

# HalluSafe at SemEval-2024 Task 6: An NLI-based Approach to Make LLMs Safer by Better Detecting Hallucinations and Overgeneration Mistakes

**Zahra Rahimi, Hamidreza Amirzadeh, Alireza Sohrabi,  
Zeinab Sadat Taghavi and Hossein Sameti**

Department of Computer Engineering, Sharif University of Technology, Tehran, Iran  
{zarahimi, hamid.amirzadeh78, sameti}@sharif.edu  
{lirezasohrabi, zeinabtaghavi1377}@gmail.com

## Abstract

The advancement of large language models (LLMs), their ability to produce eloquent and fluent content, and their vast knowledge have resulted in their usage in various tasks and applications. Despite generating fluent content, this content can contain fabricated or false information. This problem is known as hallucination and has reduced the confidence in the output of LLMs. In this work, we have used Natural Language Inference to train classifiers for hallucination detection to tackle SemEval-2024 Task 6-SHROOM (Mickus et al., 2024) which is defined in three sub-tasks: Paraphrase Generation, Machine Translation, and Definition Modeling. We have also conducted experiments on LLMs to evaluate their ability to detect hallucinated outputs. We have achieved 75.93% and 78.33% accuracy for the model-aware and model-agnostic tracks, respectively. The shared links of our models and the codes are available on GitHub<sup>1</sup>.

## 1 Introduction

Large language models are compelling in content generation. The ability of these models has led to their widespread use in various applications. Some of the use cases of these models are in sensitive fields, such as consulting in medicine and law. The eloquence of LLMs makes their content appear very acceptable, and these models respond with high confidence. An important shortcoming of these models is hallucination. Hallucination is the production of fabricated or false content (Gehman et al., 2020; Weidinger et al., 2021). Hallucination detection and mitigation are necessary to avoid the dangers of spreading false and harmful information. According to Zhang et al. (2023), hallucinations can be divided into input hallucinations, context hallucinations, and factual hallucinations.

In input hallucination, the output content of the model has data that contradicts the input content. In context hallucination, the model’s output content contradicts the content the model itself produced earlier. In the last case, factual hallucination, the output content of the model has information that contradicts the existing world knowledge. In the dataset provided for the Shroom task, each data sample has a reference to be checked with. Given that reference-based hallucination detection entails identifying contradictions between model output and the reference (either input or target), a natural language inference (NLI) approach presents an intuitive solution to detect such contradictions and consequently identify instances of hallucination, therefore we adopt an NLI approach as the foundation of our methodology.

Through this task, we have gained knowledge about hallucinations, their causes, and the various approaches to deal with them. Language model responses can be so fluent that it becomes difficult even for a human agent to detect hallucinations. Therefore, it is essential to train these models to recognize the limits of their knowledge. If they lack sufficient understanding of a subject, they should search for reliable sources and inform the human user if they are unsure of their answer. Our team ranked 19th and 30th in the model-aware and model-agnostic tracks, respectively, with a difference of 2.93% and 8.4% compared to the top-ranked team. We found that the decision boundary for detecting hallucinations can be very narrow in some cases. While our system has shown relatively good performance, there is still room for improvement.

## 2 Background

As mentioned earlier, there are three types of hallucinations. The types of hallucinations considered in this task are “factual” and “input”. The “factual”

<sup>1</sup><https://github.com/z-rahimi-r/HalluSafe-at-SemEval-Task-6-SHROOM>

type occurs in the definition modeling task, where the definition of a word or phrase must be provided, and the “input” type appears in the paraphrase generation and machine translation tasks. The hallucination detection track has two sub-tracks: model-aware and model-agnostic. In the model-aware sub-track, the model that generated the data is specified, and participants can use model parameters for diagnosis or analysis. However, our approach assumes the models are black-box and can be used for situations where we do not have access to the internal states and parameters of the model. It is important to note that overgeneration is another issue in LLM outputs. Samples with this issue should also be labeled as One, indicating the presence of hallucinations. Hallucination is not specific to LLMs, and before the emergence of these models, it has been investigated in NLP tasks such as summarization and machine translation (Azaria and Mitchell, 2023).

To deal with the hallucination problem in LLMs, it is essential to find the causes of the problem first. Two probable causes of hallucination, stated in Azaria and Mitchell (2023), are the model focusing on producing one token each time and random sampling to increase diversity in text production. Some believe overfitting to training data may lead to hallucination (McKenna et al., 2023). In contrast to this point of view, in Yao et al. (2023), they have shown that prompts consisting of only random meaningless tokens can also elicit hallucinations in LLMs. They believe that hallucinations are beyond training data and consider them as adversarial features. They have observed in their experiments that a slight change in the original prompt can produce a completely different claim by the LLM, which indicates that LLMs are very non-robust. In Rawte et al. (2023), they measure the relationship between linguistic factors such as readability, formality, and concreteness of prompts and hallucinations. Their results show that more concrete and formal prompts lead to fewer hallucinations, but no definite conclusion can be drawn regarding the effect of readability on hallucinations. According to this article, prompt engineering can be effective in reducing the problem of hallucinations. Lengthy prompts can hurt the understanding of the LLM. In some experiments, it has been observed that the LLM performs better when the critical information is placed at the beginning or end of the prompt. The performance quality decreases when the model needs to access the middle parts of the prompt for information.

Hallucination can be mitigated in different stages of an LLM’s life cycle. As we know, the life cycle of an LLM consists of Pre-training, SFT (Supervised Fine-Tuning), RLHF (Reinforcement Learning with Human Feedback), and Inference (Zhang et al., 2023). The datasets with which LLMs are pre-trained are collected without human supervision. These data can include false or outdated information, which may cause hallucinations. The training in the SFT phase should also consider the knowledge of the model, and the model should not be fine-tuned for an application that has not acquired sufficient knowledge during the pre-training. One way to reduce hallucinations in both the SFT and RLHF phases is to teach the model to be honest. The language model should be trained to avoid commenting on a subject if it does not have enough information (Zhang et al., 2023). The methods investigated in this work are related to detecting and mitigating hallucination in the inference phase. The related previous works can be categorized as white box, gray box, and black box depending on the level of access to internal parameters of the LLM. The methods that use the internal state of the language model for diagnosis are white-box approaches. Gray box approaches are methods that access the output distribution of the model, such as detecting hallucinations at the token level. Finally, Blackbox approaches only have access to the textual output of the model.

## 2.1 White-Box Approaches

In Azaria and Mitchell (2023), the SAPLMA approach (Statement Accuracy Prediction, based on Language Model Activations) has been introduced. Their approach uses the internal state of the LLM to measure the truthfulness of the statements. This applies to both the statements provided to the model and the statements produced by the model itself. They use a relatively shallow feedforward network as a classifier, which measures the truthfulness probability of a statement based on the values of the hidden layer activators.

## 2.2 Gray-Box Approaches

These approaches use the uncertainty of models to detect hallucinations. The idea of these approaches is that when the model is sure of the correctness of a sentence, the distribution probability of tokens of the sequence is sharp. Still, in uncertain conditions, this distribution will probably be flat. Kadavath et al. (2022) suggests that a model’s confidence in

answering a specific question correlates with the certainty of its response. They propose repeatedly sampling the answer at  $T = 1$ , yielding an answer distribution characterized by low entropy when the model is confident. Conversely, when the model is uncertain, it tends to produce "hallucinated" responses, resulting in an answer distribution with high entropy. Nevertheless, experimental results indicate that utilizing entropy as a metric for determining whether a model knows the answer to a question is not consistently reliable, particularly as models scale in size. Another work in this group of methods is [Yuan et al. \(2021\)](#), in which a score named BART-Score evaluates the text's quality generated by the model from different aspects such as informativeness, fluency, and factuality. Using token-level probabilities, BART-Score calculates the probability of an output sequence given a specific input sequence.

### 2.3 Black-Box Approaches

The methods presented in [Martino et al. \(2023\)](#) and [Manakul et al. \(2023\)](#) are black box methods. In [Martino et al. \(2023\)](#), where a large language model is used for the "Review Response" task, the knowledge injection method adds related information to the prompt. The relevant knowledge is extracted from a knowledge graph specific to that particular business. It includes information such as addresses, phone numbers, etc., which are naturally not available in the training data of an LLM. The target hallucination in this task is factual. Fact-based verification methods require an external database, and their inference is computationally expensive. The introduced method in [Manakul et al. \(2023\)](#) uses no external knowledge source. Their approach, self-checkGPT, is based on the idea that if an LLM knows a subject, sampled responses do not contradict each other. The proposed approach has five variants: BERTScore, question-answering, n-gram, NLI, and LLM prompting. The best-performing variant is LLM prompting, in which they ask an LLM if a sentence is supported by a context or not. This variant has a high computational cost. The second best is the NLI variant, which uses natural language inference to detect inconsistency between sampled responses.

In [Mündler et al. \(2023\)](#), a prompting-based framework is introduced to efficiently identify and address instances of self-contradiction, meaning context hallucinations. Their investigation delved into open-domain text generation utilizing a dual-

LM setup: one LM for text generation and another as an analyzer. For each sentence generated by the initial LM, a corresponding sentence is produced based on the associated context, and both are subsequently subjected to analysis by the second LM. In cases where the analyzer LM identifies a contradiction between the two sentences, it is prompted to revise the given sentences and remove the contradiction so that the output is informative and coherent with the corresponding context. ChainPoll ([Friel and Sanyal, 2023](#)) represents another recent advancement in addressing hallucinatory phenomena within LLMs. The approach adopted for hallucination detection is straightforward: employing a carefully crafted prompt, the authors prompt the GPT-3.5-turbo model to assess whether the completion contains hallucinations driven by a chain of thought (CoT) explanation. Iterating this process several times and aggregating the "yes" responses yields a probability score ranging from Zero to One, indicating the likelihood of hallucination.

In [Guerreiro et al. \(2023\)](#), hallucinations in translation models are studied concerning two different sources: perturbations and natural hallucinations. Hallucinations induced by perturbations occur when the model memorizes the training data and outputs a faulty translation triggered by a slight change in the input sequence. In contrast, natural hallucinations occur due to poor quality of training data. Natural hallucinations are divided into two categories ([Raunak et al., 2021](#)): detached and oscillatory. In the detached type, the output is fluent but inadequate. In the oscillatory type, the output has repeated n-grams. In this article, a black box method (Top N-Gram ([Raunak et al., 2021](#))) and a white box method (ALTI+ ([Ferrando et al., 2022](#))) have been used to detect natural hallucinations. It has been observed that hallucinations in translations occur more often for low-resource languages. Another work concerning detecting machine translation hallucinations is COMET ([Rei et al., 2020](#)), a reference-based neural framework with superior performance compared to conventional approaches ([Guerreiro et al., 2022](#)). It has two architectures, one of which is an estimator model, which tries to directly regress on human judgment scores for quality assessment. In contrast, the other one, a ranking model, minimizes the distance between a "better" hypothesis and its corresponding reference and original source translations.

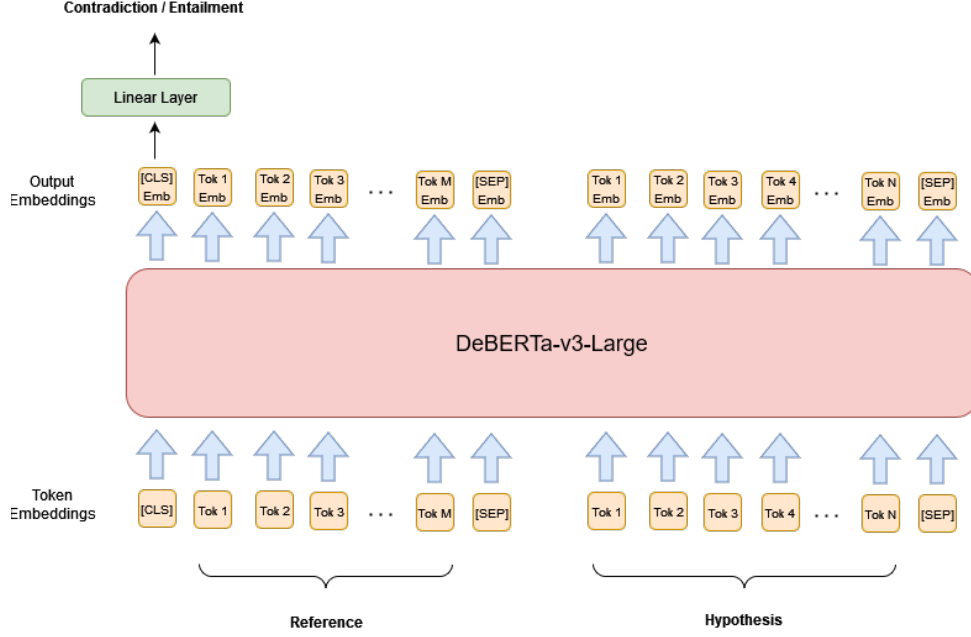


Figure 1: Formulating hallucination detection problem as an NLI task

	DM	PG	MT	Total
Train	20000	20000	20000	60000
Dev	375	250	375	1000
Trial	36	9	35	80
Test	1125	750	1125	3000

Table 1: Dataset Statistics

### 3 System Overview

In this section, we introduce our proposed system. The general system sketch is presented in Figure 1. Additionally, detailed statistics regarding the dataset are outlined in Table 1. Since the training data provided for this task was unlabeled, we labeled 3000 samples of the training data. Since LLMs have hallucination problems themselves, the labeling was done by a human agent. We have trained separate models for each task (MT, PG, and DM) to detect hallucinations. The model is DeBERTa-v3 large (He et al., 2023) and was first trained on the NLI task and then fine-tuned on the labeled data of each task. Finally, the model with the highest accuracy on validation data was saved. For training a binary classification model on the NLI task, only the data samples with labels of contradiction and entailment of the NLI dataset of Stanford University (Bowman et al., 2015) were used.

Examples of data samples for PG, MT, and DM tasks are presented in Table 2. Each sample has a source, target, and hypothesis in the MT task. The source sentence may be in languages other than English, but the target sentence is always in English. In the PG task, each sample has a source and hypothesis. We can detect hallucinations in these two tasks using the target sentence as the reference for the MT task and the source sentence as the reference for the PG task. Since the nature of hallucination in the PG and MT tasks is almost the same, the training data of both tasks were used to train the model for these two tasks. For Each task, the model with the highest accuracy on validation data was saved. The sequence classification method is utilized to detect hallucinations. The reference sentence is placed at the beginning, followed by the hypothesis sentence, separated with a "[SEP]" token. The hypothesis is the output of the LLM that may contain hallucinations. Finally, the entire sequence is fed into the NLI model, which outputs probabilities for each class, contradiction, and entailment. If the hypothesis contains information that contradicts the reference, the output label of our NLI model should be equal to 1, indicating contradiction. The probability of contradiction is considered equivalent to the probability of hallucination.

In addition to training classifier models, we have conducted tests to evaluate the performance of



PG	src	The budget cannot be adopted against the will of the European Parliament.
	hyp	The European Parliament does not approve the budget.
	label	Not Hallucination
MT	src	Doonii fayyadamuun meeshaa geejibuun namootabaay’ee fi meeshaalee galaanarra cesisuuf karaa baayee si’aataa dha.
	tgt	Using ships to transport goods is by far the most efficient way to move large amounts of people and goods across oceans.
	hyp	Using a gas-fired device is a way to stop people from using natural gas and other equipment.
	label	Hallucination
DM	src	Communistic birds. What is the meaning of communistic?
	tgt	Living or having their nests in common.
	hyp	Of or pertaining to communism.
	label	Hallucination

Table 2: Data samples of PG, MT, and DM tasks

two large language models, Falcon-7B and chat-GPT3.5, on the hallucination detection task. For this purpose, we have instruction-fine-tuned the falcon-7B model on the labeled training and validation data. For chat-GPT3.5, the accuracy was calculated on the trial set using zero and two-shot inference. For these two models, only the results on the trial set were presented.

We also thought we might find a meaningful connection between token probabilities in the output sequence and hallucination. For this, we took the top token probabilities of the output sequence of the LLM (PG, MT, and DM LLMs) with their labels. We fed them as input to an RNN model, such as LSTM, to predict hallucination based on model uncertainty of token probabilities. Unfortunately, we found out that when outputting hallucinated output, the model is as confident as non-hallucinated ones, and the classifier model could learn absolutely nothing from the token probabilities, no matter how we change the model complexity or hyperparameters.

#### 4 Experimental Setup

To provide enough labeled data to train our models, a total of 3000 of the model-aware and model-agnostic training samples were labeled. Different data splits were tested to get the best accuracy on each task. The details of the split used to train the model with the best accuracy for each task are specified in colab notebooks on GitHub<sup>1</sup>. The results

<sup>1</sup><https://github.com/z-rahimi-r/HalluSafe-at-SemEval-Task-6-SHROOM>

	DM	PG	MT
#Samples	36	9	35
Falcon7B 2-shot	47.22	44.44	45.71
Falcon7B 4-shot	33.33	55.55	48.57
Falcon7B finetuned	41.66	66.66	0.4
ChatGPT3.5 zero-shot	86.11	65.71	44.44
ChatGPT3.5 2-shot	86.11	74.28	88.88
ChatGPT3.5 4-shot	83.33	<b>82.85</b>	66.66
Best-DM-DeBERTa	<b>94.44</b>	55.55	85.71
Best-PG-DeBERTa	86.11	77.77	77.14
Best-MT-DeBERTa	91.66	55.55	<b>94.28</b>

Table 3: Results on Trial set

of the trial set are presented in Table 3.

All three models are trained for ten epochs with a learning rate equal to  $2e-5$  and batch size equal to eight samples. The base model for all three tasks is DeBERTa-v3-large (He et al., 2023), trained on the NLI task with two classes of contradiction and entailment. We have used the Hugging-Face transformers library (Wolf et al., 2020) to train DeBERTa models implemented with PyTorch. For instruction fine-tuning the Falcon-7B model, we also used the Hugging-Face library and LoRA method (Hu et al., 2022). The prompt used for fine-tuning Falcon and inference from chatGPT is similar to that used in the selfCheckGPT (Manakul et al.,

PG	src	A five, six, seven, eight.
	tgt	And 5, 6, 7, 8.
	hyp	A number between five and eight.
	gold label	Hallucination
MT	src	Est-ce que tu l’aimes?
	tgt	Do you love him?
	hyp	Do you love her?
	gold label	Not Hallucination
DM	src	Haul away, keeping strain on both parts of the halyard so that the <define> pigstick </define> remains vertical as it goes up and doesn’t foul the spreaders.
	tgt	(nautical) A staff that carries a flag or pennant above the mast of a sailboat.
	hyp	(nautical) A halyard.
	gold label	Not Hallucination

Table 4: Examples of wrongly classified samples

	acc model-agnostic	rho model-agnostic	acc model-aware	rho model-aware
Baseline	69.66	40.29	74.53	48.78
Nli-only	72.4	59.77	73.93	<b>56.33</b>
Best-models	<b>75.93</b>	<b>61.53</b>	<b>78.33</b>	53.74

Table 5: Results on Final Test set

2023). The examples can be found in the Appendix. All notebooks, labeled data, and links to saved models are present on our GitHub.

## 5 Results

We have achieved 75.93% and 78.33% accuracy for the model-aware and model-agnostic tracks of hallucination detection on final test data. We have ranked 19th and 30th in model-aware and model-agnostic tracks with a 2.93% and 8.4% difference with respect to the first-ranked team in the competition. The accuracies of the best model for each task, along with the accuracy of the base NLI model, are provided in Table 5. Also, examples of wrongly classified samples are provided in Table 4. As you can see the wrongly classified samples are challenging. The problem that exists with some samples of the MT task is that in some cases, relying only on the tgt field may result in a wrong label, and it is necessary also to consider the content of the src field as well. This is true about the MT example presented in the table. In this example, hyp and tgt are both correct translations of the source sentence, but when the content of hyp is evaluated against the tgt, it is wrongly labeled as hallucination.

## 6 Conclusion

In this work, we have trained classifiers based on Natural Language Inference to detect hallucinated outputs for the two model-aware and model-agnostic subtasks of the SemEval-2024 Task-6-SHROOM (Mickus et al., 2024). We have also conducted experiments to evaluate LLMs’ ability to perform this task. The fluency of the output of LLMs makes it difficult even for a human evaluator to recognize the hallucinated output. To train the classifiers, we labeled 3000 training data. Labels may be a little affected by the subjectivity of the annotator, and for future work, it is better to have more than one person label each data sample. Our HalluSafe classifiers have achieved 75.93% and 78.33% accuracy for the model-aware and model-agnostic tracks of hallucination detection on final test data and have outperformed official baselines. Regarding future work, enhancing the quality of training data in the pre-training and fine-tuning stages can effectively reduce hallucinations. Given the potential limitations of storing all necessary information within the memory of models, coupled with the need for regular updates to certain information, it may be beneficial to equip models with

search tools rather than relying solely on memory. It is important to train LLMs during the fine-tuning and instruction-tuning stages to refrain from answering questions if they lack sufficient knowledge on a particular subject, which needs a mechanism to be incorporated into these models to enable them to identify the boundaries of their knowledge.

## References

- Amos Azaria and Tom Mitchell. 2023. [The internal state of an LLM knows when it’s lying](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 967–976, Singapore. Association for Computational Linguistics.
- Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. [A large annotated corpus for learning natural language inference](#). In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.
- Javier Ferrando, Gerard I. Gállego, Belen Alastruey, Carlos Escolano, and Marta R. Costa-jussà. 2022. [Towards opening the black box of neural machine translation: Source and target interpretations of the transformer](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8756–8769, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Robert Friel and Atindriyo Sanyal. 2023. [Chainpoll: A high efficacy method for llm hallucination detection](#). *ArXiv*, abs/2310.18344.
- Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. 2020. [Realtoxicityprompts: Evaluating neural toxic degeneration in language models](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics.
- Nuno M. Guerreiro, Duarte M. Alves, Jonas Waldendorf, Barry Haddow, Alexandra Birch, Pierre Colombo, and André F. T. Martins. 2023. [Hallucinations in Large Multilingual Translation Models](#). *Transactions of the Association for Computational Linguistics*, 11:1500–1517.
- Nuno M. Guerreiro, Elena Voita, and André F. T. Martins. 2022. [Looking for a needle in a haystack: A comprehensive study of hallucinations in neural machine translation](#). *ArXiv*, abs/2208.05309.
- Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. [Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing](#).
- Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. [LoRA: Low-rank adaptation of large language models](#). In *International Conference on Learning Representations*.
- Saurav Kadavath, Tom Conerly, Amanda Askell, T. J. Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zachary Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislaw Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, John Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom B. Brown, Jack Clark, Nicholas Joseph, Benjamin Mann, Sam McCandlish, Christopher Olah, and Jared Kaplan. 2022. [Language models \(mostly\) know what they know](#). *ArXiv*, abs/2207.05221.
- 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.
- Ariana Martino, Michael Iannelli, and Coleen Truong. 2023. Knowledge injection to counter large language model (llm) hallucination. In *The Semantic Web: ESWC 2023 Satellite Events*, pages 182–185, Cham. Springer Nature Switzerland.
- Nick McKenna, Tianyi Li, Liang Cheng, Mohammad Hosseini, Mark Johnson, and Mark Steedman. 2023. [Sources of hallucination by large language models on inference tasks](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2758–2774, Singapore. Association for Computational Linguistics.
- Timothee Mickus, Elaine Zosa, Raúl Vázquez, Teemu Vahtola, Jörg Tiedemann, Vincent Segonne, Alessandro Raganato, and Marianna Apidianaki. 2024. [Semeval-2024 shared task 6: Shroom, a shared-task on hallucinations and related observable overgeneration mistakes](#). In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1980–1994, Mexico City, Mexico. Association for Computational Linguistics.
- Niels Mündler, Jingxuan He, Slobodan Jenko, and Martin T. Vechev. 2023. [Self-contradictory hallucinations of large language models: Evaluation, detection and mitigation](#). *ArXiv*, abs/2305.15852.
- Vikas Raunak, Arul Menezes, and Marcin Junczys-Dowmunt. 2021. [The curious case of hallucinations in neural machine translation](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1172–1183, Online. Association for Computational Linguistics.
- Vipula Rawte, Prachi Priya, S.M. Towhidul Islam Tonmoy, Islam Tonmoy, M Mehedi Zaman, A. Sheth,

- and Amitava Das. 2023. [Exploring the relationship between llm hallucinations and prompt linguistic nuances: Readability, formality, and concreteness](#). *ArXiv*, abs/2309.11064.
- Ricardo Rei, Craig Alan Stewart, Ana C. Farinha, and Alon Lavie. 2020. [Comet: A neural framework for mt evaluation](#). *ArXiv*, abs/2009.09025.
- Laura Weidinger, John Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, Zac Kenton, Sasha Brown, Will Hawkins, Tom Stepleton, Courtney Biles, Abeba Birhane, Julia Haas, Laura Rimell, Lisa Anne Hendricks, William Isaac, Sean Legassick, Geoffrey Irving, and Iason Gabriel. 2021. [Ethical and social risks of harm from language models](#).
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Jia-Yu Yao, Kun-Peng Ning, Zhen-Hui Liu, Mu-Nan Ning, and Li Yuan. 2023. Llm lies: Hallucinations are not bugs, but features as adversarial examples. *arXiv preprint arXiv:2310.01469*.
- Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. [Bartscore: Evaluating generated text as text generation](#). In *Advances in Neural Information Processing Systems*, volume 34, pages 27263–27277. Curran Associates, Inc.
- Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2023. [Siren’s song in the ai ocean: A survey on hallucination in large language models](#). *ArXiv*, abs/2309.01219.

## A Appendix

An example of instruction used for fine-tuning Falcon-7B is presented in Table 6. Also, a few-shot example for the PG task for inference from Chat-GPT and Falcon-7B is provided in table 7. Few-shot examples are selected from the development set for each task.



<human>:  
 [Context]: Being familiar with the working environment and able to intervene early is important for health care.  
 [Sentence]: Health care can be improved by being familiar with the working environment.  
 Is the Sentence supported by the Context above? Answer using ONLY yes or no:  
 <assistant>: [\[label\]: yes](#)

Table 6: Falcon-7B Fine-tuning Instruction Example

[Example 1]:  
 Context: I thought so, too.  
 Sentence: I thought you'd be surprised at me too.  
 Is the Sentence supported by the Context above? Answer using ONLY yes or no:  
 [label]: no

[Example 2]:  
 Context: I haven't been contacted by anybody.  
 Sentence: I have not been contacted.  
 Is the Sentence supported by the Context above? Answer using ONLY yes or no:  
 [label]: yes

[Example 3]:  
 Context: That was my general impression as well.  
 Sentence: I thought you'd be surprised at me too.  
 Is the Sentence supported by the Context above? Answer using ONLY yes or no:  
 [label]: no

[Example 4]:  
 Context: I said nothing of the kind.  
 Sentence: I never told you that before.  
 Is the Sentence supported by the Context above? Answer using ONLY yes or no:  
 [label]: yes

[Example 5]: [the sample to be labeled...](#)

Table 7: 4-Shot Chat-GPT Prompt Example