Trustful LLMs: Customizing and Grounding Text Generation with Knowledge Bases and Dual Decoders

Xiaofeng Zhu
Microsoft Corporation / WA, USA
Xiaofeng.Zhu@microsoft.com

Jaya Krishna Mandivarapu
Microsoft Corporation / GA, USA
jmandivarapu@microsoft.com

Abstract

Although people are impressed by the content generation skills of large language models, the use of LLMs, such as ChatGPT, is limited by the domain grounding of the content. The correctness and groundedness of the generated content need to be based on a verified context, such as results from Retrieval-Augmented Generation (RAG). One important issue when adapting LLMs to a customized domain is that the generated responses are often incomplete, or the additions are not verified and may even be hallucinated. Prior studies on hallucination detection have focused on evaluation metrics, which are not easily adaptable to dynamic domains and can be vulnerable to attacks like jailbreaking. In this work, we propose 1) a post-processing algorithm that leverages knowledge triplets in RAG context to correct hallucinations and 2) a dual-decoder model that fuses RAG context to guide the generation process.

1 Introduction

Adapting an LLM to a specific domain is challenging for several reasons: 1) Pre-trained LLMs cover general knowledge and cannot access private data (even during fine-tuning) due to privacy, copyright, and policy constraints. 2) The grounding of generated texts can change depending on specific contexts, such as domain or timestamp. Recent studies mostly focus on detecting hallucinations and using multiple LLMs when hallucinations occur. 3) Business logic and structured data, such as databases and private knowledge bases, are required when integrating customized LLMs into production systems and presenting them to customers or users.

We offer two methods for correcting hallucinations (beyond merely detecting them (Wan et al., 2024; Li et al., 2023a; Ji et al., 2023)): 1) Applying post-processing to generated texts using knowledge triplets, and 2) Proposing guided generation via Dual Decoders. Inspired by common practices like Retrieval-Augmented Generation (RAG) (Li et al., 2024), which retrieves relevant grounding context and feeds it to an LLM for text generation, we address hallucinations in generated texts from two aspects: 1) Post-editing based on knowledge graphs extracted from the context, and 2) Infusing guided context that contains important knowledge triplets into a generic LLM. Our proposed methods also provide reasoning and create consistent results from generative LLMs, benefiting from both the generation and extraction capabilities of decoder-only LLMs and the groundedness of RAG via the second decoder on the guidance (Le et al., 2020; Wang et al., 2022b).

In this work, we elaborate on our real-world commercial application scenario of using LLMs to support customers with Microsoft product inquiries in copilots, where groundedness is key to success. Pre-trained LLMs often lack the relevant knowledge or cannot adapt promptly to changes in the product database updates. Different variants of large language models (LLMs), such as Phi-3.5 (Abdin et al., 2024), ChatGPT (Mohamadi et al., 2023), LLama-3 (Dubey et al., 2024), and Gemma (Team, 2024), are proficient at producing fluent outputs for diverse user queries. Despite their human-like fluency in generating text across a wide range of prompts, large language models suffer from *hallucinations* (see examples in Figures 2, 3, 4), where parts or the entirety of the generated text lack faithfulness, factuality, or reasoning, yet are presented with a confident tone Ji et al., 2023.

To mitigate and correct hallucinations, we leverage guided text generation. Grounding guidance (Socher et al., 2013; Nickel et al., 2011; Lin et al., 2015; Wang et al., 2014; Bordes et al., 2013; Wang et al., 2022a; Grover and Leskovec, 2016), such as knowledge graphs (KGs), has been shown to significantly improve the reliability and factuality of LLMs in recent studies, e.g., KELM (Agar-

Proceedings of the 1st Workshop on Customizable NLP: Progress and Challenges in Customizing NLP for a Domain, Application, Group, or Individual (CustomNLP4U), pages 156–166
November 16, 2024 ©2024 Association for Computational Linguistics

wal et al., 2020; Lu et al., 2021), SKILL (Moiseev et al., 2022), K-DLM (Zou et al., 2023), KEPLET (Li et al., 2023b), and LUKE-Graph (Foolad and Kiani, 2023). Knowledge graphs typically consist of factual information represented explicitly in a semi-structured format, generally as [subject entity, relation, object entity] triples, e.g., (Bill Gates, was, the CEO of Microsoft) (Han et al., 2019; Gardner et al., 2017). We collect and maintain such knowledge triplets and grounded context offline for RAG.

Our contributions are as follows.
1) We correct hallucinations and out-of-domain outputs in generated texts from LLMs by leveraging a graph algorithm and provide reasoning using knowledge triplets extracted from both the guided context and the generated texts.
2) We propose a dual-decoder model that fuses guided context with natural language generation models, in which the decoders share the weights of a pre-trained LLM.
3) The proposed algorithm and model reduce the constraints on the maximum output length, in addition to correcting hallucinations, by returning or generating only outputs related to the prompt and the guided context.

2 Background and Related Work

Unlike document summarization, RAG, or traditional question answering, our approach benefits from both domain knowledge bases—particularly for groundedness—and the language understanding and generalization capabilities of various pre-trained or customized LLMs. By iterating over the knowledge triplets extracted from the generated text and comparing them to the knowledge triplets extracted from the given context (e.g., results from RAG), we can correct hallucinations (and generated phrases that lack references) using our proposed post-processing algorithm.

2.1 Guided Natural Language Generation

Prior studies have attempted multiple guidance frameworks, particularly with encoder-decoder models (See et al., 2017; Dou et al., 2020; Hokamp and Liu, 2017; Beurer-Kellner et al., 2024). Unlike GraphRAG (Edge et al., 2024), which utilizes multiple LLM calls to combine knowledge triplets from segments of RAG results, our proposed TrustfulLLM model reduces irrelevant entities and tokens in generated texts to demonstrate its efficiency.

2.2 Hallucination

Hallucination is considered one of the most prominent drawbacks of Large Language Models, as it leads models to generate inaccurate or false information (Ji et al., 2023; Wan et al., 2024). Model-generated texts may not match the true source content, and the facts presented by the model cannot always be verified from the source. These drawbacks remain significant hurdles in applying large language models (LLMs) to real-world, business-critical, and vitally important applications.

Algorithm 1 Hallucination Correction

```
Input: $\hat{Y}$, $G$
Output: $Y^*$
Construct knowledge graph $g = \{r_i\}$ from $\hat{Y}$
for knowledge triplet $t_i = (v_i^s, v_i^o, r_i)$ in $g$ do
    if $v_i^s$ not in $G$ then
        Eliminate $r_i$ from $g$ and the associated sentence in $\hat{Y}$
    else
        Replace $t_i$ and $\hat{Y}$ based on $g$
    end if
end for
Assume $\hat{G}$ is the subgraph of $G$, and $\hat{G}$ contains all the entities (nodes) in $\hat{Y}$
$Y^* = \hat{Y}$
while $Y^*$ contains cycles do
    Prune $\hat{Y}$ to $Y^*$ till $Y^*$ is a minimum spanning tree of $\hat{G}$.
end while
```

3 Methodology

Whether the generated text is factual is determined by the domain source and the given guided context. In our copilot scenario, we always retrieve related context for a user prompt/query and then utilize this context to generate the final response presented to users. The guided context can be a mix of offline or web articles and database records, from which we generate knowledge triplets (Gardner et al., 2017) for groundedness verification and hallucination correction. We propose a post-processing algorithm for correcting hallucinations that can be applied to any LLM outputs, as discussed in Section 3.1. Additionally, we propose a dual-decoder text gener-

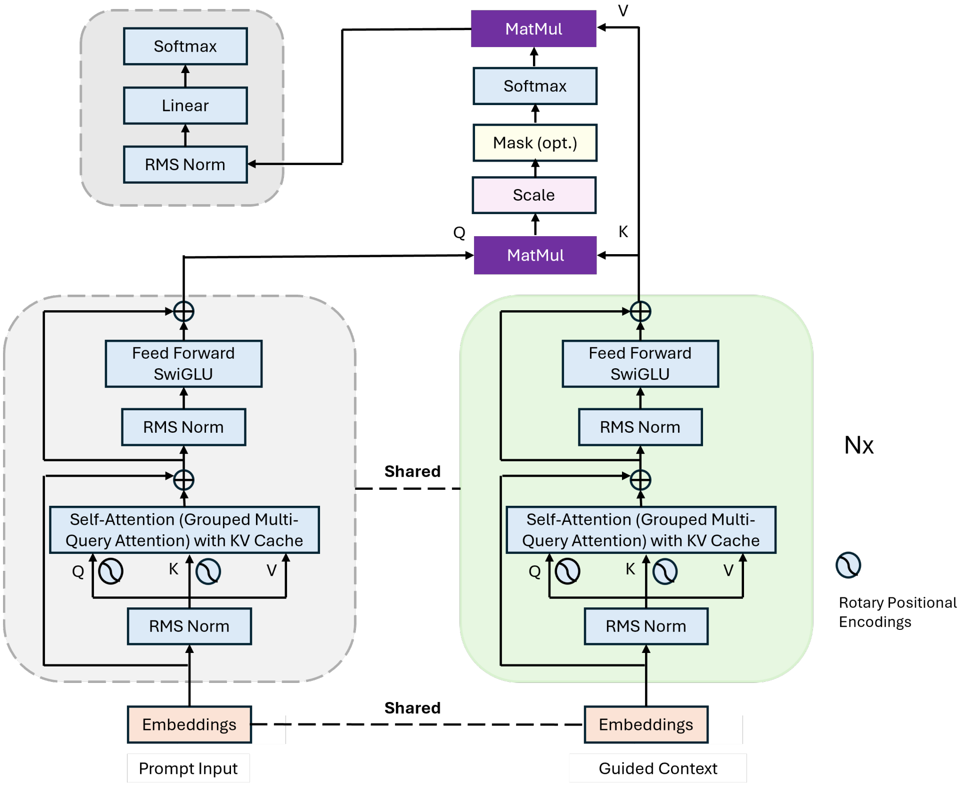


Figure 1: TrustfulLLM
The dual decoder module can be adapted to any generic LLM, and the weights are shared for the guided context and the prompt input.

ation model that takes both the prompt and guided context leveraging the RAG result content as inputs, described in Section 3.2.

3.1 Post-processing text generation by Correcting Knowledge Triplets

For generated texts from an LLM, we identify and correct potential hallucinations using knowledge triplets extracted from the RAG context and the generated text output. Specifically, we convert the extracted knowledge triplets from the guided context and the LLM output into graphs G and g, respectively, where each node v_i represents either a subject or an object, and the relations between the subject and object serve as bi-directional edges connecting the two nodes. Algorithm 1 explains the hallucination detection and correction process for a given generated text $\hat{Y}$ and the knowledge graph G extracted from the guided context. In the end, we obtain a corrected/verified output Y^*. A knowledge triplet t can be identified given a subject and a relation, or an object and a relation; i.e., we can easily locate and replace the third component when the entity or relation is incorrect in t_i, which is composed of subject v_i^s, object v_i^o, and the relation r_i. This algorithm can verify, replace, and prune triplets in $\hat{Y}$ but does not increase the number of nodes/entities. For instance, given a sentence in RAG result content: *"Microsoft 365 Business Basic is $7.2 dollars per user per month."*, we obtain knowledge triplet t_i: (v_i^s, v_i^o, r_i) is *(Microsoft 365 Business Basic, is, $7.2 dollars per user per month).* Since LLM outputs can omit or introduce additional entities, we propose a second method: guided generation via dual decoders.

3.2 TrustfulLLM and Guided Generation via Dual Decoders

In addition to the contextual embeddings used in Transformers, we embed the guidance text and apply a cross-attention calculation using the hidden states of the two decoders. In this way, we have the grounding/context source embeddings in one decoder and the user prompt in the other decoder, with both decoders sharing weights. We apply cross-attention CROSSATTN(H_p, H_g) by taking the hidden state H_p of the prompt module as the 'query' and the hidden state H_g of the guided context module as the 'key' and 'value.' The diagram of the TrustfulLLM is shown in Figure 1, and the pre-trained LLM component is generic. Only the prompt inputs are generated token by token, while the guided context contributes to the CROSSATTN(H_p, H_g) only. The fine-tuned transformer block components (the grey boxes in Figure 1) are derived from the Phi-3 and model architecture (Abdin et al., 2024; Dubey et al., 2024; Vaswani et al., 2023).

During the inference stage, the guided context is the same as the RAG context. We augment the RAG context by randomly adding additional content (shuffled from other RAG results from different prompts) as the guided context during fine-tuning, as shown in the Appendix A.2.

4 Experiments and Results

4.1 Tasks and Datasets

We elaborate the results from the public Microsoft learn.microsoft.com articles and product from www.microsoft.com [1]. The M365 dataset comprises approximately 10,000 question-and-answer pairs, including the context from which these question and answers were derived. We conducted our experiments based on that the RAG results (knowlege bases and/or domain articles) that are trustworthy. For fine-tuned LLMs, we leverage LoRA (Hu et al., 2021) and set the number of epochs to be over 400, which is relatively higher than in regular LoRA fine-tuning.

4.2 Metrics and Baseline Models

We use a combination of metrics including ROUGE-L, METEOR, GPT-Similarity, GPT-Groundedness (Appendix A.4), and BERTScore. ROUGE-L assesses the longest common subsequence between the generated and reference texts, capturing fluency and coherence. METEOR goes further by considering synonyms, stemming, and word order, providing a more nuanced evaluation. Groundedness rated 1-5 by GPT-4 ensures that the generated content is closely aligned with the source material. GPT-Similarity rated 1-5 by GPT-4 measures the semantic similarity between generated and reference texts, while BERT Score leverages pre-trained language models to evaluate the quality of the generated text on a deeper, contextual level. Together, these metrics provide a comprehensive assessment of our model performance.

We show the results of our methods, pre-trained LLMs, RAG, and Trustful LLMs on domain datasets M365 in Table 1, where boldface indicates the best scores, HC indicates applying the hallucination correction post-processsing algorithm, and TrustfulLLM indicates fine-tuning from the pre-trained model on the domain data. Leveraging the proposed HC can largely boost the groundedness score, and utilizing the TrustfulLLM dual-decoder framework and HC yield the best performance among all metrics. In particular, the percentage of eliminated entities when HC is applied to Phi-3.5 decreases from 18% to 6.9% when HC is applied to TrustfulLLM + Phi-3.5, further supporting the effectiveness of TrustfulLLM. We also explored the performance of the models on a general summarization task in Appendix A.3.

4.3 Effects of Applying HC and TrustfulLLM

We take a incorrect & incomplete statement from an LLM as a straightforward example: *"Domain registrar that support all DNS records required for Microsoft 365 are GoDaddy and Oray."* After we apply HC, HC corrects this output as follows: *"Domain registrars that support all DNS records required for Microsoft 365 are Oray , HiChina , east.net, and BIZCN."*

In our production systems, we convert the nodes at Line 4 of Algorithm 1 into embeddings using a pre-trained transformer model, allowing us to find semantically related subjects/objects using the cosine similarity and a heuristic similarity threshold. For example, *"M365 Business Basic"* can be mapped to *"Microsoft 365 Business Basic"*. When offline & pre-calibrated knowledge triplets are available, especially for user prompts related to Microsoft product information, we store the embeddings using the FAISS(Douze et al., 2024) [2] and combine them with the knowledge triplets extracted in the real-time RAG context.

LLMs can generate content that does not originate from the RAG context, which may not always be a hallucination. However, HC can make the outputs more consistent and better aligned with the RAG & guided context. For instance, given a user prompt:

What is the price of Microsoft 365 Business Basic?

[1] https://github.com/MicrosoftDocs/microsoft-365-docs

[2] https://github.com/facebookresearch/faiss

M365

Models	Rouge-L	METEOR	Groundedness	GPT-Similarity	BERTScore
TrustfulLLM + HC + Phi-3.5-mini-instruct	**0.55**	**0.51**	**5.00**	**4.68**	**0.93**
TrustfulLLM + Phi-3.5-mini-instruct	0.50	0.50	3.98	4.30	0.90
HC + Phi-3.5-mini-instruct	0.46	0.48	**5.00**	4.52	0.91
RAG + Phi-3.5-mini-instruct	0.41	0.45	3.72	3.49	0.89
RAG + Mistral-NeMo-Minitron-8B-Instruct	0.38	0.46	3.77	3.76	0.88
RAG + Llama-3.1-8B-Instruct	0.40	0.46	3.74	3.34	0.89
RAG + GPT-3.5 Turbo	0.45	0.48	3.81	3.58	0.90
RAG + GPT-4o	0.42	0.48	3.77	3.52	0.91
Phi-3.5-mini-instruct	0.17	0.26	3.33	3.60	0.84
Mistral-NeMo-Minitron-8B-Instruct	0.16	0.24	3.50	4.05	0.82
Llama-3.1-8B-Instruct	0.19	0.26	3.44	3.82	0.84
GPT-3.5 Turbo	0.23	0.31	3.70	4.10	0.85
GPT-4o	0.16	0.25	3.64	3.97	0.83

Table 1: Question Answering Benchmarking Results

and RAG context:

"Microsoft 365 Business Basic is $7.2 dollars per user per month. If you commit yearly the price is $6 dollars per user per month. Microsoft 365 Business Standard is $15 dollars per user per month ...",

a LLM can respond with:

Microsoft 365 Business Basic is priced at $7.2 dollars for each user on a monthly basis. However, if you choose to commit to a yearly plan, the price decreases to $6 dollars per user per month.

using RAG context:

Please answer using the exact following context "Microsoft 365 Business Basic is $7.2 dollars per user per month. If you commit yearly the price is $6 dollars per user per month. Microsoft 365 Business Standard is $15 dollars per user per month ...".

Utilizing TrustfulLLM and HC, we obtain the following outputs:

"Microsoft 365 Business Basic is $7.2 dollars per user per month. If you commit yearly the price is $6 dollars per user per month. Microsoft 365 Business Standard is $15 dollars per user per month."

Apparently, TrustfulLLM and HC tends to use the knowledge triplets and exact sub-sentences from the verified RAG context for grounded generation. This also makes (human) evaluations more effective in production systems.

4.4 Commercial Application and Constraints

In our commercial system, we first apply a intent detection to user prompts to filter out enquiries that are not related to our business then apply a retrieval model to obtain most relevant internal documents, records in product databases. We only reply on the groundedness and correctness of the retrieval results, i.e, phrases in AI generated texts that cannot be referenced from the RAG results are eliminated. For phrases that are semantically equivalent to the RAG results we still do a replacement using the knowledge triplet correction to keep consistent responses. We have also thoroughly conducted Red Teaming evaluations on various Responsible AI metrics such as harmful content, IP infringement, jailbreaking, groundedness, etc. Though we highligh our proposed halluciation correction algorithm and the dual decoder architecture, the upstream RAG and intent detection models can be combined in a multi-task modeling process.

5 Conclusion

We have addressed grounding issues in LLMs and proposed task-agnostic hallucination correction methods for real-world applications from two perspectives: post-processing to refine LLM outputs and trustful LLM fine-tuning via dual encoders. We have discussed hallucination correction and trustworthy text generation, demonstrating the robustness and resilience of our methods. In the future, we plan to explore heterogeneous modalities, such as structured and spatio-temporal data, knowledge-enriched representations of input tokens (Grover and Leskovec, 2016; Yu et al., 2022; Pan et al., 2023; GAO et al., 2021; Ye et al., 2021), hierarchical relation graphs, and accountability (Li et al., 2023a). We also plan to study model bias, aggregation for federated learning (Zheng et al., 2023; Hashemi et al., 2021), and privacy-preserving issues (Hashemi et al., 2021). Additionally, we aim to reduce the complexity of LLMs through parameter-efficient fine-tuning.

References

Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Qin Cai, Martin Cai, Caio César Teodoro Mendes, Weizhu Chen, Vishrav Chaudhary, Dong Chen, Dongdong Chen, Yen-Chun Chen, Yi-Ling Chen, Parul Chopra, Xiyang Dai, Allie Del Giorno, Gustavo de Rosa, Matthew Dixon, Ronen Eldan, Victor Fragoso, Dan Iter, Mei Gao, Min Gao, Jianfeng Gao, Amit Garg, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Jamie Huynh, Mojan Javaheripi, Xin Jin, Piero Kauffmann, Nikos Karampatziakis, Dongwoo Kim, Mahoud Khademi, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Yunsheng Li, Chen Liang, Lars Liden, Ce Liu, Mengchen Liu, Weishung Liu, Eric Lin, Zeqi Lin, Chong Luo, Piyush Madan, Matt Mazzola, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Swadheen Shukla, Xia Song, Masahiro Tanaka, Andrea Tupini, Xin Wang, Lijuan Wang, Chunyu Wang, Yu Wang, Rachel Ward, Guanhua Wang, Philipp Witte, Haiping Wu, Michael Wyatt, Bin Xiao, Can Xu, Jiahang Xu, Weijian Xu, Sonali Yadav, Fan Yang, Jianwei Yang, Ziyi Yang, Yifan Yang, Donghan Yu, Lu Yuan, Chengruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *Preprint*, arXiv:2404.14219.

Oshin Agarwal, Heming Ge, Siamak Shakeri, and Rami Al-Rfou. 2020. Knowledge graph based synthetic corpus generation for knowledge-enhanced language model pre-training. *arXiv preprint arXiv:2010.12688*.

Luca Beurer-Kellner, Marc Fischer, and Martin Vechev. 2024. Guiding llms the right way: Fast, non-invasive constrained generation. *Preprint*, arXiv:2403.06988.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26.

Zi-Yi Dou, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang, and Graham Neubig. 2020. Gsum: A general framework for guided neural abstractive summarization. *arXiv preprint arXiv:2010.08014*.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The faiss library.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vítor Albiero, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. 2024. From local to global: A graph rag approach to query-focused summarization. *Preprint*, arXiv:2404.16130.

Shima Foolad and Kourosh Kiani. 2023. Luke-graph: A transformer-based approach with gated relational graph attention for cloze-style reading comprehension. *arXiv preprint arXiv:2303.06675*.

HANNING GAO, LINGFEI WU, HONGYUN ZHANG, ZHIHUA WEI, PO HU, FANGLI XU, and BO LONG. 2021. Triples-to-text generation with reinforcement learning based graph-augmented structural neural networks. *arXiv preprint arXiv:2111.10545*.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke S. Zettlemoyer. 2017. Allennlp: A deep semantic natural language processing platform.

Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceed-*

ings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining, pages 855–864.

Xu Han, Tianyu Gao, Yuan Yao, Deming Ye, Zhiyuan Liu, and Maosong Sun. 2019. OpenNRE: An open and extensible toolkit for neural relation extraction. In *Proceedings of EMNLP-IJCNLP: System Demonstrations*, pages 169–174.

Hanieh Hashemi, Yongqin Wang, Chuan Guo, and Murali Annavaram. 2021. Byzantine-robust and privacy-preserving framework for fedml. *arXiv preprint arXiv:2105.02295*.

Chris Hokamp and Qun Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. *Preprint*, arXiv:1704.07138.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

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).

Hang Le, Juan Pino, Changhan Wang, Jiatao Gu, Didier Schwab, and Laurent Besacier. 2020. Dual-decoder transformer for joint automatic speech recognition and multilingual speech translation. *Preprint*, arXiv:2011.00747.

Bo Li, Peng Qi, Bo Liu, Shuai Di, Jingen Liu, Jiquan Pei, Jinfeng Yi, and Bowen Zhou. 2023a. Trustworthy ai: From principles to practices. *ACM Computing Surveys*, 55(9):1–46.

Yichuan Li, Jialong Han, Kyumin Lee, Chengyuan Ma, Benjamin Yao, and Derek Liu. 2023b. Keplet: Knowledge-enhanced pretrained language model with topic entity awareness. *arXiv preprint arXiv:2305.01810*.

Zhuowan Li, Cheng Li, Mingyang Zhang, Qiaozhu Mei, and Michael Bendersky. 2024. Retrieval augmented generation or long-context llms? a comprehensive study and hybrid approach. *arXiv preprint arXiv:2407.16833*.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the AAAI conference on artificial intelligence*, volume 29.

Yinquan Lu, Haonan Lu, Guirong Fu, and Qun Liu. 2021. Kelm: knowledge enhanced pre-trained language representations with message passing on hierarchical relational graphs. *arXiv preprint arXiv:2109.04223*.

Salman Mohamadi, Ghulam Mujtaba, Ngan Le, Gianfranco Doretto, and Donald A. Adjeroh. 2023. Chatgpt in the age of generative ai and large language models: A concise survey. *Preprint*, arXiv:2307.04251.

Fedor Moiseev, Zhe Dong, Enrique Alfonseca, and Martin Jaggi. 2022. Skill: structured knowledge infusion for large language models. *arXiv preprint arXiv:2205.08184*.

Maximilian Nickel, Volker Tresp, Hans-Peter Kriegel, et al. 2011. A three-way model for collective learning on multi-relational data. In *Icml*, volume 11, pages 3104482–3104584.

Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2023. Unifying large language models and knowledge graphs: A roadmap. *arXiv preprint arXiv:2306.08302*.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. *Advances in neural information processing systems*, 26.

Gemma Team. 2024. Gemma.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need. *Preprint*, arXiv:1706.03762.

David Wan, Koustuv Sinha, Srini Iyer, Asli Celikyilmaz, Mohit Bansal, and Ramakanth Pasunuru. 2024. Acueval: Fine-grained hallucination evaluation and correction for abstractive summarization. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 10036–10056.

Xintao Wang, Qianyu He, Jiaqing Liang, and Yanghua Xiao. 2022a. Language models as knowledge embeddings. *arXiv preprint arXiv:2206.12617*.

Ying Wang, Yuexing Peng, Wei Li, George C. Alexandropoulos, Junchuan Yu, Daqing Ge, and Wei Xiang. 2022b. Ddu-net: Dual-decoder-u-net for road extraction using high-resolution remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–12.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI conference on artificial intelligence*, volume 28.

Hongbin Ye, Ningyu Zhang, Shumin Deng, Mosha Chen, Chuanqi Tan, Fei Huang, and Huajun Chen. 2021. Contrastive triple extraction with generative transformer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 14257–14265.

Changlong Yu, Tianyi Xiao, Lingpeng Kong, Yangqiu Song, and Wilfred Ng. 2022. An empirical revisiting of linguistic knowledge fusion in language understanding tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10064–10070, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

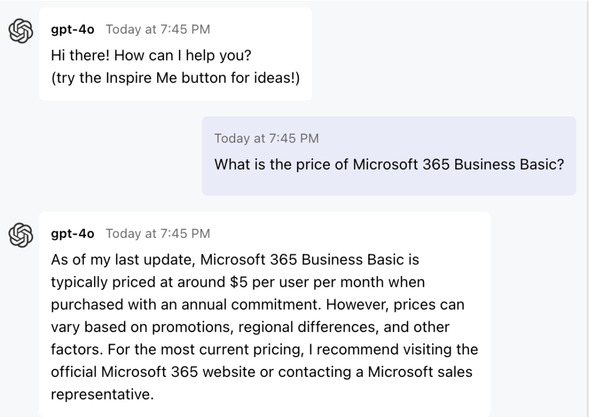
Longfei Zheng, Yingting Liu, Xiaolong Xu, Chaochao Chen, Yuzhou Tang, Lei Wang, and Xiaolong Hu. 2023. Fedpse: Personalized sparsification with element-wise aggregation for federated learning. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 3514–3523.

Jiaxin Zou, Zuotong Xie, Junhua Chen, Jiawei Hou, Qiang Yan, and Hai-Tao Zheng. 2023. K-dlm: A domain-adaptive language model pre-training framework with knowledge graph. In *International Conference on Artificial Neural Networks*, pages 447–459. Springer.

A Appendix

A.1 Hallucination Examples

We show examples where various LLMs generate hallucinations.



gpt-4o Today at 7:45 PM
Hi there! How can I help you?
(try the Inspire Me button for ideas!)

Today at 7:45 PM
What is the price of Microsoft 365 Business Basic?

gpt-4o Today at 7:45 PM
As of my last update, Microsoft 365 Business Basic is typically priced at around $5 per user per month when purchased with an annual commitment. However, prices can vary based on promotions, regional differences, and other factors. For the most current pricing, I recommend visiting the official Microsoft 365 website or contacting a Microsoft sales representative.

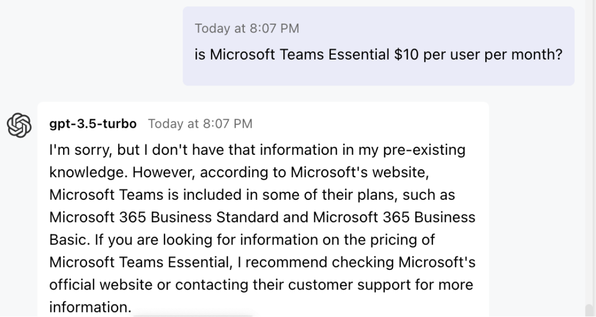
Figure 2: Hallucination Example 1
GPT-4o returns the wrong pricing information of Microsoft 365 Business Basic.

A.2 Examples of Prompt, RAG Context, and Guided Context

Prompt: *"... <|user|> How much is Microsoft 365 Business Basic? <|end|> <|assistant|> Microsoft 365 Business Basic is $7.2 dollars per user per month. <|end|>"*.

RAG context: *"Microsoft 365 Business Basic is $7.2 dollars per user per month. Microsoft 365 Business Basic ..."*.

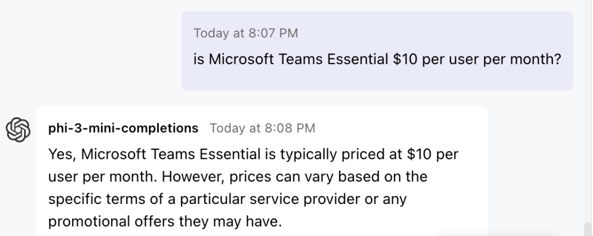
Guided context: *"Microsoft 365 Business Basic is $7.2 dollars per user per month. Microsoft 365 Business Basic ... Microsoft 365 Business Standard is ... <|end|>"*. We add additional content about, such as *"Microsoft 365 Business Standard"*, which is similar to the product *"Microsoft 365 Business Basic"* to the RAG context. This is for mimicking the potentially noisy RAG context in the retrieval stage.



Today at 8:07 PM
is Microsoft Teams Essential $10 per user per month?

gpt-3.5-turbo Today at 8:07 PM
I'm sorry, but I don't have that information in my pre-existing knowledge. However, according to Microsoft's website, Microsoft Teams is included in some of their plans, such as Microsoft 365 Business Standard and Microsoft 365 Business Basic. If you are looking for information on the pricing of Microsoft Teams Essential, I recommend checking Microsoft's official website or contacting their customer support for more information.

Figure 3: Hallucination Example 2
GPT-3.5 Turbo cannot answer questions related to Microsoft Teams Essential.



Today at 8:07 PM
is Microsoft Teams Essential $10 per user per month?

phi-3-mini-completions Today at 8:08 PM
Yes, Microsoft Teams Essential is typically priced at $10 per user per month. However, prices can vary based on the specific terms of a particular service provider or any promotional offers they may have.

Figure 4: Hallucination Example 3
Phi-3 answered incorrectly about the price of Microsoft Teams Essential.

A.3 Summarization Task

A summarization task does not have the retrieval component as in RAG. We utilize the graph building step of HC to select the salient sentences from the articles as the guided context. We first extract knowledge triplets from the articles then keep sentences where the most frequent subjects are associated with. We show the comparison of TrustfulLLM + HC + Phi-3.5-mini-instruct, where HC extract knowledge triplets from the articles and the generated texts in the inference stage, and LLM baselines in Table 2.

A.4 Prompt Template for GPT Metrics

We show the prompts of GPT Similarity and GPT Groundness addressed in Section 4.

Prompt for GPT Groundness
System:

CNN DailyMail

Models	Rouge-L	METEOR	Groundedness	GPT-Similarity	BERTScore
TrustfulLLM + HC + Phi-3.5-mini-instruct	**0.41**	**0.39**	**5.00**	**4.12**	**0.89**
TrustfulLLM + Phi-3.5-mini-instruct	0.40	**0.39**	4.68	**4.12**	0.88
HC + Phi-3.5-mini-instruct	0.35	0.36	**5.00**	3.82	0.88
Phi-3.5-mini-instruct	0.17	0.34	4.29	3.79	0.86
Mistral-NeMo-Minitron-8B-Instruct	0.20	0.35	3.32	3.87	0.86
Llama-3.1-8B-Instruct	0.32	0.37	4.61	4.10	0.87
GPT-3.5 Turbo	0.24	0.38	4.50	3.79	0.87
GPT-4o	0.18	0.36	4.42	4.10	0.87

Table 2: Summarization Benchmarking Results

You are an AI assistant. You will be given the definition of an evaluation metric for assessing the quality of an answer in a question-answering task. Your job is to compute an accurate evaluation score using the provided evaluation metric. You should return a single integer value between 1 to 5 representing the evaluation metric. You will include no other text or information.

User:

You will be presented with a CONTEXT and an ANSWER about that CONTEXT. You need to decide whether the ANSWER is entailed by the CONTEXT by choosing one of the following rating:

1. 5: The ANSWER follows logically from the information contained in the CONTEXT.
2. 1: The ANSWER is logically false from the information contained in the CONTEXT.
3. An integer score between 1 and 5, and if such an integer score does not exist, use 1: It is not possible to determine whether the ANSWER is true or false without further information.

Read the passage of information thoroughly and select the correct answer from the three answer labels. Read the CONTEXT thoroughly to ensure you know what the CONTEXT entails. Note that the ANSWER is generated by a computer system, so it can contain certain symbols, which should not be a negative factor in the evaluation.

Independent Examples:

Example Task #1 **Input:**

{"CONTEXT": "Some are reported as not having been wanted at all.", "QUESTION": "", "ANSWER": "All are reported as being completely and fully wanted."}

Example Task #1 Output:

1 **Example Task #2** **Input:**

{"CONTEXT": "Ten new television shows appeared during the month of September. Five of the shows were sitcoms, three were hourlong dramas, and two were news-magazine shows. By January, only seven of these new shows were still on the air. Five of the shows that remained were sitcoms.", "QUESTION": "", "ANSWER": "At least one of the shows that were cancelled was an hourlong drama."}

Example Task #2 Output:

5

Example Task #3 **Input:**

{"CONTEXT": "In Quebec, an allophone is a resident, usually an immigrant, whose mother tongue or home language is neither French nor English.", "QUESTION": "", "ANSWER": "In Quebec, an allophone is a resident, usually an immigrant, whose mother tongue or home language is not French."}

Example Task #3 Output:

5

Example Task #4 **Input:**

{"CONTEXT": "Some are reported as not having been wanted at all.", "QUESTION": "", "ANSWER": "All are reported as being completely and fully wanted."}

Example Task #4 Output:

1

Actual Task **Input:**

{"CONTEXT": {{context}}, "QUESTION": "", "ANSWER": {{response}}}

Reminder: The return values for each task should be correctly formatted as an integer between 1 and 5. Do not repeat the context and question.

Actual Task Output:

Prompt for GPT Similarity]

System:

You are an AI assistant. You will be given the definition of an evaluation metric for assessing the quality of an answer in a question-answering task. Your job is to compute an accurate evaluation score using the provided evaluation metric. You should return a single integer value between 1 to 5 representing the evaluation metric. You will include no other text or information.

User:

Equivalence, as a metric, measures the similarity between the predicted answer and the correct answer. If the information and content in the predicted answer is similar or equivalent to the correct answer, then the value of the Equivalence metric should be high, else it should be low. Given the question, correct answer, and predicted answer, determine the value of the Equivalence metric using the following rating scale:

- One star: the predicted answer is not at all similar to the correct answer
- Two stars: the predicted answer is mostly not similar to the correct answer
- Three stars: the predicted answer is somewhat similar to the correct answer
- Four stars: the predicted answer is mostly similar to the correct answer
- Five stars: the predicted answer is completely similar to the correct answer

This rating value should always be an integer between 1 and 5. So the rating produced should be 1, 2, 3, 4, or 5. The examples below show the Equivalence score for a question, a correct answer, and a predicted answer.

Question: What is the role of ribosomes?
Correct answer: Ribosomes are cellular structures responsible for protein synthesis. They interpret the genetic information carried by messenger RNA (mRNA) and use it to assemble amino acids into proteins.
Predicted answer: Ribosomes participate in carbohydrate breakdown by removing nutrients from complex sugar molecules.
Stars: 1

Question: Why did the Titanic sink?
Correct answer: The Titanic sank after it struck an iceberg during its maiden voyage in 1912. The impact caused the ship's hull to breach, allowing water to flood into the vessel. The ship's design, lifeboat shortage, and lack of timely rescue efforts contributed to the tragic loss of life.
Predicted answer: The sinking of the Titanic was a result of a large iceberg collision. This caused the ship to take on water and eventually sink, leading to the death of many passengers due to a shortage of lifeboats and insufficient rescue attempts.
Stars: 2

Question: What are the health benefits of regular exercise?
Correct answer: Regular exercise can help maintain a healthy weight, increase muscle and bone strength, and reduce the risk of chronic diseases. It also promotes mental well-being by reducing stress and improving overall mood.
Predicted answer: Routine physical activity can contribute to maintaining ideal body weight, enhancing muscle and bone strength, and preventing chronic illnesses. In addition, it supports mental health by alleviating stress and augmenting general mood.
Stars: 5

Question: {{query}}
Correct answer: {{ground_truth}}
Predicted answer: {{response}}
Stars: