer/DeBERTa-v3-large-mnli-fever-anli-ling-wanli

Table 5: Annotation Alignment with DelucionQA. We report F1 and Accuracy metrics on human annotated subsets of DelucionQA.

Test Set	F1	Accuracy
DelucionQA - example level	0.96	0.93
DelucionQA - span level	0.97	0.95

RAG dataset. Context documents, separated by line breaks, along with the question are slotted in for each generation sample. We set temperature to 1 for generation to encourage diversity and potential hallucinations in the responses. The prompt:

> Use the following pieces of context to answer the question.
>
> {documents}
>
> Question: {question}

C RAG QA Dataset Annotation

We leverage GPT-4-turbo to annotate the RAG QA dataset with span-level hallucination labels

Before annotation, we split the context and response into sentences using nltk (Bird and Loper, 2004). We pass the question along with the tokenized context and response sentences to GPT-4-turbo for annotation. For each sentence in the response, we instruct the LLM to identify which context sentences, if any, support the claim in the response. Tokens in sentences without any support are treated as hallucinations. We find that LLM responses often contain transition sentences and general statements that, while not supported by any specific context span, are generally grounded in the question and provided context. We instruct the annotator to label these as "generally supported", which we post-process to indicate support in every context window during training. Statements highlighting lack of sufficient information to answer the question also fall into this category.

We take measures to ensure high quality labels from our LLM annotator. First, we use chain-of-thought (Wei et al., 2022), which has been shown to increase agreement between LLM and human judgements (He et al., 2024). Next, we request both response-level and sentence-level annotations that we compare to identify potentially noisy labels. For example, if GPT-4 claims a response as supported by the context as a whole, but identifies no supporting information for one or more claims in the response, we send the example for re-annotation. We re-annotate examples up to 3 times, after which <2% of the data are still conflicting. After manual inspection, we find that the majority of the conflicts arise from partially supported sentences. Since our annotation scheme is binary on the sentence level (the full sentence is either supported or not), we resolve all tokens in partially supported sentences to "not supported" on both the sentence and example level.

C.0.1 Annotation Alignment with Human Judgements

We validate ourlabeling approach on a human annotated benchmark. DelucionQA (Sadat et al., 2023) is a curated collection of user queries on the operation of Jeep's 2023 Gladiator model. Natural language queries are first generated by an LLM, then reviewed and filtered by human annotators. Context documents are sourced from Jeep's Gladiator User Manual, and responses are generated by various LLMs. Human annotators label each response sentence as "Supported" by the context documents, "Conflicted", or "Neither". Example-level labels are derived from the span-level annotation as follows: if at least one response sentence is annotated as "Conflicted" or "Neither", the whole response receives a Hallucinated label.

In our initial investigation, we found that sentences that DelucionQA labels as "Neither" often fall into one of two categories: (1) general filler statements (e.g. "Here are the steps:"), (2) claims of missing information (e.g. "There is no mention of any problem with engine start-up in freezing weather related to DEF."). According to our annotation schema, these types of statements are generally grounded in the context and not hallucinations. Thus, we remove examples with any "Neither" sentence annotations for our analysis. We annotate the remaining 421 examples with our LLM annotator and report alignment with human annotations in Table 5.

D Cost Calculations

Costs are estimated assuming average throughput of 10 queries per second (qps), with average RAG query length of 4000 tokens, and NVIDIA L4 GPU deployment hardware. When estimating LLM cost for >1qps we assume concurrency is implemented to process multiple queries in parallel. Although we do not explicitly compare pricing against larger fine-tuned models such as Llama-2-13B, we note

that hosting a multi-billion parameter model demands substantially more compute resources than Luna, which would be reflected in the overall cost.

Luna Costs Empirically, we find that each L4 can serve up to 4qps. At the time of writing, the monthly cost of running a g6.2xlarge GPU instance on AWS cloud is $700[6]. Thus, we estimate total monthly cost for 10qps throughput as

$$\$700 * \frac{10}{4} = \$1750 \quad (6)$$

OpenAI Costs At the time of writing, querying GPT-3.5-turbo through OpenAI API costs $0.50 / 1M input tokens and $1.50 / 1M output tokens[7]. In our test set, we observe the average output token length from GPT-3.5 at 200 tokens. Using average input length of 4000 tokens, the cost of a single query is roughly

$$(4k * \$0.5 + 200 * \$1.5)/1M = \$0.0023 \quad (7)$$

Using 2,592,000 seconds/month, the monthly cost of serving 10qps with GPT-3.5 is:

$$10qps * 2,592,000 * \$0.0023 = \$59,616 \quad (8)$$

With ChainPoll ensemble, we request 3 outputs per query, bringing the cost of a single query up to

$$(4k * \$0.5 + 3 * 200 * \$1.5)/1M = \$0.0029 \quad (9)$$

And the total monthly cost for 10qps to:

$$10qps * 2,592,000 * \$0.0029 = \$75,168 \quad (10)$$

RAGAS Costs RAGAS makes 2 OpenAI API calls per an input RAG example. The first query extracts a list of claims from the response. The second requests the LLM to evaluate the faithfulness of each extracted claim to the RAG context. We estimate that the output length of the first query is roughly equal to the length of the RAG response; and the output length of the second query is roughly 3x the length of the response, since it includes the original claims followed by a faithfulness score and an explanation. Factoring in overhead token length of each prompt, we calculate the cost per query to be

$$Query1 = \$380/1M \quad (11)$$

$$Query2 = \$2730/1M \quad (12)$$

Then, the monthly cost of serving 10qps is:

$$10qps*2,592,000*(\$380+\$2730)/1M = \$79,937 \quad (13)$$

[6]https://aws.amazon.com/ec2/pricing/on-demand/

[7]https://openai.com/api/pricing/

Optimization	s/16k
baseline	3.27
TensorRT backend	2.09
efficient pre- and post- processing code	1.79
512 max model length	0.98
BLS	0.92

Table 6: Impact of latency optimizations on Luna inference speed. Reporting inference speed in seconds for processing 16k input tokens.

Trulens Costs Trulens makes 1 OpenAI per each sentence in the response. For this calculation, we estimate 3 sentences per response, which aligns with our obesrvations on the QA RAG dataset. Each query returns original sentence, a groundedness score (1-10), and an explanation. Here we assume that the token length of the explanation is roughly equal to the token length of the input sentence. The cost of a single query is roughly

$$(4k * \$0.5 + 2 * 75 * \$1.5)/1M = \$0.0022 \quad (14)$$

Using 2,592,000 seconds/month, the monthly cost of serving 10qps with Trulens is:

$$10qps*2,592,000*3*\$0.0022 = \$173,016 \quad (15)$$

E Latency Optimizations

We optimize Luna and its deployment architecture to process up to 16k input tokens in under one second on NVIDIA L4 GPU. Table 6 details the latency reductions and how they were achieved.

F Latency Comparison

We empirically estimate the latency of Luna and each baseline model. Luna latency is discussed in Appendix E. For LLm models that query OpenAI API, we calculate the average latency per query after querying the API multiple times with an input of 4000k tokens, split between 3800 tokens for the context, 25 tokens for the question, and 75 tokens for the response.

Model	s/4k	%change
Luna	0.23	-
GPT-3.5	2.5	-91%
ChainPoll n=3	3.0	-93%
Trulens	3.4	-93%
RAGAS	5.4	-96%

Table 7: Model latency (in seconds), comparing Luna to LLM baselines. We also report the % difference between Luna and LLM-based models.