HIT-MI&T Lab at SemEval-2024 Task 6: DeBERTa-based Entailment Model is a Reliable Hallucination Detector

Wei Liu[1], **Wanyao Shi**[2], **Zijian Zhang**[1], **Hui Huang**[1]*
[1]Harbin Institute of Technology, Harbin, China
[2]Northwest Normal University, Lanzhou, China
liuweihit2023@163.com, shiwanyao@qq.com,
zhangzj0318@qq.com, huanghui@stu.hit.edu.cn;

Abstract

This paper describes our submission for SemEval-2024 Task 6: SHROOM, a Shared-task on Hallucinations and Related Observable Overgeneration Mistakes. We propose four groups of methods for hallucination detection: 1) Entailment Recognition; 2) Similarity Search; 3) Factuality Verification; 4) Confidence Estimation. The four methods rely on either the semantic relationship between the hypothesis and its source (target) or on the model-aware features during decoding. We participated in both the model-agnostic and model-aware tracks. Our method's effectiveness is validated by our high rankings 3rd in the model-agnostic track and 5th in the model-aware track. We have released our code on GitHub.[1]

1 Introduction

In tasks related to natural language generation, the output of a model may be fluent but may suffer from inaccuracies or inconsistencies with the input, a phenomenon referred to as "hallucination." For instance, Lee et al. (2021) and Müller et al. (2020) noted that in machine translation tasks, translated text is regarded as a "hallucination" when it exhibits a complete disconnect from the source text. Such discrepancies can mislead users and potentially lead to severe consequences. However, current evaluation metrics such as perplexity and BLEU (Papineni et al., 2002a) concentrate more on fluency rather than the accuracy or fidelity to the original input. Therefore, hallucination detection poses a big challenge and has gathered attention from research community.

SemEval-2024 Task 6 (Mickus et al., 2024) presents a testbed to evaluate whether the model outputs are hallucinating or not. The task comprises a total of three kinds of subtasks, which are definition modeling (DM) (Noraset et al., 2017), machine translation (MT) and paraphrase generation (PG). Each subtask involves triplet data with a source, which is the input to the model; a target, which represents the "gold" text that the model is expected to produce; a hypothesis, which is the actual output of the model. For all subtasks, the objective is to evaluate whether the hypothesis exhibits hallucinations according to the source or the target. More specifically, the hallucination of the hypothesis is verified based on target for DM and MT tasks, and source for PG task.

This paper presents the participation of HIT-MI&T Lab in the shared task in detail. We introduce four distinct hallucination detection methods, which transform the problem into different tasks:

1) Entailment Recognition: Hallucination is determined by analyzing the entailment relationship between the hypothesis and its source (target). Our approach mainly involves fine-tuning large language models (LLMs) and DeBERTa (He et al., 2020). An annotation dataset is constructed automatically to address data scarcity. We also devise an optimized loss function to handle noisy annotations during the fine-tuning of DeBERTa.

2) Similarity Search: Hallucination is gauged based on the semantic similarity between the hypothesis and its source (target). We mainly leverage SBERT (Reimers and Gurevych, 2019) to derive sentence representations for similarity search.

3) Factual Verification: Hallucination is detected by identifying factual inconsistencies between the hypothesis and its source (target). We mainly employ UniEval (Zhong et al., 2022) to assess the factual consistency.

4) Confidence estimation: Hallucination is evaluated based on the model's confidence in its answer. We mainly rely on two methods to estimate the model's confidence: a) analyzing the softmax distribution during decoding; b) assessing prediction consistency among multiple samplings.

*Corresponding author.

[1] https://github.com/LiuWeiHITees/semeval2024-task6-hallucination-detection

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1788–1797
June 20-21, 2024 ©2024 Association for Computational Linguistics

Finally, different groups of methods are ensembled for further enhancement, based on the accessibility of model-aware features. With our proposed framework, we achieved the third position in the model-agnostic track and the fifth position in the model-aware track, validating its effectiveness.

2 Related Work

With the success of ChatGPT (OpenAI, 2022) and GPT-4 (OpenAI, 2023), natural language generation has gained significant prominence within the broader domain of artificial intelligence. Its applicability spans a diverse array of tasks, including machine translation, summarization, and story continuation, etc. However, these models are sometimes prone to generating outputs that are fluent yet factually inaccurate, a phenomenon referred to as "hallucination". This phenomenon poses a substantial challenge to the reliability of language generation in real-world scenarios.

In the domain of hallucination detection methods, there has been considerable work by predecessors. Some people rely on semantic similarity measures for detection, such as N-gram-based Metrics (ROUGE (Lin, 2004) and BLEU (Papineni et al., 2002b)). However, these metrics only evaluate the lexical overlap between generated texts and reference texts by measuring the n-gram co-occurrence, and cannot discern fine-grained contextual semantic mismatch. Other studies(Laurer et al., 2023; Zha et al., 2023; Vectara, 2023) have fine-tuned BERT models using entailment datasets. These fine-tuned models are then utilized to detect hallucinations in specific scenarios. However, the fine-tuning process requires annotation data and can not generalize well among different scenarios.

As the hallucination mainly comes from the decoding procedure, some people propose to rely on uncertainty measures to detect hallucination. Some research (Guerreiro et al., 2023; Fu et al., 2023) proposed to calculate the log-probability or its entropy of translations for language generation tasks, and a lower probability indicates a lack of confidence, suggesting a potential hallucination. However, access to token-level probability distributions, essential for these approaches, is limited to open-source models and unavailable for models accessed solely through APIs, such as GPT-4.

Recently, with the popularization of LLMs, several LLM-based methods have been proposed. SelfCheckGPT (Manakul et al., 2023), employs a sampling-based strategy, which involves the generation of multiple stochastic samples. This approach hypothesizes that a model with a good understanding of the concept is less likely to generate significant hallucinations. Mündler et al. (2023) has explored the examination of self-contradiction within the context generated by an LLM as another aspect of hallucination detection. Their experiments, which involved prompting the LLM to perform a detection task, have demonstrated successful detection across various LLMs.

3 Methods

In this section, we will introduce our proposed four groups of methods for hallucination detection. The overall framework is shown in Figure 1.

3.1 Entailment recognition

While the objective of hallucination detection is to discern whether there is semantic mismatch between the hypothesis and the source (target), it resembles the objective of entailment recognition. Therefore, we decide to leverage entailment recognition models for detection.

3.1.1 LLM-based Data Construction

When employing entailment recognition model for hallucination detection, task specific fine-tuning is necessary to cope with the domain difference. However, organizers only provide unannotated training data in the form of [source, target, hypothesis], which cannot be directly leveraged for fine-tuning. Therefore, we propose deriving entailment annotations ourselves, leveraging the intelligence of proprietary LLMs like GPT-4. Specifically, we provide the paired text to the LLM, and design the prompt template to utilize GPT-4 to detect hallucinations in the hypothesis[2].

3.1.2 Fine-Tuning DeBERTa

As entailment recognition is inherently a text classification problem, we believe encoder-only understanding models may be more suitable. Therefore, we propose to apply fine-tuning on DeBERTaV3 (He et al., 2021), which has achieved good results especially in the text entailment task. The hypothesis, combined with the source (target) is fed to the entailment model, and a binary label is derived as the detection result.

[2] The detailed prompt is shown in Appendix A due to space limitations.

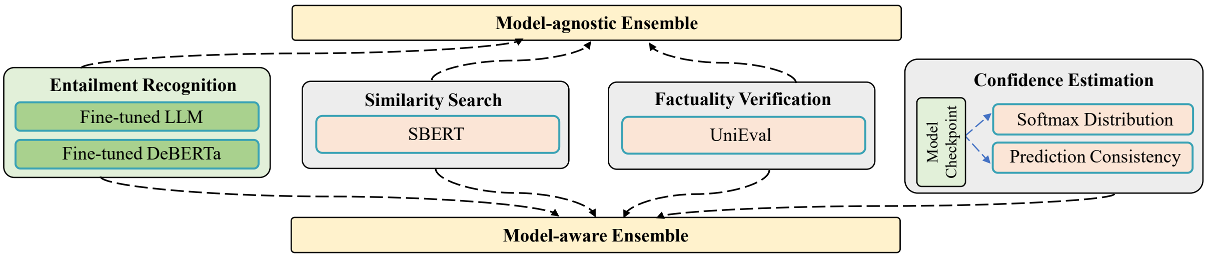


Figure 1: Overall framework of our proposed hallucination detection methods. We ensemble different groups of methods in different tracks, depending on the accessibility of model checkpoints.

As we mainly rely on automatically annotated data for fine-tuning, which the labels are generated by GPT-4 and may contain noise. Therefore, we introduced an auxiliary confidence loss that considers both the annotated labels and the difference between the model's prediction and its own confidence, following the work on weak-to-strong supervision by Burns et al. (2023). The optimized loss is formulated as follows:

$$\begin{aligned} L_{\mathbf{conf}}(f) = (1-\alpha)\cdot \mathrm{CE}(f(x), f_d(x)) \\ +\alpha \cdot \mathrm{CE}(f(x), \widehat{f}_t(x)) \end{aligned} \quad (1)$$

with symbols denoted as follows:

- $\mathrm{CE}(\cdot,\cdot)$ is the cross-entropy loss between the ground truth labels and the predicted probabilities.

- $f(x)$ belongs to [0,1] and represents the model's prediction distribution for input x.

- $f_d(x)$ represents the label for the input x.

- α is a weight used to balance the two losses.

- $\hat{f}_t(x)$ is a special version of $f(x)$, defined as follows:

$$\hat{f}_t(x) = \begin{cases} 1 & \text{if } f(x) > t \\ 0 & \text{if } f(x) \leq t \end{cases} \quad (2)$$

3.1.3 Fine-Tuning LLM

Given the superior performance of open-source LLMs across a diverse array of tasks, we also employ LLM for hallucination detection, which is fine-tuned on the annotated data.

We employ the recently released InternLM-20B (Team, 2023), due to its superior performance across various benchmarks and relatively modest parameter count. Our fine-tuning follows the instruction fine-tuning process, where the hypothesis combined with the source (target) is fed to the model to yield predictions indicative of entailment. Notice as the InternLM has gained massive linguistic knowledge, we only perform fine-tuning on the human annotated validation set. Moreover, we employ Q-LoRA (Dettmers et al., 2023), a parameter-efficient fine-tuning method to reduce the demand for training resources and time.

3.2 Similarity search

Since the hallucination mainly signifies the semantic mismatch between the hypothesis and the source (target), we believe that the mismatch can also be measured by sentence similarity. With contextual sentence embedding models, the hallucination can be discerned by a delicately designed threshold.

Specifically, we use SBERT to derive the semantic representations. SBERT model is an adapted version of BERT (Devlin et al., 2018) which is specifically designed to extract contextual text embeddings. In this work, we construct the sentence representations using the SBERT models for both hypothesis and source (target). After that, the cosine similarity scores are then calculated between the representations, to measure their semantic similarity.

3.3 Factual verification

As hallucinations often relate to factuality contradiction, we think the hallucination can be determined by evaluating the factual consistency between the hypothesis and the source (target).

Specifically, we use the UniEval framework to calculate the factual consistency score. UniEval is a comprehensive framework designed to evaluate generated text across multiple explainable dimensions, including factual consistency assessment. We feed the combined hypothesis and source (target) to the UniEval framework, and a continuous score is derived indicating the factual consistency.

3.4 Confidence Estimation

Hallucination in model output often signifies a lack of confidence. Therefore, we propose to apply confidence estimation techniques to detect hallucinations. By quantifying the model's confidence in its predictions, we can discern whether the output contains hallucination or not.

Specifically, we employ two confidence estimation techniques: analyzing softmax distribution of output tokens and assessing prediction consistency among multiple samplings. It is important to note that these methods are used only in the model-aware track due to the inherent requirement for checkpoints of the model that generates the output.

3.4.1 Softmax Distribution

One way to estimate model confidence is to analyze the softmax distribution over the vocabulary during the generation process. If the probability mass is highly concentrated on a few words, this suggests the model is confident in its predictions. Conversely, if the softmax probabilities approach a uniform distribution, where picking any word from the vocabulary is equally likely, then the quality of the hypothesis is expected to be low with hallucinations included. Therefore, we propose to incorporate the softmax distribution for hallucination detection.

In particular, we use two groups of features: token-level probability and entropy. For token-level probability, we calculate the average probability and minimum probability of each token. For entropy, we calculate the average entropy and maximum entropy at each position.

This method is primarily applied to DM and MT tasks, as these two subtasks tend to produce fixed outputs for fixed inputs.

3.4.2 Prediction Consistency

When the model lacks confidence with its own prediction, different predictions among different samplings might differ a lot. Based on this premise, we resort to the work of SelfCheckGPT (Manakul et al., 2023), using the model itself to quantify the confidence of the prediction among multiple samplings, thereby detecting hallucinations.

Specifically, we first invoke the model to generate n drawn samples S^n. For the hypothesis and the i-th S^i sample, we invoke the prompt to query LLM and discern their consistency. After that, the hallucination probabilities can be calculated as $\sum_{i=1}^{n} x_i$, with the result x_i for each sample i mapped to a value between 0 and 1. If most of the samples are consistent with the original hypothesis, then the model is confident with its own prediction, and the hypothesis is likely not hallucinated, and vice versa.

We apply this method mainly to the PG task, as this task tends to produce different outputs for fixed inputs. Notice this method does not require the accessibility of glass-box features such as softmax distribution.

4 Experiments

4.1 Experiment-Setup

4.1.1 Data

As shown in Table 1, the organizers provided a validation set with manual annotations and an unannotated training set. As described in Section 3.1.2, for the entailment recognition method, we explore using LLMs like GPT-4 to automatically annotate the unannotated training data. For similarity search, factuality verification, and confidence estimation methods, we mainly rely on the validation set.

Dataset	Track	Task	Quantity
training	model agnostic	DM	1000
		MT	750
		PG	1000
		Total	2750
validation	model agnostic	DM	187
		MT	187
		PG	125
		Total	499
	model aware	DM	188
		MT	188
		PG	125
		Total	501

Table 1: Data Statistics

4.1.2 Pretrained Checkpoints

Regarding the DeBERTa-based entailment model, we mainly rely on DeBERTa-MoritzLaurer, which has already been trained on a diverse range of entailment datasets. For the InternLM-based entailment model, we utilize both the un-instructed and instructed tuned versions for comparison. To derive sentence embeddings from SBERT, we employ three high-performing variants from the text embedding leaderboard[3]. The specific links for all incorporated models are provided in Appendix B.

4.1.3 Task Tracks

This shared task is divided into two tracks: model-agnostic and model-aware. The former operates

[3] https://huggingface.co/spaces/mteb/leaderboard

without knowledge of the hypothesis-generating model. The latter, on the other hand, is informed about the model and can access its checkpoints.

4.2 Main Results

The experimental results are shown in Table 2. The following is a detailed analysis for both model-agnostic and model-aware tracks.

1) DeBERTa-based entailment model performs the best on hallucination detection.

As can be seen, among the four groups of methods, the entailment recognition model performs the best on hallucination detection, across both model-agnostic and model-aware tracks, especially DeBERTa-based entailment model. Although DeBERTa is 50 times smaller than InternLM, it generally outperforms InternLM, possibly due to its encoder-only structure being well-suited for language understanding tasks. Additionally, as the DeBERTa we used is pre-finetuned on various entailment datasets, knowledge can be transferred from other datasets to boost its performance.

Interestingly, the un-instruction tuned InternLM performs better than its instruction tuned version. This indicates the instruction tuning process is inconsistent with our objective and may cause catastrophic forgetting.

2) Similarity-based and factuality-based methods underperform.

In contrast to the entailment-based approaches, similarity-based and factuality-based approaches markedly underperform, potentially due to their mismatches with hallucination detection.

Regarding the similarity-based model, hallucinated sentences might still be similar in the embedding space, as SBERT can only provide general semantic representations. Besides, the Siamese architecture of SBERT also disables in-depth interaction between the source (target) and hypothesis within the multi-layer neural network.

As for the factuality-based model, it mainly aims to evaluate the factual consistency between the source (target) and the hypothesis text, which is a broader task than detecting specific hallucinations. Hallucinations can sometimes be factually consistent with the source information but still contain invented details or distortions, which UniEval's factuality evaluation may not be sensitive enough to capture, leading to poor performance.

3) Confidence estimation performs noticeably worse than other methods.

In the model-aware track, we employ confidence estimation method across all subtasks. We found that this method performed poorly in terms of acc and rho for the DM and MT subtasks. It achieved good acc but poor rho for the PG subtasks. Overall, confidence estimation performed noticeably worse. This can be attributed to two main reasons:

a) The softmax distribution contains insufficient information. The softmax distribution provides a probability distribution over the output vocabulary, but it may not capture all the nuances and uncertainties present in the model's predictions, especially when it comes to hallucinated content.

b) The model fails to provide an accurate evaluation for its prediction. As the prediction is made by the model itself, it is unable to provide an accurate evaluation for the consistency. The consistency verification can only be achieved with the help of external resources among multiple samplings.

Therefore, relying solely on confidence estimation methods may not be effective in detecting hallucinations, as the model itself can be overconfident for its hallucinated outputs.

4) Ensemble of multiple models can enhance performance to some extent.

In the model-agnostic track, the ensembled model achieves an improvement of 0.6 points in acc and 1.5 points in rho. However, in the model-aware track, while the ensemble model surpasses the performance of most models, it is slightly inferior to the best result of the DeBERTa model. We think this might be due to the underperformance of some ensembled methods.

4.3 Analysis of the DeBERTa-based Entailment Model

1) Rationale for not directly utilizing the DeBERTa in entailment model.

As mentioned before, we adopted DeBERTa-MoritzLaurer which is pre-finetuned on various entailment datasets rather than the original DeBERTa model for entailment-based methods. To verify the effectiveness of the pre-finetuning, we perform an ablation study based on the training set and SNLI (Bowman et al., 2015). As can be seen in Table 4, if directly fine-tune DeBERTa on either training set or SNLI, the accuracy on the validation set can achieve only 70%. However, we observe significant performance improvement by employing a two-stage fine-tuning approach, using these datasets sequentially.

Model Type	Model	Description	model-agnostic		model-aware	
			acc	rho	acc	rho
Baseline	Mistral-7B	not train	69.66	40.29	74.53	48.78
Entailment Recognition	InternLM2-20B	train	78.86	67.30	78.20	62.70
	InternLM2-20B-sft	train	63.53	50.35	64.86	46.77
	DeBERTa-MoritzLaurer	train and loss optimization	**82.46**	**75.20**	**80.46**	**71.23**
Similarity Search	SBERT	not train	76.80	63.73	75.66	62.65
Factuality verification	UniEval	not train	72.00	58.04	73.13	54.43
Confidence Estimation	Softmax Distribution	for DM task			59.07	26.08
	Softmax Distribution	for MT task			66.07	37.87
	Prediction Consistency	for PG task			81.33	7.91
Ensemble		ensemble all model	**83.06**	**76.77**	79.73	72.37

Table 2: Experimental results were compared with the baseline of prompting the Mistral-7B model. We use accuracy (which is abbreviated as acc) as the primary evaluation metric and employ Spearman's correlation (which is abbreviated as rho) to comprehensively assess our model's performance.

Model Type	Model	Description	model-agnostic		model-aware	
			acc	rho	acc	rho
Baseline	Mistral-7B	not train	69.66	40.29	74.53	48.78
Entailment Model	DeBERTa-MoritzLaurer	not train	78.00	67.96	63.26	8.21
	DeBERTa-MoritzLaurer	train	81.20	**75.80**	80.13	**71.65**
	DeBERTa-MoritzLaurer	train separately for each task	79.00	66.60	76.40	60.27
	DeBERTa-MoritzLaurer	train and loss optimization	**82.46**	**75.20**	**80.46**	**71.23**

Table 3: Results of different DeBERTa-based entailment models with the following configurations: 1) no training, using the pre-trained model directly; 2) direct fine-tuning using cross-entropy loss; 3) separate fine-tuning on each subtask using cross-entropy loss; 4) fine-tuning with loss optimization.

Model	Description	acc
DeBERTa	fine-tuning on training set	71.23
DeBERTa	fine-tuning on SNLI	72.12
DeBERTa	two stage fine-tuning	78.21

Table 4: DeBERTa model's performance with different fine-tuning settings.

Therefore, instead of directly utilizing the original DeBERTa model, we opted for models prefinetuned on entailment tasks. Specifically, we selected the DeBERTa-MoritzLaurer model, which is pre-trained on 33 entailment-related datasets, leveraging its transferable entailment recognition knowledge for effective hallucination detection.

2) Loss optimization improves the fine-tuning on LLM-annotated data.

To demonstrate the effectiveness of our proposed loss optimization method, we contrast it with various training methods. As shown in Table 3, while the original model can perform detection to some extent, fine-tuning on annotated data improved the performance. Based on that, our proposed loss optimization method takes into account not only the label but also the model's prediction situation, effectively mitigating overfitting, thereby further improving the performance.

5 Conclusion

In this study, we aimed to address the hallucination detection problem in SemEval-2024 Task 6. We established an ensemble model that includes entailment recognition, similarity search, and factuality verification models. For the model-aware track, we further leveraged confidence estimation for augmentation. Our approach proved effective as we ranked 3rd in the model-agnostic track and 5th in the model-aware track.

Although several methods were incorporated in our experiments, we realized that the best result was achieved primarily by relying on the DeBERTa-based entailment model. Given its portability and generalizability, we plan to further explore its use in hallucination detection in the future.

References

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.

Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, et al. 2023. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. *arXiv preprint arXiv:2312.09390*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2023. Gptscore: Evaluate as you desire. *arXiv preprint arXiv:2302.04166*.

Nuno M. Guerreiro, Elena Voita, and André Martins. 2023. Looking for a needle in a haystack: A comprehensive study of hallucinations in neural machine translation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1059–1075, Dubrovnik, Croatia. Association for Computational Linguistics.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Moritz Laurer, Wouter van Atteveldt, Andreu Casas, and Kasper Welbers. 2023. Building Efficient Universal Classifiers with Natural Language Inference. ArXiv:2312.17543 [cs].

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. 2021. Deduplicating training data makes language models better. *arXiv preprint arXiv:2107.06499*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Potsawee Manakul, Adian Liusie, and Mark JF Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *arXiv preprint arXiv:2303.08896*.

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.

Mathias Müller, Annette Rios, and Rico Sennrich. 2020. Domain robustness in neural machine translation. In *Proceedings of the 14th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 151–164, Virtual. Association for Machine Translation in the Americas.

Niels Mündler, Jingxuan He, Slobodan Jenko, and Martin Vechev. 2023. Self-contradictory hallucinations of large language models: Evaluation, detection and mitigation. *arXiv preprint arXiv:2305.15852*.

Thanapon Noraset, Chen Liang, Larry Birnbaum, and Doug Downey. 2017. Definition modeling: learning to define word embeddings in natural language. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 3259–3266. AAAI Press.

OpenAI. 2022. Chatgpt blog post. `https://openai.com/blog/chatgpt`.

OpenAI. 2023. Gpt-4 technical report. `https://www.openai.com/gpt4/`.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002a. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002b. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

InternLM Team. 2023. Internlm: A multilingual language model with progressively enhanced capabilities.

Vectara. 2023. Hallucination evaluation model.

Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. 2023. Alignscore: Evaluating factual consistency with a unified alignment function. *arXiv preprint arXiv:2305.16739*.

Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Pengfei Liu, Chenguang Zhu, Heng Ji, and Jiawei Han. 2022. Towards a unified multi-dimensional evaluator for text generation. *arXiv preprint arXiv:2210.07197*.

A The Prompt of GPT-4

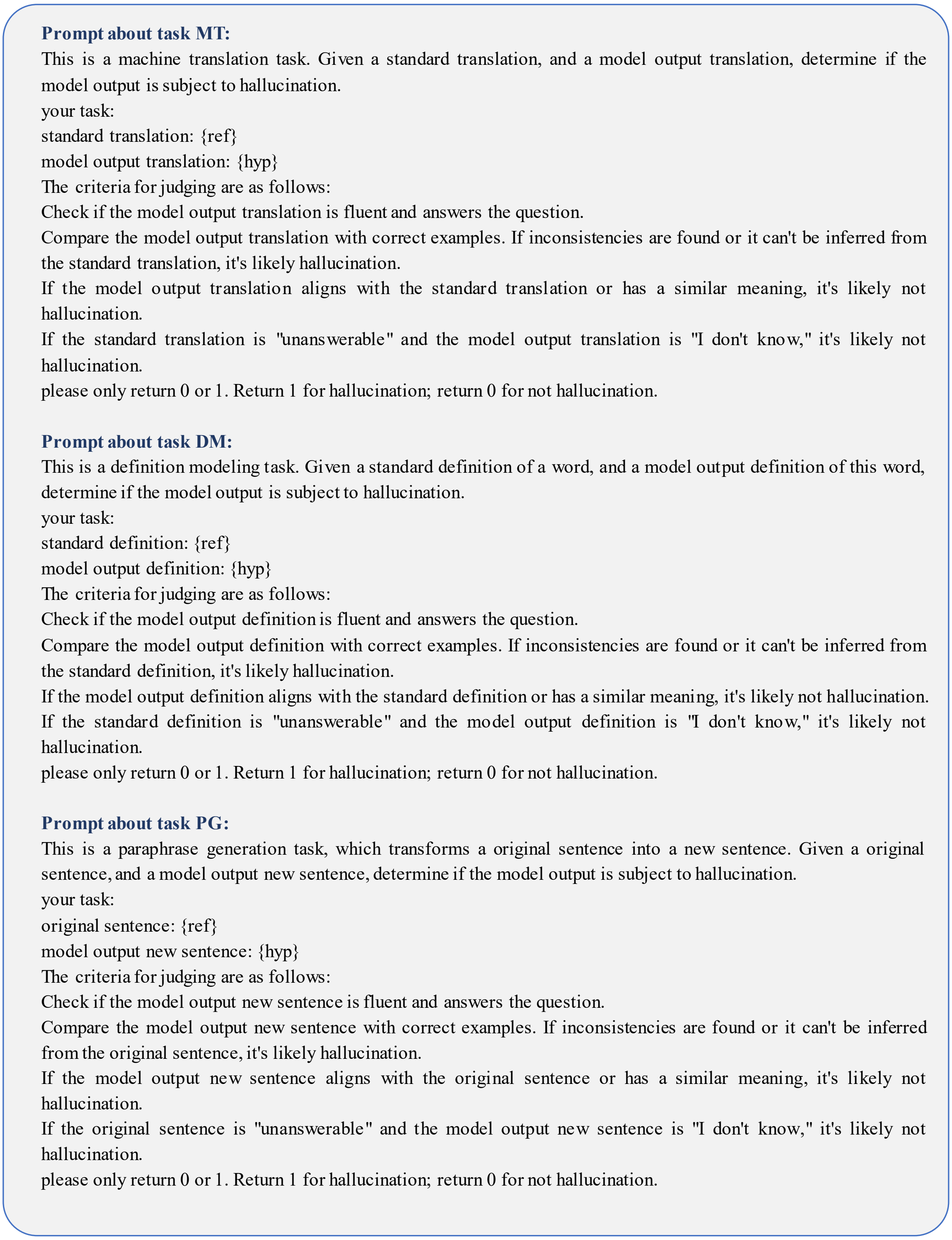
Figure 2 shows the specific prompt for asking GPT-4 to perform dataset annotation tasks.

B Utilized Model and Its URL

Table 5 shows the specific model and the corresponding download URL for the utilized model.

Model	**URL**
InternLM	https://huggingface.co/internlm/internlm2-20b https://huggingface.co/internlm/internlm2-chat-20b-sft
DeBERTa	https://huggingface.co/MoritzLaurer/deberta-v3-large-zeroshot-v1.1-all-33 https://huggingface.co/vectara/hallucination_evaluation_model
SBERT	https://huggingface.co/WhereIsAI/UAE-Large-V1 https://huggingface.co/llmrails/ember-v1 https://huggingface.co/BAAI/bge-large-en-v1.5
UniEval	https://github.com/maszhongming/UniEval
DM Task Checkpoint	https://huggingface.co/ltg/flan-t5-definition-en-base
MT Task Checkpoint	https://huggingface.co/facebook/nllb-200-distilled-600M
PG Task Checkpoint	https://huggingface.co/tuner007/pegasus_paraphrase

Table 5: Details of the utilized model and corresponding download URL



Prompt about task MT:

This is a machine translation task. Given a standard translation, and a model output translation, determine if the model output is subject to hallucination.

your task:

standard translation: {ref}

model output translation: {hyp}

The criteria for judging are as follows:

Check if the model output translation is fluent and answers the question.

Compare the model output translation with correct examples. If inconsistencies are found or it can't be inferred from the standard translation, it's likely hallucination.

If the model output translation aligns with the standard translation or has a similar meaning, it's likely not hallucination.

If the standard translation is "unanswerable" and the model output translation is "I don't know," it's likely not hallucination.

please only return 0 or 1. Return 1 for hallucination; return 0 for not hallucination.

Prompt about task DM:

This is a definition modeling task. Given a standard definition of a word, and a model output definition of this word, determine if the model output is subject to hallucination.

your task:

standard definition: {ref}

model output definition: {hyp}

The criteria for judging are as follows:

Check if the model output definition is fluent and answers the question.

Compare the model output definition with correct examples. If inconsistencies are found or it can't be inferred from the standard definition, it's likely hallucination.

If the model output definition aligns with the standard definition or has a similar meaning, it's likely not hallucination.

If the standard definition is "unanswerable" and the model output definition is "I don't know," it's likely not hallucination.

please only return 0 or 1. Return 1 for hallucination; return 0 for not hallucination.

Prompt about task PG:

This is a paraphrase generation task, which transforms a original sentence into a new sentence. Given a original sentence, and a model output new sentence, determine if the model output is subject to hallucination.

your task:

original sentence: {ref}

model output new sentence: {hyp}

The criteria for judging are as follows:

Check if the model output new sentence is fluent and answers the question.

Compare the model output new sentence with correct examples. If inconsistencies are found or it can't be inferred from the original sentence, it's likely hallucination.

If the model output new sentence aligns with the original sentence or has a similar meaning, it's likely not hallucination.

If the original sentence is "unanswerable" and the model output new sentence is "I don't know," it's likely not hallucination.

please only return 0 or 1. Return 1 for hallucination; return 0 for not hallucination.

Figure 2: The prompt of use GPT-4 to detection hallucination.