DETECTING HALLUCINATIONS IN GROUNDED LLM OUTPUTS WITH FEW-SHOT LEARNING

Rodolfo Lopez Mendoza
Department of Computer & Data Science
The University of Texas at Austin
rodolfolopez@utexas.edu

Abstract

Grounded LLM outputs are prevalent in contemporary AI applications, particularly those encompassing RAG with indexed document databases and agent-based search via web search. However, despite the advancement of capabilities in LLMs and improvements in information retrieval techniques, intrinsic hallucination remains a significant concern in these systems, as hallucinations undermine user trust, especially when they arise from within the model itself. The standard approach for hallucination detection and other quality benchmarks of deployed LLMs is the LLM as a judge method. However, this method is not accessible to everyone due to its relatively high operational costs, along with other limitations. This paper explores an efficient solution to this problem by training an embedding model using the SetFit library. SetFit fine-tunes the embedding model using contrastive loss and adds a classification head. This approach, using only three training samples achieves an F1 score of 71.87% on the RAG Truth test set adapted for binary classification using a 0.1 billion parameter model. This performance is superior to using gpt-4-turbo as a judge with a custom prompt, which achieves 63.4% and is also comparable to other similar low-parameter models, such as lettucedetect-base-v1 which achieves 76.07%, when trained on the full 15.1k sample dataset. Our approach enables the creation of hallucination detection models with just a few samples while mainting computational resources low.

***K*eywords** Hallucination Detection · Few-Shot Learning · Large Language Models · Natural Language Inference · Grounded Outputs · Retrieval-Augmented Generation · SetFit

1 Introduction

The implementation of Retrieval-Augmented Generation (RAG) systems [1] and agentic web search mechanisms [2] in real world applications has continued to grow exponentially due to two main advantages: their ability to provide the LLM with real time updated data and their ability to provide the LLM with data that was not included in the model's original training set without requiring parameter updates or additional adapters [3]. These systems rely on LLMs to synthesize knowledge from external sources, aiming to produce grounded and authoritative responses. However, despite significant advances in LLM capabilities [4] and information retrieval techniques [5], a critical vulnerability persists: hallucination. Even SOTA LLMs such as openai/gpt-5.1-high-2025-11-13 [6] shows a 12.1% hallucination rate in the Vecatara hallucination leaderboard [7]. This phenomenon, defined as the generation of plausible yet factually incorrect or unsupported content, severely compromises user trust and system reliability. Addressing this issue could advance to the safe implementation of AI systems in critical domains such as healthcare and law, ultimately benefiting humanity by enhancing the quality and accuracy of professional work [8] [9].

To precisely define the scope of this work, we adopt the hallucination taxonomy presented in the HalluLens paper [10], which distinguishes between two primary forms, extrinsic and intrinsic. Extrinsic hallucinations are generations that are inconsistent with the training data. Intrinsic hallucinations are generations that are inconsistent with the provided input context or source material. Since RAG systems are designed to strictly adhere to retrieved documents, our research focuses specifically on intrinsic hallucinations, which represent a direct failure of faithfulness to the context and can occur when the model prioritizes its training data over the retrieved information. This scenario has been extensively explored [11].

The current standard evaluation method the "LLM as a judge" paradigm can represent a major operational bottleneck due to the high computational cost of running SOTA LLMs. This approach, which uses powerful commercial LLM to evaluate generated content, can be prohibitively expensive and slow for low-resource projects. Moreover it is well known that the LLM as a judge approach is not a silver bullet method for quality testing [12] [13]. Although a review of the LLM as a judge approach is beyond this scope, we acknowledge its superiority over other methods in certain scenarios, such as when time to market is critical or when specialized expertise is unavailable.

Seeking an efficient and cost-effective solution, we draw inspiration from the Natural Language Inference approach presented by Eugene Yan, Principal Applied Scientist at Amazon [14]. In the NLI task, a "hypothesis" (the generated output) is compared with a "premise" (the source document) closely mirrors the challenge of detecting intrinsic contradictions. This suggests that models pre-trained or fine-tuned on NLI tasks have an inherent ability to evaluate textual consistency.

This paper proposes an efficient and scalable solution for binary intrinsic hallucination detection by using a lightweight, fine-tuned embedding model. Our methodology leverages the SetFit library [15], a few-shot learning framework that fine-tunes a Sentence Transformer model using contrastive loss before adding a simple differentiable classification head. We will also explore transfer learning from a fine-tuned NLI model to determine whether pre-trained consistency knowledge can further enhance the performance of our low-resource SetFit classifier. Additionally, we evaluate the effect of using larger training sets beyond the typical "few-shot" regime. The model's performance is assessed on a binary classification version of the RAGTruth test set [16]. This approach demonstrates a viable, low-cost alternative, achieving an F1 score of 71.87% with just three training samples, showing that high-performing, resource-efficient detectors are a practical solution for ensuring the quality of grounded LLM systems in low-resource settings.

2 Related Work

2.1 Hallucination in LLMs and its taxonomy

Research on hallucinations in LLMs has been increasingly active [17], driven by how widely these systems are now used in real-world industries. This interest has grown even more with the large-scale adoption of RAG systems and, more recently, agentic web-search methods. With this level of exposure, hallucinations have become one of the main limitations holding back the use of LLMs in higher-risk areas. Even with today's models, hallucinations are still very present. Some recent work even argues that hallucinations may be a built-in aspect of LLMs and can't be fully removed from within the model [18].

Hallucinations in LLMs are commonly described as content the model produces that appears clear and well-written but does not actually align with the information in its input or with real facts. In other words, the model presents a statement with strong confidence even though there is not enough support for it in the given context or in the model's own knowledge. Previous work characterizes hallucinations as a mismatch between what the model should be able to infer from the information it has and what it ultimately generates. These hallucinations may appear as invented facts, incorrect reasoning steps, or logical connections that do not properly follow from the available information. Additionally, hallucinations are not limited to obvious factual errors. They can also take the form of subtle distortions, overly specific claims, or invented relationships that the model introduces while aiming to produce a fluent response. This broader perspective is important because many harmful hallucinations arise from these quieter, harder to detect deviations rather than from clear and direct inaccuracies.

We adopt the hallucination taxonomy presented in the HalluLens paper, which is essential for defining the scope of this work. As noted in their study, there is significant confusion surrounding the distinction between extrinsic and intrinsic hallucinations.

Intrinsic Hallucination: This refers to content generated by the LLM that is inconsistent with the provided input context or source material. In the context of Retrieval-Augmented Generation (RAG) systems, intrinsic hallucination represents a failure of faithfulness to the retrieved documents—an issue that is the primary focus of this paper.

Extrinsic Hallucination: This refers to content that is inconsistent with the model's training data. It should not be confused with factuality errors, in which the model fails to provide correct factual knowledge due to factors such as knowledge cutoffs. Extrinsic hallucination falls outside the scope of this work.

With these distinctions established, we can more effectively address the challenges associated with hallucinations and advance research into practical, real-world methods for LLM-powered solutions.

2.2 NLI for intrinsic hallucination and Vectara's HHEM

Natural language inference (NLI) is the task of determining whether a hypothesis entails, contradicts, or undermines a premise. This relationship between a ground truth and a generated text applies directly to detecting intrinsic hallucinations in grounded LLM responses, where the premise is the information retrieved by any retrieval system (e.g., RAG pipelines, tool-based web search) and the hypothesis is the LLM's output.

Our approach is partly inspired by the methodology proposed by Eugene Yan, where a small model (a variant of BART [19]) was fine-tuned for natural language inference using QLoRA [20], then adapted for hallucination detection through transfer learning—first training on the USB dataset which is made up of eight summarization tasks.[21], followed by the FIB dataset [22] which is made of documents based on Wikipedia. His approach achieves a PR AUC of 0.60 when testing on the FIB test set with the USB fine-tuned model and a PR AUC of 0.85 on the FIB tested with the USB + FIB fine-tuned model. This shows that transfer learning is a viable approach when building natural language inference solutions using LLMs.

Another lightweight solution that influenced our work is Vectara's Hughes hallucination evaluation model [23], which reaches a 64.42% balanced accuracy score on the RAGTruth test set. They achieve this by modifying a flan-t5-base model [24], keeping only the encoder and adding a classifier head on top. The methodology and training data used for this model is not publicly disclosed.

These examples show that specialized neural networks can outperform general-purpose LLMs on targeted tasks. They also demonstrate that models with around 100 million parameters can surpass trillion-token models in specific areas, making AI more accessible for teams with limited computational resources.

2.3 LLM as judge

The LLM as a judge approach uses a large language model with a custom evaluation prompt to score generated text based on criteria you define. It can be used to evaluate many different qualities such as fluency, hallucinations, politeness, and more. Its biggest advantage is flexibility: you can evaluate almost any aspect of an entire conversation or a single query response pair just by adjusting the evaluation prompt. Another major advantage is development speed. If a team is willing to use a private API, all they need to do is integrate it into their system and write a custom prompt. This is far faster than training a specialized model and building the infrastructure needed to host it. Because of this, the LLM as a judge method has become a standard choice for evaluating LLM responses: it offers fast development, performance, and a lot of flexibility.

That said, this approach is far from perfect, and the hallucination detection problem is still unsolved. Even though performance is generally good and any LLM system without evaluation tools would benefit from it, research has shown that LLM-as-a-judge methods methods can be unreliable. Our Table 3 highlights this issue: GPT-4-turbo reaches an F1 score of only 63.4% in the RAGTruth test set, which is noticeably outperformed by our proposed lightweight model. Beyond reliability, another major drawback is cost. Running these SOTA models is computationally expensive, which can be a deal breaker for projects with limited budgets or for systems that need guardrails [25] to catch hallucinations during inference, before the user is exposed to them.

2.4 LettuceDetect

We want to highlight this framework because it is the closest work to our approach that we could find. The LettuceDetect framework [26] offers a state-of-the-art, resource-efficient alternative to LLM-as-a-judge methods. LettuceDetect uses ModernBERT with a classification head on top, fine-tuned for token-level classification. The model was trained on the RAGTruth dataset, where input sequences are created by concatenating the context, question, and answer segments using specialized tokens.

The lettuce-detect-base-v1 model achieves a strong 76.07% overall F1 score when trained on the full 15.1k-sample RAGTruth training dataset. This level of performance, achieved with a relatively small, specialized model, supports the general approach of using encoder-based models for this task. Our research aims to show that similarly competitive results can be achieved with only a few training samples.

3 Data

For this work, we focused on three relevant datasets: the Stanford Natural Language Inference (SNLI) [27], the Multi-Genre Natural Language Inference (MNLI) [28], and RAGTruth. SNLI and MNLI are important because they are the natural language inference datasets on which our main model was pretrained. SNLI contains 570k human-written English sentence pairs manually labeled for for clasifying entailment, contradiction or neutral text pairs. MNLI contains 433k examples spanning ten distinct genres of written and spoken English, this is its biggest improvement upon available resources in this domain. Lastly, the RAGTruth dataset is specifically designed for analyzing word-level hallucinations across different domains and tasks within standard RAG frameworks for LLM applications. It includes nearly 18k naturally generated responses from a variety of LLMs using RAG. It also supports response level hallucination detection which is the modality we are selecting for our approach.

Task	Instances	Responses	Halluc. Responses	Halluc. Spans
Summarization(CNN/DM)	628	3768	1165	1474
Summarization(Recent News)	315	1890	521	598
Question Answering	989	5934	1724	2927
Data-to-text	1033	6198	4254	9290
Overall	**2965**	**17790**	**7664**	**14289**

Table 1: Descriptive statistics divided by task in the RAGTruth dataset

To adapt this resource for efficient binary classification, we used a processed version of the original RAGTruth dataset that includes a new column indicating whether a sample contains evident conflict or baseless information. We then created another column for binary classification: samples without hallucinations (no evident conflict or baseless information) were assigned a label of 0, while samples containing hallucinations (either or both evident conflict or baseless information) were assigned a label of 1.

To adapt this resource for efficient binary classification, we used a processed version of the original RAGTruth dataset that includes a new column indicating whether a sample contains evident conflict or baseless information. We then created another column for binary classification: samples without hallucinations (no evident conflict or baseless information) were assigned a label of 0, while samples containing hallucinations (either or both evident conflict or baseless information) were assigned a label of 1.

4 Method

We use the sentence-transformer model all-mpnet-base-v2 [29], which maps sentences and paragraphs into a 768-dimensional dense vector space. We also experiment with a version of this model fine-tuned on the SNLI and MNLI datasets to test whether transfer learning from the NLI domain improves performance. Additionally, a BERT-base model [30] fine-tuned on MNLI is trained for comparison with this architecture.
Input sequences are created by concatenating the contents of the context and hypothesis columns, while the question column is ignored. The input to our model follows the structure below:

$$Input = "Premise : " + Context + "Hypothesis : " + Output$$

We use Hugging Face's SetFit library for training. SetFit updates the embedding model weights using contrastive learning [31], which pulls embeddings of positive pairs closer together in vector space and pushes embeddings of negative pairs farther apart. We also experiment with training using a triplet loss, which uses an anchor to pull the positive sample closer while pushing the negative sample away. This reshapes the embedding space to be discriminative for the new classification task, even with minimal data.

In a second step, a classification head is added on top and trained from scratch using the now fine-tuned embeddings. Training uses the AdamW optimizer [32] with a body learning rate of 1e-5, head learning rate of 1e-2, and a weight decay of 0. We train for 1–2 epochs on an NVIDIA A100 GPU with a batch size of 16–32. For our experiments, we use training sets of 2, 3, 5, 10, and 20 samples for few-shot evaluation, and also explore larger sets of 50, 100, 500, 1000, 2000, and 11k samples to test how the framework scales beyond few-shot training.

5 Results

We evaluated our models on the RAGTruth test dataset adapted for binary classification across different training set sizes. Our best-performing model is the mpnet-base model, previously fine-tuned on the MNLI dataset. For this training, we used a triplet loss function, a batch size of 16, and trained for 2 epochs.

This model achieved a 74.59% accuracy and a 71.87% F1 score, as shown in Table 2, outperforming the LLM-as-a-judge approach using GPT-4-turbo, which scored 64.4% (Table 3).

Samples	Accuracy	Macro Precision	Macro Recall	Macro F1-Score
2	0.6496	0.6448	0.6591	0.6395
3	**0.7459**	**0.7202**	**0.7173**	**0.7187**
5	0.7404	0.7143	0.7136	0.7139
10	0.5915	0.6386	0.6426	0.5912
20	0.6963	0.6766	0.6893	0.6792

Table 2: Training metrics from the mpnet-base model previously fine-tuned on the NLI dataset using triplet loss.

Method	**OVERALL**		
	Prec.	**Rec.**	**F1**
Prompt$_{\texttt{gpt-3.5-turbo}}$	37.1	92.3	52.9
Prompt$_{\texttt{gpt-4-turbo}}$	46.9	**97.9**	63.4
SelCheckGPT$_{\texttt{gpt-3.5-turbo}}$	49.7	71.9	58.8
LMVLM$_{\texttt{gpt-4-turbo}}$	36.2	77.8	49.4
Finetuned Llama-2-13B	76.9	80.7	78.7
RAG-HAT	**87.3**	80.8	**83.9**
ChainPoll$_{\texttt{gpt-3.5-turbo}}$	54.8	40.6	46.7
RAGAS Faithfulness	62.0	44.8	52.0
Trulens Groundedness	46.5	85.8	60.4
Luna	52.7	86.1	65.4
`lettucedetect-base-v1`	76.64	75.50	76.07
`lettucedetect-large-v1`	80.44	78.05	79.22
`mpnet-base-ragtruth-v1`	72.02	71.73	71.87

Table 3: Overall Performance Comparison

Table 4 shows the results of the mpnet-base model trained with a triplet loss for our task, but without prior fine-tuning on the MNLI dataset. Even in this setting, the model achieves a competitive F1 score of 68.83% after 2 epochs with a batch size of 16.

Samples	Accuracy	Macro Precision	Macro Recall	Macro F1-Score
2	0.5404	0.6072	0.6031	0.5401
3	**0.7085**	**0.6852**	**0.6950**	**0.6883**
5	0.6970	0.6772	0.6898	0.6798
10	0.5122	0.6181	0.5970	0.5072
20	0.6630	0.6577	0.6732	0.6531

Table 4: Training metrics from the mpnet-base model using triplet loss.

Table 5 presents the results of the BART-base model, fine-tuned on MNLI and then further fine-tuned on the RAGTruth dataset for 1 epoch with a batch size of 128. We tested how this model performs out of a low-resource setting, achieving a 70.18% F1 score after 11k training samples. Notably, the model reaches nearly the same performance with just 1k samples. Suggesting that our framework is not efficient outside the low-resource constraint.

Samples	Accuracy	Macro Precision	Macro Recall	Macro F1-Score
2	0.5059	0.5062	0.5059	0.5009
3	0.6036	0.6243	0.6036	0.5863
5	0.5636	0.5908	0.5636	0.5283
10	0.6272	0.6369	0.6272	0.6205
20	0.6065	0.6225	0.6065	0.5933
50	0.5636	0.5925	0.5636	0.5266
100	0.5222	0.6268	0.5222	0.3980
500	0.5533	0.6304	0.5533	0.4757
1000	0.6923	0.6927	0.6923	0.6921
2000	0.5666	0.5942	0.5666	0.5323
11000	**0.7041**	**0.7108**	**0.7041**	**0.7018**

Table 5: Training metrics from the bart base MNLI fine-tuned model out of low-resource constraints.

In table 6 we show the training metrics of the mpnet-base model out of low-resource constrains, we can see it achievesits peak performance at 500 samples with an F1 score of 70.95%, it was trained on a batch size of 64 for 1 epoch. This result again suggests that our framework is not efficient outside the low-resource constraint.

Samples	Accuracy	Macro Precision	Macro Recall	Macro F1-Score
2	0.4541	0.3997	0.4541	0.3685
3	0.5370	0.5740	0.5370	0.4708
5	0.5784	0.6022	0.5784	0.5524
10	0.6420	0.6463	0.6420	0.6394
20	0.6538	0.6556	0.6538	0.6529
50	0.6864	0.6872	0.6864	0.6861
100	0.5695	0.6053	0.5695	0.5296
500	**0.7115**	**0.7177**	**0.7115**	**0.7095**
1000	0.7101	0.7184	0.7101	0.7073
2000	0.7101	0.7189	0.7101	0.7071
11000	0.7101	0.7189	0.7101	0.7071

Table 6: Training metrics from the mpnet-base model out of low-resource constraints.

Across all these setups, we can clearly see that the metrics stop improving at a relatively low sample size. During training, we noticed that the loss function often produced very low outputs, causing the gradient signal to be almost zero and leading to an overfitting-like effect. We conclude that this sample size limitation is part of the SetFit framework, and new methods would be needed to overcome it.

6 Conclusion

Based on these results, we can conclude that our approach is a successful and viable solution for training few-shot hallucination detection systems. We want to emphasize how much more practical our method is compared to standard approaches. Achieving a good F1 score on RAGTruth or any other hallucination detection dataset does not guarantee that a model will perform well when deployed in different domains. With our approach, it is possible to create domain-specialized models for hallucination detection using just a few examples. This makes the LLM-as-a-judge method the only direct competitor, as other approaches typically require large training sets to reach similar performance. In scenarios with limited resources, or when LLM judges are not a viable option, we believe our framework represents the state of the art for hallucination detection.

We also conclude that transfer learning from a natural language inference (NLI) domain to an intrinsic hallucination detection domain is both viable and recommended when building few-shot models. Another important finding is the framework's limited ability to take advantage of larger datasets, which restricts its usefulness in situations where SOTA performance is needed, and resources are not a constraint.

Our framework is recommended for scenarios where teams want to add quality testing to their LLM-powered applications but don't have large amounts of data to train a model using a traditional approach. This model can be used temporarily while an automated pipeline collects system interactions, which can later be used to train a new model. This approach would generate a high-quality dataset focused on the target domain, and once enough samples are collected, our approach can be replaced with a higher-performing model.

References

[1] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. tau Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive nlp tasks," 2021.

[2] OpenAI, "Llm tool web search." `https://platform.openai.com/docs/guides/tools-web-search?api-mode=responses`, 2025. Accessed: 2025-12-03.

[3] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," 2021.

[4] A. Khan, M. Z. Khan, S. Jamshed, S. Ahmad, A. Zainab, K. Khatib, F. Bibi, and A. Rehman, "Advances in llms with focus on reasoning, adaptability, efficiency and ethics," 2025.

[5] C. Sharma, "Retrieval-augmented generation: A comprehensive survey of architectures, enhancements, and robustness frontiers," 2025.

[6] OpenAI, "Gpt-5.1: A smarter, more conversational chatgpt." `https://openai.com/index/gpt-5-1/`, 2025. Accessed: 2025-12-03.

[7] S. Hughes, M. Bae, M. Li, and I. Vectara, "Vectara Hallucination Leaderboard." Hugging Face Space, 2023. Continuously updated. Based on the Hughes Hallucination Evaluation Model (HHEM).

[8] D. Larson, A. Koirala, L. Chuey, M. Paschali, D. Van Veen, H. S. Na, M. Petterson, Z. Fang, and A. S. Chaudhari, "Assessing the completeness of clinical histories accompanying imaging orders using open- and closed-source large language models," *Radiology*, 2024.

[9] M. Siino, M. Falco, D. Croce, and P. Rosso, “Exploring llms applications in law: A literature review on current legal nlp approaches,” *IEEE Access*, vol. 13, pp. 18253–18276, 2025.

[10] Y. Bang, Z. Ji, A. Schelten, A. Hartshorn, T. Fowler, C. Zhang, N. Cancedda, and P. Fung, “Hallulens: Llm hallucination benchmark,” 2025.

[11] Y. Ming, S. Purushwalkam, S. Pandit, Z. Ke, X.-P. Nguyen, C. Xiong, and S. Joty, “Faitheval: Can your language model stay faithful to context, even if "the moon is made of marshmallows",” 2025.

[12] R. Kamoi, S. S. S. Das, R. Lou, J. J. Ahn, Y. Zhao, X. Lu, N. Zhang, Y. Zhang, R. H. Zhang, S. R. Vummanthala, S. Dave, S. Qin, A. Cohan, W. Yin, and R. Zhang, “Evaluating llms at detecting errors in llm responses,” 2024.

[13] D. Janiak, J. Binkowski, A. Sawczyn, B. Gabrys, R. Shwartz-Ziv, and T. J. Kajdanowicz, “The illusion of progress: Re-evaluating hallucination detection in LLMs,” in *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing* (C. Christodoulopoulos, T. Chakraborty, C. Rose, and V. Peng, eds.), (Suzhou, China), pp. 34716–34733, Association for Computational Linguistics, Nov. 2025.

[14] E. Yan, “Evaluating llms: Summarization, consistency, relevance, length.” `https://eugeneyan.com/writing/evals/`, 2023.

[15] L. Tunstall, N. Reimers, U. E. S. Jo, L. Bates, D. Korat, M. Wasserblat, and O. Pereg, “Efficient few-shot learning without prompts,” 2022.

[16] C. Niu, Y. Wu, J. Zhu, S. Xu, K. Shum, R. Zhong, J. Song, and T. Zhang, “Ragtruth: A hallucination corpus for developing trustworthy retrieval-augmented language models,” 2024.

[17] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, and P. Fung, “Survey of hallucination in natural language generation,” *ACM Comput. Surv.*, vol. 55, Mar. 2023.

[18] Z. Xu, S. Jain, and M. Kankanhalli, “Hallucination is inevitable: An innate limitation of large language models,” 2025.

[19] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, “Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension,” 2019.

[20] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, “Qlora: Efficient finetuning of quantized llms,” 2023.

[21] K. Krishna, P. Gupta, S. Ramprasad, B. C. Wallace, J. P. Bigham, and Z. C. Lipton, “Usb: A unified summarization benchmark across tasks and domains,” 2023.

[22] D. Tam, A. Mascarenhas, S. Zhang, S. Kwan, M. Bansal, and C. Raffel, “Evaluating the factual consistency of large language models through news summarization,” 2023.

[23] M. Li, R. Luo, and O. Mendelevitch, “HHEM-2.1-Open,” 2024.

[24] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, E. Li, X. Wang, M. Dehghani, S. Brahma, A. Webson, S. S. Gu, Z. Dai, M. Suzgun, X. Chen, A. Chowdhery, S. Narang, G. Mishra, A. Yu, V. Zhao, Y. Huang, A. Dai, H. Yu, S. Petrov, E. H. Chi, J. Dean, J. Devlin, A. Roberts, D. Zhou, Q. V. Le, and J. Wei, “Scaling instruction-finetuned language models,” 2022.

[25] Y. Dong, R. Mu, Y. Zhang, S. Sun, T. Zhang, C. Wu, G. Jin, Y. Qi, J. Hu, J. Meng, S. Bensalem, and X. Huang, “Safeguarding large language models: A survey,” 2024.

[26] Ádám Kovács and G. Recski, “Lettucedetect: A hallucination detection framework for rag applications,” 2025.

[27] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, “A large annotated corpus for learning natural language inference,” in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (L. Màrquez, C. Callison-Burch, and J. Su, eds.), (Lisbon, Portugal), pp. 632–642, Association for Computational Linguistics, Sept. 2015.

[28] A. Williams, N. Nangia, and S. R. Bowman, “A broad-coverage challenge corpus for sentence understanding through inference,” 2018.

[29] N. Reimers and I. Gurevych, “Sentence-bert: Sentence embeddings using siamese bert-networks,” 2019.

[30] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: pre-training of deep bidirectional transformers for language understanding,” *CoRR*, vol. abs/1810.04805, 2018.

[31] T. Gao, X. Yao, and D. Chen, “Simcse: Simple contrastive learning of sentence embeddings,” 2022.

[32] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” 2019.