

Learned Hallucination Detection in Black-Box LLMs using Token-level Entropy Production Rate

Charles Moslonka

Artefact Research Center

Paris, France

MICS, CentraleSupélec, Université Paris-Saclay
Gif-sur-Yvette, France
{name.surname}@artefact.com

Arthur Garnier

Ardian

Paris, France

{name.surname}@ardian.com

Hicham Randrianarivo

Artefact Research Center

Paris, France

MICS, CentraleSupélec, Université Paris-Saclay
Gif-sur-Yvette, France
{name.surname}@artefact.com

Emmanuel Malherbe

Artefact Research Center

Paris, France

{name.surname}@artefact.com

Abstract

Hallucinations in Large Language Model (LLM) outputs for Question Answering (QA) tasks critically undermine their real-world reliability. This paper introduces an applied methodology for robust, one-shot hallucination detection, specifically designed for scenarios with limited data access, such as interacting with black-box LLM APIs that typically expose only a few top candidate log-probabilities per token. Our approach derives uncertainty indicators directly from these readily available log-probabilities generated during non-greedy decoding. We first derive an “Entropy Production Rate” (EPR) metric that offers baseline performance, later augmented with supervised learning. Our learned model uses features representing the entropic contributions of the accessible top-ranked tokens within a single generated sequence, requiring no multiple query re-runs. Evaluated across diverse QA datasets and multiple LLMs, this estimator significantly improves hallucination detection over using EPR alone. Crucially, high performance is demonstrated using only the typically small set of available log-probabilities (e.g., top <10 per token), confirming its practical efficiency and suitability for these API-constrained deployments. This work provides a readily deployable technique to enhance the trustworthiness of LLM responses from a single generation pass in QA and Retrieval-Augmented Generation (RAG) systems, with its utility further demonstrated in a finance framework analyzing responses to queries on annual reports from an industrial dataset.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities across a wide range of tasks [1–3], heralding transformative potential in numerous domains. However, a significant and persistent challenge hindering their widespread adoption, particularly in critical industrial contexts, is their propensity to generate hallucinations. These hallucinations [4, 5], defined as LLM-generated content that is factually incorrect, nonsensical, or unfaithful to a provided source despite often appearing plausible and coherent [6, 7], pose a significant barrier to reliable deployment. These can manifest as fact-conflicting, input-conflicting, or context-conflicting statements [6], and their deceptive plausibility necessitates robust automated detection mechanisms [8]. The impact of hallucinations can

be severe, particularly in safety-critical applications like medicine or infrastructure engineering [9, 10], leading to the propagation of misinformation and erosion of user trust. Therefore, detecting and quantifying the uncertainty associated with LLM outputs is paramount to foster trust and enable responsible deployment.

The practical deployment of Uncertainty Quantification (UQ) [11] and hallucination detection methods is often constrained by limited access to LLM internals, particularly when interacting with proprietary models through APIs that may only expose a small number of top- K log-probabilities per token (where K might be around 15¹). This “black-box” setting restricts the applicability of techniques requiring access to full logits, hidden states, or extensive architectural manipulation [12]. Moreover, many real-world applications require “one-shot” (or “single-round” [13]) detection capabilities—the ability to assess the reliability of a single generated sequence without the need for multiple, often costly, model inferences for the same input query to measure output variability.

In this paper, we introduce a UQ methodology rooted in information theory, tailored for these black-box, one-shot scenarios. Our approach leverages the accessible log-probabilities to derive entropic measures of model hesitation during the generation process. We define and evaluate a metric termed the “Entropy Production Rate” (EPR) of a generated sequence. We show that EPR, calculated as the average entropy of the token probability distributions across the entire sequence (i.e., the sum of all per-token entropies divided by the sequence length), serves as an initial, unsupervised, faithful estimator of the degree of hesitation exhibited by the model during inference. While the EPR provides a valuable baseline, this work demonstrates that its indicative power can be significantly enhanced. Our second contribution is the development of a supervised learning model that utilizes more granular entropic features. By training a model on the entropic contributions of individual top-ranked tokens across the sequence, we construct a learned estimator that more accurately distinguishes between faithful and hallucinatory responses. This estimator can also be used to better highlight high-uncertainty tokens in a generated sequence.

This paper details this entropic framework, the theoretical and practical underpinnings considering the limited logprob access, and

¹The maximum value being $K = 20$ for OpenAI API at the time of writing this paper

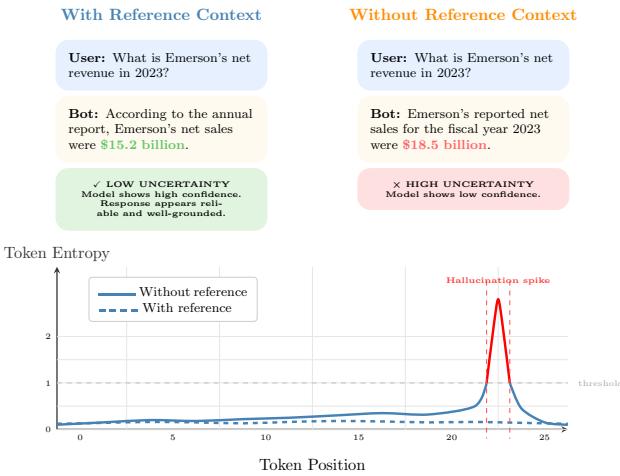


Figure 1: Illustration of token-level uncertainty detection in a financial RAG pipeline.

its empirical validation, showcasing its potential as a practical tool for improving the reliability of LLM systems, with the example of a RAG in a financial scenario.

2 Related Works

The increasing capabilities of Large Language Models (LLMs) are accompanied by significant challenges regarding their reliability, particularly in quantifying output uncertainty and detecting hallucinations. This section reviews prior work on these issues, emphasizing entropy-based methodologies relevant to constrained operational scenarios.

UQ in Large Language Models. Uncertainty Quantification (UQ) is crucial for establishing trust in LLMs, especially in high-stakes domains [11, 14]. LLM uncertainty arises from traditional aleatoric (data-inherent) and epistemic (model-based) sources [15], and additionally from unique generative characteristics like divergent multi-step reasoning paths and decoding stochasticity [13]. Traditional neural network UQ approaches, such as training modifications for aleatoric [16] or dropout-based methods for epistemic uncertainty [17, 18], are often ill-suited or prohibitively costly for LLMs due to training expense or architectural differences. The primary goal of UQ in LLMs is to enable them to signal their confidence—to “know what they do not know” [19, 20]—which is vital for risk mitigation and safe AI-assisted decision-making. Developing UQ techniques tailored to LLM intricacies remains an active research area.

Entropy-Based Approaches for UQ and Hallucination Detection

Information-theoretic entropy has emerged as a cornerstone for assessing LLM uncertainty and hallucinations, based on the premise that higher output distribution entropy often correlates with a greater likelihood of error or fabrication [21].

Varieties of Entropy Measures. Researchers have employed diverse entropy measures to capture different facets of LLM uncertainty [22]. Predictive entropy [23] assesses overall output distribution uncertainty, often applied in recommendation systems. Semantic Entropy (SE) [24] quantifies uncertainty over semantic meaning by clustering multiple generated responses, with Discrete Semantic Entropy (DSE) [25, 26] using frequency-based cluster probabilities. Answer Entropy [27], common in QA, measures variability in complete answer strings from multiple trials, sometimes enhanced with noise injection. Particularly relevant to our work, token-level entropy [19] measures per-step uncertainty from the next-token probability distribution and has been used to predict errors in QA and mathematical reasoning where errors may coincide with high-entropy tokens [12, 28].

Entropy-Driven Detection Techniques and Challenges. Direct application of entropy scores, often via thresholding, is a common approach for hallucination detection or calibration [12, 19, 24]. More sophisticated techniques include Semantic Entropy Probes (SEPs) [29]—learned models predicting semantic entropy from single-pass hidden states for efficiency—or Shifting Attention to Relevance (SAR) [30], which weights tokens by relevance in its computation. However, entropy-based methods face challenges like “high-certainty hallucinations” [31], where LLMs produce low-entropy incorrect outputs, necessitating strategies beyond simple thresholds. Furthermore, the computational cost of multi-sample methods (e.g., for Answer or Semantic Entropy) can be prohibitive, highlighting the need for efficient single-pass techniques. Approaches like SEPs [29] and Logit-induced Token Uncertainty (LogTokU) [32] aim for such single-pass efficiency, with LogTokU leveraging raw logits to potentially preserve more evidence strength.

3 Theoretical Framework for Entropic Uncertainty

Our approach to uncertainty in LLM generations is rooted in information theory [33, 34]. The core idea is to derive an “entropic score” that reflects the model’s hesitation at each token generation step. Unlike methods requiring multiple model runs to assess output variability, our methodology is designed for “one-shot” analysis, leveraging the probabilistic information available from a single generated sequence.

For a standard classification task with N_c classes, given an input q , a model outputs probabilities $p_i = p(\text{class}_i|q)$ for each class $i \in \{1, \dots, N_c\}$. The uncertainty can be quantified by the Shannon entropy (measured in bits) of this distribution:

$$H(q) = - \sum_{i=1}^{N_c} p_i \log_2(p_i) \geq 0. \quad (1)$$

This entropy $H(q)$ is minimized (zero) when the distribution is sharply peaked (high certainty) and maximized at:

$$H_{\max}(q) = \log_2(N_c) \quad (2)$$

for a uniform distribution (maximum uncertainty).

Extending this to LLM sequence generation, let $V = \{v_i\}_{i=1}^{|V|}$ denote the model's vocabulary, with size² $|V|$. For a given input query q , the LLM generates a sequence of tokens $\mathcal{T}_q = \{t_1, t_2, \dots, t_{L_q}\}$, where L_q is the length of the sequence. At each generation step j , the LLM internally computes a probability distribution $\{p(v_i|q, t_{<j})\}_{i=1}^{|V|}$ over the entire vocabulary for the next token t_j , conditioned on the preceding tokens $t_{<j} := \{t_1, \dots, t_{j-1}\}$ and the input q . The complete Shannon entropy for the j -th token's distribution is:

$$H(q, t_{<j}) = - \sum_{i=1}^{|V|} p(v_i|q, t_{<j}) \log_2 (p(v_i|q, t_{<j})). \quad (3)$$

Note that this quantity does not depend on the predicted token t_j . Moreover, in typical “black-box” scenarios involving proprietary LLMs accessed via APIs, only the log-probabilities (or probabilities) for a small number, denoted by K , of top-ranked candidate tokens are exposed (e.g., $K \leq 20$). Thus, we can only compute an estimated per-token entropy based on these top K probabilities. We define $r : \{1, \dots, |V|\} \rightarrow \{1, \dots, |V|\}$ the ranking operator on the vocabulary indices, meaning $r(k)$ is the index i of the k -th token in decreasing order of $p(v_i|q, t_{<j})$. The estimator can then be written as:

$$\tilde{H}_K(q, t_{<j}) = - \sum_{k=1}^K p_{r(k),j} \log_2 (p_{r(k),j}), \quad (4)$$

where $p_{r(k),j} := p(v_{r(k)}|q, t_{<j})$ is the probability of the k -th ranked token at generation step j .

3.1 Sufficiency of Top- K Log-Probabilities for Entropy Estimation

A critical question is whether $\tilde{H}_K(q, t_{<j})$, calculated from only the top K probabilities, serves as a reliable proxy for the true per-token entropy $H(q, t_{<j})$. The discrepancy arises from the missing tail of the distribution, $\Delta H_K(q, t_{<j}) = H(q, t_{<j}) - \tilde{H}_K(q, t_{<j})$, given by:

$$\Delta H_K(q, t_{<j}) = - \sum_{k=K+1}^{|V|} p_{r(k),j} \log_2 (p_{r(k),j}). \quad (5)$$

We can establish an upper bound, $\Delta H_{K,\max}(q, t_{<j})$, for this missing entropy. This bound corresponds to the scenario where the remaining probability mass $Q_K(q, t_{<j}) = 1 - \sum_{k=1}^K p_{r(k),j}$ is distributed uniformly among the remaining tokens considered to form the tail of the distribution. The maximum missing entropy from this uniform tail is given by:

$$\begin{aligned} \Delta H_{K,\max}(q, t_{<j}) &= -(|V| - K) \cdot \left(\frac{Q_K(q, t_{<j})}{|V| - K} \log_2 \frac{Q_K(q, t_{<j})}{|V| - K} \right) \\ &= -Q_K(q, t_{<j}) \log_2 \left(\frac{Q_K(q, t_{<j})}{|V| - K} \right) \\ &= h(Q_K(q, t_{<j})) + Q_K(q, t_{<j}) \log_2 (|V| - K), \end{aligned} \quad (6)$$

where $h(x) = -x \log_2 x$. The effective number of tokens in the tail, N_{tail} , depends on the specific assumptions:

- (1) **Full Vocabulary Tail:** Theoretically, if one considers the entire remaining vocabulary beyond the top K exposed logprobs, then $N_{\text{tail}} = |V| - K$. The maximum missing

²e.g., $|V| = 2^{17} = 131,072$ for some contemporary models, such as Mistral-Small-3.1-24B [35].

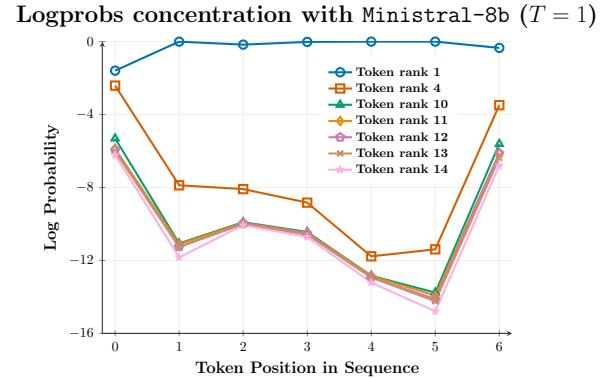


Figure 2: Illustrative plot of log-probabilities $\log p_{r(k),j}$ for several fixed ranks k (e.g., $k = 1, 4, \dots$) as a function of token position j in a sentence.

entropy is given by Eq. (6) with this N_{tail} . If $|V| \gg K$, then $N_{\text{tail}} \approx |V|$ and the bound can be approximated:

$$\Delta H_{K,\max}(q, t_{<j}) \approx h(Q_K(q, t_{<j})) + Q_K(q, t_{<j}) \log_2 |V| \quad (8)$$

using the expanded form (Eq. (7)).

- (2) **Truncated Tail due to Sampling:** In practice, LLM inference often employs sampling strategies (using a parameter K_{samp}), where probabilities for tokens outside this set are effectively considered zero for generation. If we are interested in the entropy within this practically sampleable set of K_{samp} tokens, and we only have access to the top K logprobs ($K < K_{\text{samp}}$), then the relevant tail for entropy estimation comprises $N_{\text{tail}} = K_{\text{samp}} - K$ tokens. Substituting this into the expanded form (Eq. (7)) yields:

$$\Delta H_{K,\max}(q, t_{<j}) = h(Q_K(q, t_{<j})) + Q_K(q, t_{<j}) \log_2 (K_{\text{samp}} - K). \quad (9)$$

This bound is substantially tighter than that derived from considering the full vocabulary tail (Eq. (8)), especially when $K_{\text{samp}} \ll |V|$ (e.g., typically $K_{\text{samp}} = 50$). It makes $\tilde{H}_K(q, t_{<j})$ a more meaningful estimate of the entropy over the set of tokens the model is practically considering for generation.

The quality of $\tilde{H}_K(q, t_{<j})$ as an estimator of the entropy within the relevant token pool (either full vocabulary or K_{samp} limited) can be assessed by comparing $\tilde{H}_K(q, t_{<j})$ to the corresponding $\Delta H_{K,N_{\text{tail}},\max}(q, t_{<j})$; a large ratio $\tilde{H}_K/\Delta H_{K,N_{\text{tail}},\max}$ suggests that $\tilde{H}_K(q, t_{<j})$ captures most of the relevant uncertainty.

Empirically, we observe a concentration in the log-probabilities of lower-ranked tokens (e.g., for ranks $i \approx 10 - 15$), as illustrated in Figure 2. This means that differences between successive $p_{r(k),j}$ tend to diminish for larger k within the accessible K . When combined with K_{samp} sampling, this characteristic further supports the idea that the entropy computed from the top K logprobs, $\tilde{H}_K(q, t_{<j})$, can capture the dominant portion of the uncertainty within the set of practically “sampleable” tokens, especially if K is reasonably close to K_{samp} .

3.2 Influence of Sampling Temperature

In practice, a LLM first predicts raw logits $l_{j,i}$ at each generation step j using a linear layer. These logits are then converted to probabilities $p_{j,i}$ by a softmax function with a temperature parameter T_{samp} , such that: $p_{j,i}(T_{\text{samp}}) = \exp(l_{j,i}/T_{\text{samp}})/\sum_{m=1}^{|V|} \exp(l_{j,m}/T_{\text{samp}})$. The resulting per-token information entropy, $H(q, t_{<j}; T_{\text{samp}})$, is thus dependent on T_{samp} . The behavior at extreme temperatures is well-understood:

- As $T_{\text{samp}} \rightarrow 0^+$, $p_{j,i}(T_{\text{samp}})$ approaches 1 for the token with the highest logit (assuming a unique maximum) and 0 for others. Consequently, $H(q, t_{<j}; T_{\text{samp}}) \rightarrow 0$, reflecting high certainty.
- As $T_{\text{samp}} \rightarrow \infty$, $p_{j,i}(T_{\text{samp}})$ approaches $1/|V|$ for all tokens, leading to a uniform distribution. Therefore, the entropy $H(q, t_{<j}; T_{\text{samp}}) \rightarrow \log_2 |V|$, its maximum possible value, reflecting maximum uncertainty.

These limits highlight that T_{samp} controls the sharpness of the output distribution. For our method, non-greedy decoding ($T_{\text{samp}} > 0$, and typically of the order of the unity) is essential to observe meaningful variations in log-probabilities and thus in the derived entropic scores.

3.3 Entropy Production Rate (EPR)

Based on the per-token entropy estimate $\tilde{H}_K(q, t_{<j})$, we define the “Entropy Production Rate” (EPR) for a generated sequence \mathcal{T}_q of length L_q in response to query q . It is the average estimated entropy over the top K accessible log-probabilities, across all tokens in the sequence:

$$\text{EPR}_K(q) = \frac{1}{L_q} \sum_{j=1}^{L_q} \tilde{H}_K(q, t_{<j}) = -\frac{1}{L_q} \sum_{j=1}^{L_q} \sum_{k=1}^K p_{r(k),j} \log_2(p_{r(k),j}). \quad (10)$$

This $\text{EPR}_K(q)$ serves as a global measure of hesitation or uncertainty for the entire generated sequence, given the black-box constraints. Yet, we propose to improve its veracity alignment by statistical learning on annotated generations, as described below.

3.4 Supervision on Entropic Contributions Ranks

In the following, we consider the scenario where we have access to a dataset \mathcal{D} where each entry is a query q , the generated sequence \mathcal{T} and an annotation $Y \in \{0, 1\}$ expressing whether \mathcal{T} is a correct answer to q . Please note that this annotation depends on the LLM used to generate \mathcal{T} . We consider that a realistic size for such a dataset is hundreds or thousands of entries.

To build a more nuanced supervised detector, we define features based on the entropic contributions of tokens at specific ranks. For the j -th generation in the sequence and for the rank $k \in \{1, \dots, K\}$, the entropic contribution of the k -th ranked token is

$$s_{k,j} = -p_{r(k),j} \log_2(p_{r(k),j}).$$

Note that the actually sampled token at step j may not be the highest-ranked token ($k = 1$), especially with $T_{\text{samp}} \gtrsim 1$. Our features are computed based on the probabilities of the tokens at

fixed ranks k , regardless of which token was sampled. We consider weighing each term of $\tilde{H}_K(q, t_{<j})$ (Eq. (4)) according to its rank k :

$$S_\beta(q, t_{<j}) = \beta_0 + \sum_{k=1}^K \beta_k s_{k,j}, \quad (11)$$

where $\beta \in \mathbb{R}^{K+1}$ are learned weights that aim at adapting the entropy measure thanks to the annotations of \mathcal{D} . This token-wise quantity $S_\beta(q, t_{<j})$ is close to $\tilde{H}_K(q, t_{<j})$, but can be negative. In practice, we learn the β parameters by maximizing:

$$\max_{\beta} \sum_{q, \mathcal{T}, Y \in \mathcal{D}} Y \log \left(\sigma \left(\sum_{j=1}^{L_q} \frac{S_\beta(q, t_{<j})}{L_q} \right) \right) + (1 - Y) \log \left(\sigma \left(1 - \sum_{j=1}^{L_q} \frac{S_\beta(q, t_{<j})}{L_q} \right) \right) \quad (12)$$

Eq. (12) corresponds to a cross-entropy loss, of a logistic regression on the average contributions $S_\beta(q, t_{<j})$ across tokens, in order to align the signal provided by the entropy to the ground truth annotations.

Because of its similarity with $\text{EPR}_K(q)$ (Eq. (10)), we denote the averaged contributions for the answer to a query $\sum_{j=1}^{L_q} \frac{S_\beta(q, t_{<j})}{L_q}$ as the Weighted Entropy Production Rate (WEPR $_K$). Moreover, $\sigma \left(\sum_{j=1}^{L_q} \frac{S_\beta(q, t_{<j})}{L_q} \right) \in [0, 1]$ can be interpreted as a probability for the generated sequence to be valid, with no hallucination. In particular, contrary to $\text{EPR}_K(q)$, this quantity is scaled, and can easily be displayed as a normalized score to the user.

While the annotations are at the sequence level, we can also consider a token-level measure:

$$\sigma(S_\beta(q, t_{<j})) \in [0, 1] \quad (13)$$

that can be interpreted as a probability that the j -th token causes the whole sequence to be an hallucination. This quantity is also scaled and may indicate to the user which tokens seems less reliable.

4 Experiments

4.1 Experimental Setup

4.1.1 Models and Datasets. We evaluated our methods on a suite of contemporary instruction-finetuned LLMs, including Mistral-Small-3.1-24B-Instruct-2503 [35], Falcon3-10B-Instruct [36], Phi-4 [37], and Ministrall-8B-Instruct-2410 [38]. For Question Answering (QA) tasks, we primarily utilized the TriviaQA [39] and WebQuestions [40] datasets. We also explored the applicability of our entropic measures in a Retrieval-Augmented Generation (RAG) context using a specialized dataset based on financial reports from the Argimi-Ardian dataset [41], to observe how entropy varies with the provision of relevant context (see Section 5 for this particular application). We focused on models generating relatively short answers, typical of non-extended reasoning QA, rather than lengthy outputs.

4.1.2 Answer Generation and Log-probability Extraction. For generating answers, LLMs were served using the v11m inference library [42]. To ensure variability in outputs and access to meaningful

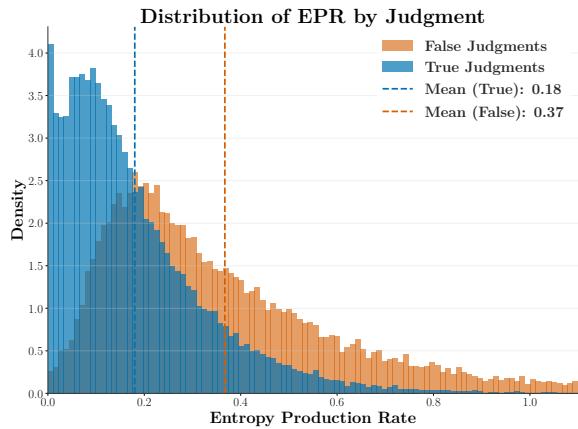


Figure 3: Illustrative distributions of Entropy Production Rate (EPR) for “True” (non-hallucinated) and “False” (hallucinated) judgments on responses from Falcon-10B on the TriviaQA dataset ($T_{\text{samp}} = 1$).

probability distributions for entropy calculation, non-greedy decoding was employed with a sampling temperature $T_{\text{samp}} = 1.0$. The top_p sampling parameter was set to 1.0 to consider the full vocabulary distribution as initially filtered by the model, while the K_{samp} sampling parameter could be set to a specific value (e.g., $K_{\text{samp}} = 50$) to effectively limit the vocabulary considered for generation and, consequently, for the calculation of the maximum possible missing entropy $\Delta H_{K,\max}(q, t_{<j})$ as defined in Section 3. For each generated token, we extracted the accessible top- K log-probabilities (typically $K \leq 15$ or $K \leq 20$ depending on the API/model).

4.1.3 Hallucination Annotation Protocol. The generated answers for the QA tasks were labeled as “True” (non-hallucinated) or “False” (hallucinated). This labeling was performed using an LLM-as-a-judge [43–46] approach, employing Gemma-3-27b-it [47] for its excellent performance. The judge LLM semantically compared the generated answer to the ground-truth answer (and any available aliases). This automated annotation method is scalable and suitable for our one-shot analysis framework, though we acknowledge potential limitations such as mislabeling of random correct guesses with high uncertainty or failing to flag confident but incorrect statements (high-certainty hallucinations).

4.2 Evaluation Protocol

4.2.1 Metrics. The primary metrics used to evaluate the performance of our hallucination detection methods (both EPR baseline and the supervised logistic regression model) are the Area Under the Precision-Recall Curve (AUC-PR) and the Area Under the Receiver Operating Characteristic Curve (ROC-AUC). AUC-PR is particularly informative for imbalanced datasets.

4.2.2 Training and Evaluation Strategy. For the supervised logistic regression model, data points (generated sequences and their labels) were grouped by their original query ID during the creation of training and testing splits. This ensures that no data points originating from the same query ID appear in both the training and

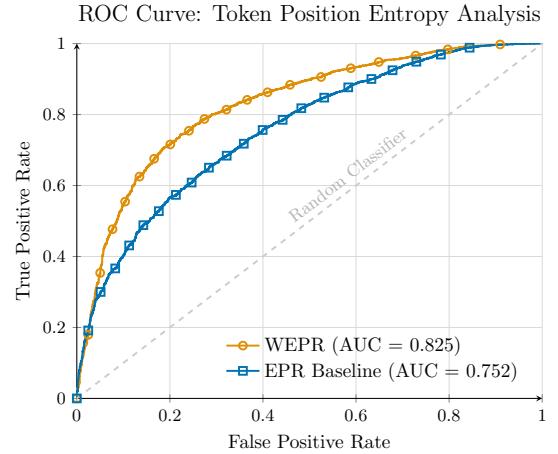


Figure 4: Sample ROC curves of the two methods applied on Falcon-3-10B, for $K = 15$ logprobs, on the TriviaQA dataset.

testing sets for a given fold, mitigating potential data leakage and overestimation of generalization performance. To ensure stability and robust estimation of performance, particularly given potential variability in smaller datasets or specific model-dataset pairings, we employed bootstrapping (e.g., ≈ 1000 iterations) during model evaluation where appropriate. We compare the performance of the logistic regression model against the baseline performance achieved using only $\text{EPR}_K(q)$ as the discriminating feature.

4.3 Results

4.3.1 True/False Classification Performance. We evaluated the capability of both the EPR baseline and our supervised Weighted Entropy Production Rate (WEPR) model to classify LLM-generated answers as “True” (non-hallucinated) or “False” (hallucinated). The WEPR models were trained on a subset of the TriviaQA dataset.

Performance metrics on the TriviaQA are presented in Table 1. The EPR-base method demonstrates notable discriminative ability, with PR-AUC scores ranging from 0.853 for Falcon-3-10B to 0.916 for Phi-4. Yet, WEPR consistently yields superior performance across all LLMs; for example, for Mistral-Small-3.1-24B, PR-AUC increases from 0.915 to 0.935, and ROC-AUC improves from 0.746 to 0.796. Similar enhancements are observed for other models, underscoring the benefit of the learned approach.

To assess generalization, the models (with WEPR still trained on TriviaQA) were also evaluated on the WebQuestions dataset, with results shown in the center part of Table 1. While performance is generally lower on this dataset, the WEPR model maintains its advantage over the EPR baseline for all LLMs. The improved classification performance of the WEPR model is further illustrated by the sample Receiver Operating Characteristic (ROC) curves for Falcon-3-10B on TriviaQA (Figure 4), where the WEPR curve consistently dominates that of the EPR baseline.

Table 1: Comparative hallucination and missing context detection performance (PR-AUC and ROC-AUC) of the baseline Entropy Production Rate (EPR-base) and our proposed Weighted EPR (WEPR) method across different LLMs, using $K = 15$ accessible logprobs. The table details: (i) True/False statement classification results on the TriviaQA dataset, which was also used for training the WEPR model; (ii) Generalization performance for the same True/False classification task on the WebQuestions dataset, using the WEPR model trained on TriviaQA; and (iii) Performance for a Missing Context Detection task on the ArGiMi-Ardian dataset, also utilizing the WEPR model trained on TriviaQA.

Model	Method	TriviaQA [39]		WebQuestions [40] (Evaluation)		Ardian[41] (Missing Context Detection)	
		PR-AUC	ROC-AUC	PR-AUC	ROC-AUC	PR-AUC	ROC-AUC
Mistral-Small-3.1-24B	EPR-base	0.915	0.746	0.847	0.625	0.802	0.810
	WEPR	0.935	0.796	0.860	0.651	0.860	0.850
Falcon-3-10B	EPR-base	0.853	0.754	0.809	0.682	0.839	0.828
	WEPR	0.884	0.820	0.831	0.729	0.905	0.901
Phi-4 (14.7B)	EPR-base	0.916	0.776	0.849	0.652	0.914	0.910
	WEPR	0.925	0.795	0.852	0.658	0.931	0.928
Minstral-8B-2410	EPR-base	0.869	0.807	0.769	0.654	0.802	0.810
	WEPR	0.888	0.835	0.798	0.697	0.844	0.843

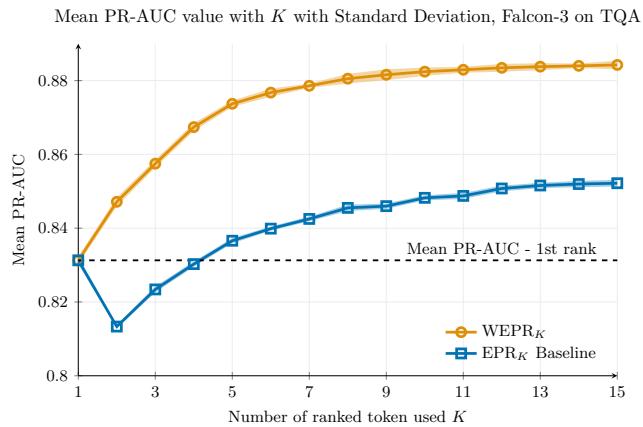


Figure 5: Evolution of the mean PR-AUC value with the number of considered ranked tokens K . The filling around the curves correspond to their respective standard deviations

4.3.2 Performance with respect to the Number of Accessible Logprobs (K). We investigated the impact of K on hallucination detection performance. Figure 5 displays the evolution of the mean PR-AUC for Falcon-3-10B on TriviaQA as K varies. The results indicate that performance generally improves with an increasing number of ranked token contributions up to a certain point. Notably, for several models, the PR-AUC tends to plateau when K reaches approximately 8 to 10. This suggests that incorporating entropic contributions beyond the top ~ 10 ranks may yield diminishing returns for hallucination detection. This finding has practical significance, as it implies that effective detection can be achieved even when API access is limited to a relatively small number of top logprobs, thereby reducing data requirements and potentially API costs.

An analysis of the learned coefficients (β_k) in the WEPR model revealed a consistent pattern: the coefficient associated with the second-ranked token’s entropic contribution (S_2) frequently exhibited an opposite sign compared to those of other prominent ranks. This was observed across all tested models and datasets, suggesting a distinct role for the second token’s uncertainty in the learned detection mechanism, that compensates the decrease of performance observed for $K = 2$ in Figure 5.

5 Application in RAG Output Analysis

Beyond general QA, we explored the applicability of our entropic methods for analyzing outputs in Retrieval-Augmented Generation (RAG) [48] pipelines, specifically for detecting the need for, or assessing the impact of, retrieved context. The premise is that uncertainty, as captured by EPR or related learned scores, might increase when an LLM generates an answer without sufficient supporting context, potentially indicating a higher risk of hallucination or irrelevance.

We demonstrated this capability using the ArGiMi-Ardian dataset [41], from which we derived specific questions regarding public companies annual reports³. Generated answers were assessed with and without the provision of relevant contextual documents. The task was framed as detecting "missing context" based on the entropic signature of the answer. Table 1 presents the performance of both EPR-base and our WEPR model (trained on the general TriviaQA dataset) for this missing context detection task. The results show that the WEPR model significantly outperforms the EPR baseline in this domain-specific RAG scenario. For example, with Falcon-3-10B, the ROC-AUC for detecting missing context increased by approximately 0.073 points. Figure 6 visually confirms this improvement.

This indicates that an entropic detector trained on a general QA dataset can transfer, to a notable extent, to specialized domains

³These questions are available online at https://huggingface.co/datasets/argimi/emerson_2023_questions.

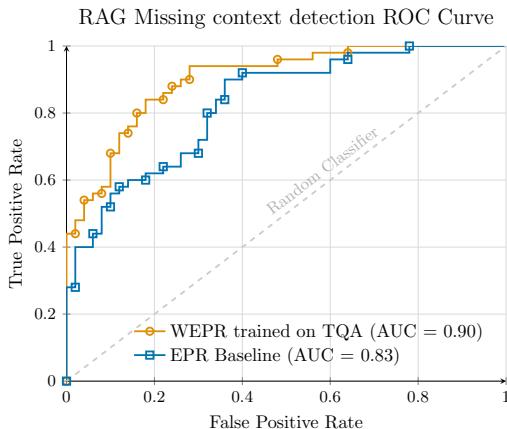


Figure 6: ROC Curve of a missing context detection in a financial RAG pipeline. The EPR baseline is compared to a WEPR trained on TriviaQA data with the True/False judgments. The LLM used was Falcon-3-10B, with $T_{\text{sample}} = 1$ and $K = 15$.

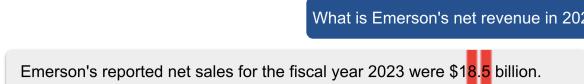


Figure 7: Token-level WEPR hallucination detection, in a deployment-ready chat-bot interface. The interface flags tokens such that $\sigma(S_\beta(q, t_{<j})) > 0.5$, with the flagged tokens appearing more red if the values tend to 1.

for identifying responses generated with insufficient context. Such a mechanism can be readily applied in RAG pipelines to flag outputs where context retrieval may have been inadequate or where the model is otherwise struggling, thereby triggering deeper information retrieval or alerting the user. Furthermore, the learned coefficients β_k from our WEPR model can be used to compute a weighted entropic contribution for each token j in a sequence $S_\beta(q, t_{<j})$ (Eq. (11)), offering a finer-grained view of uncertainty hotspots within the generated text. This WEPR-based token flagging was proved to be readily implementable for in-house chat-bots, as displayed in Figure 7.

6 Limitations

While our proposed methods demonstrate promising results for one-shot, black-box hallucination detection, several limitations warrant acknowledgment. First, our experiments were conducted on contemporary mid-sized LLMs (ranging from approximately 8B to 24B parameters). The performance of our methods on very large-scale models (e.g., 100B+ parameters) remains untested and may differ. Second, the current study focuses on QA tasks that typically yield relatively short, factual answers. We did not study tasks requiring extensive multi-step reasoning or very lengthy outputs, where the aggregated EPR signal might become less distinct or where different

types of uncertainty could dominate. Finally, our approach, like many other uncertainty-based methods, is inherently limited in its ability to detect "high-certainty hallucinations" [31]—instances where the model generates incorrect information with low output entropy (i.e., high confidence). If a LLM has strongly learned erroneous facts during its pre-training or fine-tuning, entropic measures based on output probabilities may not flag such confident fabrications.

7 Conclusion

This paper addressed the critical challenge of detecting hallucinations in Large Language Model outputs, particularly for "black-box" systems where access to internal states or full log-probability distributions is limited. We introduced an applied methodology for robust, one-shot hallucination detection rooted in information-theoretic principles, leveraging only the top- K token log-probabilities typically exposed by LLM APIs.

Our primary contribution is a supervised learning approach that utilizes features representing the mean entropic contributions of individual top-ranked tokens. Experimental results across diverse Question Answering datasets and multiple LLMs demonstrate that this learned estimator significantly outperforms a baseline Entropy Production Rate (EPR) metric in identifying hallucinatory content. Notably, we found that high detection performance can be achieved using a small number of accessible logprobs (typically $K < 10$), underscoring the practical efficiency and deployability of our method in API-constrained environments. Furthermore, we showcased its utility in a specialized finance framework for analyzing responses to queries on annual reports and highlighted its potential for identifying high-uncertainty tokens within a generated sequence.

This work provides a practical and effective technique for enhancing the trustworthiness of LLM responses in both general QA and Retrieval-Augmented Generation systems from a single generation pass. By offering a reliable method for uncertainty estimation and hallucination detection under common operational constraints, we contribute to the broader effort of developing more dependable and safely deployable LLM technologies. Future work may explore extending this approach to more complex reasoning tasks and investigating the nuances of the learned entropic feature weights for deeper interpretability.

Acknowledgments

This work was done as part of the ArGiMi project, funded by BPIFrance under the France2030 national effort towards numerical common goods. CM wishes to thank Antoine Delaby for preliminary discussions on RAG applications, along with Gabriel Trier, Lucas Tramonte and Rayane Bouaita for their help on the question dataset.

References

- [1] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. Language Models as Knowledge Bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China, 2019. Association for Computational Linguistics.
- [2] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H.

- Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent Abilities of Large Language Models, October 2022.
- [3] OpenAI et. al. GPT-4 Technical Report, March 2024.
- [4] Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. Object Hallucination in Image Captioning, March 2019.
- [5] S. Baker and T. Kanade. Hallucinating faces. In *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition* (Cat. No. PR00580), pages 83–88, March 2000.
- [6] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yeqin Bang, Delong Chen, Wenliang Dai, Ho Shu Chan, Andrea Madotto, and Pascale Fung. Survey of Hallucination in Natural Language Generation. *ACM Computing Surveys*, 55(12):1–38, December 2023.
- [7] Gaurav Sriramanan, Siddhant Bharti, Vinu Sankar Sadasivan, Shoumik Saha, Priyatham Kattakinda, and Soheil Feizi. LLM-Check: Investigating Detection of Hallucinations in Large Language Models.
- [8] Ming Jin, Qingsong Wen, Yuxuan Liang, Chaoli Zhang, Siqiao Xue, Xue Wang, James Zhang, Yi Wang, Haifeng Chen, Xiaoli Li, Shirui Pan, Vincent S. Tseng, Yu Zheng, Lei Chen, and Hui Xiong. Large Models for Time Series and Spatio-Temporal Data: A Survey and Outlook, October 2023.
- [9] Yubin Kim, Hyewon Jeong, Shan Chen, Shuyue Stella Li, Mingyu Lu, Kumail Alhamoud, Jimin Mun, Cristina Grau, Minseok Jung, Rodrigo Gameiro, Lizhou Fan, Eugene Park, Tristan Lin, Joonseok Yoon, Wonjin Yoon, Maarten Sap, Yulia Tsvetkov, Paul Liang, Xuhai Xu, Xin Liu, Daniel McDuff, Hyeonhoon Lee, Hae Won Park, Samir Tulebaev, and Cynthia Breazeal. Medical Hallucinations in Foundation Models and Their Impact on Healthcare, February 2025.
- [10] Fang Liu, Yang Liu, Lin Shi, Houkun Huang, Ruijing Wang, Zhen Yang, Li Zhang, Zhongqi Li, and Yuchi Ma. Exploring and Evaluating Hallucinations in LLM-Powered Code Generation, May 2024.
- [11] T. J. Sullivan. *Introduction to Uncertainty Quantification*. Springer, December 2015.
- [12] Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. SelfCheckGPT: Zero-Resource Black-Box Hallucination Detection for Generative Large Language Models, October 2023.
- [13] Xiaou Liu, Tiejin Chen, Longchao Da, Chacha Chen, Zhen Lin, and Hua Wei. Uncertainty Quantification and Confidence Calibration in Large Language Models: A Survey, March 2025.
- [14] Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U. Rajendra Acharya, Vladimir Makarenkov, and Saeid Nahavandi. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, 76:243–297, December 2021.
- [15] Yarin Gal. *Uncertainty in Deep Learning*. PhD thesis, University of Cambridge, 2016.
- [16] Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine learning*, 110(3):457–506, 2021.
- [17] Janis Postels, Francesco Ferroni, Huseyin Coskun, Nassir Navab, and Federico Tombari. Sampling-free epistemic uncertainty estimation using approximated variance propagation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2931–2940, 2019.
- [18] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pages 1050–1059. PMLR, 2016.
- [19] Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schieber, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Jackson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. Language Models (Mostly) Know What They Know, November 2022.
- [20] Amos Azaria and Tom Mitchell. The Internal State of an LLM Knows When It's Lying, October 2023.
- [21] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, July 1948.
- [22] Marylou Gabrié, Andre Manoel, Clément Luneau, Jean Barbier, Nicolas Macris, Florent Krzakala, and Lenka Zdeborová. Entropy and mutual information in models of deep neural networks. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.
- [23] Wonbin Kweon, Sanghwan Jang, SeongKu Kang, and Hwanjo Yu. Uncertainty Quantification and Decomposition for LLM-based Recommendation. In *Proceedings of the ACM on Web Conference 2025*, pages 4889–4901, April 2025.
- [24] Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic Uncertainty: Linguistic Invariances for Uncertainty Estimation in Natural Language Generation, April 2023.
- [25] Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630, June 2024.
- [26] Nicola Cecere, Andrea Bacci, Ignacio Fernández Tobías, and Amin Mantrach. Monte Carlo Temperature: A robust sampling strategy for LLM's uncertainty quantification methods, February 2025.
- [27] Litian Liu, Reza Pourreza, Sunny Panchal, Apratim Bhattacharyya, Yao Qin, and Roland Memisevic. Enhancing Hallucination Detection through Noise Injection, February 2025.
- [28] Eojin Joo, Young-Jun Lee, and Ho-Jin Choi. Entropy-based Sentence-level Hallucination Score in Large Language Models. In *2025 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 77–78, February 2025.
- [29] Jannik Kossen, Jiatong Han, Muhammed Razzak, Lisa Schut, Shreshth Malik, and Yarin Gal. Semantic Entropy Probes: Robust and Cheap Hallucination Detection in LLMs, June 2024.
- [30] Jinhao Duan, Huan Cheng, Shiqi Wang, Alex Zavalny, Chenan Wang, Renjing Xu, Bhavya Kaikhura, and Kaidi Xu. Shifting Attention to Relevance: Towards the Predictive Uncertainty Quantification of Free-Form Large Language Models, May 2024.
- [31] Adi Simhi, Itay Itzhak, Fazl Barez, Gabriel Stanovsky, and Yonatan Belinkov. Trust Me, I'm Wrong: High-Certainty Hallucinations in LLMs, February 2025.
- [32] Huan Ma, Jingdong Chen, Joey Tianyi Zhou, Guangyu Wang, and Changqing Zhang. Estimating LLM Uncertainty with Evidence, May 2025.
- [33] Leon Brillouin. *Science and Information Theory*. Courier Corporation, July 2013.
- [34] Marc Mezard and Andrea Montanari. *Information, Physics and Computation*. 2008.
- [35] Mistral Small 3.1 | Mistral AI. <https://mistral.ai/news/mistral-small-3-1>.
- [36] TII Team. The falcon 3 family of open models, December 2024.
- [37] Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Cao C. T. Menden, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Xin Wang, Rachel Ward, Yue Wu, Dingli Yu, Cyril Zhang, and Yi Zhang. Phi-4 Technical Report, December 2024.
- [38] Un Ministral, des Ministraux | Mistral AI. <https://mistral.ai/fr/news/ministraux>.
- [39] Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension, May 2017.
- [40] Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. Semantic Parsing on Freebase from Question-Answer Pairs. In David Yarowsky, Timothy Baldwin, Anna Korhonen, Karen Livescu, and Steven Bethard, editors, *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, Seattle, Washington, USA, October 2013. Association for Computational Linguistics.
- [41] Hicham Randrianarivo, Charles Moslonka, Arthur Garnier, and Emmanuel Malherbe. The ArGiMi datasets. <https://huggingface.co/datasets/artefactory/Argimi-Ardian-Finance-10k-text>, 2024.
- [42] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.
- [43] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena, December 2023.
- [44] Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. Large Language Models are not Fair Evaluators, August 2023.
- [45] Vaibhav Adlakha, Parishad BehnamGhader, Xing Han Lu, Nicholas Meade, and Siva Reddy. Evaluating Correctness and Faithfulness of Instruction-Following Models for Question Answering. *Transactions of the Association for Computational Linguistics*, 12:681–699, May 2024.
- [46] Hossein A. Rahmani, Emine Yilmaz, Nick Craswell, Bhaskar Mitra, Paul Thomas, Charles L. A. Clarke, Mohammad Aliannejadi, Clemencia Siro, and Guglielmo Faggioli. LLMJudge: LLMs for Relevance Judgments, August 2024.
- [47] Gemma et. al. Team. Gemma 3 Technical Report, March 2025.
- [48] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks, April 2021.