HalluciBERT: Efficiently Predicting Likelihood of Hallucation Pre-Inference

Anthony Baez
acbaez@mit.edu

Audrey Douglas
adouglas@mit.edu

Michael Wong
mwong21@mit.edu

Nathan Guntvedt
nathang0@mit.edu

Abstract

Recently developed Large Language Models (LLMs) have unprecedented natural language ability. However, they can be prone to hallucinate, where the LLM relays false information with the same authority as true information. Among current hallucination mitigation techniques, there are relatively few that attempt to predict hallucinations before inference. Additionally, most methods rely on models that are similar in computational size to the LLM. In order to address this, we developed HalluciBERT, a BERT-based model that scores prompts according to their likelihood of producing a hallucination in a LLM response. HalluciBERT outperformed our naive model on confidence score prediction with high correlation between predicted and ground truth scores, but had limited accuracy in the task, with only 51.79% of scores at most 10 points from the ground truth.

1 Introduction

Recently developed generative Large Language Models, such as GPT-3 [4], have made significant improvements in language generation over their predecessors. Modern LLMs outperform early models on various benchmarks and are able to process and respond in natural language to a nearly human degree [14]. They can also be trained on billion-token datasets and recall information in a large range of topics with accuracy and depth.

However, these LLMs can sometimes produce hallucinations, in which false information is relayed with the same authority as true information in the model's response [16]. As these LLMs evolve and are given more real-world responsibilities, it becomes increasingly vital that they respond truthfully, preventing possible harm and limiting the spread of misinformation. Therefore, methods must be developed to prevent or mitigate hallucinations to fully realize the potential of LLMs.

Many methods currently exist to detect and mitigate hallucinations in LLMs, but most involve models that are equally as large as the original LLM, increasing computational and monetary need [16]. To meet this need some hallucination detection is preventative, wherein the prompt is judged before inference. This offers the benefit of preventing the output of inaccurate information and saving the computational resources that would have been used on inference. Notably, hallucinations may also sometimes be desired, particularly in creative applications [10].

One method of hallucination mitigation uses scores to quantify the likelihood of a model hallucinating [16]. These scores are usually calculated from the content of a LLM response. Using a scoring system offers more insight and control to users of LLMs. Prediction of a score would also be a simpler task for a language model rather than proposing changes in the language of a LLM response.

In response to these challenges, our work focuses on developing a novel method for predicting the likelihood of hallucinations in LLMs based on the content of prompts before model inference. We aim to construct an smaller auxiliary model using BERT [6] to reduce the cost of hallucination prediction. We created a dataset of LLM prompts and corresponding confidence scores, trained HalluciBERT, our BERT-based model, to predict the likelihood of hallucinations based on prompt content, and evaluated its effectiveness.

2 Related Works

2.1 Mitigation of Hallucinations

Hallucinations in LLMs are a recognized issue, and detection, explanation, and mitigation of hallucinations is an ongoing area of research. Hallucination mitigation techniques include curating training

data, honesty-oriented fine-tuning, and altering inference with knowledge retrieval or exploitation of uncertainty [16]. One method to improve factuality in the training set includes implementation of a heuristic to more carefully select which internet data is included in the training set [9]. Another utilizes an LLM to annotate and filter a dataset for fine-tuning, allowing the model to view factual examples to decrease prevalence of hallucination [5]. At inference, retrieval augmentation, where the LLM fact checked itself at multiple steps to improve its response, has also been explored [11].

2.2 Detection of Hallucinations

Within the field of hallucination mitigation is hallucination detection. Hallucination detection methods have been classified into inference classifiers, uncertainty metrics, self-evaluation, and evidence retrieval [15]. Among the methods, LLMs have been used to judge the truthfulness of a prompt, with broad agreement marking an output with a lower likelihood of hallucination [18]. Without needing LLM inference, the truthfulness of an output can also be assessed using the model's internal state to train a classifier with its activation values [2]. Another method tagged LLM outputs that were identified to be outside of their domain knowledge. When a tagged model was supplied with additional context, hallucination was reduced [7].

2.3 Uncertainty Scoring

Scoring is a commonly used method to identify hallucination, as it allows for easy interpretability and comparison. Measuring the confidence of a language model's output through its logits was first discussed in the context of machine translation using a multi-layer perceptron [3]. Scoring metrics also include Answer Uncertainty Disparty, or AUD, POLAR, and Probability Scoring. AUD is calculated from the average of the differences of semantic features between answers provided to previously known and unknown questions [1]. POLAR, Pareto optimal learning assessed risk, relies less on natural language features, and uses a Pareto optimal self-supervision framework to produce a risk score for LLM responses [17]. The scoring method we used, called probability scoring, calculated the score by taking the minimum of the log probabilities, or logits, of important tokens. [14].

3 Method

3.1 Dataset

Curating a Prompt Dataset To create a dataset of prompts, we used a subset of Alpaca's question-answer dataset [12]. We chose this source because it has prompts with varying types of questions that would have different likelihoods to result in hallucinations. The Alpaca dataset contains 52K question-answer pairs, generated by ChatGPT through a self-instruction method seeded with human prompts. We made a subset of 12K questions from the question-answer pairs of the Alpaca dataset to be our prompt dataset. These questions were filtered to vary less in length, so length of prompt did not significantly influence results. To generate confidence scores, we used the probability score method described in Varshney et al., 2023 [14].

Generating Output Tokens and Probabilities Using OpenAI's API, we ran each of our 12K prompts through text-davinci-003 to get the output tokens and their corresponding log probabilities. We chose this model because it is one of a few OpenAI models that returns output log probabilities along with the response. Further, using their API allowed us to generate results faster than if we used our own compute.

Identifying Important Concepts We define a *concept* as the concatenation of one or more consecutive output tokens in a response, capturing the main ideas of the text. In our approach to identifying important concepts, we initially experimented with various methods, including Keyword Extraction and Entity Extraction [14]. However, these techniques did not yield satisfactory results in accurately capturing the comprehensive set of concepts as per our definition. As with the work that proposed this method, the Instructing Model method demonstrated superior performance in concept extraction tasks [14]. Consequently, we utilized OpenAI's API with their gpt-3.5-turbo-1106 model for this purpose. To generate the concepts, we passed each output response through this model with the prompt. Examples are provided in Appendix A. The model outputs a list of important concepts which are used in calculating the confidence score. However, for some responses, the model would output invalid concepts that were not aligned with our definition. We ignored these responses, reducing our dataset by 367 data points.

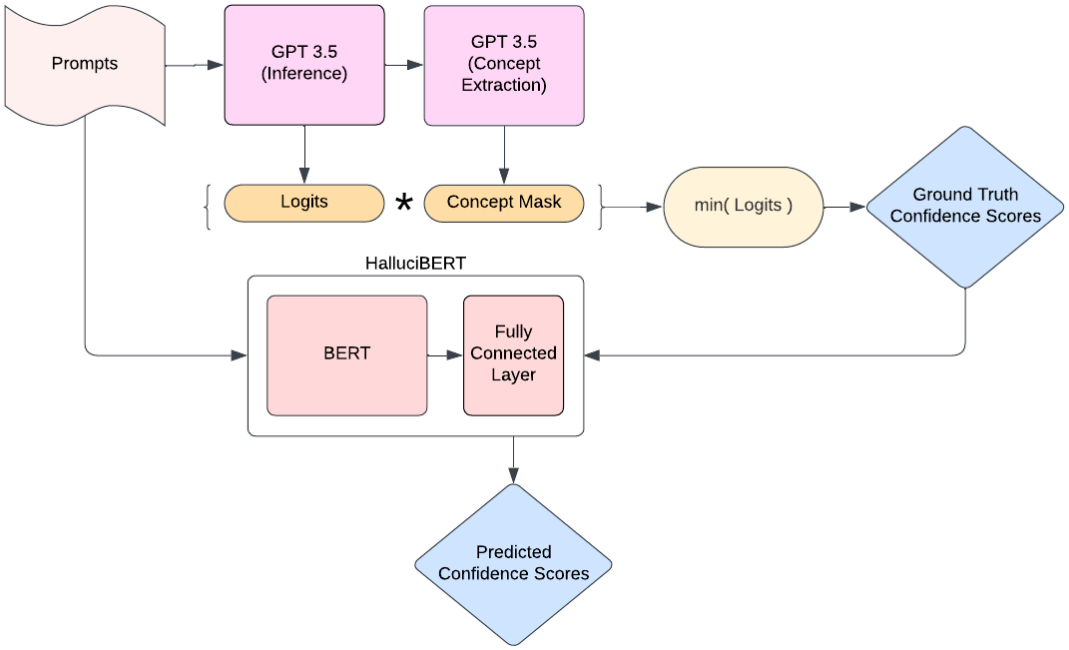

Figure 1: Dataset Generation and Model Pipeline.

Calculating Confidence Scores As input to get prompt confidence scores, we used the log probabilities of the tokens, identified as concepts by gpt-3.5-turbo-1106, from the output from text-davinci-003. This was done because gpt-3.5-turbo-1106 is the superior model, but only text-davinci-003 gave us output logits.

To calculate confidence score, we define C as the set of output token indices that are included in *any* identified concept for a given response. Additionally, we define p_i as the (normalized) output token probability of token i. We define a confidence score, S, for the corresponding prompt as:

$$S = \min_{i \in C} \{p_i\} \tag{1}$$

Each score is in the range [0,1]. The minimum of the log probabilities of the concept tokens was taken to be the confidence score of the whole prompt. This has previously been shown to be the most effective, as we want to define the prompt's confidence at the point where the model is the least confident in its answer [14]. When a model begins to hallucinate, the first token in its hallucination likely has a low probability, but the subsequent tokens may have high probabilities. As such, taking the average of logits has little correlation to hallucination. Taking the minimum more accurately aligns with the idea of predicting if a hallucination ever occurs. After calculating all scores, we scaled the confidence scores to a range of [0,100].

Analysis of Distribution of Confidence Scores When we created our original dataset, we noticed that the distribution of the confidence scores was very skewed toward the lower end of the distribution. The remnant of this can be seen in Figure 2A. To remedy this, we created a more uniform subset of 2K total prompt-confidence score pairs, which was used for the training method. The remaining 10K data points were kept for additional model testing.

3.2 Models

To predict LLM hallucinations with a smaller model, we selected BERT to be the basis of the model. BERT is an extensively used masked language model. At 110M parameters [6], it is significantly smaller than modern LLMs like GPT-3, which is 175B parameters [4]. BERT allows us to learn contextual word embeddings of our prompts. To transform these embeddings into a confidence score, we added a fully connected layer to the end of BERT. We refer to this model at HalluciBERT.

We also created a naive model to act as a baseline for comparison to HalluciBERT. Our naive model predicts the mean of all confidence scores for the expected distribution, as this is the most effective

simple approach. The expected distribution we used was the training set when ran on the 2K model, and the average of all 12K data points for the 10K dataset, as there was no training set for the 10K dataset.

3.3 Training

The generated 2K dataset was split with an $80/10/10$ training-validation-testing ratio. The loss function used was Mean Squared Error (MSE). Training was done for 10 epochs. Further details can be found in Appendix A.

3.4 Evaluation

Post-training, we evaluated the model using several metrics, including MSE, Mean Average Error (MAE), and accuracy, which we define as the percentage of predictions within a certain range of the actual value. These metrics helped us understand the model's effectiveness at the prediction task the reliability of its predictions. We also analyzed the model's predictions compared to the actual data, looking at the best and worst 10% of predictions to find any patterns or correlations, such as the relationship between the number of important tokens in a prompt and the confidence score. We also compared word count differences between the highest and lowest confidence scores.

4 Results

Table 1 contains the MAE, MSE, and accuracy of our two models: HalluciBERT and the naive model. The table shows their performances on our two datasets: the 2K Uniform and 10K datasets. Accuracy was was defined based on whether the predicted confidence score was within 10 of the ground truth. On the 2K Uniform dataset, HalluciBERT had an MAE of 15.80, MSE of 454.98, and accuracy of 0.4336 and outperformed the naive model on all metrics, which were an MAE of 23.95, MSE of 786.91, and accuracy of 0.2358. On the 10K dataset, HalluciBERT had an MAE of 13.63, MSE of 357.36, and accuracy of 0.5282 which also outperforms the naive model on all metrics at an MAE of 14.82, MSE of 488.70, and accuracy of 0.5179.

The first column of Figure 2 (defined as Figure 2A) shows histograms of the distribution of confidence scores in the testing set of the 2K Uniform and 10K datasets. The second column (defined as Figure 2B) shows confusion matrices of accuracies of our HalluciBERT model. The third column (defined as Figure 2C) shows the cumulative distribution function (CDF) of the absolute difference between the predicted and ground truth confidence scores for the naive and HalluciBERT models on both testing sets.

The histograms show that in the 10K dataset, the classes of scores appear to be skewed toward the lower end of the distribution. For the 2K Uniform dataset, the distribution appears significantly more uniform than the 10K dataset. Additionally, the predicted confidence scores are more skewed toward lower values than the ground truth confidence scores in the 10K dataset. This result also justified our creation of the 2K Uniform dataset.

The confusion matrices allow us to scrutinize which data points are being misclassified. They also allow us to easily view how many data points are being misclassified into the next lowest or next highest confidence score range. We can see how often HalluciBERT predicts the correct class by looking at the diagonal in the confusion matrix. The minimum value on this diagonal on the 2K Uniform is 0.27 and the maximum is 0.40, so HalluciBERT's accuracy was between 27% and 40% when using the range in the matrix. Results improve even more on the 10K dataset, with a range of accuracy of 44% to 68%.

The CDFs allow us to visualize the absolute error across the testing set. We notice that most of the CDF of our HalluciBERT model more left than the naive model on the 2K Uniform dataset, indicating that HalluciBERT has lower error on most data points, but also contains significant error outliers toward the top of the curve. On the 10K dataset, there is not much difference between both models, indicating less of an advantage of HalluciBERT

In order to verify that our confidence score metric was capturing a meaningful features in the prompts, we analyzed the content of the prompts. The results of this is shown in Table 2. We discovered that some words in the prompts were more often associated with a lower confidence score, and some words were associated with a higher confidence score.

5 Discussion

HalluciBERT Does Not Learn Distributions One of the concerns with the original 12K dataset's skew towards low confidence scores was that HalluciBERT would simply learn to always predict

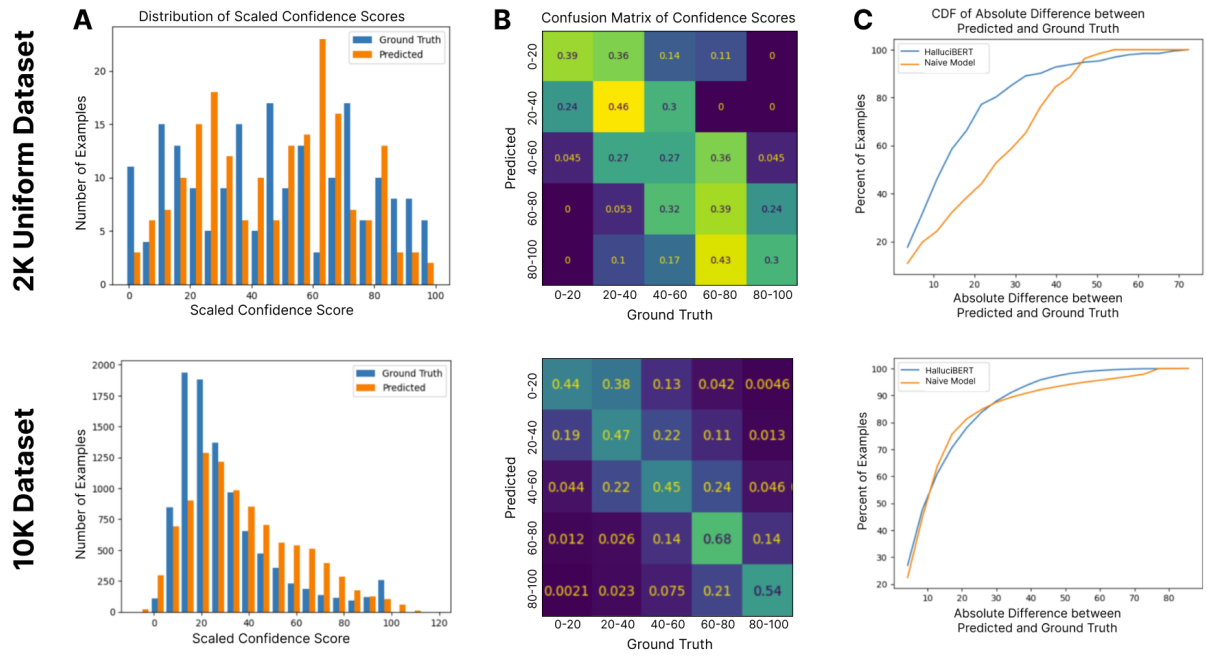

Figure 2: Performance of HalluciBERT and our Naive Model. (A) Charts show the distribution of scaled confidence scores between HalluciBERT's predicted scores versus the ground truth distribution. (B) Rows of the confusion matrices show the normalized distribution of predicted confidence scores when compared to the corresponding ground truth values. (C) The graphs represent the cumulative distribution function of the absolute difference between HalluciBERT's predicted confidence scores and the ground truth values.

Test Dataset	HalluciBERT			Naive		
	MAE	MSE	Accuracy	MAE	MSE	Accuracy
2K Uniform	15.80	454.98	0.4336	23.93	786.91	0.2358
10K	13.63	357.36	0.5282	14.82	488.70	0.5179

Table 1: MAE, MSE, and Accuracy for HalluciBERT and our Naive Model on both datasets.

Confident Words		Uncertain Words	
The	649	A	-479
Name	172	Generate	-207
What	99	Describe	-147
Identify	55	Poem	-95
One	39	Create	-77

Table 2: Weighted frequencies of words corresponding to high and low confidence scores. To calculate this, we add 1 each time a word appears in a prompt in the top 10 percent of all scores. We subtract 1 when they appear in prompts in the bottom 10 percent.

values near the mean of the dataset. When training on this non-uniform dataset, we saw this occur, as the performance of HallluciBERT was very close to that of the naive model on the 12K testing set. This meant that the model was not discouraged enough for always making the same prediction. This outcome led to us creating the 10K and 2K Uniform datasets.

Once the 2K Uniform dataset was created and trained on, we saw the range of predicted confidence scores greatly increase. As seen Figure 2A, the distribution of predicted confidence scores closely matched that of ground truth. Additionally, as can be seen in Figure 2C, the performance of HalluciBERT and the naive model on this dataset are quite different, indicating that HalluciBERT is not just guessing the mean. However, the histogram does not provide any information of whether the corresponding predicted and ground truth values

are in the same bucket. It only shows the total number of predicted and ground truth in some bucket, so we are limited in the conclusions we can draw from it.

The skew in the predicted confidence scores is most striking in our 10K dataset, where its distribution shares a leftward skew and similar visual behavior. The predicted confidence scores from HalluciBERT does seem to follow the same general pattern as the ground truth, but the distribution is less extreme. Additionally, HalluciBERT had a tendency to avoid low or high outlier scores, as the loss reduction from guessing them would not be as high as guessing the middle of the distribution.

HalluciBERT Predicts Confidence Scores with High Correlation to Ground Truth In addition to having the correct distribution, it is important to note what kind of values HalluciBERT is better at predicting and if there any noticable pattern in its failure to predict correct confidence scores.

As such, looking at the confusion matrix in Figure 2B of ground truth versus predicted scores help us accomplish this. Having 5 different ranges, instead of the 20 in the histogram, could be more consistent with our understanding of an accurate prediction. We then see there is high correlation between predicted confidence scores and ground truth.

Looking at the numbers on the confusion matrices, we see that the accuracy values for HalluciBERT on both datasets are all higher than the expected 20% accuracy of the naive approach. The confusion matrix also allows us to easily explore increasing our definition of accuracy. If we were to expand accuracy to include those in the next higher and next lower predicted class, by adding up the value in the higher and lower neighbor of the confusion matrix, HalluciBERT ranges from 75% to 96% accuracy on the 10K dataset. This is an improvement from 44% to 68% with only the boxes in the diagonal. These are good accuracies, and these new bounds are still meaningful. Overall, our model rarely predicted values that were highly inaccurate, which implies that correlation between prompt and confidence score in HalluciBERT does exist.

Accuracy Improves Rapidly with Tolerance The goal of our work was to predict confidence scores within 10 of their actual score. We fell short of this on both of our testing sets, with HalluciBERT on the 2K Uniform data having a MAE of 15.80 and on the 10K dataset having a MAE of 13.63, as seen in Table 1. Furthermore, our accuracy for neither dataset was very high, with the 2K dataset at 0.4336 and the 10K dataset at 0.5282. These would not be acceptable if we were to deploy this model in real use cases, but it would insightful to determine how our model performs if we relax the definition of being accurate.

As previously discussed, we were hoping to have our model accurate within a score of 10. However, depending on the specific use case of such a hallucination prediction model, the tolerance, or how close the confidence score needs to be to the ground truth value, could be higher. By relaxing tolerance to 20, which could be thought of as a $1-5$ scale, or even to a binary classification task, the model may be useful and applicable. In particular, we see that both CDF curves in Figure 2C increase steeply near 0% of examples, but the slope decreases steadily near 80%. This means that a larger accuracy threshold would greatly improve results. We determined that at a tolerance of 15, HalluciBERT would have an accuracy of 0.5974 on the 2K Uniform dataset and an accuracy of 0.6591 on the 10K dataset. At a tolerance of 20, HalluciBERT would have an accuracy of 0.7282 on the 2K Uniform dataset and an accuracy of 0.7614 on the 10K dataset. By making the definition of accuracy a bit more lenient, it can be understood that HalluciBERT could be accurate enough to be useful and deployable.

Words in Prompt Correlate to Confidence Score We analyzed the content of the prompts, and found a relationship between the presence of certain words and confidence score. In Table 2, 'the' is associated with confident prompts and 'a' is associated with unconfident prompts. This makes sense, as 'the' is the definite article and specifies some noun, and 'a' is the indefinite article and does not specificy some noun. This pattern also occurs with other words. Words like 'what' and 'identify' would have specific correct answers. Assuming the model would be able to come up with the correct answer or say that it does not know, it would be confident in it answer. On the other hand, words such as 'describe' or 'create' would not have specific correct answers. Since there would be many different options of response, the single response it chooses would not have a high confidence. However, the correlation between a word and confidence score was not always consistent, so purely using the

words as a feature would not be accurate. Therefore, a language model such as BERT is needed to take into account the nuances of the language of the prompt.

Limitations Our study faced several limitations. First, it relies on a pre-existing dataset, limiting the scope and diversity of our training data. This reliance restricts our findings to the specific contexts represented in the dataset, which affects the generalizability of our model. Second, the size of our training set was constrained by the costs of the API calls. This limitation restricted the amount of data available for training, which limits the effectiveness of our model.

Further, our approach is dependent on LLMs for concept extraction, specifically GPT-3.5. GPT-3.5 occasionally failed in accurately mapping extracting concepts from tokens. This was addressed by removing prompts where accurate concept extraction failed, but this further limited the size of our training set.

Since HalluciBERT was trained on a dataset from GPT-3.5, its effectiveness is limited to this LLM. Testing HalluciBERT on other models like LLaMA yielded significantly different results, suggesting that our approach may not be directly applicable to other models. Thus, fine-tuning the base architecture again for each specific model could be needed to accommodate these differences.

Lastly, our confidence score calculation is based on the minimum logit value, focusing only on the probability of at least one hallucination and not explicitly considering the probability of multiple hallucinations. To address this and simplify the task further, HalluciBERT could use binary classification to determine whether a hallucination is more probable than not, as was done with the model that first used the minimum logit confidence score technique [14].

6 Conclusion

In this paper we created a novel model, HalluciBERT, to predict confidence scores from prompts before inference. HalluciBERT outperformed our naive model on MAE, MSE, and accuracy, defined as having a prediction within 10 of the ground truth confidence score. Despite this, HalluciBERT did not achieve a high accuracy with this definition. However when the accuracy threshold is loosened to 15 or 20, which can be useful bounds under different conditions, the accuracy notably improves. These results demonstrate that while HalluciBERT did not achieve good accuracy with the desired the threshold, our method and approach hold promise, and additional work should be done to better realize our goal of training a smaller, more computationally efficient model to predict confidence scores pre-inference.

For future work, our goal is to not only to enhance HalluciBERT's ability to predict hallucinations in other LLMs but to also align it closely with the core NLP values of reliability and generalizability. To this end, we could create a larger and more diverse dataset made up of content from different LLMs for fine-tuning. This approach could improve the model's performance across more varied data distributions and LLM architectures. Additionally, we intend to evaluate other BERT variants and transformer encoders. This exploration would further reduce model size and optimize computational efficiency at the hallucination prediction task.

7 Impact Statement

Our research attempts to build a more efficient model that can predict the likelihood of hallucinations. This work addresses hallucinations, in which LLMs can output falsehoods with the same confidence as truths. As LLMs and AI models more broadly become more prevalent and powerful, such models could come to worsen the already prevalent problem of misinformation. Our work stands to address this issue, by empowering users of LLMs to gauge the factuality of their prompts. The preventative aspect also saves time and compute that would otherwise be spent on LLM inference. If such a prediction model was instituted on a large scale, it could lessen resource and electricity consumption, essential to preventing the worsening of the climate crisis.

Our research involved using previously created tools and procedures, so the ethics of these need to be taken into account. BERT, the model we used to constructing contextualized word embeddings, has demonstrated tendencies for racial, gender, and other biases, and our modifications do not directly address those issues [8]. Consequently, there is a risk of these biases influencing our model's predictions in hallucination detection. This could potentially lead to biased treatment of content related to specific racial or gender groups.

Furthermore, neither the Alpaca dataset nor the generated dataset we used was not inspected manu-

ally for any significant bias. Any bias in the dataset would influence the model's learning process and its outcomes [13]. The Alpaca dataset is also limited in the types and category of prompts it has, so if our model is presented with questions outside of the training data distribution, the way it would assign confidence scores may be biased.

In light of these considerations, our research should be understood as a step in the ongoing journey of AI development. We highlight the need for further research to ensure that we advance AI models that are equitable, sustainable, and beneficial for humanity.

References

[1] Alfonso Amayuelas et al. *Knowledge of Knowledge: Exploring Known-Unknowns Uncertainty with Large Language Models*. 2023. arXiv: 2305.13712 [cs.CL].

[2] Amos Azaria and Tom Mitchell. "The internal state of an llm knows when its lying". In: *arXiv preprint arXiv:2304.13734* (2023).

[3] John Blatz et al. "Confidence estimation for machine translation". In: *Coling 2004: Proceedings of the 20th international conference on computational linguistics*. 2004, pp. 315–321.

[4] Tom Brown et al. "Language models are few-shot learners". In: *Advances in neural information processing systems* 33 (2020), pp. 1877–1901.

[5] Lichang Chen et al. *AlpaGasus: Training A Better Alpaca with Fewer Data*. 2023. arXiv: 2307.08701 [cs.CL].

[6] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL].

[7] Philip Feldman, James R. Foulds, and Shimei Pan. *Trapping LLM Hallucinations Using Tagged Context Prompts*. 2023. arXiv: 2306.06085 [cs.CL].

[8] Michael A Lepori. "Unequal representations: Analyzing intersectional biases in word embeddings using representational similarity analysis". In: *arXiv preprint arXiv:2011.12086* (2020).

[9] Guilherme Penedo et al. *The RefinedWeb Dataset for Falcon LLM: Outperforming Curated Corpora with Web Data, and Web Data Only*. 2023. arXiv: 2306.01116 [cs.CL].

[10] Vipula Rawte, Amit Sheth, and Amitava Das. *A Survey of Hallucination in Large Foundation Models*. 2023. arXiv: 2309.05922 [cs.AI].

[11] Ruiyang Ren et al. *Investigating the Factual Knowledge Boundary of Large Language Models with Retrieval Augmentation*. 2023. arXiv: 2307.11019 [cs.CL].

[12] Rohan Taori et al. *Stanford Alpaca: An Instruction-following LLaMA model*. https://github.com/tatsu-lab/stanford_alpaca. 2023.

[13] Antonio Torralba and Alexei A Efros. "Unbiased look at dataset bias". In: *CVPR 2011*. IEEE. 2011, pp. 1521–1528.

[14] Neeraj Varshney et al. "A stitch in time saves nine: Detecting and mitigating hallucinations of llms by validating low-confidence generation". In: *arXiv preprint arXiv:2307.03987* (2023).

[15] Hongbin Ye et al. "Cognitive mirage: A review of hallucinations in large language models". In: *arXiv preprint arXiv:2309.06794* (2023).

[16] Yue Zhang et al. "Siren's Song in the AI Ocean: A Survey on Hallucination in Large Language Models". In: *arXiv preprint arXiv:2309.01219* (2023).

[17] Theodore Zhao et al. *Automatic Calibration and Error Correction for Generative Large Language Models via Pareto Optimal Self-Supervision*. 2023. arXiv: 2306.16564 [cs.CL].

[18] Lianmin Zheng et al. "Judging LLM-as-a-judge with MT-Bench and Chatbot Arena". In: *arXiv preprint arXiv:2306.05685* (2023).

A Appendix

Prompt for Concept Identification Using gpt-3.5-turbo-1106 we used the following few-shot prompt to generate concepts [14].

```
Identify all the important
keyphrases in order from the
following text and return a comma
separated list.

Text:  John Russell Reynolds
was an English physician and
neurologist who made significant
contributions to the field of
neurology.

Response:  John Russell Reynolds,
English, physician, neurologist,
neurology

Text:  He was born in London in
1820 and studied medicine at the
University of London.

Response:  London, 1820,
medicine, University of London

Text:  After college, he worked
as a lawyer for the PGA Tour,
eventually becoming the Tour's
Deputy Commissioner in 1989.
Response:  college, lawyer, PGA
Tour, Deputy Commissioner, 1989

Text:  <INPUT>

Response:
```

Training Set-up We used PyTorch and Google Colab to write the code for our paper. For training, we used the AdamW optimizer, a learning rate of 5e-5, and a batch size of 2.