Black-Box Hallucination Detection: A Tunable Ensemble Approach

Dylan Bouchard[1] * **Mohit Singh Chauhan**[1]
[1]CVS Health®

Abstract

Hallucinations are a persistent problem with Large Language Models. As these models become increasingly used in high-stakes domains, such as healthcare and finance, the need for effective hallucination detection is crucial. While many conventional approaches rely on measuring faithfulness to source content or external search, the need for state-of-the-art zero-resource hallucination detection still exists. To this end, we introduce a versatile, tunable-ensemble approach to zero-resource hallucination. Our ensemble includes various components to measure hallucination likelihood, including semantic entropy, LLM-as-a-judge, BertScore and various other scoring methods. As different components will be more informative for different use cases, users can select which component to include, and tune the weights based on their specific use cases. Our experiments reveal that our ensemble approach outperforms state of the art approaches across various benchmarks.

1 Introduction

Large language models (LLMs) are being increasingly used in production-level applications, often in high-stakes domains such as healthcare or finance. Consequently, there is an increasing need to monitor these systems for the accuracy and factual correctness of model outputs. Mistakes in these sensitive use cases can lead to high financial costs and reputation damage. A particularly concerning risk for LLMs is hallucination, where LLM outputs sound plausible but contain content that is factually incorrect. Many studies have investigated hallucination risk for LLMs (see Huang et al. [2023], Tonmoy et al. [2024], Shorinwa et al. [2024], Huang et al. [2024] for surveys of the literature). Among these, experiments have found model-specific hallucination rates as high as 4.1% [Hughes et al., 2023].

One approach to hallucination detection is human-in-the-loop, where a reviewer investigates LLM outputs for correctness. However, these systems are often implemented at a scale that is prohibitively large for exhaustive human review. In high-risk LLM applications, sampling-based human-in-the-loop, where a human manually reviews a random subset of LLM outputs, is unlikely to suffice. This motivates the need for an automated, response-level method to identify LLM outputs that are most likely to contain hallucinations. Such a technique enables filtering or targeted human review of responses with low confidence.

Computational hallucination detection methods typically involve comparing ground truth texts to generated content, comparing source content to generated content, or quantifying uncertainty. Assessments that compare ground truth texts to generated content are typically conducted pre-deployment in order to quantify hallucination risk of an LLM for a particular use case. While important, this collection of techniques does not lend itself well to real-time evaluation and monitoring of systems already deployed to production. In contrast, techniques that compare source content to generated content or quantifying uncertainty can compute response-level assessments and hence can be used for real-time monitoring of production-level applications.

*Correspondence to dylan.bouchard@cvshealth.com
Code available at: https://github.aetna.com/analytics-org/uqlm

Another difficulty with a unified approach to hallucination detection is the versatility of LLMs in handling a wide variety of tasks. For instance, source content comparison may be able to detect some hallucination in retrieval augmented generation (RAG) tasks, but not all use cases contain applicable source content. Another example is LLM-as-a-judge, for which effectiveness is likely to vary greatly across use cases and largely depend on prompt and the model of choice. While these approaches may be effective for specific use cases, they lack flexibility and customizability across a broad spectrum of LLM use cases.

To address these gaps, we propose a black-box, tunable-ensemble-approach to zero-resource hallucination detection. Our approach incorporates a diverse suite of components that can detect hallucinations across a wide variety of use cases. Each component of our ensemble outputs a confidence score ranging from 0 to 1. The ensemble output is a simple weighted average of these individual components, for which the weights can be tuned using a user-provided set of graded LLM responses. Further, our ensemble is extensible, meaning practitioners can expand to include new components as research on hallucination detection evolves.

To validate the effectiveness of our approach, we conduct an extensive set of experiments on various LLM question-answering benchmarks. We find that our ensemble consistently outperforms the individual components in hallucination detection. Further, we use our optimized weights from our experiments to offer task-specific guidelines on weights in the absence of tuning.

2 Related Work

In this section, we discuss a collection of studies that propose zero-resource hallucination detection techniques. These techniques compute response-level confidence or uncertainty scores without requiring access to source content, external resources (such as the internet), or ground truth texts. Importantly, we focus on black-box uncertainty quantification, where access to internal model states is not required.[2]

2.1 Semantic Similarity

Several studies have proposed methods for quantifying uncertainty using semantic similarity between an original LLM response and a set of candidate responses generated from the same prompt. These methods typically involve pairwise comparison using exact match comparisons, text similarity metrics, model-based text similarity, or natural language inference (NLI) models.

Cole et al. [2023] propose evaluating similarity between an original response and candidate responses using exact match-based metrics. In particular, they propose two metrics: repetition, which measures the proportion of candidate responses that match the original response, and diversity, which penalizes a higher proportion of unique responses in the set of candidates. While these metrics have the advantage of being intuitive, they have two notable disadvantages. First, minor phrasing differences are penalized, even if two answers have the same meaning. Second, these metrics are poorly suited for tasks that do not have a unique correct answer, such as summarization tasks.

Text similarity metrics assess response consistency in less stringent manner. Manakul et al. [2023] propose using n-gram-based evaluation to evaluate text similarity. Similar metrics such as ROUGE [Lin, 2004], BLEU [Papineni et al., 2002], and METEOR [Banerjee and Lavie, 2005] have also been proposed [Shorinwa et al., 2024]. These metrics, while widely adopted, have the disadvantage of being highly sensitive to token sequence orderings and often fail to detect semantic equivalence when two texts have different phrasing.

Sentence embedding-based metrics such as cosine similarity [Qurashi et al., 2020], computed using a sentence transformer such as Sentence-Bert [Reimers and Gurevych, 2019], have also been proposed [Shorinwa et al., 2024]. These metrics have the advantage of being able to detect semantic similarity in a pair of texts that are phrased differently. In a similar vein, Manakul et al. [2023] propose using BERTScore [Zhang et al., 2020], based on the maximum cosine similarity of contextualized word embeddings between token pairs in two candidate texts. BLEURTScore [Sellam et al., 2020] and BARTScore [Yuan et al., 2021] are notable alternatives.

[2]For white-box uncertainty quantification techniques, we refer the reader to Ling et al. [2024], Bakman et al. [2024], Fadeeva et al. [2024], Guerreiro et al. [2023], Zhang et al. [2023], Varshney et al. [2023], Luo et al. [2023], Ren et al. [2023], van der Poel et al. [2022], Wang et al. [2023].

Lastly, NLI models are another popular method for evaluating similarity between an original response and candidate responses. These models classify a pair of texts as either *entailment*, *contradiction*, or *neutral*. Several studies propose using NLI estimates of $1 - P(\text{contradiction})$ or $P(\text{entailment})$ between the original and a candidate responses to quantify uncertainty [Chen and Mueller, 2023, Lin et al., 2024]. Zhang et al. [2024] follow a similar approach but instead average across sentences and exclude $P(\text{neutral})$ from their calculations. Other studies compute semantic entropy using NLI-based clustering [Kuhn et al., 2023, Kossen et al., 2024, Farquhar et al., 2024]. Qiu and Miikkulainen [2024] estimate density in semantic space for candidate responses.

2.2 LLM-as-a-Judge

Another popular approach for uncertainty quantification is to use LLM-as-a-judge [Gu et al., 2025], using either the same LLM that was used for generating the original responses or a different LLM. The nature of the judgement may be a numerical score [Xiong et al., 2024a, Kadavath et al., 2022, Jones et al., 2024, Li et al., 2023, Zhu et al., 2023, Xiong et al., 2024b, Bai et al., 2023] or a binary yes/no [Shinn et al., 2023, Tian et al., 2024, Sun et al., 2024].

For uncertainty quantification, several studies concatenate a question-answer pair and ask an LLM to score or classify the answer's correctness. Chen and Mueller [2023] propose using an LLM for self-reflection certainty, where the same LLM is used to judge correctness of the response. Specifically, the LLM is asked to score the response as incorrect, uncertain, or correct, which map to scores of 0, 0.5, and 1, respectively. Similarly, Kadavath et al. [2022] ask the same LLM to state $P(Correct)$ given a question-answer concatenation. Xiong et al. [2024a] explore several variations of similar prompting strategies for LLM self-evaluation. More complex variations such as multiple choice question answering generation [Manakul et al., 2023], multi-LLM interaction [Cohen et al., 2023], and follow-up questions [Agrawal et al., 2024] have also been proposed.

2.3 Ensemble Approaches

Chen and Mueller [2023] propose a two-component algorithm for zero-resource hallucination known as BSDetector. The first component, known as observed consistency, computes a weighted average of two comparison scores between an original response and a set of candidate responses, one based on exact match, and another based on NLI-estimated contradiction probabilities. The second component is self-reflection certainty component, which uses the same LLM to judge correctness of the response. In their ensemble, response-level confidence scores are computed using a weighted average observed consistency and self-reflection certainty.

Fallah et al. [2024] leverage an ensemble of judges to evaluate uncertainty of LLM responses. Specifically, they propose three ensemble variations, each comprised of three LLM judges. The first two involve having two LLMs comment on a response's correctness and having of the LLMs form a judgment based on the comments from both. The third involves having two commentator LLMs with a third LLM forming the final judgment based on the comments of the first two. Similarly, Verga et al. [2024] propose using a Panel of LLM evaluators (PoLL) to assess LLM responses. Rather than using a single large LLM as a judge, their approach leverages a panel of smaller LLMs. Their experiments find that PoLL outperforms large LLM judges, having less intra-model bias in the judgments.

3 Hallucination Detection Methodology

3.1 Problem Statement

We aim to model the binary classification problem of whether an LLM response contains a hallucination, which we define as any content that is nonfactual. To this end, our goal is to construct an extensible ensemble classifier for black-box hallucination detection, comprised K binary classifiers. In our ensemble, each classifier maps an LLM response $y_i \in \mathcal{Y}$, generated from prompt x_i, to a 'confidence score' between 0 and 1, where $\mathcal{Y}$ is the set of possible LLM outputs. We denote hallucination classifier k as $\hat{s}_k : \mathcal{Y} \rightarrow [0, 1]$.

Several of our ensemble components exploit variation in LLM responses to the same prompt. For a given prompt x_i, this approach involves generating m responses $\tilde{\mathbf{y}}_i = \{\tilde{y}_{i,1}, ..., \tilde{y}_{i,m}\}$, using a non-zero temperature, from the same prompt and comparing these responses to the original response y_i. Hence, our ensemble is parameterized by $\theta = (\tilde{\mathbf{y}}_i, \mathbf{w})$, where $\mathbf{w}$ denote the ensemble weights. For original response y_i, we can write our ensemble classifier as follows:

$$\hat{s}(y_i; \tilde{\mathbf{y}}_i, \mathbf{w}) = \sum_{k=1}^{K} w_k \hat{s}_k(y_i; \tilde{\mathbf{y}}_i), \tag{1}$$

where $\mathbf{w} = (w_1, ..., w_K), \sum_{k=1}^{K} w_k = 1$, and $w_k \in [0, 1]$ for $k = 1, ..., K$. Note that although we write each classifier to be parameterized by the set of candidate responses, some of the classifiers depend only on the original response.

Given a classification threshold τ, we denote binary hallucination predictions from our ensemble as $\hat{h}(y; \mathbf{w}, \tau)$:

$$\hat{h}(y_i; \tilde{\mathbf{y}}_i, \mathbf{w}, \tau) = \mathbb{I}(\hat{s}(y_i; \tilde{\mathbf{y}}_i, \mathbf{w}) < \tau). \tag{2}$$

Note that $\hat{h}(\cdot) = 1$ implies a hallucination is predicted. We denote the corresponding ground truth value, indicating whether or not the original response y_i actually contains a hallucination, as $h(y_i)$, where h represents a process to 'grade' LLM responses.

$$h(y_i) = \begin{cases} 1 & y_i \text{ contains a hallucination} \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

The components of our ensemble are adapted from various techniques proposed in the literature, each of which outputs response-level confidence scores, ranging from 0 to 1, to be used for hallucination detection. Below, we provide details of components we include our ensemble, noting that practitioners may extend our ensemble to include other components as well.

3.2 Semantic Similarity Components

These components assess contradiction likelihood, semantic volatility, or text similarity across responses to the same prompt. We provide detailed descriptions of each below.

Exact Match Rate. For LLM tasks that have a unique, closed-form answer, exact match rate can be a useful hallucination detection approach. Under this approach, an indicator function is used to score pairwise comparisons between the original response and the candidate responses. Given an original response y_i and candidate responses $\tilde{\mathbf{y}}_i$, generated from prompt x_i, exact match rate (ERM) is computed as follows:

$$EMR(y_i; \tilde{\mathbf{y}}_i) = \frac{1}{m} \sum_{j=1}^{m} \mathbb{I}(y_i = \tilde{y}_{ij}). \tag{4}$$

Mean Pairwise Contradiction Score. Mean pairwise contradiction score (MPCS), is a similar, but less-stringent approach. MPCS, a component of the BSDetector approach proposed by Chen and Mueller [2023], also conducts pairwise comparison between the original response and each candidate response. In particular, a natural language inference (NLI) model is used to classify each pair (y_i, y_{ij}) as *entailment*, *neutral*, or *contradiction* and contradiction probabilities are saved. MPCS for original response y_i is computed as the average NLI-based non-contradiction probabilities across pairings with all candidate responses:

$$MPCS(y_i; \tilde{\mathbf{y}}_i) = \frac{1}{m} \sum_{j=1}^{m} (1 - p_j) \tag{5}$$

$$p_j = \frac{\eta(y_i, \tilde{y}_{ij}) + \eta(\tilde{y}_{ij}, y_i)}{2}. \tag{6}$$

Above, $\eta(y_i, y_{ij})$ denotes the contradiction probability of (y_i, y_{ij}) estimated by the NLI model. Following Chen and Mueller [2023] and Farquhar et al. [2024] we use `microsoft/deberta-large-mnli` for our NLI model.

Normalized Semantic Negentropy. Semantic entropy, proposed by Farquhar et al. [2024], exploits variation in multiple responses to compute a measure of response volatility. In contrast to the EMR and MPCS, semantic entropy does not distinguish between an original response and candidate responses. Instead, this approach computes a single metric value on a list of responses generated from the same prompt. Under this approach, responses are clustered using an NLI model based on mutual entailment. We consider the discrete version of semantic entropy (SE), where the final set of clusters is defined as follows:

$$SE(y_i; \tilde{\mathbf{y}}_i) = -\sum_{C \in \mathcal{C}} P(C|y_i, \tilde{\mathbf{y}}_i) \log P(C|y_i, \tilde{\mathbf{y}}_i), \tag{7}$$

where $P(C|y_i, \tilde{\mathbf{y}}_i)$ denotes the probability a randomly selected response $y \in \{y_i\} \cup \tilde{\mathbf{y}}_i$ belongs to cluster C, and $\mathcal{C}$ denotes the full set of clusters of $\{y_i\} \cup \tilde{\mathbf{y}}_i$.[3]

To ensure that we have a normalized confidence score with $[0, 1]$ support and with higher values corresponding to higher confidence, we implement the following normalization to arrive at *Normalized Semantic Negentropy* (NSN):

$$NSN(y_i; \tilde{\mathbf{y}}_i) = 1 - \frac{SE(y_i; \tilde{\mathbf{y}}_i)}{\log m}, \tag{8}$$

where $\log m$ is included to normalize the support.

BERT Score Another approach for measuring text similarity between two texts is BERTScore Zhang et al. [2020]. Let a tokenized text sequence be denoted as $\mathbf{t} = \{t_1, ...t_L\}$ and the corresponding contextualized word embeddings as $\mathbf{E} = \{\mathbf{e}_1, ..., \mathbf{e}_L\}$, where L is the number of tokens in the text. The BERTScore precision, recall, and F1-scores between two tokenized texts $\mathbf{t}, \mathbf{t}'$ are respectively defined as follows:

$$BertPrec(\mathbf{t}, \mathbf{t}') = \frac{1}{|\mathbf{t}|} \sum_{t \in \mathbf{t}} \max_{t' \in \mathbf{t}'} \mathbf{e} \cdot \mathbf{e}' \tag{9}$$

$$BertRec(\mathbf{t}, \mathbf{t}') = \frac{1}{|\mathbf{t}'|} \sum_{t' \in \mathbf{t}'} \max_{t \in \mathbf{t}} \mathbf{e} \cdot \mathbf{e}' \tag{10}$$

$$BertFScore(\mathbf{t}, \mathbf{t}') = 2 \frac{BertPrec(\mathbf{t}, \mathbf{t}') BertRec(\mathbf{t}, \mathbf{t}')}{BertPrec(\mathbf{t}, \mathbf{t}') + BertRec(\mathbf{t}, \mathbf{t}')}, \tag{11}$$

where e, e' respectively correspond to t, t'. We compute our BERTScore-based confidence scores as follows:

$$BertConfidence(y_i; \tilde{\mathbf{y}}_i) = \frac{1}{m} \sum_{j=1}^{m} BertFScore(y_i, \tilde{y}_{ij}), \tag{12}$$

i.e. the average BERTScore F1 across pairings of the original response with all candidate responses.

Mean Cosine Similarity Lastly, we include mean cosine similarity (MCS) to measure semantic similarity Shorinwa et al. [2024]. This component leverages a sentence transformer to map LLM outputs to an embedding space and measure similarity using those sentence embeddings. Let $V : \mathcal{Y} \to \mathbb{R}^d$ denote the sentence transformer, where d is the dimension of the embedding space. Confidence scores for this component are computed as the average cosine similarity across pairings of the original response with all candidate responses:

$$MCS(y_i; \tilde{\mathbf{y}}_i) = \frac{1}{m} \sum_{i=1}^{m} \frac{\mathbf{V}(y_i) \cdot \mathbf{V}(\tilde{y}_{ij})}{\|\mathbf{V}(y_i)\| \|\mathbf{V}(\tilde{y}_{ij})\|}. \tag{13}$$

[3] If token probabilities of the LLM responses are available, the values of $P(C|y_i, \tilde{\mathbf{y}}_i)$ can be instead estimated using mean token probability. However, unlike the discrete case, this version of semantic entropy is unbounded and hence does not lend itself well to normalization.

3.3 LLM-as-a-Judge Components

Below we outline two ensemble components that use LLM-as-a-Judge. We follow the approach proposed by Chen and Mueller [2023] in which an LLM is instructed to score a question-answer concatenation as either *incorrect*, *uncertain*, or *correct* using a carefully constructed prompt. These categories are respectively mapped to numerical scores of 0, 0.5, and 1. We denote the LLM-as-a-judge scorers as $J : \mathcal{Y} \to \{0, 0.5, 1\}$. Formally, we can write these scorer functions as follows:

$$J(y_i) = \begin{cases} 0 & \text{LLM states response is incorrect} \\ 0.5 & \text{LLM states that it is uncertain} \\ 1 & \text{LLM states response is correct.} \end{cases} \tag{14}$$

In a slight modification from the prompt used by [Chen and Mueller, 2023], we use the following prompt:

> Question: [question], Proposed Answer: [answer].
>
> Your task is to look at the question and answer provided and determine if the answer is correct. You are to respond with ONLY one of: "Correct", "Incorrect", or "I am not sure". YOUR ANSWER MUST ONLY CONTAIN ONE OF "Correct", "Incorrect", or "I am not sure". DO NOT ANSWER THE QUESTION AGAIN. ONLY DETERMINE IF THE ANSWER TO THE QUESTION IS "Correct", "Incorrect", or "I am not sure".

The capitalization and repeated instructions, inspired by Wang et al. [2024], are included to ensure the LLM correctly follows instructions. The instruction part of the prompt is also used as the LLM's system prompt for the generation of LLM judgments.

Self-Judge. Under the self-judge approach, the same LLM used to generate the original responses is used to score the responses with the aforementioned approach. Importantly, the query asking the LLM to judge the response is passed independently and does not include the chat history including the original prompt and response.

External Judge. A generalization of of the self-reflection certainty approach proposed by Chen and Mueller [2023], our external LLM-as-a-judge approach follows the same setup as the self-judge, but instead leverages a *different* LLM to classify the response as correct, incorrect, or uncertain. Note that multiple LLM-as-a-grader components can be included if multiple LLMs are available to the practitioner.

3.4 Ensemble Tuning

Lastly, we outline a method for tuning ensemble weights for improved hallucination detection accuracy. This approach allows for customizable component-importance that can be optimized for a specific use case. In practice, tuning the ensemble weights requires having a 'graded' set of n original LLM responses which indicate whether a hallucination is present in each response.[4] For a set of n prompts, we denote the vector of original responses as $\mathbf{y}$

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, \tag{15}$$

and candidate responses across all prompts with the matrix $\tilde{\mathbf{Y}}$:

$$\tilde{\mathbf{Y}} = \begin{pmatrix} \tilde{\mathbf{y}}_1 \\ \tilde{\mathbf{y}}_2 \\ \vdots \\ \tilde{\mathbf{y}}_n \end{pmatrix} = \begin{pmatrix} \tilde{y}_{11} & \tilde{y}_{12} & \cdots & \tilde{y}_{1m} \\ \tilde{y}_{21} & \tilde{y}_{22} & \cdots & \tilde{y}_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{y}_{n1} & \tilde{y}_{n2} & \cdots & \tilde{y}_{nm} \end{pmatrix}. \tag{16}$$

[4] Grading responses may be accomplished computationally for certain tasks, e.g. multiple choice questions. However, in many cases, this will require a human grader to manually evaluate the set of responses.

Analogously, we denote the vectors of ensemble confidence scores, binary hallucination predictions, and corresponding ground truth values respectively as

$$\hat{\mathbf{s}}(\mathbf{y}; \tilde{\mathbf{Y}}, \mathbf{w}) = \begin{pmatrix} \hat{s}(y_1; \tilde{\mathbf{y}}_1, \mathbf{w}) \\ \hat{s}(y_2; \tilde{\mathbf{y}}_2, \mathbf{w}) \\ \vdots \\ \hat{s}(y_n; \tilde{\mathbf{y}}_n, \mathbf{w}) \end{pmatrix}, \tag{17}$$

$$\hat{\mathbf{h}}(\mathbf{y}; \tilde{\mathbf{Y}}, \mathbf{w}, \tau) = \begin{pmatrix} \hat{h}(y_1; \tilde{\mathbf{y}}_1, \mathbf{w}, \tau) \\ \hat{h}(y_2; \tilde{\mathbf{y}}_2, \mathbf{w}, \tau) \\ \vdots \\ \hat{h}(y_n; \tilde{\mathbf{y}}_n, \mathbf{w}, \tau) \end{pmatrix}, \tag{18}$$

and

$$\mathbf{h}(\mathbf{y}) = \begin{pmatrix} h(y_1) \\ h(y_2) \\ \vdots \\ h(y_n) \end{pmatrix}. \tag{19}$$

Modeling this problem as binary classification enables us to tune the weights of our ensemble classifier using traditional classification objective functions. Following this approach, we consider two distinct strategies to tune ensemble weights $w_1, ..., w_K$: threshold-agnostic optimization and threshold-aware optimization.

Threshold-Agnostic Weights Optimization. Our first approach to ensemble tuning uses a threshold-agnostic objective function for tuning the ensemble weights. Given a set of n prompts, corresponding original LLM responses and candidate responses, the optimal set of weights, $\mathbf{w}^*$, is the solution to the following problem:

$$\mathbf{w}^* = \underset{\mathbf{w} \in \mathcal{W}}{\arg\max}\, \mathcal{S}(\hat{\mathbf{s}}(\mathbf{y}; \tilde{\mathbf{Y}}, \mathbf{w}), \mathbf{h}(\mathbf{y})), \tag{20}$$

where

$$\mathcal{W} = \{(w_1, ..., w_K) : \sum_{k=1}^{K} w_k = 1, w_k \in [0, 1], k = 1, ..., K\} \tag{21}$$

is the support of the ensemble weights and $\mathcal{S}$ is a threshold-agnostic classification performance metric, such as area under the receiver-operator curve (AUROC).

After optimizing the weights, we subsequently tune the threshold using a threshold-dependent objective function. Hence, the optimal threshold, τ^*, is the solution to the following optimization problem:

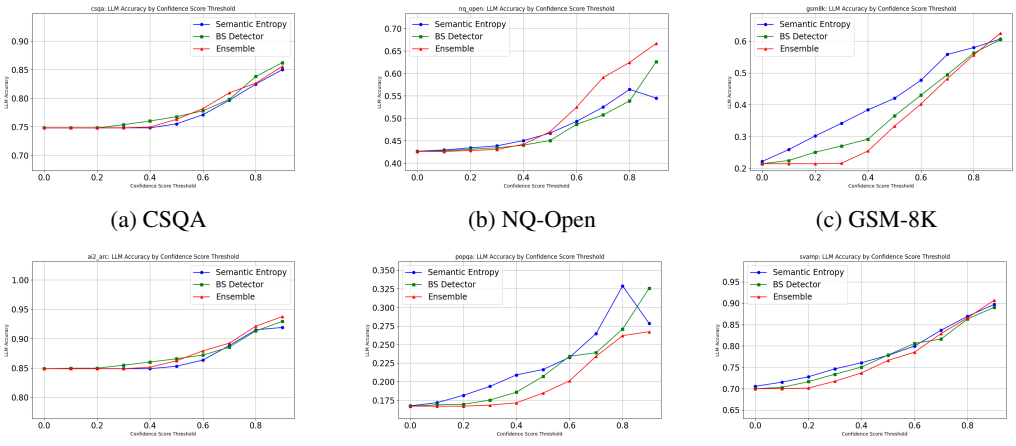
$$\tau^* = \underset{\tau \in (0,1)}{\arg\max}\, \mathcal{B}(\hat{\mathbf{h}}(\mathbf{y}; \tilde{\mathbf{Y}}, \mathbf{w}^*, \tau), \mathbf{h}(\mathbf{y})), \tag{22}$$

where $\mathcal{B}$ is a threshold-dependent classification performance metric, such as F1-score.

Threshold-Aware Weights Optimization. Alternatively, practitioners may wish jointly optimize ensemble weights and classification threshold using the same objective. This type of optimization relies on a threshold-dependent objective. We can write this optimization problem as follows:

$$\mathbf{w}^*, \tau^* = \underset{\mathbf{w} \in \mathcal{W}, \tau \in (0,1)}{\arg\max}\, \mathcal{B}(\hat{\mathbf{h}}(\mathbf{y}; \tilde{\mathbf{Y}}, \mathbf{w}, \tau), \mathbf{h}(\mathbf{y})), \tag{23}$$

where $\mathcal{B}$, $\hat{\mathbf{h}}$, $\mathbf{h}$, and $\mathcal{W}$ follow the same definitions as above.



(a) CSQA (b) NQ-Open (c) GSM-8K

(d) AI2-ARC (e) POPQA (f) SVAMP

Figure 1: Accuracy-at-τ Results: Ensemble vs. BSDetector vs. Semantic Entropy

4 Experiments

4.1 Experiment Setup.

We conduct a series of experiments to assess the hallucination detection accuracy of our approach. To accomplish this, we leverage a set of publicly available benchmark datasets that contain questions and answers. To ensure that our answer format has sufficient variation, we use two benchmarks with numerical answers (*GSM8k* [Cobbe et al., 2021] and *SVAMP* [Patel et al., 2021]), two with multiple choice answers (*CSQA* [Talmor et al., 2022] and *ARC* [Clark et al., 2018]), and two with open-ended text answers (*PopQA* [Mallen et al., 2023] and *NQ-Open* [Lee et al., 2019]).

For each benchmark, we draw samples of 1000 questions and generate original and candidate responses using `gpt-3.5-16k-turbo`. Using the candidate responses, we compute exact match, mean pairwise contradiction score, normalized semantic negentropy, BertConfScore, and mean cosine similarity scores for each original response. We use the same LLM for a self-judge component and another LLM, `gemini-1.0-pro` as an external judge. We evaluate the performance of our ensemble-based confidence scores in hallucination detection for each of the benchmarks using various metrics. For baseline comparison, we also evaluate these metrics for BSDetector [Chen and Mueller, 2023] and Normalized Semantic Negentropy (based on Farquhar et al. [2024]).

4.2 Evaluation Approach.

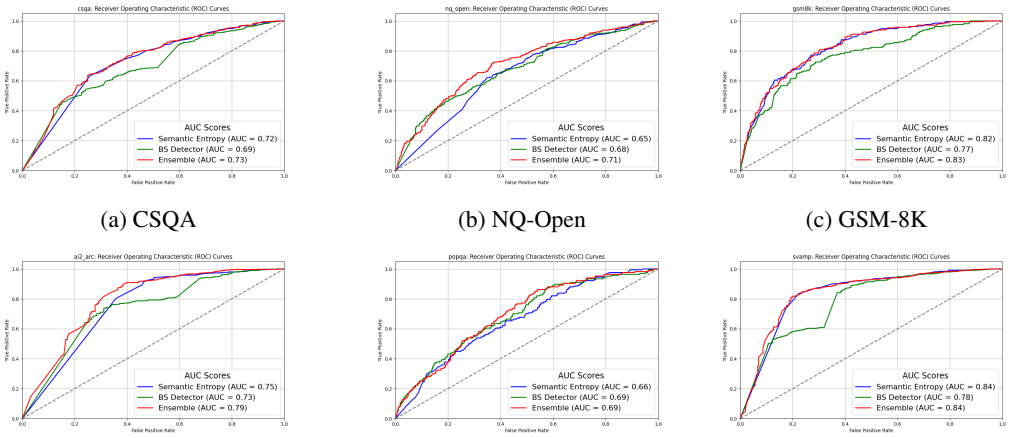
Accuracy-at-τ To assess the reliability of the confidence scores, we first compute model accuracy on the subset of model responses having confidence scores exceeding a specified threshold τ. Since the model accuracy depends on the choice of the threshold τ, we repeat the calculation for multiple values of the confidence score threshold ($\tau \in [0, 0.1, \ldots, 0.9]$). Following our earlier notation, we define accuracy-at-τ as follows:

$$acc(\mathbf{y}; \tilde{\mathbf{Y}}, \tau) = \frac{\sum_{i=1}^{N} \mathbb{I}(\hat{s}(y_i; \tilde{\mathbf{y}}_i, \mathbf{w}^*) \geq \tau) \cdot (1 - h(y_i))}{\sum_{i=1}^{N} \mathbb{I}(\hat{s}(y_i; \tilde{\mathbf{y}}_i, \mathbf{w}^*) \geq \tau)}.$$

Note that accuracy at $\tau = 0$ uses the full sample without score-based filtering.

Threshold-Agnostic Evaluation To further assess the performance of ensemble classifier, we evaluate the performance of the confidence scores in a threshold-agnostic fashion. Specifically, we focus on AUROC and compare the performance of our ensemble to the two baseline approaches. Under this setting, we use the AUROC-optimized ensemble weights.

Threshold-Optimized Evaluation Lastly, we compare our ensemble classifier to the two baselines using threshold-dependent classification performance metrics: precision, recall, and F1-score. Under this setting, we use jointly-optimized weights and thresholds, using F1-score as the objective function.



(a) CSQA (b) NQ-Open (c) GSM-8K

(d) AI2-ARC (e) POPQA (f) SVAMP

Figure 2: ROC Curves: Ensemble vs. BSDetector vs. Semantic Entropy

Table 1: Precision, Recall, and F1-Scores: Ensemble vs. BSDetector vs. Semantic Entropy

		MC Questions		*Open-Ended Questions*		*Math Questions*	
		CSQA	AI2-ARC	POPQA	NQ-Open	SVAMP	GSM8K
Entropy	Precision	0.765	0.906	0.259	0.514	0.857	0.558
	Recall	0.987	0.948	0.539	0.796	0.900	0.605
	F1-Score	0.862	0.924	0.350	0.624	0.878	0.580
BS Detector	Precision	0.757	0.865	0.283	0.482	0.818	0.502
	Recall	0.993	0.980	0.527	0.904	0.911	0.558
	F1-Score	0.860	0.924	0.368	0.629	0.862	0.529
Ensemble	Precision	0.758	0.891	0.281	0.538	0.864	0.517
	Recall	0.997	0.980	0.605	0.824	0.887	0.628
	F1-Score	0.862	0.933	0.384	0.651	0.875	0.567

4.3 Experiment Results.

We first consider our accuracy-at-τ results, displayed in Figure 1. For all three techniques, LLM accuracy increases with the threshold approximately monotonically. We see particularly large improvements in LLM accuracy with higher confidence scores for the NQ-Open and GSM8K benchmarks, with accuracy rates improving from 43% to 67% and 22% to 62%, respectively, when comparing the entire sample to responses in the top decile of confidence scores. Improvements in accuracy-at-τ are similar for all three techniques, although improvements were most pronounced for NQ-Open with the ensemble.

Next, we compare the ROC curves for the three techniques across the six benchmarks, displayed in Figure 2. For all six benchmarks, the area under the ROC curve is highest using our ensemble technique. This result is unsurprising, given that our ensemble weights in this setting are optimized on AUROC. However, all three techniques demonstrate impressive AUROC values, ranging from .65 to .84.

Lastly, we compute the values of precision, recall, and F1-score for all six benchmarks and compare across our three techniques. The full suite of metric values is displayed in Table 1. To give a more balanced view of performance, we pay particular attention to F1-scores in our comparisons. All three techniques have very strong performance for the CSQA, AI2-ARC, and SVAMP benchmarks, with F1-scores ranging from .86 to .93. Performance on the POPQA is notably lower, with F1-scores ranging from .35 to .38. Comparing across techniques, we see our ensemble outperforms the other techniques on multiple choice and open-ended benchmarks, while semantic entropy has the best performance for the math benchmarks. Overall, all three techniques have reasonably similar performance.

5 Discussion

In practice, users of our ensemble approach may wish to use confidence scores for various purposes. First, practitioners can use our confidence scores for response filtering, where responses with low confidence are blocked. Our experimental evaluations of accuracy-at-τ demonstrate the efficacy of this approach, illustrating the strong correlation between confidence scores and LLM accuracy. Second, practitioners may also consider implementing 'targeted' human-in-the-loop practices. Under this approach, manual review would happen for any response that yields a low confidence score. Again, our accuracy-at-τ evaluations indicate that this approach will offer far more efficient review compared to arbitrary human-in-the-loop, where random samples of responses are manually reviewed. Additionally, this approach helps address scenarios where exhaustive human-in-the-loop is infeasible due to the scale of responses being generated.

6 Conclusions

In this paper, we introduce a novel, ensemble-based approach for black-box, zero-resource hallucination detection. To achieve this, we model hallucination detection as a binary classification problem, and use a tunable ensemble of binary classifiers to construct optimized response-level confidence scores. Our experiments on various benchmarks demonstrate favorable performance compared to current state-of-the-art approaches. For scenarios in which practitioners are not able to tune the ensemble weights, due to a sample of 'graded' responses being unavailable, we recommend weights for use cases with closed-form answers and use cases with open-ended answers.

Our approach has many advantages over existing techniques. First, our ensemble components are all black-box scorers, meaning they do not require access to any internal model states, only model outputs. This is particularly useful for practitioners using APIs that do not enable access to such information. Second, our ensemble is highly flexible in that practitioners can select any subset of our proposed components or include additional components if they wish. Third, our approach is versatile in terms of the use cases to which it applies. While some of our components are well-suited for use cases closed-form, unique answers (e.g. multiple choice or math questions), other components handle scenarios where the correct answer may be articulated in a variety of ways.

References

A. Agrawal, M. Suzgun, L. Mackey, and A. T. Kalai. Do language models know when they're hallucinating references?, 2024. URL https://arxiv.org/abs/2305.18248.

Y. Bai, J. Ying, Y. Cao, X. Lv, Y. He, X. Wang, J. Yu, K. Zeng, Y. Xiao, H. Lyu, J. Zhang, J. Li, and L. Hou. Benchmarking foundation models with language-model-as-an-examiner, 2023. URL https://arxiv.org/abs/2306.04181.

Y. F. Bakman, D. N. Yaldiz, B. Buyukates, C. Tao, D. Dimitriadis, and S. Avestimehr. Mars: Meaning-aware response scoring for uncertainty estimation in generative llms, 2024. URL https://arxiv.org/abs/2402.11756.

S. Banerjee and A. Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In J. Goldstein, A. Lavie, C.-Y. Lin, and C. Voss, editors, *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. URL https://aclanthology.org/W05-0909/.

J. Chen and J. Mueller. Quantifying uncertainty in answers from any language model and enhancing their trustworthiness, 2023. URL https://arxiv.org/abs/2308.16175.

P. Clark, I. Cowhey, O. Etzioni, T. Khot, A. Sabharwal, C. Schoenick, and O. Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge, 2018. URL https://arxiv.org/abs/1803.05457.

K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, C. Hesse, and J. Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

R. Cohen, M. Hamri, M. Geva, and A. Globerson. Lm vs lm: Detecting factual errors via cross examination, 2023. URL https://arxiv.org/abs/2305.13281.

J. R. Cole, M. J. Q. Zhang, D. Gillick, J. M. Eisenschlos, B. Dhingra, and J. Eisenstein. Selectively answering ambiguous questions, 2023. URL https://arxiv.org/abs/2305.14613.

E. Fadeeva, A. Rubashevskii, A. Shelmanov, S. Petrakov, H. Li, H. Mubarak, E. Tsymbalov, G. Kuzmin, A. Panchenko, T. Baldwin, P. Nakov, and M. Panov. Fact-checking the output of large language models via token-level uncertainty quantification, 2024. URL https://arxiv.org/abs/2403.04696.

P. Fallah, S. Gooran, M. Jafarinasab, P. Sadeghi, R. Farnia, A. Tarabkhah, Z. S. Taghavi, and H. Sameti. SLPL SHROOM at SemEval2024 task 06 : A comprehensive study on models ability to detect hallucination. In A. K. Ojha, A. S. Doğruöz, H. Tayyar Madabushi, G. Da San Martino, S. Rosenthal, and A. Rosá, editors, *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1148–1154, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.semeval-1.167. URL https://aclanthology.org/2024.semeval-1.167/.

S. Farquhar, J. Kossen, L. Kuhn, and Y. Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630, Jun 2024. ISSN 1476-4687. doi: 10.1038/s41586-024-07421-0. URL https://doi.org/10.1038/s41586-024-07421-0.

J. Gu, X. Jiang, Z. Shi, H. Tan, X. Zhai, C. Xu, W. Li, Y. Shen, S. Ma, H. Liu, S. Wang, K. Zhang, Y. Wang, W. Gao, L. Ni, and J. Guo. A survey on llm-as-a-judge, 2025. URL https://arxiv.org/abs/2411.15594.

N. M. Guerreiro, E. Voita, and A. F. T. Martins. Looking for a needle in a haystack: A comprehensive study of hallucinations in neural machine translation, 2023. URL https://arxiv.org/abs/2208.05309.

H.-Y. Huang, Y. Yang, Z. Zhang, S. Lee, and Y. Wu. A survey of uncertainty estimation in llms: Theory meets practice, 2024. URL https://arxiv.org/abs/2410.15326.

L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin, and T. Liu. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions, 2023. URL https://arxiv.org/abs/2311.05232.

S. Hughes, M. Bae, and M. Li. Vectara Hallucination Leaderboard, Nov. 2023. URL https://github.com/vectara/hallucination-leaderboard.

J. Jones, L. Mo, E. Fosler-Lussier, and H. Sun. A multi-aspect framework for counter narrative evaluation using large language models, 2024. URL https://arxiv.org/abs/2402.11676.

S. Kadavath, T. Conerly, A. Askell, T. Henighan, D. Drain, E. Perez, N. Schiefer, Z. Hatfield-Dodds, N. DasSarma, E. Tran-Johnson, S. Johnston, S. El-Showk, A. Jones, N. Elhage, T. Hume, A. Chen, Y. Bai, S. Bowman, S. Fort, D. Ganguli, D. Hernandez, J. Jacobson, J. Kernion, S. Kravec, L. Lovitt, K. Ndousse, C. Olsson, S. Ringer, D. Amodei, T. Brown, J. Clark, N. Joseph, B. Mann, S. McCandlish, C. Olah, and J. Kaplan. Language models (mostly) know what they know, 2022. URL https://arxiv.org/abs/2207.05221.

J. Kossen, J. Han, M. Razzak, L. Schut, S. Malik, and Y. Gal. Semantic entropy probes: Robust and cheap hallucination detection in llms, 2024. URL https://arxiv.org/abs/2406.15927.

L. Kuhn, Y. Gal, and S. Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation, 2023. URL https://arxiv.org/abs/2302.09664.

K. Lee, M.-W. Chang, and K. Toutanova. Latent retrieval for weakly supervised open domain question answering, 2019. URL https://arxiv.org/abs/1906.00300.

J. Li, S. Sun, W. Yuan, R.-Z. Fan, H. Zhao, and P. Liu. Generative judge for evaluating alignment, 2023. URL https://arxiv.org/abs/2310.05470.

C.-Y. Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013/.

Z. Lin, S. Trivedi, and J. Sun. Generating with confidence: Uncertainty quantification for black-box large language models, 2024. URL https://arxiv.org/abs/2305.19187.

C. Ling, X. Zhao, X. Zhang, W. Cheng, Y. Liu, Y. Sun, M. Oishi, T. Osaki, K. Matsuda, J. Ji, G. Bai, L. Zhao, and H. Chen. Uncertainty quantification for in-context learning of large language models, 2024. URL https://arxiv.org/abs/2402.10189.

J. Luo, C. Xiao, and F. Ma. Zero-resource hallucination prevention for large language models, 2023. URL https://arxiv.org/abs/2309.02654.

A. Mallen, A. Asai, V. Zhong, R. Das, D. Khashabi, and H. Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In A. Rogers, J. Boyd-Graber, and N. Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.546. URL https://aclanthology.org/2023.acl-long.546/.

P. Manakul, A. Liusie, and M. J. F. Gales. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models, 2023. URL https://arxiv.org/abs/2303.08896.

K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://doi.org/10.3115/1073083.1073135.

A. Patel, S. Bhattamishra, and N. Goyal. Are nlp models really able to solve simple math word problems?, 2021. URL https://arxiv.org/abs/2103.07191.

X. Qiu and R. Miikkulainen. Semantic density: Uncertainty quantification for large language models through confidence measurement in semantic space, 2024. URL https://arxiv.org/abs/2405.13845.

A. W. Qurashi, V. Holmes, and A. P. Johnson. Document processing: Methods for semantic text similarity analysis. In *2020 International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*, pages 1–6, 2020. doi: 10.1109/INISTA49547.2020.9194665.

N. Reimers and I. Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks, 2019. URL https://arxiv.org/abs/1908.10084.

J. Ren, Y. Zhao, T. Vu, P. J. Liu, and B. Lakshminarayanan. Self-evaluation improves selective generation in large language models, 2023. URL https://arxiv.org/abs/2312.09300.

T. Sellam, D. Das, and A. P. Parikh. Bleurt: Learning robust metrics for text generation, 2020. URL https://arxiv.org/abs/2004.04696.

N. Shinn, F. Cassano, E. Berman, A. Gopinath, K. Narasimhan, and S. Yao. Reflexion: Language agents with verbal reinforcement learning, 2023. URL https://arxiv.org/abs/2303.11366.

O. Shorinwa, Z. Mei, J. Lidard, A. Z. Ren, and A. Majumdar. A survey on uncertainty quantification of large language models: Taxonomy, open research challenges, and future directions, 2024. URL https://arxiv.org/abs/2412.05563.

J. Sun, C. Xu, L. Tang, S. Wang, C. Lin, Y. Gong, L. M. Ni, H.-Y. Shum, and J. Guo. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph, 2024. URL https://arxiv.org/abs/2307.07697.

A. Talmor, O. Yoran, R. L. Bras, C. Bhagavatula, Y. Goldberg, Y. Choi, and J. Berant. Commonsenseqa 2.0: Exposing the limits of ai through gamification, 2022. URL https://arxiv.org/abs/2201.05320.

Y. Tian, A. Ravichander, L. Qin, R. L. Bras, R. Marjieh, N. Peng, Y. Choi, T. L. Griffiths, and F. Brahman. Macgyver: Are large language models creative problem solvers?, 2024. URL https://arxiv.org/abs/2311.09682.

S. M. T. I. Tonmoy, S. M. M. Zaman, V. Jain, A. Rani, V. Rawte, A. Chadha, and A. Das. A comprehensive survey of hallucination mitigation techniques in large language models, 2024. URL https://arxiv.org/abs/2401.01313.

L. van der Poel, R. Cotterell, and C. Meister. Mutual information alleviates hallucinations in abstractive summarization, 2022. URL https://arxiv.org/abs/2210.13210.

N. Varshney, W. Yao, H. Zhang, J. Chen, and D. Yu. A stitch in time saves nine: Detecting and mitigating hallucinations of llms by validating low-confidence generation, 2023. URL https://arxiv.org/abs/2307.03987.

P. Verga, S. Hofstatter, S. Althammer, Y. Su, A. Piktus, A. Arkhangorodsky, M. Xu, N. White, and P. Lewis. Replacing judges with juries: Evaluating llm generations with a panel of diverse models, 2024. URL https://arxiv.org/abs/2404.18796.

B. Wang, W. Chen, H. Pei, C. Xie, M. Kang, C. Zhang, C. Xu, Z. Xiong, R. Dutta, R. Schaeffer, S. T. Truong, S. Arora, M. Mazeika, D. Hendrycks, Z. Lin, Y. Cheng, S. Koyejo, D. Song, and B. Li. Decodingtrust: A comprehensive assessment of trustworthiness in gpt models, 2024. URL https://arxiv.org/abs/2306.11698.

X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery, and D. Zhou. Self-consistency improves chain of thought reasoning in language models, 2023. URL https://arxiv.org/abs/2203.11171.

M. Xiong, Z. Hu, X. Lu, Y. Li, J. Fu, J. He, and B. Hooi. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms, 2024a. URL https://arxiv.org/abs/2306.13063.

T. Xiong, X. Wang, D. Guo, Q. Ye, H. Fan, Q. Gu, H. Huang, and C. Li. Llava-critic: Learning to evaluate multimodal models, 2024b. URL https://arxiv.org/abs/2410.02712.

W. Yuan, G. Neubig, and P. Liu. Bartscore: Evaluating generated text as text generation, 2021. URL https://arxiv.org/abs/2106.11520.

C. Zhang, F. Liu, M. Basaldella, and N. Collier. Luq: Long-text uncertainty quantification for llms, 2024. URL https://arxiv.org/abs/2403.20279.

T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi. Bertscore: Evaluating text generation with bert, 2020. URL https://arxiv.org/abs/1904.09675.

T. Zhang, L. Qiu, Q. Guo, C. Deng, Y. Zhang, Z. Zhang, C. Zhou, X. Wang, and L. Fu. Enhancing uncertainty-based hallucination detection with stronger focus, 2023. URL https://arxiv.org/abs/2311.13230.

L. Zhu, X. Wang, and X. Wang. Judgelm: Fine-tuned large language models are scalable judges, 2023. URL https://arxiv.org/abs/2310.17631.