On the Role of Unobserved Sequences on Sample-based Uncertainty Quantification for LLMs

Lucie Kunitomo-Jacquin[1] **and Edison Marrese-Taylor**[1,2] **and Ken Fukuda**[1]
[1] National Institute of Advanced Industrial Science and Technology (AIST), Tokyo, Japan
[2] Graduate School of Engineering, The University of Tokyo, Tokyo, Japan
kunitomo-jacquin.lucie@aist.go.jp edison.marrese@aist.go.jp
emarrese@weblab.t.u-tokyo.ac.jp ken.fukuda@aist.go.jp

Abstract

Quantifying uncertainty in large language models (LLMs) is important for safety-critical applications because it helps spot incorrect answers, known as hallucinations. One major trend of uncertainty quantification methods is based on estimating the entropy of the distribution of the LLM's potential output sequences. This estimation is based on a set of output sequences and associated probabilities obtained by querying the LLM several times. In this paper, we advocate and experimentally show that the probability of unobserved sequences plays a crucial role, and we recommend future research to integrate it to enhance such LLM uncertainty quantification methods.

1 Introduction

The advent of large language models (LLMs) has revolutionized numerous fields by demonstrating remarkable capabilities across a diverse array of tasks. However, despite their impressive performance, these models often struggle with reliability issues, particularly due to factual inaccuracies in their outputs. In this context, quantifying their confidence and adjusting them for various tasks can reduce risks and enhance the quality of outputs.

However, uncertainty quantification (UQ) on LLMs remains challenging since the output possibilities for these models are substantially greater than those of discriminative models. As the generation length increases, the number of potential outcomes grows exponentially, making it unfeasible to evaluate all possible answers (Geng et al., 2024). We can distinguish two types of uncertainty in LLMs: aleatoric uncertainty, stemming from inherent randomness, and epistemic uncertainty, resulting from a lack of information (Kendall and Gal, 2017). Following previous work, we aim to quantify a measure of total uncertainty, i.e., aleatory and/or epistemic, as both types of uncertainty contribute to model errors.

Among the methods of uncertainty quantification for LLMs, we identify black-box methods, which assume access only to the generations, and white-box methods, which also utilize internal states of the LLM or token-level probabilities. In this paper, we focus on the latter, utilizing token-level probabilities. Concretely, we study sampling-based estimation methods, that is, approaches that rely on information (e.g. probability) obtained from multiple answers generated by the LLM, in order to quantify uncertainty.

Sample-based uncertainty quantification methods via entropy estimation, like Predictive Entropy (E) (Malinin and Gales, 2020) and the recently proposed Semantic entropy (SE) (Kuhn et al., 2023; Farquhar et al., 2024), have succeeded recently perhaps due to their simplicity, as they do not require any special training or architectural modifications. However, we note that these methods are themselves subject to epistemic uncertainty, as they rely on only a glimpse of the probability distribution of possible answers due to practical constraints. We highlight that methods like E and SE, in particular, do not account for this epistemic uncertainty, as they only consider the estimated probability of sampled sequences, neglecting the remaining probability of possible but unobserved answers.

Recent work by Abbasi Yadkori et al. (2024) has moved in a similar direction and explored the concept of missing mass in UQ. However, their approach directly compares the distributions of the generated answers against the ground truth. Instead, here we present work focusing on modeling the probability of unobserved answers without the need for ground truth. Concretely, our aim is to propose a framework that enables us to incorporate this probability into existing formulations for estimation based on entropy. We provide technical considerations for the calculation of such probability and evaluate the relevance of one such implementation by using it as a UQ method.

Proceedings of the 2nd Workshop on Uncertainty-Aware NLP (UncertaiNLP 2025), pages 179–183
November 9, 2025 ©2025 Association for Computational Linguistics

2 Proposed Approach

Let us denote by x the object about which we quantify uncertainty; in our case study, x refers to the input given to the LLM which often consists on a question and potentially a prompt. We denote by $(\mathcal{S}, \mathbb{P})$ the probability space, where $\mathcal{S}$ is the set of all possible sequences, and $\mathbb{P}$ is the probability measure over $\mathcal{S}$. The entropy for the random output sequence of the LLM and the input x is defined as Equation 1 shows, below, where $p(s|x)$ is the probability of the sequence s conditioned on the input x.

$$E^*(x) = -\sum_{s \in \mathcal{S}} p(s|x) \log p(s|x). \quad (1)$$

As it is not realistic to compute the probability of all answers in $\mathcal{S}$, entropy-based UQ methods for LLMs estimate $E^*(x)$ on a set of M sequences sampled from the model denoted $s_1, \ldots, s_M$. Let us denote $A \subset \mathcal{S}$ the set of unique sampled answers and note that $|A| \leq M$ because some identical answers might be sampled multiple times. Each answer $s \in A$ consists of a sequence of length N in the set of vocabulary tokens $\mathcal{T}$. The probability of $s = (t_1, \ldots t_N)$ is obtained by the product of conditional token probabilities via the language model, as follows.

$$p(s|x) = \prod_i p(t_i|t_{<i}, x). \quad (2)$$

Some works have considered adjusting the calculation of sequence probabilities to account for varying sequence lengths. This is due to the tendency for longer sequences to exhibit lower joint likelihoods. To address this, a length normalized probability, which we denote p' was proposed (Malinin and Gales, 2020) as follows.

$$\log p'(s|x) = \frac{1}{N} \sum_i \log p(t_i|t_{<i}, x). \quad (3)$$

We now focus on the probability of sequences not observed in the set A of sequences provided by the LLM for a given input x. This probability is given by

$$\mathbb{P}(\bar{A}|x) = 1 - \mathbb{P}(A|x) \quad (4)$$

$$= 1 - \sum_{s \in A} p(s|x), \quad (5)$$

where $\bar{A}$ denotes the complement set of A.

We believe that the probability of unobserved sequences can capture some of the uncertainty associated with an input x. When uncertainty is low, the model's output probabilities tend to be higher, leading to a lower probability for the unobserved sequences. Conversely, when uncertainty is high, the model's output probabilities are lower, resulting in a higher probability for the unobserved sequences. In case of maximum uncertainty, all sequences in $\mathcal{S}$ are equally likely, with each having a probability of $1/|\mathcal{S}|$. As a result, $\mathbb{P}(\bar{A}|x) = 1 - M/|\mathcal{S}|$ approaches 1, especially when the set of possible sequences is very large. Conversely, in situations of minimal uncertainty, $\mathbb{P}(\bar{A}|x) = 0$.

In practice, we have two technical concerns related to the accurate calculation of probabilities for unobserved answers. Firstly, to the best of our knowledge, it is not always clear whether the last token, specifically the end-of-sequence (EOS) token, is considered in sequence probability calculations presented in Equation 2. If sequences do not include the EOS token, this raises concerns about the construction of the sample space, as two unfinished sequences are not mutually exclusive. Let us introduce a small example to illustrate our discussion about the sequence probability calculation.

Example. For the question input $x =$"*Where are St. Peter's Basilica and the Sistine Chapel?*", let us assume we observed two output sequences such that $A = \{$"*vatican*", "*vatican city*"$\}$ and consider the token conditional probabilities presented in Figure 1. If we do not include the end-of-sequence token, the probability value of 0.8 may be incorrectly interpreted as the probability of the sequence "*vatican*". In fact, this represents the probability that the sequence starts with "*vatican*", which also includes the possibility of the sequence being "*vatican city*". Essentially, the events of the sequence beginning with "*vatican*" and "*vatican city*" are not mutually exclusive.

In addition to this issue, we also note that sequence length normalization techniques as shown in Equation 3, and often used approaches like **SE**, can distort probabilities, potentially leading to the sum of output probabilities differing from 1.

Due to the issues discussed above, we highlight that we cannot properly estimate the probability of unobserved answers with the usually-adopted sequence probability calculations. Thus, we compute the probability of sequences without sequence length normalization and considering the EOS to-

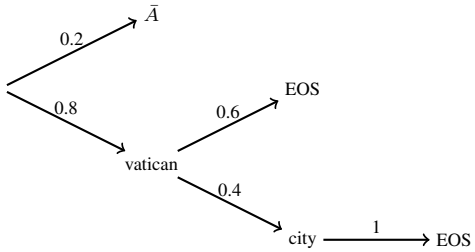

Figure 1: Example of tree of possible sequences with token conditional probabilities.

ken. Formally, we consider the probability of a sequence $s = (t_1, \ldots, t_N, \text{EOS})$ as

$$p(s|x) = \prod_i p(t_i|t_{<i}, x) \times p(\text{EOS}|t_{\leq N}, x). \quad (6)$$

Example (revisited). Looking back at our previous example, if we consider the EOS token in the computation of the probability, we obtain $p(\text{“}vatican\text{”}|x) = 0.8 \times 0.6 = 0.48$, $p(\text{“}vatican\ city\text{”}|x) = 0.8 \times 0.4 \times 1 = 0.32$ and the probability of the unobserved samples is $\mathbb{P}(\bar{A}|x) = 1 - 0.48 - 0.32 = 0.2$.

Based on this framework, here we present an alternative method for computing the uncertainty of an LLM where we directly use the value $\mathbb{P}(\bar{A}|x)$. We note that this approach, which we call *Unobserved Probability* (UP), is arguably a very simple way to capture some part of the LLM uncertainty, as derived from our analysis.

- EOS-Inclusive UP (EOS-UP): this approach consist of quantifying the LLM uncertainty using $\mathbb{P}(\bar{A}|x)$ in the way we consider most suitable or recommended, i.e., accounting for the EOS token in calculating the sequence probabilities as in Equation 6.

- Length-Normalized UP (LN-UP): we propose to quantify the LLM uncertainty using $\mathbb{P}(\bar{A}|x)$ as above, but considering the usual way for calculating the sequence probabilities, i.e., without accounting for EOS token and performing sequence length normalization, following Equation 3.

3 Experiments and Results

In this section, we detail our experimental setup to evaluate the relevance of using the probability of unobserved answers for LLM uncertainty quantification via our proposed approach UP. We compare its performance with three entropy-based methods and also include, for reference, the probability of unobserved answers calculated using the conventional method for sequence probabilities.

Model and dataset. Our experiments focused on the uncertainty quantification for the *falcon-40b-instruct* model (Almazrouei et al., 2023) and were performed on a general knowledge dataset, TriviaQA (Joshi et al., 2017). This model and dataset were recently used by Nikitin et al. (2024). TriviaQA was also originally used by Kuhn et al. (2023) for their seminal work on SE.

Sampling. We conducted our sampling using two styles of prompts. On the one hand, we adopt a prompt that pushes the model to produce short answers (SHORT), “*Answer the following question as briefly as possible*”. This prompt was used on a more recent implementation of SE, presented by Farquhar et al. (2024).[1] On the other hand, we also experiment with the original prompt (NORMAL) presented by Kuhn et al. (2023), and was also considered by Nikitin et al. (2024), “*Answer the following question in a single brief but complete sentence.*”. Following the methodology of previous studies (Farquhar et al., 2024; Nikitin et al., 2024), we employed top-K sampling with $K = 50$ and nucleus sampling with $p = 0.9$ at a temperature of $T = 1$.

Evaluation Metric. In line with previous works (Farquhar et al., 2024), we evaluated the model’s accuracy by sampling an additional answer at a lower temperature ($T = 0.1$). Then we used another LLM, *Meta-Llama-3-8B-Instruct* (AI@Meta, 2024), to compare this answer with the ground truth answers from the datasets. The prompts for checking answers correctness are provided in the appendix. We evaluate uncertainty quantification methods by measuring their ability in predicting model output accuracy using the Area under the Receiver Operating Curve (AUROC).

UQ methods. We considered the following baseline methods in our experiments.

- Predictive Entropy (E) (Malinin and Gales, 2020; Kuhn et al., 2023) is a Monte-Carlo estimation of predictive entropy, shown by Equation 7, below. As per the original implementation, this uses sentence length normal-

[1] https://github.com/jlko/semantic_uncertainty

ization as in Equation 3.

$$E(x) \approx -\frac{1}{M}\sum_{m=1}^{M}\log p'(s_m|x) \quad (7)$$

- Semantic Entropy (SE) (Kuhn et al., 2023; Farquhar et al., 2024) is defined on a set of clusters capturing the distinct meaning, denoted $\mathcal{C}$. This consists in a sub-σ-algebra of the event-space of all possible answers $\mathcal{S}$. The uncertainty quantification is calculated by an approximation of the semantic entropy involving the normalization of the cluster probabilities (Farquhar et al., 2024), as shown in Equation 8 and Equation 10, below, where $C \in \mathcal{C}$.

$$p'(C|x) = \sum_{s\in C} p'(s|x) \quad (8)$$

$$p''(C|x) = \frac{p'(C|x)}{\sum_{C\in\mathcal{C}} p'(C|x)} \quad (9)$$

$$SE(x) \approx -\sum_{c\in\mathcal{C}} p''(C|x)\log p''(C|x) \quad (10)$$

- Discrete Semantic Entropy (DSE) (Kuhn et al., 2023; Farquhar et al., 2024) consists in a variant of SE where cluster probabilities are approximated by $p(C|x) \approx |\{s : s\in C\}|/M$.

The results in terms of AUROC are presented in Figure 2. We observe that the probability of unobserved answers EOS-UP is indeed relevant for quantifying uncertainty, achieving performance comparable to the Predictive Entropy (E) method.

Moreover, we note that while state-of-the-art baselines (E, SE, and DSE) are affected by the number of available samples, the probability of unobserved answers maintains its performance even with a single sample. Sampling more answers from the LLM can generally lead to larger answer variability, and hence as M grows, the effect of the probability of unobserved answers on the estimation decreases. Therefore, our results suggest that incorporating the probability of unobserved samples in the estimation of uncertainty can be of critical importance when the number of samples is limited (e.g. $M = 1$). Note that when $M = 1$, $A = \{s_1\}$, E method reduces to $-\log p'(s_1|x)$, and LN-UP method to $1 - p'(s_1|x)$. Since these quantity are strictly decreasing and monotonic with respect to $p'(s_1|x)$, they yield the same ranking over input instances and thus the same AUROC performance, as shown in Figure 2.

Finally, we observe the poor performance of our proposed probability of unobserved answers, considering length-normalization and no EOS token

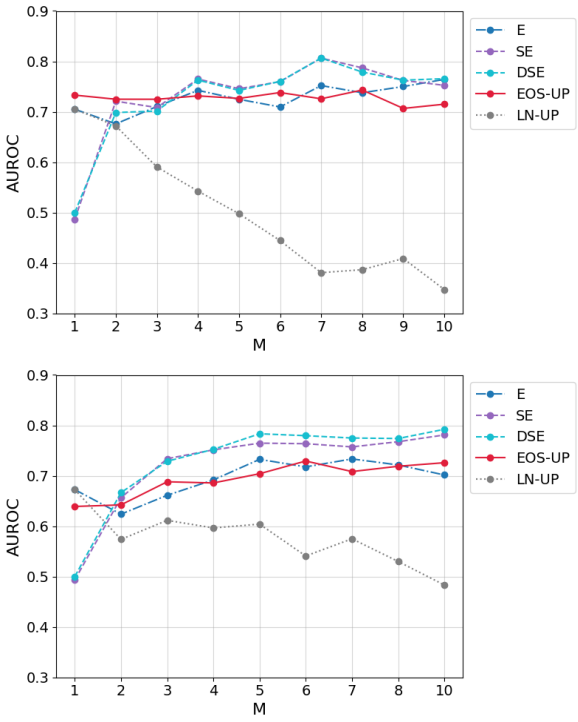

Figure 2: Influence of the number of samples (M) for the LLM uncertainty quantification in terms of AUROC, for the SHORT (top) and NORMAL (bottom) answer length scenarios. We compare the performance of our proposed approach variations (UP) against relevant baselines. Results were computed on 500 pairs of questions and ground truth answers on the *falcon-40b-instruct* model.

probability (LN-UP), not only remains the worst performing method for all M values, but also that its performance decreases dramatically as M grows. We think that, as shown by our technical considerations, our suggested way to compute this probability (EOS-UP) is necessary to obtain an adequate estimation.

4 Conclusion

In this work, we aimed to focus on the probability of unobserved answers, which we note have been overlooked by existing entropy-based LLM UQ methods. We acknowledge that this probability captures only a portion of the uncertainty. For instance, hesitation between observed answers is not considered since the probability of each separate observed answer is not used.

Our empirical results are encouraging and in the future we plan to integrate this quantity into existing entropy estimation methods. To achieve this, we believe a theoretical framework that considers both aleatoric and epistemic uncertainty, such

as the Evidence Theory (Shafer, 1976; Smets and Kennes, 1994) would be suitable.

We also note that current approaches of entropy-based UQ, present other issues and limitations. Although the work of (Nikitin et al., 2024) has made progress in this regard, we think further improvements are necessary, for example, by more directly modeling hypernymy and hyponymy relationships across answers, and/or clusters of answers.

Acknowledgement

This paper is based on results obtained from a project, JPNP25006, commissioned by the New Energy and Industrial Technology Development Organization (NEDO).

References

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

AI@Meta. 2024. Llama 3 model card.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, and 1 others. 2023. The falcon series of open language models. *arXiv preprint arXiv:2311.16867*.

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

Jiahui Geng, Fengyu Cai, Yuxia Wang, Heinz Koeppl, Preslav Nakov, and Iryna Gurevych. 2024. A survey of confidence estimation and calibration in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6577–6595, Mexico City, Mexico. Association for Computational Linguistics.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Alex Kendall and Yarin Gal. 2017. What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

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

Andrey Malinin and Mark Gales. 2020. Uncertainty estimation in autoregressive structured prediction. *arXiv preprint arXiv:2002.07650*.

Alexander Nikitin, Jannik Kossen, Yarin Gal, and Pekka Marttinen. 2024. Kernel language entropy: Fine-grained uncertainty quantification for llms from semantic similarities. *Advances in Neural Information Processing Systems*, 37:8901–8929.

Glenn Shafer. 1976. *A mathematical theory of evidence*, volume 42. Princeton university press.

Philippe Smets and Robert Kennes. 1994. The transferable belief model. *Artificial intelligence*, 66(2):191–234.

Appendix

To check the correctness of the answers, we used the same prompts as previous studies presented in Figure 3.

Prompt (single answer)

*We are assessing the quality of answers to the following question: {**question**} \n The expected answer is: {**correct_answer**}. \n The proposed answer is: {**predicted_answer**} \n Within the context of the question, does the proposed answer mean the same as the expected answer? \n Respond only with yes or no.\n Response:*

Prompt (multiple answers)

*We are assessing the quality of answers to the following question: {**question**} \n The following are expected answers to this question: {**correct_answers**}. \n The proposed answer is: {predicted_answer} \n Within the context of the question, does the proposed answer mean the same as any of the expected answers? \n Respond only with yes or no.\n Response:*

Figure 3: Prompts fed to the model in our experiments when providing a single (top) and many correct answers (bottom), where **placeholders** are denoted in bold.