OctavianB at SemEval-2024 Task 6: An exploration of humanlike qualities in hallucinated LLM texts

Octavian Brodoceanu
University of Bucharest Faculty of Mathematics and Informatics
octavian.brodoceanu@gmail.com

Abstract

The objective of the SHROOM shared task (Mickus et al., 2024) is to identify sequences of text generated by Large Language Models that contain incorrect, nonfactual, or fabricated information. These sequences, referred to as 'hallucinations', are characterized by lower probabilities assigned to the outputs, as demonstrated by research (Varshney et al., 2023). This discrepancy highlights a possible contrast in the language used between hallucinated and non-hallucinated texts. The aim of this paper is to investigate whether hallucinated responses exhibit phrasing and patterns that more closely resemble those of machine-generated text rather than coherent, human-like language.

1 Introduction

The SHROOM shared task, as described by Mickus et al. (2024), has as its objective 'detecting grammatically sound output that contains incorrect semantic information (i.e. unsupported or inconsistent with the source input), with or without having access to the model that produced the output'. This type of output is encompassed by generations often referred to as "hallucinations". According to Varshney et al.,in the context of language models, hallucinations refer to the generation of text or responses that seem syntactically sound, fluent, and natural but are factually incorrect, nonsensical, or unfaithful to the provided source input. (Maynez et al., 2020; Holtzman et al., 2023; Koehn and Knowles, 2017) With the advent of mainstream Large Language Models brought upon by OpenAIs ChatGPT (Brown et al., 2020), it is a relatively new and important topic in this context. Apart from the possible spread of misinformation, being able to make this distinction is crucial for the adoption of Large Language Models in domains highly sensitive to misinformation, such as the medical, jurisdictional or financial fields. Possible repercussions include medical misdiagnosis, fictitious financial or legal advice, or exploitation by bad actors in order to deceive users.

In our approach to solving this task, we employ two methods. The first method involves utilizing models trained for detecting machine-generated text in order to distinguish between regular and hallucinated sequences. The other involves using looking at the loss of the hypothesis as scored by an LLM(Large Language Model), in the hope that generations with low probabilities can be properly tagged as hallucinations. This method was introduced by Fu et al. (2020) as GPTScore as a way to get a numerical assessment of an aspect in a given text.

When it comes to the first method, the hypothesis to be tested is that patterns which help differentiate machine generated texts will be transferable to the task at hand. The rationale is as follows: the training data of the model is human-written text, therefore deviations from the training set could be detected in this manner. From the experimental results on the model aware track, the performance of this method yielded a score of 0.483. This is below the baseline achieved using an LLM to label the generations. These results could stem from the hypothesis itself, or the fact that the model is not able to differentiate the texts of newer LLMs.

The second method was employed after the end of the competition, as a way to further explore the dataset and its characteristics. It was based on the success of Ji et al., who used a similar approach in a reprompting system meant to reduce hallucinations. On the validation data, it yielded an accuracy of 0.686 without the target reference and 0.702 when it was included in the prompt on the model aware track.

2 Background and Dataset

The objective of the SHROOM shared task is to detect hallucinations in two distinct datasets: one that is model agnostic and one that is model aware.

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1160–1165

Both of the datasets consist of text pertaining to 3 tasks: DM - 'Definition Modeling' - which involves providing the definition of a word given surrounding context, PG - 'Paraphrase Generation' - in which the generated text is meant to be a paraphrase of the input, and MT - 'Machine Translation' - in which the task is to translate a given sequence. The text provided for the definition modeling and the paraphrase generation task types is in English. For the machine translation task, the prompt is provided in the native language and the hypothesis and target are both in English. The model-aware validation dataset consists of 499 datapoints, while the model-agnostic version has 501 datapoints. The test sets both have 1500 samples.

Train Model Aware		
Column Name	**Description**	**Data Type**
hyp	Generated sequence	String
tgt	Desired target sequence	String
src	Prompt sequence	String
ref	Target column	Categorical
task	Prompt task type	Categorical
model	LLM model name	Categorical

Table 1: Description of Model Aware Dataset Columns

The main difference between the datasets is that the 'model' column is not present in the model agnostic version. This distinction is not relevant to the experiments presented in this paper, as no data is used apart from the 'hyp' - generated sequence column.

The datasets are split by the organizers in train, validation and test sets respectively, with the validation and test sets containing human-annotated labels. The probability of hallucination is defined as the average of the label given by each annotator, and the final label is chosen by majority vote from said labels. The validation and test set have 5 such labels per entry.

An example datapoint consists of the input 'Resembling or characteristic of a weasel.' - corresponding to this input, the output is structured as per Table 2.

Output		
label	**p(Hallucination)**	**id**
Not Hallucination	0.470	1

Table 2: Example of Model Aware Dataset Row

3 Related Work

Due to the importance and the relative novelty of the LLM hallucination detection task, there are many recently proposed ways to alleviate the issue. Ji et al. proposed a system for preventing hallucinations via self reflection, by using GPTScore as a way to gauge aspects such as factuality and consistency. Using a community sourced body of knowledge, for example wikipedia, in order to greatly enhance context (Semnani et al., 2023). Perturbations to the input to check for model self consistency (Zhang et al., 2023). Segmenting the generations and reprompting to check for consistency also appears to have lead to good results. (Wei et al., 2023; Zhou et al., 2023; Khot et al., 2023) Looking at the log probabilities of the output words to detect low-confidence generations (Varshney et al., 2023) has also been proposed, an approach very similar to one of the two methods used.

4 Methodology

4.1 Method 1: Generated Text Detection

The first method involves using a pretrained model for distinguishing machine-generated text. The decision to use this type of model stemmed in part from the similarity of the two tasks. Considering the fact that the training set for Large Language Models is often entirely human-written, deviations from the dataset - which are a possible cause of hallucinations - should appear as machine-like generations.

The model used during inference is 'roberta-large-openai-detector' (Solaiman et al., 2019). It is a a RoBERTa-large (Liu et al., 2019) model that has been trained in order to differentiate between texts generated by the Large Language Model GPT2 (Radford et al., 2019) after its inception. As the authors explain, it is able to distinguish texts generated by the LLM with 95% accuracy. The use of this model is, however, a limitation of the experiment. As cited by the authors (Solaiman et al., 2019) 'The model developers also report finding that classifying content from larger models is more difficult, suggesting that detection with automated tools like this model will be increasingly difficult as model sizes increase.' It should also be noted

that due to the nature of the MT - Machine Translation task, hallucinations of this type are unlikely to be picked up by the model.

Input is taken as the 'hyp' hypothesis column in the dataset. Since it is under the form of simple text, it will be tokenized using the 'roberta-large-openai-detector' tokenizer with padding and truncation. No other changes were made to the text.

Outputs are represented by the logits resulting from passing the tokenized input sequences through the model. The logits are then passed through a softmax function in order to obtain probabilities attributed to each class (0 - not generated/not hallucinated, 1 - generated/hallucinated). The class with the highest probability is saved as the 'label' and the probability of the input belonging to the 'hallucinated' label is 'p(hallucination)'. In the case of the test set, the id of the sequence is added to the structure to be added to the json.

4.2 Method 2: GPTScore

The second method involves prompting a pre-trained LLM with a task and checking the loss attributed to the predefined output.

The prompt is comprised of: instruction, demos, input and output.

Instruction prompts were constructed for each of the 3 tasks in the dataset, for example: "The following is a Definition Modeling task. Please focus on capturing the correct meaning based on the surrounding context in the original text. "

For each of the 3 tasks, demos were constructed by randomly sampling 3 datapoints from a subset of the validation dataset. This subset involves rows labeled "Not hallucination" by all five human annotators, in the case of the positive examples, and "Hallucination" in the case of the negative examples.

The input is the prompt sequence provided in the dataset. The output is the response provided in the same datapoint.

As an example, a prompt with no demos would be: Give the definition for the specified words in the given context. The answer for "The sides of the casket were covered with heavy black broadcloth , with velvet caps , presenting a deep contrast to the rich surmountings . What is the meaning of surmounting ?" is "A sloping top ."

The resulting output of the method is defined as the average of the logprobs of the output sequence (i.e. "A sloping top"). Naturally, the output score is predicated on the model doing the evaluation, with more accurate models having a higher chance of giving better results.

Optionally, the target sequence can be added to the prompt. Although this increases performance, as we would expect, it changes the use of the method to that of evaluation.

The models used include a quantized version of **Mistral-7B** and **OpenHermes-13B**. After generating the scores for each of the inputs, the goal is to employ simple binary classification. Logistic regression and SVM were tested, with logistic regression consistently giving superior results.

5 Experimental Setup

The first method is fully unsupervised, and therefore does not require calibration on the training set. The second method requires us to have a subset of labeled data to determine the score threshold.

For evaluating our models, we used the metrics proposed by the organizers: accuracy, based on the labels and Spearman's Rho, based on the probabilities assigned to each entry.

6 Results

The outputs on the test set model-aware track yielded a score of 0.483. The results on the validation dataset were the following:

Validation Set Results		
Track	**Accuracy**	**Rho**
Model Agnostic	0.545	0.033
Model Aware	0.465	-0.145

Table 3: Valdiation dataset results

The results for method 2 are shown in Tables 4 and 5:

Model Aware Track Validation Set Results				
Model	**Total**	**PG**	**MT**	**DM**
Mistral-7B	0.686	**0.776**	0.696	0.638
OpenHermes-13B	0.688	**0.808**	0.691	0.643

Table 4: Validation model aware dataset accuracy results

Model Aware Track Validation Set Results				
Model	**Total**	**PG**	**MT**	**DM**
Mistral-7B	0.687	0.704	**0.812**	0.657
OpenHermes-13B	0.701	0.688	**0.802**	0.625

Table 5: Validation model agnostic dataset accuracy results

Valdiation task aware results			
Track	**DM**	**PG**	**MT**
Model Agnostic	0.540	**0.624**	0.497
Model Aware	0.489	**0.608**	0.345

Table 6: Valdiation dataset task-aware accuracy results

6.1 Track results analysis

As evident, the model agnostic accuracy surpasses that of the model-aware track by a considerable margin. This difference could be due to statistical noise, as both results seem to be within 5% of the expected value for random binary attribution i.e. 50%. One competing hypothesis would be that the hidden distribution of the agnostic track allows for the model to better differentiate between the two classes.

Inferred sample label distributions in the form of 'Hallucinated'/'Not Hallucinated' are the following: 177/322 for model agnostic and 240/261 for model aware. For comparison, the real distributions are 218/281 model-agnostic track and 206/295. The fact that the model-aware results exhibit a near 50-50 split, in contrast to the model-agnostic track, whose distribution is closer to that of the real set, leads some credence to the hypothesis that the inference model is be able to detect relevant patterns.

Spearman correlation is calculated using the 'p(hallucination)' column. In the context of the proposed model, this is the probability assigned to class 1 ('Hallucination'). From the resulting values, it is evident that the probabilities of the reference and input display little to no correlation, with both values being near 0. From this, we come to the conclusion that the proposed method is not suitable for inferring the probability of hallucination.

6.2 Task-aware results

In order to investigate if the task had any impact on the performance of the model, standard accuracy was calculated for each separate subset of sequences. This was done on both the model aware and model agnostic track. The results are as per Table 6:

6.3 Task-aware results analysis

DM - 'Definition Modeling' showcases better performance on the agnostic dataset. As stated above, this could be attributed to random noise, or a difference in the distribution of the dataset.

PG - 'Paraphrase Generation' displayed the highest accuracy out of all three tasks on both the model-agnostic and model-aware tracks. It is a consistent and large enough improvement from the random baseline to be considered significant.

MT - 'Machine Translation' task results were the lowest, with the model reaching the expected random outcome of 50% on the model agnostic track. The results for the Model Aware Track showed an unexpected and significant difference, 15.5% from the random baseline. Low performance is to be expected, as this task requires the least free generation. These results could be due to the fact that the LLM is simply translating sequences that have been written by humans, and this requires less 'creative' generation on its part.

One plausible explanation for why the method has displayed superior performance on the PG task could be attributed to its inherently free-form nature compared to the other tasks. Definition Modeling is an information retrieval adjacent task, and Machine Translation leaves little room for interpretation, apart from cases of ambiguous wording. This suggests that the proposed method may have potential applications in specific tasks given to LLMs.

As per the results showcased in 4 and 5, we notice the increased performance when using OpenHermes-13B. This is to be expected, as it is the larger model, and the efficacy of the method is predicated on the quality of the certainty attributed by each model. We notice a leap in accuracy for certain tasks, Paraphrase Generation in the case of the model aware track, and Machine Translation in the case of the model agnostic track. This may once again be due to a difference in the distributions of the two tracks.

7 Additional experiments

Post competition, in addition to method 2, we have attempted to further finetune the RoBERTa model

to see if we can improve performance. In order to do this, we have used the dataset provided by Liang et al.. It is comprised of 749 datapoints, containing text generated by GPT3 and GPT4, as well as human written text. Before any additional operations, the model has an accuracy of 0.539 on this set, further confirming the fact that more powerful models and methods of detecting machine generated text are needed for use on newer LLMs. The dataset was split in a 9-1 ratio of training-validation data. After finetuning for 3 epochs the accuracy on the evaluation dataset reached an accuracy of 0.591. From this we can conclude the model is not able to properly learn. The results of the finetuning are shown in Table 7.

Validation Set Results	
Track	**Accuracy**
Model Agnostic	0.436
Model Aware	0.411

Table 7: Validation Set Results

The reason for why the model does not seem to learn can either be due to the model itself, or the generations are too humanlike to be told apart. As for the results of the finetuning, the performance of the model has dropped significantly on both datasets.

8 Conclusions

We have proposed two methods, the first based on using models pretrained on generated text detection, and the second based on looking at the confidence displayed by the LLM under the form of logits. Reviewing the results, we can assess that the tasks of generated text detection and hallucination detection showcase too large of a divergence for the approaches to generally be used interchangeably. However, the results on the Paraphrase Generation task may warrant further investigation into the use of models pretrained for text generation detection for the hallucination detection task. Concerning the second method we have explored, it has showcased promising results on specific tasks in each track, which may warrant use in an ensemble method.

8.1 Limitations

The main limitation of this experiment was the model used for inference. As it was trained to discriminate the generations of GPT2, which is a significantly smaller model compared to the current Language Models.

8.2 Future Work

In future work, we might explore model finetuning on newer datasets used for discerning between human and machine-generated texts, as well as finetuning pretrained models on labeled hallucination related tasks.

Another simple improvement would be utilizing pretrained models able to better distinguish between generated and human written texts.

We may test the first method on other free-form generation tasks, as this seems to be a strong suit.

We may also look into newer methods for detecting machine generated text, that account for the leaps made by the recent advancements in LLMs.

We may further investigate the discrepancy in accuracy for the provided datasets when using GPTScore.

Acknowledgements

I would like to thank Ana Sabina Uban for kickstarting the idea for this work.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *arXiv:2005.14165v4*.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2020. Gptscore: Evaluate as you desire.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2023. The curious case of neural text degeneration.

Ziwei Ji, Tiezheng Yu, Yan Xu, Nayeon Lee, Etsuko Ishii, and Pascale Fung. 2023. Towards mitigating llm hallucination via self reflection.

Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2023. Decomposed prompting: A modular approach for solving complex tasks.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation.

Weixin Liang, Mert Yuksekgonul, Yining Mao, Eric Wu, and James Zouv. 2023. Gpt detectors are biased against non-native english writers. *arXiv:2304.02819*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv:1907.11692*.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, , and Ryan McDonaldr. 2020. On faithfulness and factuality in abstractive summarization.

Timothee Mickus, Elaine Zosa, Raúl Vázquez, Teemu Vahtola, Jörg Tiedemann, Vincent Segonne, Alessandro Raganato, and Marianna Apidianaki. 2024. SemEval-2024 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)*, Mexico City, Mexico. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Sina Semnani, Violet Yao, Heidi Zhang, and Monica Lam. 2023. Wikichat: Stopping the hallucination of large language model chatbots by few-shot grounding on wikipedia.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, Miles McCain, Alex Newhouse, Jason Blazakis, Kris McGuffie, and Jasmine Wang. 2019. Release strategies and the social impacts of language models. *arXiv:1908.09203v2*.

Neeraj Varshney, Wenlin Yao, Hongming Zhang, Jianshu Chen, , and Dong Yu. 2023. A stitch in time saves nine: Detecting and mitigating hallucinations of llms by validating low-confidence generation. *arXiv preprint arXiv:2307.03987*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

Jiaxin Zhang, Zhuohang Li, Kamalika Das, Bradley A. Malin, and Sricharan Kumar. 2023. Sac3: Reliable hallucination detection in black-box language models via semantic-aware cross-check consistency.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V Le, and Ed H. Chii. 2023. Least-to-most prompting enables complex reasoning in large language models.