Two-tiered Encoder-based Hallucination Detection for Retrieval-Augmented Generation in the Wild

Ilana Zimmerman[*], **Jadin Tredup**[†], **Ethan Selfridge**[§], **Joseph Bradley**[‡]
[*]Numa, [†]Dickinson-Wright PLLC., [§]LivePerson Inc., [‡]IDG
ilana@numa.com jtredup@dickinson-wright.com
eselfridge@liveperson.com joseph.bradley@idg.com

Abstract

Detecting hallucinations, where Large Language Models (LLMs) are not factually consistent with a Knowledge Base (KB), is a challenge for Retrieval-Augmented Generation (RAG) systems. Current solutions rely on public datasets to develop prompts or fine-tune a Natural Language Inference (NLI) model. However, these approaches are not focused on developing an enterprise RAG system; they do not consider latency, train or evaluate on production data, nor do they handle non-verifiable statements such as small talk or questions. To address this, we leverage the customer service conversation data of four large brands to evaluate existing solutions and propose a set of small encoder models trained on a new dataset. We find the proposed models to outperform existing methods and highlight the value of combining a small amount of in-domain data with public datasets.

1 Introduction

In the last year, Large Language Models (LLMs) have exploded in popularity, in part due to their ability to convincingly answer arbitrary questions. Retrieval-Augmented Generation (RAG), which injects portions of external knowledge bases into the prompt, is an effective method for introducing specific information for a given brand or use case. However, hallucinations, where the system provides an ungrounded response, threatens the viability of this application in an industry setting.

This paper proposes and evaluates a novel encoder-based classifier for hallucination detection tailored for enterprise customers. Our model, RAGHalu, is an encoder-based two-tiered solution that leverages one binary classifier in each tier. RAGHalu first identifies factually verifiable statements and then determines whether each verifiable statement is supported or unsupported by the KB. Whereas other works either do not handle non-verifiable (e.g. small talk or information gathering) statements (Honovich et al., 2022; Gekhman et al., 2023; Muhlgay et al., 2023), or group them with other types of verifiable claims (Gupta et al., 2022), we developed a 3-label taxonomy to distinguish between the two. Our model is trained on both re-annotated and original public datasets, and internal in-domain data. Although there are recent studies such as Wang et al. (2023) using ChatGPT in a similar two-tiered solution, to the best of our knowledge, this is the first hallucination detection solution developed that explicitly identifies verifiable claims and leverages them as atomic claims (cf. Min et al. 2023).

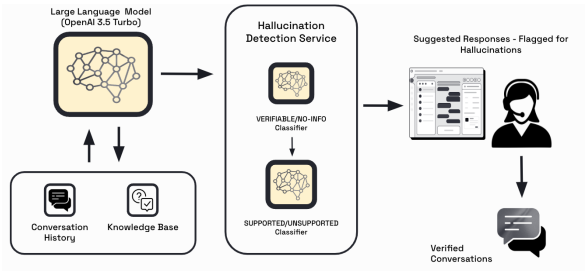


Figure 1: RAG customer service system with RAGHalu, the two-tiered hallucination detection service, and human agent in the loop.

We compare RAGHalu against a number of baselines: prompt-engineering OpenAI's GPT-3.5-turbo-0613[1] (OpenAI, 2023), a hallucination detection fine-tuned Mistral-7B-Instruct LLM, and open source hallucination detection models by Google Honovich et al., 2022 and Vectara.[2] We find our two-tiered solution which further fine-tunes a natural language inference (NLI) [3] DeBERTa (He et al., 2021) cross-encoder model performs best and generalizes both across customer service domains and open source data. Figure 1 shows RAGHalu inte-

[1]We refer to this model as ChatGPT throughout.

[2]`Mistral-7B-Instruct-v0.1`, `t5_xxl_true_nli_mixture`,`hallucination_evaluation_model`

[3]`cross-encoder/nli-deberta-v3-large`

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track, pages 8–22
November 12-16, 2024

grated into a customer service RAG system.

The paper is organized as follows. We first present our model architecture and the data used to train it. We then outline the baselines, followed by the results and discussion.

2 Related Work

Hallucination Detection in Language Model Generated Text Recently there has been work around factual consistency detection in relation to LLM summarization (Gekhman et al., 2023; Wu et al., 2023). In these works they discuss the shortage of annotated data for this task and attempt to mitigate the issue by using model-generated soft labels. In Gekhman et al. (2023) they improved upon the 11B parameter T5 model in Honovich et al. (2022), and speculate that LLM-produced data leads to improved performance over the human-perturbed data that was used for the original model.

There has also been work aimed at judging the factual precision of LLMs without retrieval. Muhlgay et al. (2023) assess LM factualness as related to generated token perplexity. They find that while perplexity is related to factualness, it is not enough to identify hallucinations on its own. Tian et al. (2023) fine-tunes LLMs for factualness using model uncertainty.

Within the area of question-answering and RAG, there has been a variety of work aimed at using LLMs to self-verify factual consistency with prompting (Min et al., 2023; Wang et al., 2023; Manakul et al., 2023). Though these prompts were shown to be effective, using an LLM to self-judge remains impractical and expensive in a large scale industry setting.

Fact Verification Similar to other works, we judge factual consistency on a sentence level (Thorne et al., 2018; Honovich et al., 2022), we consider a "checkworthiness"/verifiable statement type (Wang et al., 2023; Gupta et al., 2022; Mishra et al., 2024), and we fine-tune an NLI model. However, unlike the aforementioned works, we train and evaluate on real commercial data, we train a model to distinguish between verifiable and non-verifiable claims, we fine-tune our model on a new collection of LLM generated texts, and we produce an end-to-end solution that does not rely on prompting of an LLM for classification.

3 RAGHalu

3.1 Architecture

RAGHalu input includes the user question, retrieved knowledge articles, and LLM response and outputs a prediction of whether each sentence in the LLM response is supported by the knowledge articles. See Table 1 for an example[4]. RAGHalu uses two sequential classifiers involving binary models where the first acts as a filter to the second. The first model (RAGHalu-1) classifies statements according to whether they contain information that can be proven true or false, resulting in two labels: VERIFIABLE and NO-INFO. Statements such as "we can look into that for you," "please visit a branch for assistance," or small talk, would be classified as NO-INFO since they do not contain information that can be checked for validity. The second model (RAGHalu-2) classifies all VERIFIABLE statements as SUPPORTED or UNSUPPORTED based on whether there is corroborating information in the retrieved articles. We considered a single 3-label encoder model, however it often confused UNSUPPORTED and NO-INFO claims.[5]

We chose to use a DeBERTa encoder model for each binary model for several reasons. First, from an industry practicality standpoint - they are much smaller and faster. Further, encoders are known to be more powerful than decoder-only models because they encode relative context both in front of and behind each token (Devlin et al., 2019). While LLMs are often preferred for their high token limits, the relative position embeddings of the DeBERTa encoder model allows for a theoretical maximum token limit of 24,528.[6]

3.2 Data

RAGHalu is trained on both public and private brand data. Data statistics are shown in Table 2. Test-train splits for the data are shown, all of which involved a random stratified split across labels.[7] All fine-tuned model results use the same training splits described in Table 2 unless explicitly stated.

Public Data Between 2022 and 2023, several LLM hallucination datasets have been released including TruthfulQA(Lin et al., 2022),

[4]Exact input formatting in A.3

[5]Details in A.5.3

[6]See section A.5 of He et al. (2021))

[7]In addition to the train-test split, a random 2% was taken from the train set for model evaluation/monitoring during training making for a 78.4-1.6-20 train-eval-test split.

Conversation History	USER: loan assistance ASSISTANT: Ok great, is this for a new loan? USER: Yes ASSISTANT: OK, what kind of loan? USER: i need an auto loan
Question	i need an auto loan
KB	We offer various low interest home, auto, and personal loans. As of January 2024, a typical 30-year mortgage rate for first time home owners is 5.5%. To learn more and apply for a loan, visit us at www.yourbank.com/loans
LLM Response	*Sure I can help with that.* As of January 2024 you can expect an auto loan with interest rates as low as 5.5%. To learn more check out our website at www.yourbank.com/loans.
Classification	*NO-INFO* UNSUPPORTED SUPPORTED

Table 1: Synthetic example inputs (question, KB), LLM response, and expected sentence-level classification. The formatting of the LLM Response matches that of its respective classification.

Data Set Name	**NO-INFO**	**SUPPORTED**	**UNSUPPORTED**	**Train/Test Split**	**Avg. Input Length** (tokens)± std
FactScore (fs)	0	10,106	4,419	80/20	1219±311
HaluEval (halu)	0	10,000	10,000	80/20	105±39
TruthfulQA (tqa)	35	406	1226	80/20	630±374
Dolly	62	415	414	80/20	207±288
TRUE**	0	47,680	57,441	0/100	48±151
wiki-bio-gpt3	0	516	1392	0/100	285±147
Bank	70	64	68	80/20	144±63
Credit Union	180	53	78	80/20	106±131
Telecom	330	71	159	80/20	322±121
FinTech	230	104	168	80/20	204±137

Table 2: Open-source and brand data statistics showing support numbers per label. Brand statistics are below the horizontal line with italicized names. The relative train-test split used for model development and testing, along with the average input lengths in tokens are also show (DeBERTa-v3 tokenizer). **For more information about the breakdown of the TRUE dataset see Honovich et al. (2022)

FactScore(Min et al., 2023), HaluEval(Li et al., 2023a), ExpertQA(Malaviya et al., 2023), and Wiki-Bio-GPT3(Manakul et al., 2023). The combined dataset TRUE described in Honovich et al. (2022), consists of data across domains including paraphrasing, summarization, dialogue, and QA.

We used four public datasets for model development: FactScore, HaluEval, TruthfulQA and Databricks Dolly(Conover et al., 2023). We filtered and re-annotated subsets of data from TruthfulQA and Dolly to align with our taxonomy and better reflect an emphasis on hallucinations relative to retrieved knowledge instead of absolute truth. We released this data including formatted training/test sets.[8] For more details on the changes made to these datasets see Appendix.A.1.

Brand Data We annotated conversations across four large brands: a bank (*Bank*), broadband provider (*Telecom*), credit union (*Credit Union*), and a crypto-currency software company (*FinTech*), all using RAG in production today. For each brand, we annotated ∼50 historical conversations each with one or more retrieved (KB) and LLM generated response in the conversation.[9] Data procurement and annotation consisted of several steps. First we queried for historical conversations where

[8] github.com/ilanazim/RAGHalu_public_data

[9] KBs are only used for RAG when the article has an embedding match score above a brand-specified threshold

the brand used LLM suggestions in a RAG setting. Currently, all brands in production use GPT-3.5-turbo, however, to get more variation in LLM responses that are also usable for commercializeable model development (e.g. RAGHalu), we prompted Xwin-LM-70b, llama2-70b-chat, falcon-7b-instruct, and llama2-13b to respond as the AI Assistant given the conversation history and KBs.[10]

The historical brand conversation along with retrieved articles and the generated LLM responses were span-annotated by three domain expert annotators. Annotators were instructed to annotate sentences according to the above taxonomy, and to skip any incomplete sentences that may have arisen due to LLM token limits. Across the four brands, the average Fleiss' kappa (Fleiss, 1971) for inter-annotator agreement was 0.79, indicating substantial agreement. Brand data is proprietary and will not be released.

4 Experimental Setup

In addition to evaluating three open source NLI-based models on the SUPPORTED/UNSUPPORTED examples, we compare the performance of RAGHalu with prompting ChatGPT and fine-tuning Mistral-7b. Similar to other works (Thorne et al., 2018; Honovich et al., 2022; Wang et al., 2023) we split the response into sentences using the NLTK sentence tokenizer(Bird et al., 2009) for classification.

4.1 Baselines

Prompt Engineering ChatGPT's zero-shot performance has proven to be a competitive baseline for hallucination detection systems (Huang et al., 2023). Though cost and latency remain a concern, we chose to use prompt-engineering as a baseline and interim production solution.

We developed both a 3-label (SUPPORTED, UNSUPPORTED, NO-INFO) prompt and a similar binary prompt (SUPPORTED, UNSUPPORTED) to classify LLM sentences with respect to a set of retrieved KBs[11]. All prompt-engineered results shown are for GPT-3.5-turbo. While generative models like ChatGPT have the ability to classify more than one statement at a time, we found that performance is significantly better when the model classifies a single statement at a time.[12] For this reason, all ChatGPT results shown in Section 4.2 are for single-sentence classification.

Decoder LLM Fine-tuning In addition to prompt-engineering instruction-following LLMs, there has been recent work such as Li et al. (2023b) which researches the affect of fine-tuning LLMs for classification. LLMs are acclaimed for their learned world knowledge and large token limits. Because grounding context for hallucination detection can vary widely in length, we chose to compare fine-tuned LLMs to an encoder based solution in order to judge if the fine-tuned LLM would outperform the encoder on longer inputs.

Using the same prompts developed for zero-shot prompting, we experimented with fine-tuning several open source LLMs. In the context of RAG, the model was given the input prompt including the user's question, retrieved KBs, and a sentence from the LLM Response (the statement being classified for factual consistency, see Table 1), and was trained to produce one of the labels from our taxonomy. The mistral model was fine-tuned using Deepspeed Zero Stage 1 optimization (Rajbhandari et al., 2020), batch size of 1, gradient accumulation steps of 4, floating point 16 precision, a learning rate of 5e-6, and 4 epochs. The maximum token limit for this model is 8000.

4.2 Results

As shown in Table 3 our tier two (RAGHalu-2) model performs best on production brand data with an average UNSUPPORTED F1 of 0.93, followed closely by the fine-tuned binary Mistral model (mistral-7b-ft-binary). Surprisingly, google/t5_xxl_true_nli_mixture outperforms all other models on the *Bank* test set with a high score of 0.96, and RoBERTa-large-mnli performs best on *Credit Union* data by a significant margin with an F1 of 0.97. While zero-shot prompting (ChatGPT-binary) performs well on brand data, the fine-tuned LLM and encoder models show significant improvements (10% F1 on UNSUPPORTED claim detection on average). RAGHalu-2 also performs best across the board on open source data with an average UNSUPPORTED F1 of 0.82.

Our model's largest performance gain relative to other models on open source data is on the FactScore test set. We hypothesize this is due to the long grounding context/KB lengths in the

[10] `Xwin-LM/Xwin-LM-70B-V0.1`, `meta-llama/Llama-2-70b-chat-hf`, `tiiuae/falcon-7b-instruct`, `meta-llama/Llama-2-13b-chat-hf`

[11] Prompts found in A.2

[12] For details see A.5.2

Data Set	ChatGPT -binary§	Vectara§	google/t5_xxl_ true_nli_mixture§	mistral-7b -ft-binary†	RoBERTa -mnli**§	RAGHalu-2†
HaluEval	0.71	0.8	0.79	0.79	0.68	**0.95**
FactScore	0.66	0.6	0.35	0.73	0.45	**0.8**
TruthfulQA	0.81	0.84	0.68	**0.87**	0.85	0.84
Dolly	0.68	0.77	**0.8**	0.63	0.74	0.65
Wiki-Bio-GPT3	**0.9**	0.88	0.81	0.85	0.85	0.88
PAWS	**0.74**	-	-	0.15	0.57	0.64
VitaminC	**0.76**	-	-	0.71	0.74	0.71
FEVER	**0.91**	-	-	0.73	0.89	0.86
TRUE*	0.85	**0.87**	0.78	0.81	0.78	0.79
Avg*	0.77	0.79	0.70	0.78	0.73	**0.82**
Avg (open source all)	0.78	-	-	0.70	0.73	**0.79**
Bank	0.8	0.83	**0.96**	0.87	0.61	0.95
Telecom	0.85	0.81	0.9	0.96	0.83	**0.97**
FinTech	0.82	0.7	0.73	0.86	0.55	**0.87**
Credit Union	0.86	0.85	0.87	0.95	**0.97**	0.92
Avg (brand data)	0.83	0.8	0.87	0.91	0.74	**0.93**

Table 3: Binary SUPPORTED/UNSUPPORTED model results. F1 score for the UNSUPPORTED class shown. *Vectara and google/t5_xxl_true_nli_mixture were trained using PAWS, VitaminC, and FEVER so we calculate average scores without those results. TRUE performance is TRUE data minus FEVER,PAWS,VitaminC. **Note: RoBERTa-NLI "neutral" predictions were mapped to "UNSUPPORTED". † Indicates the model was fine-tuned in this work. § Indicates the model was used without fine-tuning, either with prompting or following expected input format.

Label	ChatGPT	RAGHalu
NO-INFO	0.71	**0.92**
VERIFIABLE	0.85	**0.91**
SUPPORTED	0.84	**0.94**
UNSUPPORTED	0.75	**0.93**
NO-INFO	0.71	**0.91**
SUPPORTED	0.77	**0.89**
UNSUPPORTED	0.60	**0.85**

Table 4: End-to-end systems: Average F1 scores across brand test sets comparing RAGHalu to GPT-3.5-turbo using the 3-label prompt. 3-label model performance is mapped to the 2 binary label sets by converting (SUPPORTED/UNSUPPORTED) labels to VERIFIABLE.

FactScore dataset relative to others as shown in Table 2. We explore the relationship between input length and model correctness/max token limits in the Discussion 4.3 below.[13]

End-to-end model performance including filtering of NO-INFO labels in the first tier of RAGHalu resulted in a performance gain of 0.25 relative to the 3-label ChatGPT baseline for flagging unsupported claims as shown in Table 4.

4.3 Discussion

Training Data and Model Generalization We further explored the effects of training data by comparing performance of three further fine-tuned DeBERTa-based NLI models: one trained on only production brand data, one trained on only the open source data specified in Table 2, and finally one trained first on the open source data and further fine-tuned on brand data.

We found that the model trained on brand only data does not generalize to the open source data, but performs equally as well if not better across the four brand test sets. The open source only model performs well on brand data, but the addition of brand data pushed performance up across all brand test sets. These results again highlight the importance of domain specific training data.[14]

Input Length Analysis As shown in Figure 2 there is a clear relationship between model correctness and input length - the longer the input, the more incorrect predictions. The models with lower token limits such as RoBERTa-large-mnli, google/t5_xxl_true_nli_mixture, and Vectara all suffer more than the models that have longer max token limits. RoBERTa-large-mnli likely suffers the most due to a combination of input lengths seen at training time, domain,[15] and token limits. The Spearman's correlation[16] between the number of input tokens and proportion incorrect predictions is statistically significant across all models.[17]

[13] Additional error analysis found in A.6.

[14] Appendix Figure 3 shows the ROC plots of these results

[15] See information about the MNLI training data here

[16] `scipy.stats.spearmanr`

[17] correlations range from 0.76–0.97 with p-values ≤ 0.0017

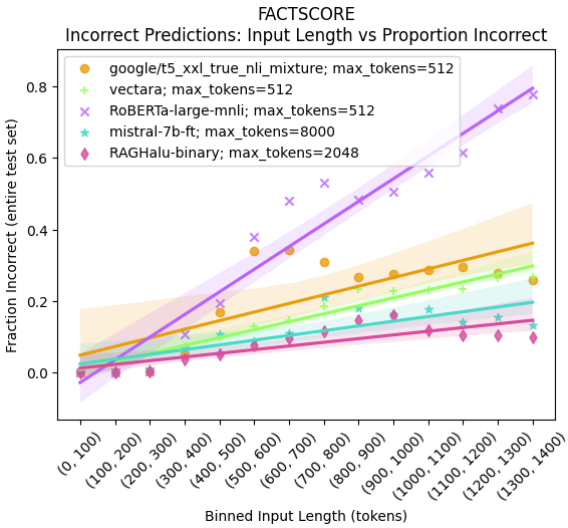


Figure 2: Plot showing prompt length bins (tokens) versus fraction incorrect prediction by model for the test set of the FactScore dataset.

Impact of Model Size and Architecture RAGHalu-2, a 304M parameter encoder-based model, outperforms the 7B parameter decoder-only mistral-7b-ft on almost all datasets tested. These findings are consistent with Zhang et al. (2023); Benayas et al. (2024) which highlight the shortcomings of decoder-based LLMs for classification tasks that smaller encoder models excel in and the importance of relative position encoding.

Error Analysis We found three types of recurring errors in RAGHalu predictions: mostly-supported statements, inconsistent taxonomies, and incorrect labels. Mostly-supported statement errors occur when a majority of the information is correct save for minor details, and in statements where the information is technically all supported but there is implied information that would be unsupported. An example of the latter is: "After the Revolutionary War, Blair returned to South Carolina and served in the state legislature." This statement implies that Blair was alive during the Revolutionary War when in fact they were not. Others have used an LLM to generate atomic claims to avoid classifying sentences with multiple statements like these, however that approach is less practical in production.[18]

Practicality in Production The relative cost of using an in-house model versus a third party such as OpenAI is multi-faceted: one must consider performance, inference speeds, costs, and model monitoring (Howell et al., 2023). In addition to performance gains, we estimated the cost savings of using RAGHalu versus ChatGPT as a hallucination classifier and find that RAGHalu is at least 5x less expensive per inference. For a real telecom brand with 2 million conversations per month and an average of 5 LLM responses per conversation, expected savings is upwards of $105k per year.[19] The same framework can be used to compare self-hosted LLMs to smaller encoders.

5 Future Work

Future work could include developing a more fine-grained hallucination detection model as done in Mishra et al. (2024). Examples include distinguishing between unsupported and contradicting claims and identifying statements of action such as "I found your account number", which could indicate a need for an API integration. Correcting or mitigating hallucinations by improving KB chunking are also important considerations.

6 Conclusion

We developed a novel encoder-based hallucination classifier optimized for performance on customer service RAG bots in enterprise. Our models are trained on a new collection of open source and private data that generalizes and outperforms other models tested. We demonstrated the need for domain specific training data for hallucination detection, as well as the importance of KB lengths used in RAG.

Limitations

The relevance of the hallucination detection model for RAG systems is only as useful as the KB articles and their retrieval system. If all retrieved articles are ill-fitting to the conversation, most all statements will be flagged for hallucination. Further, this model was developed specifically for customer service RAG systems and has been shown to underperform on other types of data such as paraphrases or summarization.

Ethics Statement

While the hallucination detection system is developed to act as a safety net/guardrail for information produced by LLMs, if the model fails to detect a hallucination, it is possible that misinformation is

[18] See A.6 for more error analysis examples

[19] Details on calculations are found in A.7

spread to users. Privacy concerns related to personally identifiable information (PII) are also very important when using customer service chat data. We pseudo-anonymized all customer data prior to model training and evaluation.

Acknowledgements

We want to thank Michael Goodman for proof reading and sharing the final public dataset, and Julianne Marzulla, Cecilia Moran, and Daniel Gilliam who annotated the data for RAGHalu. We also appreciate engineering assistance that we received from Tyson Chihaya and Fazlul Shahriar.

References

Alberto Benayas, Miguel Sicilia, and Marçal Mora-Cantallops. 2024. A comparative analysis of encoder only and decoder only models in intent classification and sentiment analysis: Navigating the trade-offs in model size and performance.

Steven Bird, Edward Loper, and Ewan Klein. 2009. Natural Language Processing with Python. In *Natural Language Processing with Python*. O'Reilly Media Inc.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378–382.

Zorik Gekhman, Jonathan Herzig, Roee Aharoni, Chen Elkind, and Idan Szpektor. 2023. TrueTeacher: Learning factual consistency evaluation with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2053–2070, Singapore. Association for Computational Linguistics.

Prakhar Gupta, Chien-Sheng Wu, Wenhao Liu, and Caiming Xiong. 2022. DialFact: A benchmark for fact-checking in dialogue. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3785–3801, Dublin, Ireland. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTa: Decoding-enhanced BERT with disentangled attention. *arXiv preprint arXiv:2006.03654*.

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.

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.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232*.

Junyi Li, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023a. HaluEval: A large-scale hallucination evaluation benchmark for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6449–6464, Singapore. Association for Computational Linguistics.

Zongxi Li, Xianming Li, Yuzhang Liu, Haoran Xie, Jing Li, Fu-lee Wang, Qing Li, and Xiaoqin Zhong. 2023b. Label supervised LLaMA finetuning. *arXiv preprint arXiv:2310.01208*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods. pages "3214 – 3252".

Chaitanya Malaviya, Subin Lee, Sihao Chen, Elizabeth Sieber, Mark Yatskar, and Dan Roth. 2023. ExpertQA : Expert-curated questions and attributed answers. *arXiv 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.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore:

Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.

Abhika Mishra, Akari Asai, Vidhisha Balachandran, Yizhong Wang, Graham Neubig, Yulia Tsvetkov, and Hannaneh Hajishirzi. 2024. Fine-grained hallucination detection and editing for language models. *arXiv preprint arXiv:2401.06855*.

Dor Muhlgay, Ori Ram, Inbal Magar, Yoav Levine, Nir Ratner, Yonatan Belinkov, Omri Abend, Kevin Leyton-Brown, Amnon Shashua, and Yoav Shoham. 2023. Generating benchmarks for factuality evaluation of language models. *arXiv preprint arXiv:2307.06908*.

OpenAI. 2023. Models: GPT-3. `https://platform.openai.com/docs/models/gpt-3-5`. Accessed: 2023-12-03.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020. ZeRO: memory optimizations toward training trillion parameter models. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '20. IEEE Press.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

Katherine Tian, Eric Mitchell, Huaxiu Yao, Christopher Manning, and Chelsea Finn. 2023. Fine-tuning language models for factuality. In *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*.

Yuxia Wang, Revanth Gangi Reddy, Zain Muhammad Mujahid, Arnav Arora, Aleksandr Rubashevskii, Jiahui Geng, Osama Mohammed Afzal, Liangming Pan, Nadav Borenstein, Aditya Pillai, Isabelle Augenstein, Iryna Gurevych, and Preslav Nakov. 2023. Factcheck-gpt: End-to-end fine-grained document-level fact-checking and correction of llm output. *arXiv preprint arXiv:2311.09000*.

Wenhao Wu, Wei Li, Xinyan Xiao, Jiachen Liu, Sujian Li, and Yajuan Lyu. 2023. WeCheck: Strong factual consistency checker via weakly supervised learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 307–321, Toronto, Canada. Association for Computational Linguistics.

Ying Xu, Xu Zhong, Antonio Jose Jimeno Yepes, and Jey Han Lau. 2020. Forget me not: Reducing catastrophic forgetting for domain adaptation in reading comprehension. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

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

A Appendix

A.1 Data Annotation

We filtered the original TruthfulQA dataset of 817 unique questions to a set of 206 questions based on: a) our ability to retrieve the related Wikipedia articles, and b) examples within the 2048 token limit that many LLMs are restricted to. The resultant dataset consists of 206 unique questions, their related Wikipedia articles, and a list of responses to the question. We annotated the responses according to our taxonomy and resulting support numbers are shown in Table 2.

Data from Dolly was procured as follows. First we sampled from the closed_qa portion of the Dolly dataset. This data was generated by crowd workers who were given a context and instructed to generate questions and answers based on that context. To generate examples of hallucinations, we split each response into individual sentences. Then we made each sentence an example of a hallucination by altering the context so that it either contradicts the answer or does not contain the answer. This is the only dataset used for training where LLMs did not produce the hallucinations.

The content of the remaining datasets was unmodified, save for formatting as described in Appendix A.2 and A.3

A.2 Prompts

VERIFIABLE/NO-INFO Prompt:

```
  The "Fact List" below repre-
sents responses to a user ques-
tion.  Your job is to determine
whether each response in the
"Fact List" can be factually ver-
ified.  If the response can be
factually verified mark the re-
sponse 'VERIFIABLE', otherwise
mark the response 'NO-INFO'. 'NO-
INFO'statements include responses
like "Is there anything else I
can help you with?", as well as
greetings and small talk that is
not intended to convey verifiable
```

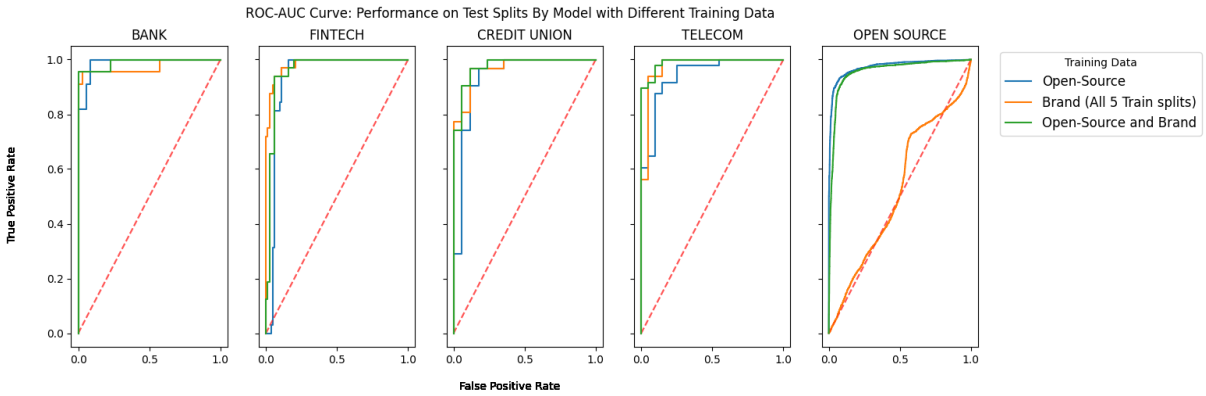


Figure 3: Receiver-operator curve showing the impact of various training data on performance by Brand. Opensource data here is performance on (HaluEval, FactScore, TruthfulQA, Dolly test splits, combined)

truths or falsehoods.

"Fact List": {agent_turn}

Fact Check:

SUPPORTED/UNSUPPORTED Prompt:

The "Fact List" below represents responses to a user question. Your job is to determine whether each response in the "Fact List" is supported by the information in the "Provided Text". Apply one of the following labels to each response in the "Fact List":

* SUPPORTED: use this label if the response is found in the "Provided Text".
* UNSUPPORTED: use this label if the response is either not found or contradicted in the "Provided Text".

"Question": {user_turn}

"Provided Text": {retrieved_knowledge}

"Fact List": {agent_turn}

Fact Check:

3-Label Prompt:

The "Fact List" below represents responses to a user question. Your job is to determine whether each response in the "Fact List" is supported by the information in the "Provided Text". Apply one of the following labels to each response in the "Fact List":

* SUPPORTED: use this label if the response is found in the "Provided Text".
* UNSUPPORTED: use this label if the response is either not found or contradicted in the "Provided Text".
* NO-INFO: use if the response does not present information that can be factually verified. This includes responses like "Is there anything else I can help you with?", as well as greetings and small talk that is not intended to convey verifiable truths or falsehoods.

Examples:
1. How are you today? - NO-INFO

"Question": {user_turn}

"Provided Text": {retrieved_knowledge}

"Fact List": {agent_turn}

Fact Check:

Training Data	*Bank*	*Telecom*	*FinTech*	*Credit Union*
fs-halu-dolly-tqa	**0.99**	**0.94**	0.93	0.93
fs-halu-dolly	0.88	0.81	0.88	0.73
fs-halu-tqa	0.73	0.83	0.81	0.56
fs-dolly-tqa	0.97	0.91	**0.94**	**0.97**
halu-dolly-tqa	0.87	0.83	0.87	0.76
Telecom-FinTech-Credit Union	0.97	0.92	**0.99**	**0.98**
Bank-FinTech-Credit Union	**0.99**	0.92	0.97	**0.98**
Bank-Telecom-Credit Union	0.97	**0.99**	0.96	0.96
Bank-Telecom-FinTech	0.97	0.95	0.96	0.95

Table 5: Ablation Study: Comparing ROC-AUC on Brand data - ablating one training data source at a time. Models trained are binary (SUPPORTED/UNSUPPORTED) DeBERTa cross-encoder (similar to RAGHalu-2).

A.3 Encoder Inputs

Input format with only single user turn of context:

```
"Question": {user_turn}

{context}[SEP]{claim}
```

Input format with 3 previous turns of context[20]:

```
"Conversation":
USER:{prev_user_turn}
ASSISTANT:{agent_turn}
USER:{user_turn}

{context}[SEP]{claim}
```

A.4 Ablation Study

We performed an ablation study in which we systematically held-out different open source and brand data. Results are shown in Table 5. We found that all open source datasets used for training plays a role in model performance on brand test sets. Surprisingly, the biggest change in performance we see is when holding out the Dolly dataset. Performance drops over 10 points across the brand test sets. We hypothesize that Dolly has a big impact on performance because it was manually annotated according to our taxonomy and is less likely to deviate from our strict definition than other datasets used.

Finally, we found that the best performance on each of the four brands occurs when using training data from the respective brand. While the generalized model performs well, this result supports the opportunity to train brand-optimized hallucination detection models for improved performance.

A.5 Additional Experimentation

A.5.1 Multi-Staged Fine-Tuning

Given that the volume of brand data annotated is quite small for model fine-tuning, especially relative to the volume of open source data, we evaluated the impact of training with a mixture of brand and open source data versus a multi-stage fine-tuning approach of first fine-tuning with open source data followed by fine-tuning on brand data. A concern with this method is related to the issue of "catastrophic forgetting" (Xu et al., 2020); the further fine-tuned model tends to unlearn and underperform on tasks relative to the original model. Our findings, summarized in Table 6 below, reinforce this known issue.

The model fine-tuned on open source data only (stage 1) outperforms the same model that is then further fine-tuned on customer service brand data (stage 2) on FactScore, HaluEval, Dolly, and TruthfulQA test sets. When comparing the multi-stage fine-tuning to a single stage, multi-stage fine-tuning does however improve domain specific performance on our customer service brand datasets, and is statistically significant.[21]

Test Data	Singe-Stage	Multi-Stage
Bank	0.95	**0.97**
FinTech	0.9	**0.92**
Telecom	0.9	**0.96**
Credit Union	**0.9**	0.89
fs-halu-dolly-tqa	**0.93**	0.91

Table 6: Comparing performance of single vs multi-stage RAGHalu-2 fine-tuning. Single-stage was trained with a mixture of the open source and brand data whereas multi-stage was trained with only open source data first, then further with only brand data. Micro F1 score reported.

[20] We experimented with using conversation context in training and saw no meaningful impact on model performance.

[21] Both the Wilcoxon Signed Rank Test and McNemar's t-test result in p-values < 0.05

A.5.2 Single Versus Multi-Sentence Prompting

To reduce inference time and overall cost, we also prompted/trained and evaluated the decoder models (GPT-3.5-Turbo and Mistral-7b) to classify multiple sentences at a time. Most generated messages consist of multiple sentences, each requiring hallucination classification. Classifying multiple sentences at once reduces the amount of required model calls and thus decreases inference time per message. We found, however, that classifying a single sentence at a time consistently outperformed classifying multiple sentences in one call for both Turbo and Mistral. Further, with multi-sentence classification we found the decoder model failed to produce a classification for all statements more often. Preferring high performance over latency, we ultimately chose to move forward with single sentence classification only. Examples of statements that were misclassified by multi-sentence models but correctly classified by single-sentence models can be seen in Appendix A.6.5.

Model	F1 (Average across brands)
Mistral Single-Sentence	**0.94**
Mistral Multi-Sentence	0.87
ChatGPT Single-Sentence	**0.88**
ChatGPT Multi-Sentence	0.83

Table 7: Micro average F1 performance comparison of models trained to classify single sentences vs. multiple sentences in a single model call.

A.5.3 Two-Tier vs 3-Label Model

We experimented with a single 3-label model instead of the two-tiered RAGHalu solution presented in this paper. We found that the 3-label solution consistently under-performed relative to the two-tiered approach. After performing error analysis comparing the two systems we found this was mainly because NO-INFO and UNSUPPORTED claims were confused with one another. A comparison is of end-to-end performance is shown in Table 8.

Label	RAGHalu	RAGHalu-3-label
NO-INFO	**0.92**	0.91
VERIFIABLE	0.91	0.89
SUPPORTED	**0.94**	0.90
UNSUPPORTED	**0.93**	0.82
NO-INFO	0.91	0.89
SUPPORTED	0.89	0.90
UNSUPPORTED	**0.85**	0.79

Table 8: Comparing end-to-end systems: Average F1 scores across brand test sets for double binary vs 3-label solutions. We extrapolate 3-label model performance across the 3 labels to the 2 binary label sets by converting (SUPPORTED/UNSUPPORTED) labels to VERIFIABLE.

A.6 Error Analysis

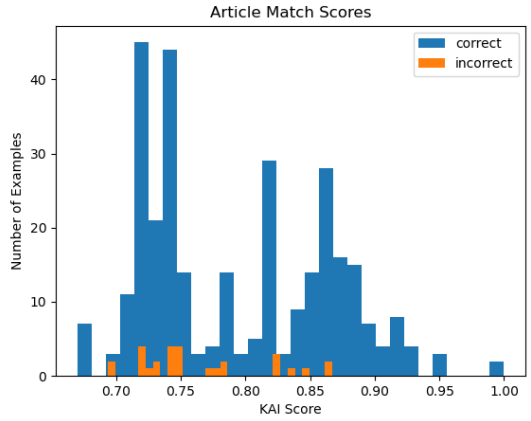
Fine-grained error analysis helps provide insight as to where our model is under-performing. This analysis is helpful to understand where our model under-performs, and whether or not incorrect classifications are a fault of the model or simply due to annotation errors or differences in taxonomy. Below are are various type of errors we analyzed along with examples from selected datasets.

A.6.1 KB Similarity Score

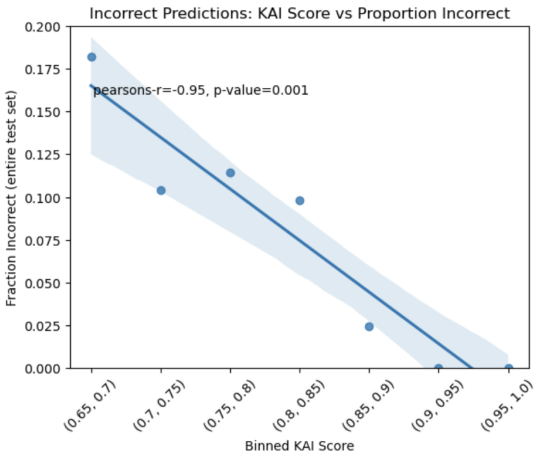
Each KB article retrieved in the brand datasets, has a score (KAI score) based on its similarity and/or relevance to the consumer's question which the AI Assistant is responding to. In Figure 4a we can see that a majority of the examples our model predicted incorrectly fall under a KAI score threshold of roughly 0.85. Corroborating this observation, we show in Figure 4b that there is a strong, statistically significant negative correlation ($r = -0.95$, $p = 0.001$) between KAI scores and the fraction of incorrect classifications. This correlation supports the possibility that we can improve hallucination detection model performance by introducing more relevant KBs to each brand's RAG database resulting in higher scoring articles.

A.6.2 Mostly-Supported Statements

Errors due to mostly-supported statements occur when the provided statement contains multiple verifiable independent pieces of information within a single sentence. Mostly-supported statement errors tend to come in two types: statements where a majority of the information is correct save for a few minor details and statements where the information is technically all supported but the language implies information that would be unsupported. Examples for both versions of this error can be seen in Table 9. In the first example, it can be verified through the evidence that information about the subjects racing career is correct, the only inaccuracies here are the dates spanning the subject's life. In the second example, the entire statement is technically true and supported in the text, however, there is an implication in the statement (that Blair was alive during the Revolutionary War) that is unsupported



(a) Article retrieval match scores for all examples.



(b) Correlation of article retrieval match scores compared with the fraction of examples classified incorrectly.

Figure 4: Article match scores compared to incorrect classifications.

Error Type: Mostly-Supported *(from wiki-bio-gpt3)*
Statement: Freddie Frith (1917–1994) was an English motorcycle racer who competed in the Isle of Man TT races and other international events. **Evidence:** "Frederick Lee "Freddie" Frith OBE (born 30 May 1909 in Grimsby, Lincolnshire, England – 24 May 1988) . . . five-time winner of the Isle of Man TT. . . Freddie also has the distinction of being the first ever 350 cc World Champion in 1949" **Gold Label:** UNSUPPORTED **Prediction:** SUPPORTED **Justification:** The information about his racing accomplishments is correct, the only inaccuracy are the birth and death dates.
Statement: After the Revolutionary War, Blair returned to South Carolina and served in the state legislature. **Evidence:** "James Blair (September 26, 1786 - April 1, 1834) was a United States Representative from South Carolina." **Gold Label:** UNSUPPORTED **Prediction:** SUPPORTED **Justification:** Blair did serve in the South Carolina legislature, and although this did occur after the revolutionary war, the implication is that he fought in the war hen he was actually born 3 years after its conclusion.

Table 9: Examples of mostly-supported statements.

and actually refuted by the provided evidence. Any single inaccuracy in either of these statements qualifies them for an UNSUPPORTED label, however our models tend to predict these as SUPPORTED given the prevalence of correct information.

A.6.3 Inconsistent Taxonomies

Another common source of errors are inconsistencies between the taxonomy we used to train our model and those used to create other datasets. A good example is the DialFact dataset which provides two classes of labels which do not have enough information for judgement, one for verifiable statements that do not have enough evidence to support or refute the claim, and another for personal statements (such as opinions) that are factually verifiable. For our purposes, we classify both of these statements as VERIFIABLE in tier one, and further classify each statement as supported or unsupported according to the evidence. For our use case, NO-INFO statements (greetings, small-talk, etc.) are not flagged as hallucinations because they do not effect the factual accuracy of the message as a whole. However, we do not want LLM responses to include opinionated statements which may bias a consumer. By classifying opinionated statements as VERIFIABLE, we allow them to be flagged as UNSUPPORTED by the second binary model, and possibly removed from the original LLM response. For reference, see the first example in Table 10.

Another example of inconsistent taxonomies are samples where one label is applied to a message with multiple statements. In these examples, the individual statements could potentially have conflicting labels, but by applying one to the entire message, we are unable to accurately evaluate the true performance of our model. For an example of a multi-statement message that should have conflicting labels see the second example in Table 10.

Error Type: Inconsistent Taxonomies *(from DialFact)*
Statement: I would have to say olive green is the worst. olive green and lavender are very closely related and look nothing alike. **DialFact Label:** NOT ENOUGH INFO **Gold Label:** NO-INFO **Prediction:** VERIFIABLE **Justification:** The statement is a personal opinion stated like a fact. Our taxonomy does not have a distinction for these types of messages and so it would be labeled as an UNSUPPORTED VERIFIABLE statement in our taxonomy.
Statement: I've never been to an actual blues festival, but i do like jazz. it's influenced by blues, country, folk, and many other genres. **DialFact Label:** NOT ENOUGH INFO **Gold Label:** NO-INFO **Prediction:** VERIFIABLE **Justification:** Our taxonomy is built to classify single sentences at a time and this example contains multiple sentences; one is a personal opinion stated similarly to a fact and another is a factual statement. Multiple sentences with potentially different labels can confuse a model built to classify single sentences at a time.

Table 10: Examples of inconsistent taxonomy errors.

Error Type: Incorrectly Labeled *(from wiki-bio-gpt3)*
Statement: He also oversaw the introduction of the FedEx Cup, a season-long points competition that culminates in a four-tournament playoff. **Evidence:** Timothy W. Finchem (born April 19, 1947) is the current Commissioner of Golf's PGA Tour. Finchem was born... received the 2001 Old Tom Morris Award from the Golf Course Superintendents Association of America, GCSAA's highest honor. He is a single-figure handicap golfer. **Gold Label:** SUPPORTED **Prediction:** UNSUPPORTED **Justification:** There is no indication in the evidence that he had anything to do with the FedEx cup and thus the correct gold label should actually be UNSUPPORTED.
Statement: Mahler was drafted by the Braves in the first round of the 1975 amateur draft. **Evidence:** Richard Keith Mahler... signed by the Braves as an amateur free agent in 1975... He was survived by his wife, Sheryl, and five children Ricky, Robby, Timothy, Tyler and Shannon. **Gold Label:** SUPPORTED **Prediction:** UNSUPPORTED **Justification:** Mahler was signed to the braves as an unsigned free agent and was not in the 1975 amateur draft, thus the correct gold label should be UNSUPPORTED.

Table 11: Examples of errors due to incorrect data labelling.

A.6.4 Incorrect Labels

Incorrect labelling errors are a simple case where the original label provided for the sample is deemed to be incorrect. These incorrect labels can sometimes come from information that is factually correct but not provided in the evidence (see the first examples in Table 11) or alternatively from information that was perhaps misinterpreted by the annotator (see the second example in Table 11). In these examples our model is scored as being incorrect in the automatic evaluation process, however we found it to be correct during manual evaluation. The presence of these examples artificially lowers the evaluation metrics for our model in certain datasets.

A.6.5 Single Statement vs. Multi Statement

Generated responses often consist of multiple statements making up a single message. While developing our model training and evaluation strategy for decoder-based models, we compared the results of models when they either classify a single statement at a time, or provide classifications for all of the statements comprising the message at once. We found that the single-sentence case outperformed the multi-sentence case and provide a few examples in Table 9 where classifying single statements at a time resulted in better predictions than classifying all statements at once.

A.7 Cost Analysis

We estimated the cost per inference of our RAGHalu model by determining the price of deploying one instance of a Google Kubernetes Engine (GKE) Node Pool with a NVIDIA L4 GPU using the Google Cloud Pricing Calculator[22] for a G2 accelerator-optimized machine. We approximate 8hr/day of consistent use which results in a cost of $172/month at 243.33 hours/month which equates to $0.707/hour.

[22] https://cloud.google.com/products/calculator

Error Type: Single Statement vs. Multi Statement Classification *(Internal Datasets)*
Statement List: ['For security reasons, we are unable to provide account numbers over the phone or online', 'To obtain your account number, please contact our call centre on <PHONE NUMBER> or visit your nearest branch'] **Gold Labels:** ['UNSUPPORTED', 'UNSUPPORTED'] **Single Sentence Predictions:** ['UNSUPPORTED', 'UNSUPPORTED'] **Multi Statement Predictions:** ['UNSUPPORTED', 'SUPPORTED']
Statement List: ['You can return online purchases at any of our *Sporting Goods* store locations or through the mail', 'We offer free returns on most items, but some exclusions do apply'] **Gold Labels:** ['SUPPORTED', 'UNSUPPORTED'] **Single Statement Prediction:** ['SUPPORTED', 'UNSUPPORTED'] **Multi Statement Prediction:** ['UNSUPPORTED', 'SUPPORTED']

Table 12: Examples of performance differences between single-sentence and multi-sentence models.

To estimate GPT-3.5-Turbo cost, we use the average input prompt length and output tokens across the four brands test sets. Each brand prompt input is approximately ∼360 tokens[23] with an average of 5 output tokens, resulting in $0.00037/inference.[24] Inference speeds and cost estimates are shown in Table 13.

We estimate each RAG event to have three factually verifiable claims, so, letting *S* denote savings per year, *C* denote cost, and *V* denote events/year we can estimate savings is as follows:

$$S = (C_{ChatGPT} - C_{RAGHalu})(3V)$$

Using inference cost estimates on a real brand with 2 million monthly conversations and roughly 5 LLM responses per conversation, relative to zero-shot GPT-3.5-Turbo hallucination detection, each year RAGHalu will save the brand:

$$S = (0.00037 - 0.000077) \times 3(2,000,000 \times 5 \times 12) = \$105,480$$

[23]The prompt template itself without KBs or LLM statements is 184 tokens

[24]We assume 1x concurrent requests evenly distributed across 8 hours/day. One NVIDIA L4 meets throughput demands. RAGHalu model throughput is ∼5.2 requests/second per tier

Model	Inference Speed (ms) $\pm$ std	Model Cost	Cost/Inference ($)
ChatGPT*	295 ± 131	0.0010$/1k tokens Input 0.0020$/1k tokens Out	0.00037
mistral-7b-ft*	1023 ± 83	$0.707/hr	0.0002
RAGHalu	391 ± 77	$0.707/hr	0.000077

Table 13: Inference speed in milliseconds/iteration - tests performed using either OpenAI API, Huggingface TGI or MLServer for Inference on 1xNVIDIA-L4 GPU on GCP. Both RAGHalu models can fit on the same GPU. *Both models are end-to-end and use the 3-label prompt in Appendix A.2