Hallucination Detection on a Budget: Efficient Bayesian Estimation of Semantic Entropy

Anonymous authors
Paper under double-blind review

Abstract

Detecting whether an LLM hallucinates is an important research challenge. One promising way of doing so is to estimate the semantic entropy (Farquhar et al., 2024) of the distribution of generated sequences. We propose a new algorithm for doing that, with two main advantages. First, due to us taking the Bayesian approach, we achieve a much better quality of semantic entropy estimates for a given budget of samples from the LLM. Second, we are able to tune the number of samples adaptively so that 'harder' contexts receive more samples. We demonstrate empirically that our approach systematically beats the baselines, requiring only 59% of samples used by Farquhar et al. (2024) to achieve the same quality of hallucination detection as measured by AUROC. Moreover, quite counterintuitively, our estimator is useful even with just one sample from the LLM.

1 Introduction

Detecting hallucinations in LLMs is a task of huge practical significance (Ji et al., 2023). An important subset of hallucinations, called 'confabulatory', amounts to the model making up confabulations or statements with made-up meanings (Filippova, 2020; Maynez et al., 2020). Recently, semantic entropy (Farquhar et al., 2024) has been introduced as an important indicator for detecting if a model exhibits this type of hallucination. Semantic Entropy is based on two principles. The first one is to measure a type of *Shannon entropy* of the sequences generated by a model, reflecting the idea that large entropy indicates confounding or a lack of knowledge. The second principle is to do the measurement in the space of *meanings* rather than directly operating on raw token sequences. By doing so, one can leverage the insight that in many cases, distinct token sequences can have the same meaning. It turns out that combining these insights to estimate semantic entropy and then thresholding on it value amounts to a highly competitive hallucination detection method (Farquhar et al., 2024).

While semantic entropy is a state-of-the art method of hallucination detection, computing it has a high cost. It first requires the generation of several independent answers to the same question and then a quadratic number of calls to the function determining if two meanings are the same. In fact, the work of Farquhar et al. (2024) used ten generations per prompt, which is prohibitively expensive in many practical cases. We address this bottleneck by making semantic entropy estimation much cheaper. We achieve this by leveraging insights from Bayesian literature about entropy estimation (Wolpert & Wolf, 1994; Hausser & Strimmer, 2009; Archer et al., 2014), building up a probabilistic belief about the underlying distribution over meanings and reasoning about how the belief in the space of meaning distributions affects the belief in the space of possible values of the entropy. We further study a novel, adaptive, setting, where 'harder' prompts are afforded a larger budget of samples. In this setting, the efficiency of our estimator can be increased even further.

Contributions We develop a new system for measuring semantic entropy, based on Bayesian principles. Compared to the work of Farquhar et al. (2024), we outperform all other ways of measuring semantic entropy, reducing the sample complexity measured as the number of LLM generations needed to achieve the same performance by 41% on average across datasets. We also release several datasets for semantic entropy estimation, enabling researchers without access to GPU resources to work on even better estimators.

2 Preliminaries

Language Generation Denote with X the set of prompts[1] the LLM can be asked to respond to. One instance[2] of a prompt $\mathbf{x} \in X$ would be

$$\mathbf{x} = \textit{`Where is the Eiffel Tower?'}$$

Denote with $S_{\mathbf{x}}$ the set of all reply sequences given a prompt $\mathbf{x}$.[3] Denote the probability of the LLM generating a sequence $s \in S_{\mathbf{x}}$ in response to the prompt $\mathbf{x} \in X$ with $p(\mathbf{s}|\mathbf{x})$. One possible continuation is

$$\mathbf{s} = \textit{`It's Paris.'}$$

We model both $\mathbf{s}$ and $\mathbf{x}$ as random variables, denoting them in bold.

Meanings Semantic entropy is always conditioned on a prompt. We are going to consider a given generic context $\mathbf{x}$. Denote the set of meanings (also known as meaning classes or semantic classes) with $M_{\mathbf{x}}$. The set $M_{\mathbf{x}}$ is a partition of the set of $S_{\mathbf{x}}$.[4] We assume that the set of meanings is finite (although we do not necessarily know its cardinality). Denote with

$$f^{\mathbf{x}}(\mathbf{s}) : S_{\mathbf{x}} \to M_{\mathbf{x}}$$

the function that determines the meaning of the sequence $\mathbf{s}$ in context $\mathbf{x}$. While $f^{\mathbf{x}}$ is typically implemented using calls to an entailment oracle, we abstract away this implementation detail in this paper. For a random sequence $\mathbf{s} \sim p(\cdot|\mathbf{x})$, we consider the random variable

$$\mathbf{m} = f^{\mathbf{x}}(\mathbf{s}).$$

Semantic Entropy The semantic entropy corresponding to the context $\mathbf{x}$ is defined as the Shannon entropy of the random variable $\mathbf{m}$:

$$\mathrm{SE}_{\mathbf{x}} = \mathbb{H}[\mathbf{m}].$$

In the remainder of the paper, we will occasionally drop the subscript $\mathbf{x}$ where the dependence on the context $\mathbf{x}$ is obvious.

The Estimation Problem The aim of this paper is to estimate semantic entropy from a finite dataset, based on N calls to the target LLM. For a given context $\mathbf{x}$, our samples are represented as a list of independently generated sequences

$$\mathbf{s}_1, \dots, \mathbf{s}_N \sim p(\cdot|\mathbf{x}).$$

For each sequence, we can determine its meaning, obtaining the corresponding list of meanings

$$\mathbf{m}_1, \dots, \mathbf{m}_N, \text{ where } \mathbf{m}_i = f^{\mathbf{x}}(\mathbf{s}_i).$$

We are also given the probabilities of generating $\mathbf{s}_1, \dots, \mathbf{s}_N$, which we denote with $p(\mathbf{s}_i|\mathbf{x})$. Note that these probabilities can be obtained at no extra cost when generating sequences from the LLM. Our overall dataset is defined as

$$\mathcal{D} = (\mathbf{s}_1, \mathbf{m}_1, p(\mathbf{s}_1|\mathbf{x})), \dots, (\mathbf{s}_N, \mathbf{m}_N, p(\mathbf{s}_N|\mathbf{x})).$$

The dataset can in general have repeated elements. It is important to note that we only have the probabilities for the sequences that were actually generated, which might represent a very small fraction of all possible sequences. Moreover, an important feature of our problem is that we want to achieve reasonable estimates using as few samples as possible. When generating the dataset, we do have the ability to ask for more data, i.e. increase N until we are satisfied that our estimate of semantic entropy is good enough. We will make this precise in Section 3.

---

[1] Occasionally, the notion of a 'context' is used in addition to the prompt, to model the phenomenon that the same prompt can have different meanings in different contexts. To keep our notation simple, we don't explicitly model contexts. However, if one wants to generalize our results to contexts, it can be done by considering $\mathbf{x}$ to be a context-prompt tuple.

[2] We use examples borrowed from Farquhar et al. (2024).

[3] Typically, both X and $S_{\mathbf{x}}$ are the set of natural language sequences. However, whether $X = S_{\mathbf{x}}$ is immaterial for this paper.

[4] The term 'partition' is used in the mathematical way so that a sequence $s \in S_x$ always has exactly one meaning.

3 A Bayesian Estimator for Semantic Entropy

We now give a sketch of our estimation process. Since we have finite data, our estimate of semantic entropy will be noisy. Adopting the Bayesian philosophy, we construct a random variable $\mathbf{h}$ that represents our belief about the value of the semantic entropy $\mathrm{SE}_{\mathbf{x}}$, based on limited available data contained in a dataset $\mathcal{D}$ (we define $\mathbf{h}$ formally later on in the Section). Since we are motivated by detecting hallucinations, our focus is on measuring the quantities

$$\mathbb{E}\left[\mathbf{h}\right] \text{ and } \mathrm{Var}\left[\mathbf{h}\right], \tag{1}$$

i.e. the mean and variance of our Bayesian belief about what the value of the semantic entropy is.

In this Section, we describe a Bayesian process for forming a probabilistic belief over $\mathbf{h}$. For presentation purposes, we first derive our estimator under the assumption that that the number of meaning classes is known, i.e. $|M_{\mathbf{x}}| = K$. Under this assumption, in Section 3.1, we describe the basic variant of the estimator, which only uses the list of meanings $\mathbf{m}_1, \ldots, \mathbf{m}_N$. In Section 3.2, we then extend the estimator to also make use of the probabilities of the generated sequences $p(\mathbf{s}_1|\mathbf{x}), \ldots, p(\mathbf{s}_N|\mathbf{x})$. In Section 3.3, we remove the assumption that K is known, defining a hierarchical Bayesian system that maintains a belief about K. In Section 3.4, we summarize our methodology in the form of pseudo-code.

3.1 Basic Variant of the Estimator

We first summarize the dataset, counting how many times we sampled each meaning. The counter for meaning $j \in M$ is defined as

$$\mathbf{c}_j = \left|\{i \,:\, \mathbf{m}_i = j\}\right|. \tag{2}$$

We have $\sum_j \mathbf{c}_j = N$. We seek to use the information from the counters to get an idea about how the true distribution over meanings looks like. We adopt the Bayesian modeling philosophy, using a belief distribution. Specifically, our Bayesian belief about the probability distribution over meanings is modeled as

$$B_p = \mathrm{Dirichlet}(\alpha + \mathbf{c}_1, \ldots, \alpha + \mathbf{c}_K), \tag{3}$$

where we used the letter B_p to indicate that the probability distribution is used as a belief and K is the number of meanings. The value α represents a prior of the Dirichlet distribution and is a hyper-parameter of our method[5]. Equation 3 has a Bayesian interpretation as the posterior distribution, when the prior is chosen to be $\mathrm{Dirichlet}(\alpha, , \ldots, \alpha)$, and the likelihood is categorical. Consider a random variable distributed according to B_p:

$$\mathbf{b} \sim B_p.$$

Here $\mathbf{b} \in \Delta^K$ is itself a probability distribution, representing the fraction of the total probability mass assigned to each meaning. A belief about $\mathbf{b}$ induces a belief about its entropy, represented with the random variable

$$\mathbf{h} \triangleq \mathbb{H}[\mathbf{b}].$$

The expectation as per equation 1 can be computed analytically as

$$\mathbb{E}\left[\mathbf{h}\right] = \int_{\mathbf{b}} \mathbb{H}[\mathbf{b}] p_{B_p}(\mathbf{b}) d\mathbf{b} = \psi\left(1 + K\alpha + \sum_j \mathbf{c}_j\right) - \sum_j \frac{\alpha + \mathbf{c}_j}{K\alpha + \sum_j' \mathbf{c}_j'} \psi\left(\alpha + \mathbf{c}_j + 1\right),$$

where ψ is the digamma function (see Appendix A.1 for proof and a derivation of a similar expression for the variance integral).

---

[5] We study the sensitivity of our method to the choice of α in Appendix D.

3.2 An Estimator that Also Uses Sequence Probabilities

An LLM doesn't just give us meaning-classes but also probabilities of the generated continuations. Conditioning on this additional information can make our estimates of semantic entropy much better. Recall that the probability of generating $\mathbf{s}$ is denoted with $p(\mathbf{s}_i|\mathbf{x})$. We can define a constraint bounding the probability of each meaning, writing

$$\text{constr}(\mathbf{b}, \mathcal{D}) \;:=\; \left\{ \mathbf{b}_j \geq \sum_{\mathbf{s} \in \{\mathbf{s}\,:\, \mathbf{s} \in \mathcal{D}\;,\; f^{\mathbf{x}}(\mathbf{s}) \,=\, j\}} p(\mathbf{s}|\mathbf{x}) \right\}_{j=1,\ldots,K} . \tag{4}$$

Intuitively, equation 4 holds because the probability of a meaning j is at least equal to the sum of probabilities of distinct sequences with that meaning. The bound is not an equality because it is possible (and typically the case) that we didn't generate all sequences that correspond to this meaning. Probabilistically, the constraint can be interpreted as an event, i.e. something we can condition on. We in fact do that, modifying the estimator to sample from belief conditional on the constraint:

$$\mathbf{b} \sim B_p \mid \text{constr}.$$

This conditioning is in fact key to obtaining good empirical results. While the conditioning operation allows us to leverage all information at our disposal, it also makes the process of computing the expectation in equation 1 more complicated. In practice, the integrals for $\mathbb{E}[\mathbf{h}]$ and $\text{Var}[\mathbf{h}]$, which are defined as:

$$\begin{aligned} \mathbb{E}\left[\mathbf{h}\right] &= \int_{\mathbf{b}} \mathbb{H}[\mathbf{b}] p_B^{\text{trunc}}(\mathbf{b}; \mathcal{D}) d\mathbf{b}, \\ \text{Var}\left[\mathbf{h}\right] &= \left(\int_{\mathbf{b}} \mathbb{H}[\mathbf{b}]^2 p_B^{\text{trunc}}(\mathbf{b}; \mathcal{D}) d\mathbf{b} \right) - \left(\mathbb{E}\left[\mathbf{h}\right]\right)^2, \end{aligned}$$

will have to be computed approximately using a Monte Carlo method. Here, the density of a truncated Dirichlet random variable is defined as

$$p_B^{\text{trunc}}(\mathbf{b}; \mathcal{D}) = \begin{cases} \frac{p_B(\mathbf{b})}{\int_{\mathbf{b} \in \text{constr}(\mathbf{b}, \mathcal{D})} p_B(\mathbf{b}) d\mathbf{b}}, & \text{if } \mathbf{b} \in \text{constr}(\mathbf{b}, \mathcal{D}), \\ 0 & \text{if } \mathbf{b} \notin \text{constr}(\mathbf{b}, \mathcal{D}). \end{cases}$$

We treat a particular choice of the integration algorithm as an implementation detail and defer its discussion to Appendix A.2. Note that, even though we are using a Monte Carlo method, obtaining estimates of semantic entropy is is still relatively cheap. This is because the integration routine is orders of magnitude cheaper than increasing N. In other words, sampling meanings from an LLM is expensive while MC integration has negligible cost.

3.3 Unknown Number of Meanings

Previously, we assumed that we know the number of meanings possible for a given context x, i.e. $|M_x| = K$ for a known value of K that can be used to design the estimator. This is not a realistic assumption since the number of possible meanings can be vastly different for each context and is not typically known a priori. We resolve this dilemma in a Bayesian way, representing our Bayesian belief about the number of meanings using a probability distribution.

$$B_K = \text{Discrete}((K_1, \lambda_1), \ldots, (K_M, \lambda_M)). \tag{5}$$

Here, the parameters $\lambda_1, \ldots, \lambda_M$ are relative frequencies of each support size K_i. The parameters can be computed using a small (separate) training dataset and M is the maximum observed support size.

Our belief about the number of meanings can be used to estimate the entropy in a hierarchical way. Specifically, given we have observed that there are at least K_{min} different meanings, we obtain the probability distribution

$$\mathbf{K} \sim B_K \mid \left(\mathbf{K} > K_{\text{min}}\right).$$

Algorithm 1 Estimate of Semantic Entropy for a prompt $\mathbf{x}$.

```
$\mathcal{D} \leftarrow [\,]$
repeat
    $\mathbf{s}, p(\mathbf{s}|\mathbf{x}) \leftarrow$ llmSample()    ▷ Sample $\mathbf{s}$ and store the corresponding probability.
    $\mathbf{m} \leftarrow f^{\mathbf{x}}(\mathbf{s})$    ▷ Determine the meaning.
    $\mathcal{D}$.append$(\mathbf{s}, \mathbf{m}, p(\mathbf{s}|\mathbf{x}))$
    $K_{\text{min}} \leftarrow |\{\mathbf{m} \in \mathcal{D}\}|$
    for $j \in \{1, \ldots, K_{\text{min}}\}$ do
        $\mathbf{c}_j \leftarrow |\{i \,:\, \mathbf{m}_i = j, \ \ \mathbf{m}_i \in \mathcal{D}\}|$    ▷ Use equation 2.
    end for
    for $K \in \text{Support}\,(B_K \mid (\mathbf{K} > K_{\text{min}}))$ do
        $\widehat{e}_K \leftarrow$ numericallyIntegrate$(\int_{\mathbf{b}} \mathbb{H}[\mathbf{b}] p_B^{\text{trunc}}(\mathbf{b}; \mathcal{D}; K) d\mathbf{b})$
        $\widehat{e^2}_K \leftarrow$ numericallyIntegrate$(\int_{\mathbf{b}} \mathbb{H}[\mathbf{b}]^2 p_B^{\text{trunc}}(\mathbf{b}; \mathcal{D}; K) d\mathbf{b})$
        $\widehat{var}_K = \widehat{e^2} - (\widehat{e})^2$
    end for
    $\widehat{e}, \widehat{var} \leftarrow$ aggregateSupport$(\widehat{e}_k, \ \widehat{var}_k)$    ▷ Apply equation 6 for $k \in \{K_{\text{min}}, \ldots, K_{\text{max}}\}$.
until $\widehat{var} \geq \gamma$
return $\widehat{e}$
```

Here, we used bold font for $\mathbf{K}$ to denote it is a random variable. We can use probabilities of this discrete distribution to take an expectation of entropy estimates conditioned on particular values of K:

$$\mathbb{E}[\mathbf{h}] = \mathbb{E}_{\mathbf{K}}\left[\mathbb{E}\left[\mathbf{h}|\mathbf{K}\right]\right], \quad \text{Var}[\mathbf{h}] = \mathbb{E}_{\mathbf{K}}\left[\text{Var}\left[\mathbf{h}|\mathbf{K}\right]\right] + \text{Var}_{\mathbf{K}}\left[\mathbb{E}\left[\mathbf{h}|\mathbf{K}\right]\right], \tag{6}$$

where the quantities $\mathbb{E}\left[\mathbf{h}|\mathbf{K}\right]$ and $\text{Var}\left[\mathbf{h}|\mathbf{K}\right]$ can be obtained as described in Section 3.2. In our pseudo-code (see the next Section), equation 6 is assumed to be implemented using a procedure aggregateSupport.

3.4 Algorithm

Having specified the estimator for the quantities $\mathbb{E}[\mathbf{h}]$ and $\text{Var}[\mathbf{h}]$, we still need to specify the stopping rule for determining the right number of samples N. It is natural to keep drawing more samples until

$$\text{Var}[\mathbf{h}] \geq \gamma, \tag{7}$$

where γ is a desired level of precision. Increasing γ indicates we are satisfied with lower-quality semantic entropy estimates, which allows us to use smaller N. Intuitively, this stopping rule reflects the fact that we only want to know the semantic entropy to a certain level of precision. We summarize the ideas introduced in Sections 3.1, 3.2 and 3.3 in Algorithm 1.

4 Baselines

4.1 Other Estimators For Semantic Entropy

There are two existing baselines for measuring semantic entropy, both of which coming from the paper by Farquhar et al. (2024). They are called *discrete semantic entropy* and *semantic entropy* in that paper, although we use the term *histogram semantic entropy* for the former and *rescaled semantic entropy* for the latter to avoid confusion with the concept of semantic entropy itself, which is independent of the estimator used.

Histogram Semantic Entropy This estimator samples a fixed number of sequences, computes the meaning of each sequence and then computes the entropy of the empirical histogram of the meaning distribution. It is computed from the meaning counts $\mathbf{c}_j$ as $-\sum_j (\frac{\mathbf{c}_j}{N}) \log \frac{\mathbf{c}_j}{N}$.

Rescaled Semantic Entropy This estimator also samples a fixed number of sequences and then computes the meaning of each. However, it assigns probabilities to each meaning in a different way. First, one defines the un-normalized probability distribution

$$q(\mathbf{m}|\mathbf{x}) = \sum_{\mathbf{s} \in \{\mathbf{s}\,:\,\mathbf{s} \in \mathcal{D}\ ,\ f^{\mathbf{x}}(\mathbf{s})\,=\,\mathbf{m}\}} p(\mathbf{s}|\mathbf{x}).$$

This is then normalized as

$$p(\mathbf{m}|\mathbf{x}) = \frac{q(\mathbf{m}|\mathbf{x})}{\sum_{\mathbf{m}} q(\mathbf{m}|\mathbf{x})},$$

and the semantic entropy is computed as the Shannon entropy of this distribution. In addition, the probabilities $p(\mathbf{s}|\mathbf{x})$, which normally correspond to the multiplication of the probabilities of each token conditioned on past tokens, are heuristically replaced by the exponent of the mean log probability of each token, a process known as length normalization. We report results for the rescaled estimator both with and without the (heuristic) length normalization.

4.2 Other Baselines for Hallucination Detection

Semantic Entropy is not the only way of detecting hallucinations. We consider two non-entropy baselines.

P(True) It has been shown that LLMs have surprising introspective abilities, i.e. one can often see if the LLM is hallucinating simply by simply asking it (Kadavath et al., 2022). We use the same procedure for measuring P(True) as was employed by Farquhar et al. (2024).

Sequence Log Likelihood Recently, Aichberger et al. (2024) suggested using the log probability of the sequence generated greedily as a predictor of hallucinations, justifying it using the notion of zero-one scoring rule (Hofman et al., 2024). Unfortunately, this method is not directly compatible with the evaluation protocol of Farquhar et al. (2024), which does not perform greedy generation. In order to stay within the boundaries of that protocol, we instead used the log likelihood of a sequence generated with a low (but nonzero) temperature.

5 Prior Work

Hallucination Detection We do not provide a complete survey of hallucinations in Large Language Models, instead referring the reader to the work of Ji et al. (2023). We focus our work on combating 'confabulatory' hallucinations also addressed by Farquhar et al. (2024), i.e. situations where the LLM is randomly adding spurious facts to its replies. This is the same kind of hallucinations that was considered by Filippova (2020) and Maynez et al. (2020).

Semantic Entropy We build on the works that pioneered semantic entropy (Kuhn et al., 2023; Farquhar et al., 2024) by providing a more statistically efficient estimator. Our work is different from entropy probes (Kossen et al., 2024), which attempt to distill a thresholded version of semantic entropy into a classifier, although the ideas can certainly be used in conjunction with each other (similar ideas were also explored by Chen et al. (2024)). Our estimators do not attempt to leverage similarity in the meaning clusters (Nikitin et al., 2024; Qiu & Miikkulainen, 2024), instead focusing on obtaining as accurate estimates of vanilla semantic entropy for a given sample budget as possible.

Bayesian Entropy Estimation Wolpert & Wolf (1994) have provided the foundations for Bayesian estimators of entropy for arbitrary priors, and also pioneered the specialization to the case of the Dirichlet prior. Hausser & Strimmer (2009) have provided an explicit summary of the equivalences between the Dirichlet-Bayesian estimator and various pre-existing entropy estimators, for different values of the Dirichlet prior parameter. Archer et al. (2014) have provided an overview of past work on Bayesian entropy estimation, in addition to extending the framework to distributions with countably infinite support.[6]

Epistemic Uncertainty in LLMs The distinction between epistemic and aleatoric uncertainty (Gal et al., 2016; 2017; Kendall & Gal, 2017) has been proposed as a useful idea in modeling the behavior of LLMs (Abbasi Yadkori et al., 2024). In this paper, we do not distinguish between aleatoric and epistemic uncertainty, instead staying in the framework of Farquhar et al. (2024) and modeling the combined predictive uncertainty. While an accurate model of epistemic uncertainty would almost certainly lead to improved hallucination detection, we leave such extensions to further work.

Human Perception of Hallucinations Hallucinations are related to how confident LLMs are about their outputs. Recent research (Steyvers et al., 2025) studies how such self-confidence intrinsic in LLMs relates to how humans perceive it. Our work is largely orthogonal to this effort. Indeed, we treat the definition of semantic entropy as a given and focus on finding the statistically most efficient way to estimate it.

6 Experiments

6.1 Experimental Setup

Evaluation Methodology Our goal is to measure the quality of semantic entropy estimates as quantified with AUROC on hallucination detection tasks. To do so, we follow the methodology from the paper by Farquhar et al. (2024) as much as possible, deviating from it only by (1) separating out the dataset generation phase and the entropy estimation phase, (2) varying the sample budget N and (3) removing bugs from the dataset generation code. We defer the detailed discussion of these changes to Appendix B.

LLMs and Source Datasets We investigate the behavior of three LLMs. We use Llama-2-7b-chat for comparability with Farquhar et al. (2024). We also evaluate on the much more modern Llama-3.2-3B-Instruct and on Mistral-Small-24B-Instruct-2501. These LLMs are referred to as Llama 2, Llama 3 and Mistral in our figures. We used the TriviaQA (Joshi et al., 2017), SQUAD (Rajpurkar et al., 2016), SVAMP (Patel et al., 2021) and NQ (Lee et al., 2019) datasets.

Derivative Entropy Estimation Datasets For each combination of LLM and dataset, we generated a derivative dataset of 100 LLM generations per prompt for 1000 prompts, which we then used to estimate semantic entropy. We will release these derivative datasets upon acceptance allowing researches without GPU access to work on even better estimators for semantic entropy.

Train and Test Our Bayesian Semantic Entropy estimator requires a training set to estimate the prior on the size of the support of the meaning distribution as in equation 5. We use the first 200 prompts from each derivative dataset as the training set and the remaining 800 as the test set.

Temperature following the methodology of Farquhar et al. (2024), the N LLM responses are generated with temperature 1.0. On the other hand, the LLM response about which we seek to determine if it is a hallucination is generated with temperature 0.1. Because GPU response generation is expensive, we did not tune those temperatures.

6.2 Results

We performed two types of experiment. First, we studied the setting of a fixed budget of samples per prompt. Second, we varied the number of samples per prompt, giving harder prompts more data. In both cases, we note that we can support $N = 1$ because we have access to the probability of the generates sequence, which already gives us an imperfect but still useful handle on the entropy (for example, if the probability is close to one, we know that the entropy is almost zero).

---

[6] We do not use their infinite support framework, instead modeling unknown support using techniques described in Section 3.3.

$N = 2$

LLM	Dataset	LL	P(true)	SE-Bayes	SE-Histogram	SE-Rescaled	SE-Rescaled (h)
Llama-2	NQ	0.583 ± 0.000	0.461 ± 0.000	$\mathbf{0.723 \pm 0.007}$	0.652 ± 0.010	0.654 ± 0.013	0.644 ± 0.014
	SVAMP	0.631 ± 0.000	0.469 ± 0.000	$\mathbf{0.855 \pm 0.022}$	0.748 ± 0.024	0.749 ± 0.027	0.759 ± 0.025
	Squad	0.624 ± 0.000	0.441 ± 0.000	$\mathbf{0.735 \pm 0.007}$	0.654 ± 0.012	0.659 ± 0.017	0.649 ± 0.010
	Trivia QA	0.594 ± 0.000	0.436 ± 0.000	$\mathbf{0.737 \pm 0.006}$	0.670 ± 0.012	0.675 ± 0.012	0.673 ± 0.010
Llama-3.2	NQ	0.627 ± 0.000	0.615 ± 0.000	$\mathbf{0.728 \pm 0.008}$	0.640 ± 0.013	0.663 ± 0.017	0.649 ± 0.020
	SVAMP	0.647 ± 0.000	0.414 ± 0.000	$\mathbf{0.845 \pm 0.022}$	0.761 ± 0.032	0.774 ± 0.030	0.771 ± 0.028
	Squad	0.610 ± 0.000	0.555 ± 0.000	$\mathbf{0.664 \pm 0.019}$	0.608 ± 0.024	$\mathbf{0.642 \pm 0.029}$	$\mathbf{0.621 \pm 0.027}$
	Trivia QA	0.614 ± 0.000	0.710 ± 0.000	$\mathbf{0.768 \pm 0.004}$	0.699 ± 0.005	0.706 ± 0.007	0.708 ± 0.008
Mistral	NQ	0.695 ± 0.000	$\mathbf{0.731 \pm 0.000}$	0.705 ± 0.013	0.647 ± 0.007	0.700 ± 0.007	0.654 ± 0.009
	SVAMP	0.645 ± 0.000	0.843 ± 0.000	$\mathbf{0.876 \pm 0.011}$	0.793 ± 0.023	0.815 ± 0.028	0.812 ± 0.024
	Squad	$\mathbf{0.698 \pm 0.000}$	0.687 ± 0.000	0.667 ± 0.010	0.618 ± 0.005	0.671 ± 0.017	0.631 ± 0.010
	Trivia QA	0.672 ± 0.000	0.647 ± 0.000	$\mathbf{0.682 \pm 0.008}$	0.638 ± 0.011	0.645 ± 0.013	0.643 ± 0.012

$N = 5$

LLM	Dataset	LL	P(true)	SE-Bayes	SE-Histogram	SE-Rescaled	SE-Rescaled (h)
Llama-2	NQ	0.583 ± 0.000	0.461 ± 0.000	$\mathbf{0.752 \pm 0.006}$	0.730 ± 0.009	0.701 ± 0.012	0.725 ± 0.010
	SVAMP	0.631 ± 0.000	0.469 ± 0.000	$\mathbf{0.871 \pm 0.011}$	$\mathbf{0.849 \pm 0.012}$	$\mathbf{0.853 \pm 0.013}$	$\mathbf{0.858 \pm 0.013}$
	Squad	0.624 ± 0.000	0.441 ± 0.000	$\mathbf{0.774 \pm 0.008}$	0.756 ± 0.004	0.711 ± 0.012	0.752 ± 0.005
	Trivia QA	0.594 ± 0.000	0.436 ± 0.000	$\mathbf{0.763 \pm 0.009}$	0.734 ± 0.007	0.737 ± 0.006	0.735 ± 0.006
Llama-3.2	NQ	0.627 ± 0.000	0.615 ± 0.000	$\mathbf{0.760 \pm 0.008}$	0.732 ± 0.012	0.691 ± 0.005	0.733 ± 0.012
	SVAMP	0.647 ± 0.000	0.414 ± 0.000	$\mathbf{0.870 \pm 0.009}$	$\mathbf{0.860 \pm 0.010}$	0.850 ± 0.004	$\mathbf{0.864 \pm 0.009}$
	Squad	0.610 ± 0.000	0.555 ± 0.000	$\mathbf{0.710 \pm 0.017}$	$\mathbf{0.707 \pm 0.012}$	0.667 ± 0.021	$\mathbf{0.705 \pm 0.013}$
	Trivia QA	0.614 ± 0.000	0.710 ± 0.000	$\mathbf{0.792 \pm 0.004}$	0.775 ± 0.002	0.763 ± 0.005	0.777 ± 0.004
Mistral	NQ	0.695 ± 0.000	0.731 ± 0.000	$\mathbf{0.780 \pm 0.006}$	0.762 ± 0.007	0.728 ± 0.008	0.756 ± 0.007
	SVAMP	0.645 ± 0.000	0.843 ± 0.000	$\mathbf{0.880 \pm 0.019}$	$\mathbf{0.866 \pm 0.021}$	$\mathbf{0.855 \pm 0.027}$	$\mathbf{0.878 \pm 0.022}$
	Squad	0.698 ± 0.000	0.687 ± 0.000	$\mathbf{0.731 \pm 0.008}$	$\mathbf{0.719 \pm 0.010}$	0.699 ± 0.008	0.712 ± 0.008
	Trivia QA	0.672 ± 0.000	0.647 ± 0.000	$\mathbf{0.691 \pm 0.005}$	$\mathbf{0.688 \pm 0.005}$	$\mathbf{0.684 \pm 0.004}$	$\mathbf{0.690 \pm 0.005}$

Table 1: Experimental results for a fixed budget of $N = 2$ and $N = 5$ samples per prompt.

Fixed Budget Per Prompt Results for $N = 2$ and $N = 5$ samples per prompt[7] are shown in Table 1. Bold font is applied as follows: the estimator with the best mean performance is put in bold, together with all the others with overlapping confidence bars. It can be seen that the Bayesian estimator mostly outperformed or tied with other approaches to measuring semantic entropy, with the difference being greater for small N. We can also see that it is difficult to conclude which version of the rescaled estimator is better.

Main Experiment: Adaptive Budget Per Prompt As described in Section 3.4, the Bayesian framework gives us an additional handle on sample complexity in that we can use the variance of the belief about the semantic entropy as a proxy for confidence. Results are shown in Figures 1, 2 and 3. All confidence bars for the AUROC estimates in our paper represent 1.96 times the standard error. It can be seen that our Bayesian estimator is nearly Pareto-optimal in the sense that we achieve better AUROC than other approaches to semantic entropy, regardless of the value of N. Note that performance of the adaptive Bayesian estimator for a given N will in general be better than performance for the same fixed value of N. This is because, while the number of prompts is still N on average, harder prompts will get more samples (and easier prompts will get fewer). Concerning the non-semantic-entropy baselines, we outperform them for all N for Llama 2 and 3, while needing $N \geq 3$ for Mistral. We stress one additional take-away from the experiment: our Bayesian estimator is often competitive even for $N = 1$. This is completely counterintuitive since the entailment oracle (a crucial component of semantic entropy) is not needed in that case.

---

[7] See Appendix C for results for other values of N.

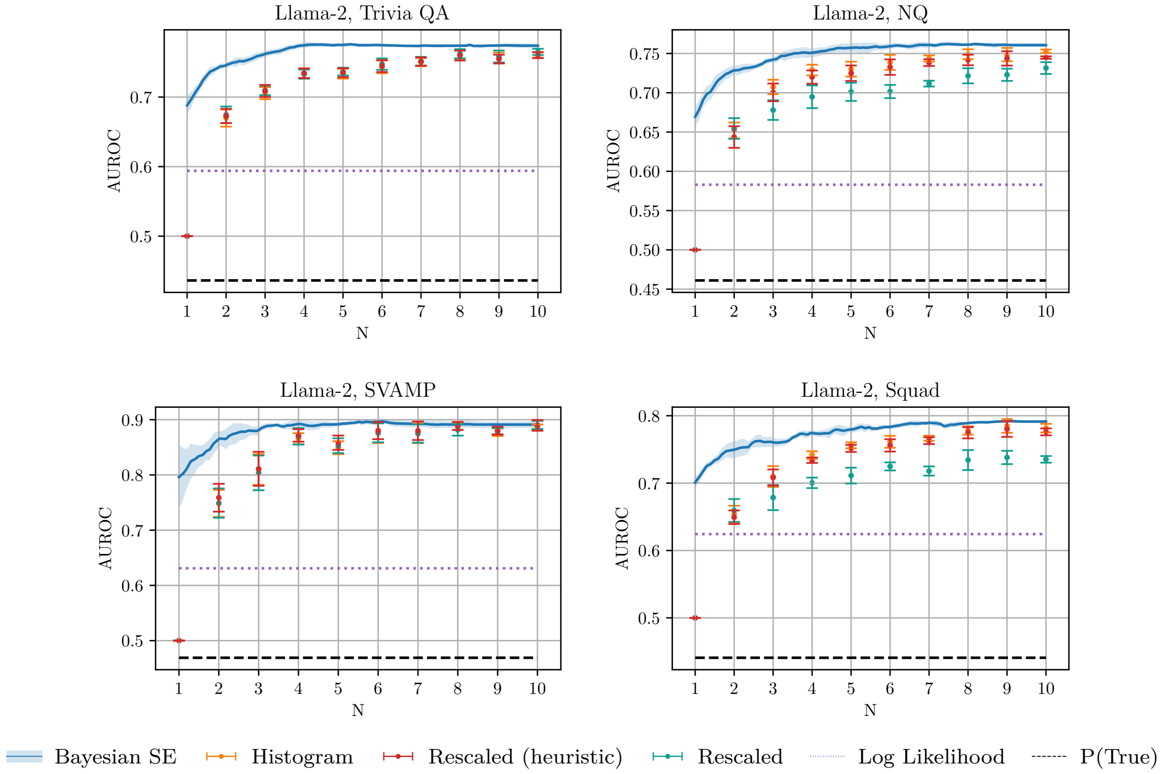

Figure 1: Results in the adaptive budget setting (Llama 2).

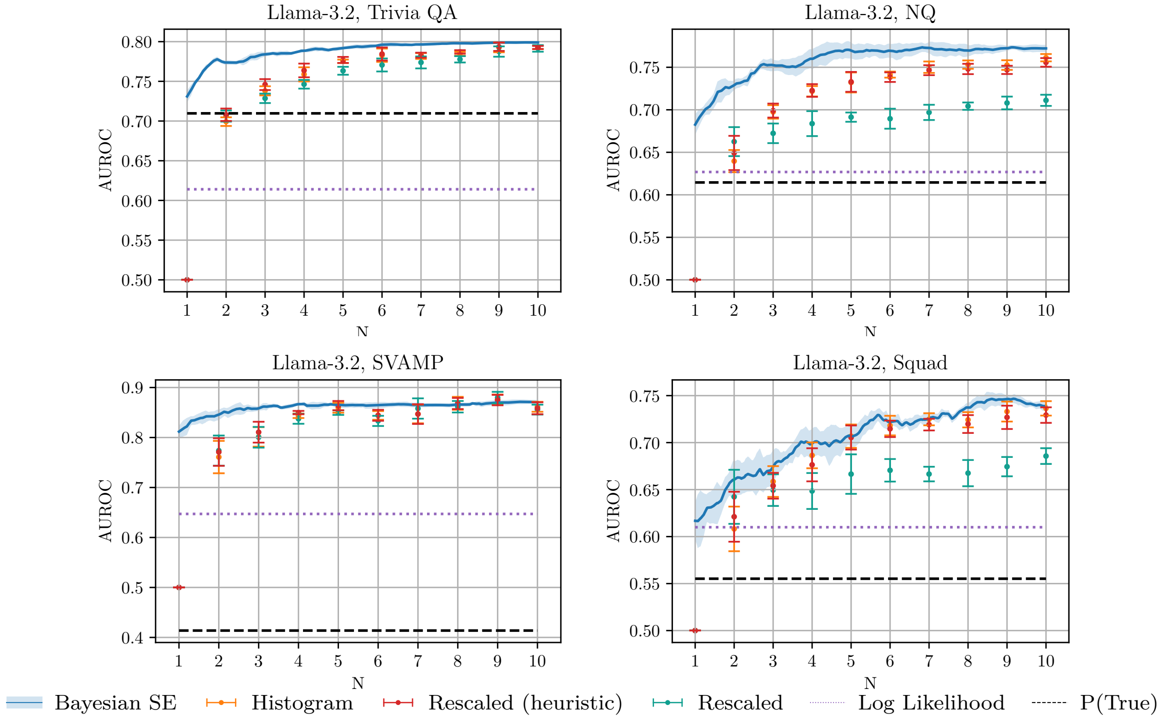

Figure 2: Results in the adaptive budget setting (Llama 3).

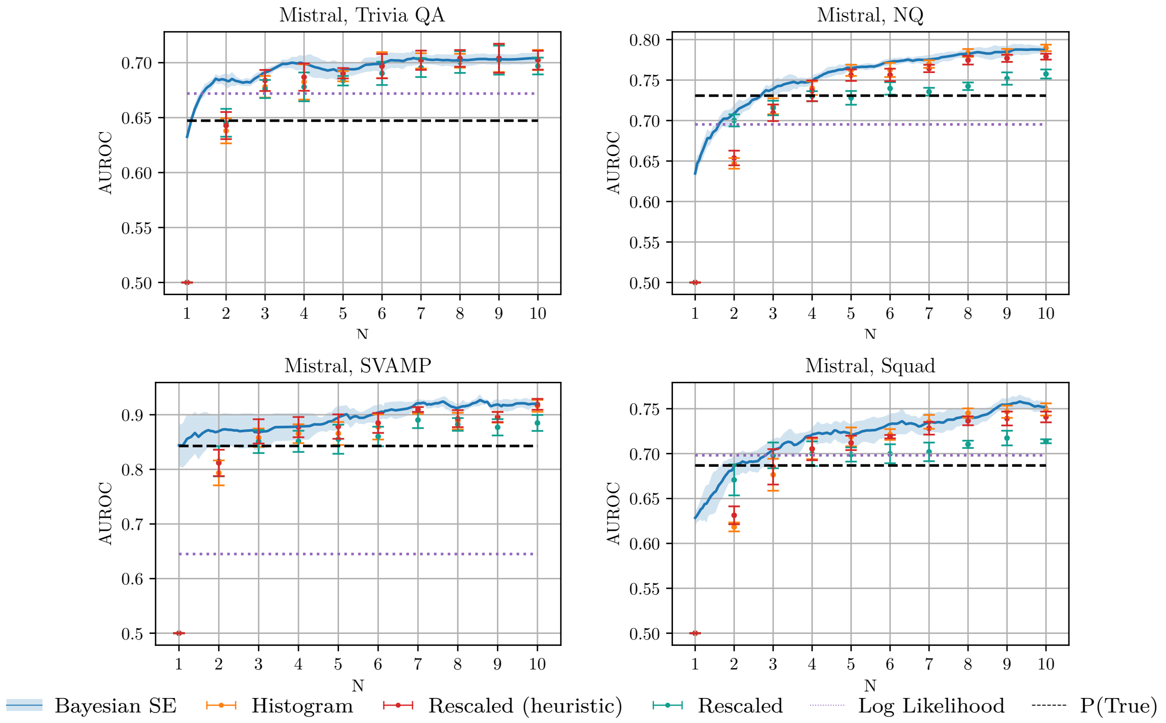

Figure 3: Results in the adaptive budget setting (Mistral).

7 Conclusions

We have described a new Bayesian estimator for measuring semantic entropy. The proposed estimator has systematically outperformed other semantic entropy baselines in several practical settings.

References

Yasin Abbasi Yadkori, Ilja Kuzborskij, András György, and Csaba Szepesvari. To believe or not to believe your llm: Iterative prompting for estimating epistemic uncertainty. *Advances in Neural Information Processing Systems*, 37:58077–58117, 2024.

Lukas Aichberger, Kajetan Schweighofer, and Sepp Hochreiter. Rethinking uncertainty estimation in natural language generation. *arXiv preprint arXiv:2412.15176*, 2024.

Evan Archer, Il Memming Park, and Jonathan W Pillow. Bayesian entropy estimation for countable discrete distributions. *The Journal of Machine Learning Research*, 15(1):2833–2868, 2014.

Narayanaswamy Balakrishnan and Valery B Nevzorov. *A primer on statistical distributions*. John Wiley & Sons, 2004.

Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. INSIDE: LLMs' internal states retain the power of hallucination detection. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=Zj12nzlQbz`.

William G. Cochran. Sampling techniques. *Proceedings of the Edinburgh Mathematical Society*, 13(4):342–343, 1963.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630, 2024.

Katja Filippova. Controlled hallucinations: Learning to generate faithfully from noisy data. *arXiv preprint arXiv:2010.05873*, 2020.

Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In *International conference on machine learning*, pp. 1183–1192. PMLR, 2017.

Yarin Gal et al. *Uncertainty in deep learning*. PhD thesis, University of Cambridge, 2016.

Jean Hausser and Korbinian Strimmer. Entropy inference and the james-stein estimator, with application to nonlinear gene association networks. *Journal of Machine Learning Research*, 10(7), 2009.

Till Hoffmann. Moments of the dirichlet distribution. `https://web.archive.org/web/20160214015422/https://tillahoffmann.github.io/Moments-of-the-Dirichlet-distribution/`, 2015. Accessed: 2025-04-28.

Paul Hofman, Yusuf Sale, and Eyke Hüllermeier. Quantifying aleatoric and epistemic uncertainty with proper scoring rules. *arXiv preprint arXiv:2404.12215*, 2024.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12), March 2023. ISSN 0360-0300. doi: 10.1145/3571730. URL `https://doi.org/10.1145/3571730`.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. triviaqa: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. *arXiv e-prints*, art. arXiv:1705.03551, 2017.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017.

Jannik Kossen, Jiatong Han, Muhammed Razzak, Lisa Schut, Shreshth Malik, and Yarin Gal. Semantic entropy probes: Robust and cheap hallucination detection in llms. *arXiv preprint arXiv:2406.15927*, 2024.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664*, 2023.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. Latent retrieval for weakly supervised open domain question answering. *arXiv preprint arXiv:1906.00300*, 2019.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. On faithfulness and factuality in abstractive summarization. *arXiv preprint arXiv:2005.00661*, 2020.

Alexander Nikitin, Jannik Kossen, Yarin Gal, and Pekka Marttinen. Kernel language entropy: Fine-grained uncertainty quantification for llms from semantic similarities. *arXiv preprint arXiv:2405.20003*, 2024.

Art B Owen. Monte carlo theory, methods and examples, 2013.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. Are nlp models really able to solve simple math word problems? *arXiv preprint arXiv:2103.07191*, 2021.

Xin Qiu and Risto Miikkulainen. Semantic density: Uncertainty quantification for large language models through confidence measurement in semantic space. *arXiv preprint arXiv:2405.13845*, 2024.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In Jian Su, Kevin Duh, and Xavier Carreras (eds.), *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1264. URL `https://aclanthology.org/D16-1264`.

Mark Steyvers, Heliodoro Tejeda, Aakriti Kumar, Catarina Belem, Sheer Karny, Xinyue Hu, Lukas W Mayer, and Padhraic Smyth. What large language models know and what people think they know. *Nature Machine Intelligence*, pp. 1–11, 2025.

Adith Swaminathan and Thorsten Joachims. The self-normalized estimator for counterfactual learning. *advances in neural information processing systems*, 28, 2015.

David H Wolpert and David R Wolf. Estimating functions of probability distributions from a finite set of samples, part 1: Bayes estimators and the shannon entropy. *arXiv preprint comp-gas/9403001*, 1994.