RETHINKING UNCERTAINTY ESTIMATION IN NATURAL LANGUAGE GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) are increasingly employed in real-world applications, driving a need to determine when their generated text can be trusted or should be questioned. To assess the trustworthiness of the generated text, reliable uncertainty estimation is essential. Current LLMs generate text through a stochastic process that can lead to different output sequences for the same prompt. Consequently, leading uncertainty measures require generating multiple output sequences to estimate the LLM's uncertainty. However, generating additional output sequences is computationally expensive, making these uncertainty estimates impractical at scale. In this work, we challenge the theoretical foundations of the leading measures and derive an alternative measure that eliminates the need for generating multiple output sequences. Our new measure is based solely on the negative log-likelihood of the most likely output sequence. This vastly simplifies uncertainty estimation while maintaining theoretical rigor. Empirical results demonstrate that our new measure achieves state-of-the-art performance across various models and tasks. Our work lays the foundation for reliable and efficient uncertainty estimation in LLMs, challenging the necessity of the more complicated methods currently leading the field.

1 INTRODUCTION

Language models are increasingly adopted in a wide range of real-world applications. Despite the advancements in language models, determining whether a generated text can be trusted remains a significant challenge. To address this challenge, it is crucial to reliably assess the level of certainty a language model has regarding its generated text. While uncertainty estimates do not guarantee factuality for generated text based on consistent but erroneous training data, they are a reliable indicator of errors at present (Kuhn et al., 2023; Aichberger et al., 2024; Farquhar et al., 2024).

Assessing predictive uncertainty in language models is inherently difficult due to their autoregressive nature. For a given input sequence, language models predict the next token probabilities, based on which a specific token is selected and appended to the sequence. This stochastic process is repeated for each new token. Selecting different tokens at specific steps during the generation leads to varying output sequences for the same input sequence with the same language model. Consequently, the space of possible output sequences is vast and computationally intractable to fully explore (Sutskever et al., 2014; Vaswani et al., 2017; Radford et al., 2018).

Current uncertainty estimation methods rely on assessing the probability distribution over all possible output sequences. However, the generation of each additional token is computationally expensive, and practical methods can only sample a small fraction of possible output sequences (Malinin & Gales, 2021; Kadavath et al., 2022). Moreover, even after having generated multiple likely output sequences, the question remains whether these indicate high uncertainty. A language model that likely generates different output sequences is not necessarily uncertain about the underlying meaning if the output sequences are semantically equivalent. Leading uncertainty measures address this fact by considering the semantics of the output sequences, utilizing separate language inference models (Kuhn et al., 2023; Farquhar et al., 2024). While these measures improve the performance of the uncertainty estimates, they also further add complexity and computational overhead. These factors make current uncertainty estimation methods impractical at scale, hindering their broad adoption in real-world applications. There is a need for efficient uncertainty estimation methods that give clear insights into the reliability of language models without incurring substantial computational costs.

In this work, we assess whether we can theoretically motivate an uncertainty measure that does not rely on the probability distribution over all possible output sequences. Building on insights from the principled framework of proper scoring rules (Kotelevskii & Panov, 2024; Hofman et al., 2024), we adopt the zero-one score as an alternative to the currently used logarithmic score for uncertainty measures in NLG. This leads to a theoretically motivated measure that does not require generating multiple output sequences but solely relies upon a single output sequence. Our proposed measure is straightforward: it simply is the negative log-likelihood of the most likely output sequence. By eliminating the need to generate and semantically cluster multiple output sequences, our measure significantly reduces computational costs and complexity.

Experimental results demonstrate that our new measure matches or even exceeds the performance of current state-of-the-art uncertainty estimation methods across various model classes, model sizes, model stages, tasks, datasets, and evaluation metrics. In summary, our new measure not only preserves theoretical rigor but also provides a more scalable solution for uncertainty estimation in language models, making it highly practical for real-world applications.

Our main contributions are:

- We introduce the negative log-likelihood of the most likely output sequence as an efficient and practical measure of uncertainty in NLG.
- We provide a rigorous theoretical foundation for our measure, building upon established principles in uncertainty theory and proper scoring rules.
- We conduct extensive experiments demonstrating that our measure achieves strong performance, matching or surpassing state-of-the-art methods while significantly reducing computational costs.

2 Predictive Uncertainty in NLG

Preliminaries. We assume a fixed training dataset $\mathcal{D} = \{\boldsymbol{s}_i\}_{i=1}^N$ consisting of ordered tokens $s_t \in \mathcal{V}$, with $\mathcal{V}$ being a given vocabulary. Each token at step t is assumed to be sampled according to the predictive distribution $p(s_t \mid \boldsymbol{s}_{<t}, \boldsymbol{w}^*)$, conditioned on the sequence of preceding tokens $\boldsymbol{s}_{<t}$ and the true (but unknown) language model parameters $\boldsymbol{w}^*$. We assume that the given model class can theoretically represent the true predictive distribution, a common and usually necessary assumption (Hüllermeier & Waegeman, 2021). How likely language model parameters $\tilde{\boldsymbol{w}}$ match $\boldsymbol{w}^*$ is determined by the posterior distribution $p(\tilde{\boldsymbol{w}} \mid \mathcal{D}) = p(\mathcal{D} \mid \tilde{\boldsymbol{w}})p(\tilde{\boldsymbol{w}})/p(\mathcal{D})$.

In language model inference, the input to a given language model parameterized by $\boldsymbol{w}$ is a sequence $\boldsymbol{x} = (x_1, ..., x_M)$ and the output is a sequence $\boldsymbol{y} = (y_1, ..., y_T) \in \mathcal{Y}_T$, with $x, y \in \mathcal{V}$ and $\mathcal{Y}_T$ being the set of all possible output sequences with a sequence length smaller equal to T. The likelihood of a token $y_t \in \boldsymbol{y}$ being generated by the language model is conditioned on both the input sequence and all previously generated tokens, denoted as $p(y_t \mid \boldsymbol{x}, \boldsymbol{y}_{<t}, \boldsymbol{w})$. The likelihood of output sequences $\boldsymbol{y} \in \mathcal{Y}_T$ being generated by the language model is then the product of the individual token probabilities, denoted as $p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w}) = \prod_{t=1}^T p(y_t \mid \boldsymbol{x}, \boldsymbol{y}_{<t}, \boldsymbol{w})$ (Sutskever et al., 2014), while the heuristic length-normalized variant is $\bar{p}(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w}) = \exp\left(\frac{1}{T}\sum_{t=1}^T \log p(y_t \mid \boldsymbol{x}, \boldsymbol{y}_{<t}, \boldsymbol{w})\right)$ (Malinin & Gales, 2021).

Computing the likelihood of a specific output sequence $\boldsymbol{y}$ being generated by the language model parameterized by $\boldsymbol{w}$ – or in other words, being sampled from the probability distribution over possible output sequences $\boldsymbol{y} \sim p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w})$ – is straightforward. The language model directly provides the individual token likelihoods. However, determining the full probability distribution over possible output sequences is considerably more challenging, since $\mathcal{Y}_T$ scales exponentially with the sequence length T. The computational complexity of evaluating all possible sequences grows as $\mathcal{O}(|\mathcal{V}|^T)$. Since modern language models even exceed a vocabulary size $|\mathcal{V}|$ of one hundred thousand tokens, this distribution becomes intractable to compute, even for relatively short maximal sequence lengths T (Dubey et al., 2024).

Uncertainty Measures and Proper Scoring Rules. We now derive measures to estimate uncertainty in NLG. Throughout this work, the focus is on estimating the predictive uncertainty of a single, given "off-the-shelf" model. We assume to that a given language model parameterized by $\boldsymbol{w}$ is the *predicting model* used to sample output sequences $\boldsymbol{y} \sim p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w})$. Furthermore, we assume that any language model parameterized by $\tilde{\boldsymbol{w}}$ is an *approximation of the true predictive distribution* according to its posterior probability $p(\tilde{\boldsymbol{w}} \mid \mathcal{D})$. Together, these two assumptions give rise to specific uncertainty measures (Schweighofer et al., 2023; 2024), as elaborated on in more detail below. Aichberger et al. (2024) shows that established uncertainty measures in NLG, such as Predictive Entropy (PE) (Malinin & Gales, 2021) and Semantic Entropy (SE) (Kuhn et al., 2023; Farquhar et al., 2024), naturally emerge under this assumption. In general, the information-theoretic entropy has become the standard measure to assess predictive uncertainty. However, recent studies by Lahlou et al. (2023); Gruber & Buettner (2023); Kotelevskii & Panov (2024) and Hofman et al. (2024) have shown that these information-theoretic measures are not the only viable options. A broader class of *proper scoring rules* provides a principled framework for predictive uncertainty measures. In the following, we leverage this framework to derive our alternative measure that relies solely on a single output sequence. We begin by discussing the concept of proper scoring rules.

In general, proper scoring rules are a class of functions that evaluate the quality of probabilistic predictions by assigning a numerical score based on the predictive distribution and the actual observations (Gneiting & Raftery, 2007). For uncertainty estimation in NLG, the general notion of proper scoring rules assigns a numerical score to how well the predicted distribution of output sequences $p(\boldsymbol{y} \mid \boldsymbol{x}, \cdot)$ aligns with the observed output sequence $\boldsymbol{y}'$. In particular, a proper scoring rule is an extended real-valued function $\mathrm{S} : \mathcal{P} \times \mathcal{Y} \rightarrow [-\infty, \infty]$, such that $\mathrm{S}(p, \cdot)$ is $\mathcal{P}$-quasi-integrable over a convex class of probability measures $\mathcal{P}$. The expected score over possible output sequences $\boldsymbol{y}'$ is given by

$$\mathrm{E}_{\boldsymbol{y}' \sim p(\boldsymbol{y}' \mid \boldsymbol{x}, \cdot)} \left[\mathrm{S} \left(p(\boldsymbol{y} \mid \boldsymbol{x}, \cdot), \boldsymbol{y}' \right) \right] \tag{1}$$

Given this general formulation, we now incorporate the assumptions outlined above to establish the connection to uncertainty measures (Schweighofer et al., 2024). First, under the assumption about the *predicting model*, the distribution giving rise to the observed output sequences $p(\boldsymbol{y}' \mid \boldsymbol{x}, \cdot)$ corresponds to the predictive distribution of the given language model, denoted as $p(\boldsymbol{y}' \mid \boldsymbol{x}, \boldsymbol{w})$. Second, under the assumption about the *approximation of the true predictive distribution*, a sampled output sequence $\boldsymbol{y}'$ has to be compared to all possible language models parameterized by $\tilde{\boldsymbol{w}}$, according to their posterior distribution $p(\tilde{\boldsymbol{w}} \mid \mathcal{D})$. This captures how much the sampled output sequence aligns with all possible predictive distributions $p(\boldsymbol{y} \mid \boldsymbol{x}, \tilde{\boldsymbol{w}})$. Therefore, we take a posterior expectation over Eq. (1), which can be additively decomposed into an entropy term and a divergence term (Gneiting & Raftery, 2007; Kull & Flach, 2015):

$$\begin{aligned}
&\underbrace{\mathrm{E}_{\tilde{\boldsymbol{w}} \sim p(\tilde{\boldsymbol{w}} \mid \mathcal{D})} \left[\mathrm{E}_{\boldsymbol{y}' \sim p(\boldsymbol{y}' \mid \boldsymbol{x}, \boldsymbol{w})} \left[\mathrm{S} \left(p(\boldsymbol{y} \mid \boldsymbol{x}, \tilde{\boldsymbol{w}}), \boldsymbol{y}' \right) \right] \right]}_{\text{expected score}} \\
&\quad = \underbrace{\mathrm{E}_{\boldsymbol{y}' \sim p(\boldsymbol{y}' \mid \boldsymbol{x}, \boldsymbol{w})} \left[\mathrm{S} \left(p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w}), \boldsymbol{y}' \right) \right]}_{\text{entropy term}} \\
&\qquad + \underbrace{\mathrm{E}_{\tilde{\boldsymbol{w}} \sim p(\tilde{\boldsymbol{w}} \mid \mathcal{D})} \left[\mathrm{E}_{\boldsymbol{y}' \sim p(\boldsymbol{y}' \mid \boldsymbol{x}, \boldsymbol{w})} \left[\mathrm{S} \left(p(\boldsymbol{y} \mid \boldsymbol{x}, \tilde{\boldsymbol{w}}), \boldsymbol{y}' \right) - \mathrm{S} \left(p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w}), \boldsymbol{y}' \right) \right] \right]}_{\text{divergence term}} .
\end{aligned} \tag{2}$$

Aleatoric and Epistemic Uncertainty. In terms of predictive uncertainty, this general framework can be interpreted as follows. The expected score over possible output sequences and language model parameters captures the *total* uncertainty of the given language model. The entropy term reflects *aleatoric* uncertainty, which is the uncertainty inherent in the data generation process, arising from the inherent variability and randomness in natural language (Gal, 2016; Kendall & Gal, 2017). The divergence term reflects *epistemic* uncertainty, which quantifies the uncertainty due to lack of knowledge about the true language model parameters, arising from limited data or model capacity (Houlsby et al., 2011; Gal, 2016; Malinin, 2019; Hüllermeier & Waegeman, 2021).

The concrete *total*, *aleatoric*, and *epistemic* uncertainty measures depends on the choice of proper scoring rule. For instance, the logarithmic score is the most common proper scoring rule that gives rise to the well-known information-theoretic uncertainty measures in both classification tasks (Houlsby et al., 2011; Gal, 2016) and NLG (Malinin & Gales, 2021; Kuhn et al., 2023).

In the following, we first revisit these uncertainty measures that are based on the logarithmic score and analyze their effectiveness in estimating aleatoric and epistemic uncertainty. Thereafter, we propose uncertainty measures that are based on another proper scoring rule, the zero-one score. This score has not yet been considered for uncertainty estimation in NLG. We show that utilizing uncertainty measures based on the zero-one score offers certain advantages.

2.1 Established Uncertainty Measures in NLG based on Logarithmic Score

The logarithmic score is usually assumed implicitly to derive uncertainty measures, due to the foundation of resulting measures in information theory (Lahlou et al., 2023; Gruber & Buettner, 2023; Hofman et al., 2024; Kotelevskii & Panov, 2024). In the context of NLG, it considers the negative log-likelihood of a generated output sequence $\boldsymbol{y}'$:

$$\mathrm{S}_{\log}\left(p(\boldsymbol{y} \mid \boldsymbol{x}, \cdot), \boldsymbol{y}'\right) = -\log p(\boldsymbol{y} = \boldsymbol{y}' \mid \boldsymbol{x}, \cdot) \ . \tag{3}$$

Using the logarithmic score in Eq. (2) results in the cross-entropy $\mathrm{CE}(\cdot\, ; \cdot)$ between the output sequence distribution of the given language model and that of every possible language model according to their posterior $p(\tilde{\boldsymbol{w}} \mid \mathcal{D})$ (Schweighofer et al., 2023; Aichberger et al., 2024):

$$\begin{aligned} &\underbrace{\mathrm{E}_{\tilde{\boldsymbol{w}} \sim p(\tilde{\boldsymbol{w}} \mid \mathcal{D})}\left[\mathrm{CE}(p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w}); p(\boldsymbol{y} \mid \boldsymbol{x}, \tilde{\boldsymbol{w}}))\right]}_{\text{total}} \\ &= \underbrace{\mathrm{H}(p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w}))}_{\text{aleatoric}} + \underbrace{\mathrm{E}_{\tilde{\boldsymbol{w}} \sim p(\tilde{\boldsymbol{w}} \mid \mathcal{D})}\left[\mathrm{KL}(p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w}) \,\|\, p(\boldsymbol{y} \mid \boldsymbol{x}, \tilde{\boldsymbol{w}}))\right]}_{\text{epistemic}} \ . \end{aligned} \tag{4}$$

The epistemic uncertainty is a posterior expectation of the Kullback-Leibler divergence $\mathrm{KL}(\cdot \,\|\, \cdot)$ between the output sequence distribution of the given model and that of all possible models. This requires considering every possible model parametrization. Since modern language models have billions of parameters (Radford et al., 2018; Zhang et al., 2022; Touvron et al., 2023; Zuo et al., 2024; Dubey et al., 2024), the epistemic uncertainty is particularly challenging to estimate.

Current work usually solely considers the aleatoric uncertainty, which is the Shannon entropy $\mathrm{H}(\cdot)$ of the output sequence distribution of the given language model (Malinin & Gales, 2021; Kuhn et al., 2023; Aichberger et al., 2024). Computing the output sequence distribution still requires considering the whole set of possible output sequences $\mathcal{Y}_T$. Thus, the primary objective of uncertainty estimation based on the logarithmic score is to closely approximate this output sequence distribution.

Predictive Entropy. The aleatoric uncertainty under a given language model is the entropy of the output sequence distribution, commonly referred to as Predictive Entropy (PE). Intuitively, high PE implies that the language model is likely to generate different output sequences from the same input sequence, indicating high uncertainty of the language model. PE usually is estimated via Monte Carlo (MC) sampling (Malinin & Gales, 2021):

$$\begin{aligned} \mathrm{H}(p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w})) \;&=\; \mathrm{E}_{\boldsymbol{y} \sim p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w})}\left[-\log p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w})\right] \\ &\approx\; \frac{1}{N} \sum_{n=1}^{N} -\log p(\boldsymbol{y}^n \mid \boldsymbol{x}, \boldsymbol{w}) \ , \qquad \boldsymbol{y}^n \sim p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w}) \ . \end{aligned} \tag{5}$$

Semantic Entropy. Semantic Entropy (SE) builds on the fact that output sequences may be different on a token level but equivalent on a semantics level. In such cases, the PE can be misleading, as it reflects high uncertainty even when different output sequences have the same semantic meaning. PE also captures the uncertainty of the language model in expressing the semantically same statement, which is often not the focus of uncertainty estimation in NLG. Thus, instead of the entropy of the output sequence distribution, the entropy of the semantic cluster distribution is considered, denoted as $p(c \mid \boldsymbol{x}, \boldsymbol{w}) = \sum_{\mathcal{Y}} p(c \mid \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{w})\, p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w})$. The probability of an output sequence belonging to a semantic cluster is usually approximated with a separate natural language inference model. High SE implies that the language model is likely to generate output sequences that have high semantic diversity, indicating high semantic uncertainty (Kuhn et al., 2023; Farquhar et al., 2024).

$$\begin{aligned} \mathrm{H}(p(c \mid \boldsymbol{x}, \boldsymbol{w})) \;&=\; \mathrm{E}_{c \sim p(c \mid \boldsymbol{x}, \boldsymbol{w})}\left[-\log p(c \mid \boldsymbol{x}, \boldsymbol{w})\right] \\ &\approx\; \frac{1}{N} \sum_{n=1}^{N} -\log p(c^n \mid \boldsymbol{x}, \boldsymbol{w}) \ , \qquad c^n \sim p(c \mid \boldsymbol{x}, \boldsymbol{w}) \ . \end{aligned} \tag{6}$$

Each of these uncertainty measures based on the logarithmic score considers the distribution over all possible output sequences $p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w})$, which is defined over the entire set of possible output sequences $\mathcal{Y}_T$. To approximate this distribution, it requires sampling output sequences from $\mathcal{Y}_T$. This requires generating multiple output sequences, which is computationally expensive. In the following, we eliminate this requirement by considering an alternative proper scoring rule.

2.2 New Uncertainty Measures in NLG based on Zero-One Score

Next, we introduce measures based on the zero-one score, which has not yet been considered as a proper scoring rule for deriving uncertainty measures in NLG. The zero-one score considers the predictive distribution for the most likely output sequence:

$$\mathrm{S}_{0\text{-}1}\left(p(\boldsymbol{y} \mid \boldsymbol{x}, \cdot), \boldsymbol{y}'\right) = \begin{cases} 1 - p(\boldsymbol{y} = \boldsymbol{y}' \mid \boldsymbol{x}, \cdot) & \text{if } \boldsymbol{y}' = \operatorname{argmax}_{\boldsymbol{y}} p(\boldsymbol{y} \mid \boldsymbol{x}, \cdot), \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

Using the zero-one score in Eq. (2) results in the total uncertainty being the expected confidence of the given language model about the most likely output sequences generated by all language models according to their posterior probability $p(\boldsymbol{w} \mid \mathcal{D})$:

$$\begin{aligned} &\underbrace{\mathrm{E}_{\tilde{\boldsymbol{w}} \sim p(\tilde{\boldsymbol{w}} \mid \mathcal{D})}\left[1 - p(\boldsymbol{y} = \tilde{\boldsymbol{y}}^* \mid \boldsymbol{x}, \boldsymbol{w})\right]}_{\text{total}} \\ &= \underbrace{1 - p(\boldsymbol{y} = \boldsymbol{y}^* \mid \boldsymbol{x}, \boldsymbol{w})}_{\text{aleatoric}} + \underbrace{p(\boldsymbol{y} = \boldsymbol{y}^* \mid \boldsymbol{x}, \boldsymbol{w}) - \mathrm{E}_{\tilde{\boldsymbol{w}} \sim p(\tilde{\boldsymbol{w}} \mid \mathcal{D})}\left[p(\boldsymbol{y} = \tilde{\boldsymbol{y}}^* \mid \boldsymbol{x}, \boldsymbol{w})\right]}_{\text{epistemic}}, \end{aligned} \tag{8}$$

with $\boldsymbol{y}^* = \operatorname{argmax}_{\boldsymbol{y}} p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w})$ and $\tilde{\boldsymbol{y}}^* = \operatorname{argmax}_{\boldsymbol{y}} p(\boldsymbol{y} \mid \boldsymbol{x}, \tilde{\boldsymbol{w}})$. Similar to Eq. (4), the epistemic uncertainty is a posterior expectation that remains challenging to estimate. However, we again focus on the aleatoric uncertainty, which solely considers the likelihood of the most likely output sequence under the given language model.

While aleatoric uncertainty derived from the logarithmic score requires approximating the entire output sequence distribution by sampling multiple sequences (as seen in Eq. (5) and Eq. (6)), the aleatoric uncertainty based on the zero-one score (see Eq. (8)) requires approximating the most likely output sequence under the given language model. This distinction is crucial, as approximating the most likely output sequence aligns directly with standard inference techniques widely used in language models, such as greedy decoding, beam search (Sutskever et al., 2014), top-k sampling, or nucleus sampling (Holtzman et al., 2020). For numerical stability, we consider the negative log-likelihood of the most likely output sequence that is proportional to the measure of aleatoric uncertainty in Eq. (8). We propose to estimate this quantity using the greedily decoded output sequence as an efficient and effective measure of aleatoric uncertainty:

$$\mathsf{NLL} := -\sum_{t=1}^{T} \log\left(\max_{y_t} p(y_t \mid \boldsymbol{x}, \boldsymbol{y}_{<t}, \boldsymbol{w})\right) \approx -\log p(\boldsymbol{y} = \boldsymbol{y}^* \mid \boldsymbol{x}, \boldsymbol{w}) \tag{9}$$

Discussion. Our proposed uncertainty measure challenges the prevailing reliance on multi-sequence sampling and semantic clustering for uncertainty estimation in NLG. By solely relying on the output sequences generated with greedy decoding, our approach significantly reduces computational overhead while maintaining theoretical rigor through its foundation in proper scoring rules. While uncertainty measures based on the logarithmic score could theoretically excel if the full distribution over output sequences $p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{w})$ were accessible – as in standard classification tasks – this distribution is intractable for NLG tasks due to their sequential nature. As a result, sampling-based methods often yield crude approximations, constrained by computational limits and sampling variability. In contrast, our uncertainty measure, based on the zero-one score, offers a more rigorous alternative while eliminating the need for extensive sampling. In Sec. 4, we demonstrate that using our measure of uncertainty yields performance that is superior to or at least on par with uncertainty measures based on the logarithmic score. This makes our method more practical for large-scale applications.

3 Related Work

In the previous section, we discussed uncertainty estimation methods based on the logarithmic score. Beyond these, there is a body of work that extends the concept of Semantic Entropy (Kuhn et al., 2023; Farquhar et al., 2024), for instance by either improving the semantic clustering (Nikitin et al., 2024; Qiu & Miikkulainen, 2024), improving the sampling of output sequences (Aichberger et al., 2024), or directly approximating the measure from hidden states of the language model (Kossen et al., 2024). Also, there is a body of work that builds upon the concept of Predictive Entropy (Malinin & Gales, 2021), for instance by considering a weighting factor for individual token and sequence likelihoods to account for the importance on a semantic level (Duan et al., 2023; Bakman et al., 2024).

There is also work on uncertainty estimation in NLG that is not grounded in proper scoring rules. For instance, several approaches leverage the language model itself to directly predict uncertainty, whether through numerical estimates or verbal explanations (Mielke et al., 2022; Lin et al., 2022; Kadavath et al., 2022; Cohen et al., 2023a; Ganguli et al., 2023; Ren et al., 2023; Tian et al., 2023). Cohen et al. (2023b) employ cross-examination, where one language model generates an output sequence and another model acts as an examiner to assess uncertainty. Zhou et al. (2023) explore the behavior of language models when expressing their uncertainty, providing insights into how models articulate confidence in their predictions. Also, Manakul et al. (2023) propose using sampled output sequences as input for another language model to assess uncertainty, offering a unique perspective on sequence evaluation. Additionally, Xiao et al. (2022) provide an empirical analysis of how factors such as model architecture and training data influence uncertainty estimates. Finally, conformal prediction (Quach et al., 2023) offers another approach by calibrating a stopping rule for output sequence generation, providing a statistical framework for uncertainty estimation.

4 Experiments

We aligned the evaluation of uncertainty estimation methods with related work by focusing on free-form question-answering tasks (Kuhn et al., 2023; Duan et al., 2023; Bakman et al., 2024; Nikitin et al., 2024; Aichberger et al., 2024; Kossen et al., 2024). While Farquhar et al. (2024) additionally concerns experiments with paragraph-length generations, their approach involves breaking down the entire paragraph into factual claims and reconstructing corresponding questions. Since the performance is expected to correlate with the performance on free-form question answering, we decided to focus specifically on free-form question answering tasks for a more direct assessment and less ambiguity in the evaluation.

Datasets. We evaluated uncertainty estimation methods on three different datasets. We used the over 3,000 test instances from *TriviaQA* (Joshi et al., 2017) concerning trivia questions, the over 300 test instances from *SVAMP* (Patel et al., 2021) concerning elementary-level math problems, and the over 3,600 test instances from *NQ-Open* (Lee et al., 2019) to assess natural questions aggregated from Google Search. Each dataset was utilized for two distinct tasks: (1) generating concise answers in the form of short phrases, and (2) producing more detailed answers in the form of full sentences (Farquhar et al., 2024). The resulting six tasks span a broad range of scenarios, ensuring a comprehensive evaluation of the uncertainty estimation methods.

Models. We conducted our evaluations on six distinct language models across different architectures, sizes, and training stages. Specifically, we used the Transformer-based model series *Llama-3.1* (Vaswani et al., 2017; Dubey et al., 2024) and the state-space model series *Falcon Mamba* (Gu & Dao, 2024; Zuo et al., 2024), representing two prominent language model paradigms. To assess the effect of training stage model scale on uncertainty estimation in NLG, we considered pre-trained (*PT*) and instruction-tuned (*IT*) language models with 7, 8, and 70 billion parameters, together covering a wide spectrum of model characteristics.

Baselines. We compare our method against the commonly used uncertainty measures based on the logarithmic score as of Eq. (5) and Eq. (6). These include Predictive Entropy (*PE*), length-normalized Predictive Entropy (*LN-PE*) (Malinin & Gales, 2021), Semantic Entropy (*SE*), length-normalized Semantic Entropy (*LN-SE*), and Discrete Semantic Entropy (*D-SE*) (Kuhn et al., 2023; Farquhar et al., 2024). For a given output sequence $\boldsymbol{y}'$, the length-normalized variants consider $\bar{p}(\boldsymbol{y}' \mid \boldsymbol{x}, \boldsymbol{w})$ instead of $p(\boldsymbol{y}' \mid \boldsymbol{x}, \boldsymbol{w})$ to compute the uncertainty estimates. The discrete variant of Semantic

Table 1: Average AUROC across TriviaQA, SVAMP and NQ datasets, using uncertainty estimates of different measures to distinguish between correct and incorrect answers. Varying model architectures (*transformer*, *state-space*), model sizes (*7B*, *8B*, *70B*), and model stages (*PT*, *IT*) are considered for generating answers. The reference answer is generated using *greedy decoding*, either as a whole sentence (*long*) or a short phrase (*short*). The reference answer's correctness is assessed by checking if the F1 score of the commonly used SQuAD metric exceeds 0.5 (*F1*) or if the LLM-as-a-judge considers it as correct (*LLM*). Predictive Entropy (*PE*), length-normalized Predictive Entropy (*LN-PE*), Semantic Entropy (*SE*), length-normalized Semantic Entropy (*LN-SE*), and discrete Semantic Entropy (*D-SE*) use 10 output sequences to assign an uncertainty estimate, each generated via multinomial sampling. NLL solely uses the reference answer to assign an uncertainty estimate.

Uncertainty measure based score					*Logarithmic*					*Zero-One*
Model			**Gen.**	**Metric**	**PE**	**LN-PE**	**SE**	**LN-SE**	**D-SE**	**NLL**
Transformer	**8B**	PT	short	F1	.776	.795	.775	.793	<u>.804</u>	**.824**
			short	LLM	.698	.714	.690	.706	<u>.719</u>	**.726**
			long	LLM	.562	.555	.545	.553	<u>.600</u>	**.649**
		IT	short	F1	.772	.801	.805	<u>.814</u>	.806	**.838**
			short	LLM	.676	.697	.704	<u>.709</u>	.694	**.722**
			long	LLM	.551	.548	.599	.601	<u>.609</u>	**.615**
	70B	PT	short	F1	.775	.790	.793	<u>.803</u>	.791	**.820**
			short	LLM	.693	.709	.718	<u>.722</u>	.715	**.723**
			long	LLM	.552	.534	.558	.569	<u>.571</u>	**.649**
		IT	short	F1	.748	.781	.790	**.799**	.783	.792
			short	LLM	.681	.698	.703	**.709**	.699	.699
			long	LLM	.555	.557	.568	.595	**.600**	.562
State-Space	**7B**	PT	short	F1	.811	.815	.809	.822	<u>.828</u>	**.843**
			short	LLM	.705	.711	.701	.711	<u>.716</u>	**.728**
			long	LLM	.567	.597	.574	.611	**.624**	.612
		IT	short	F1	.793	.814	.797	.816	<u>.829</u>	**.838**
			short	LLM	.690	.701	.689	.699	<u>.711</u>	**.719**
			long	LLM	.588	.587	.597	.618	**.629**	.615

Entropy entirely disregards the output sequence likelihood and only considers the proportion of output sequences that belong to the same semantic cluster (Farquhar et al., 2024).

Evaluation. Effective uncertainty measures should accurately reflect the reliability of answers generated by the language model. Higher uncertainty more likely leads to incorrect generations. Thus, to evaluate the performance of an uncertainty estimator, we assess how well it correlates with the correctness of the language model's answers; correct answers should be assigned a lower uncertainty estimator than incorrect answers. To determine whether an answer is correct, it has to be compared to the respective ground truth answer. To do so, we check if the F1 score of the commonly used SQuAD metric exceeds 0.5 (Rajpurkar et al., 2016). Although there are some limitations to using such a simple metric, it has relatively small errors in standard data sets and, therefore, remains widely used in practice However, this metric is only applicable for short-phrase generations that align with the ground truth answer. Therefore, we additionally employ Llama-3.1 with 70 billion parameters (Dubey et al., 2024) as an LLM-as-a-judge to assess the correctness of both short-phrase and full-sentence generations. Subsequently, to measure the correlation between incorrectness of answers and the respective uncertainty estimates, we use the Area Under the Receiver Operating Characteristic (AUROC). Higher AUROC values indicate better performance of the uncertainty estimator, as it reflects a stronger alignment between the correctness of the language model's answers and their respective uncertainty estimates. Overall, this evaluation process follows established methodologies for assessing the performance of uncertainty measures in NLG (Kuhn et al., 2023; Duan et al., 2023; Bakman et al., 2024; Farquhar et al., 2024; Nikitin et al., 2024; Aichberger et al., 2024; Kossen et al., 2024).

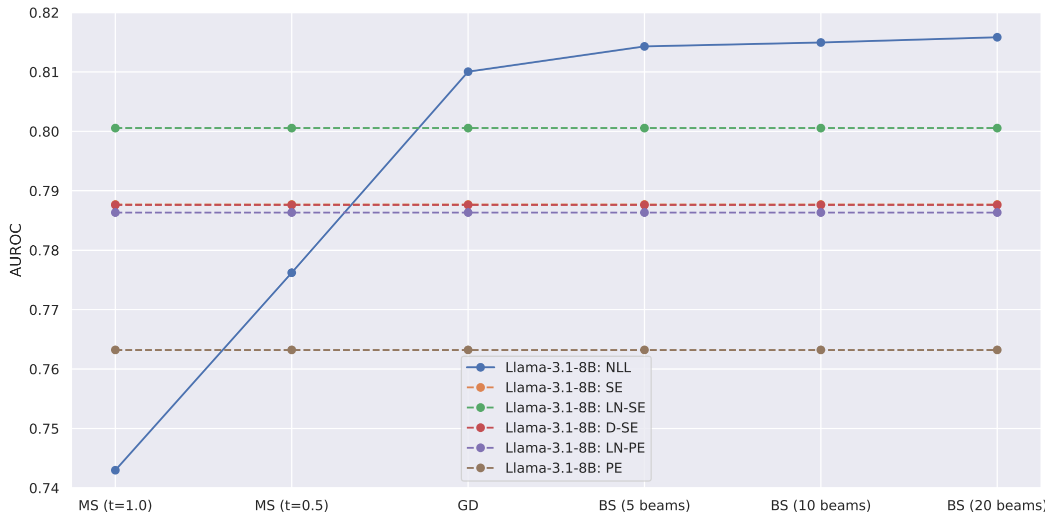

Figure 1: Average AUROC for the TriviaQA dataset, using the Llama-3.1-8B model to generate short phrase answers. The reference answer is generated using multinomial sampling (MS) with different temperature values (t), Greedy decoding (GD), and beam search (BS) with a different number of beams.

Analysis of results. Tab. 1 summarizes the performance of uncertainty measures across six different language models and six different tasks. Our proposed measure (NLL) largely outperforms current state-of-the-art uncertainty measures, particularly in tasks that involve generating short phrases. This suggests that our measure is highly effective when focusing on the critical part of the output sequence that contains the actual answer to a question. In practical scenarios, the reliability of the specific answer is often more relevant than the uncertainty of the entire generated sentence. Thus, our measure provides targeted and computationally efficient uncertainty estimates, delivering enhanced performance where it is most critical, especially in real-world applications.

Approximating the most likely output sequence. Figure 1 illustrates the performance of our uncertainty measure when considering different inference techniques for generating answers. The reference answer, generated via beam search with a size of 20, is used to assess correctness, as it provides the best approximation of the most likely answer generated by the language model. Since the baselines are evaluated on output sequences generated using their optimal hyperparameter settings, their performance remains consistent. The results show that as the approximation to the most likely answer improves, so does the performance of our measure. However, while multinomial sampling significantly degrades the performance of our uncertainty measure, greedy decoding achieves performance comparable to more precise methods, such as beam search, reinforcing its validity as an effective approximation of the most likely output sequence.

Further experimental results and insights into the behavior of the uncertainty estimators can be found in Sec. A and Sec. B in the appendix.

5 Conclusion

We introduced a computationally efficient, theoretically grounded uncertainty measure, the negative log-likelihood of the most likely output sequence under a given language model. This measure is motivated by the general notion of proper scoring rules, providing a theoretically justified measure that is well aligned with the practical usage of LLMs. The experiments show that our measure performs extremely well with just a single generated output sequence, compared to previous measures that require multiple costly sequences to estimate the uncertainty. As a result, our approach represents a significant advance toward providing reliable uncertainty estimates that can be effectively applied at scale.

Although our proposed measure effectively captures uncertainty, it currently does not consider the semantics of the generated output sequence. Future work should investigate how it could be extended to also account for semantic meaning, to further enrich the uncertainty estimator while preserving its computational efficiency.. Furthermore, all measures based on proper scoring rules depend on heuristics such as length normalization to deal with varying sequence lengths (Malinin & Gales, 2021; Duan et al., 2023; Bakman et al., 2024). Investigating theoretically justified means to account for these varying generation characteristics is another promising direction for future work. While there remain opportunities for refinement, our proposed measure establishes a solid foundation for reliable and scalable uncertainty estimation in NLG.

Ethics and Reproducibility Statement

We acknowledge that language models can generate biased or harmful content if not properly managed. While our uncertainty estimation method enhances reliability, we encourage the responsible use of our approach in conjunction with bias mitigation and content moderation techniques.

We have made concerted efforts to ensure the reproducibility of our results. We report the raw average scores across held-out test datasets without standard error, as the distributional characteristics are more reflective of the models and datasets selected than the uncertainty estimation method itself. Theoretical derivations are provided in Sec. 2. All datasets are publicly available, and standard benchmarks are utilized to facilitate replication. The source code for reproducing all experiments will be made available upon publication.

References

Lukas Aichberger, Kajetan Schweighofer, Mykyta Ielanskyi, and Sepp Hochreiter. Semantically diverse language generation for uncertainty estimation in language models. *arXiv*, 2406.04306, 2024.

Yavuz Faruk Bakman, Duygu Nur Yaldiz, Baturalp Buyukates, Chenyang Tao, Dimitrios Dimitriadis, and Salman Avestimehr. Mars: Meaning-aware response scoring for uncertainty estimation in generative llms. *arXiv*, 2402.11756, 2024.

Roi Cohen, Mor Geva, Jonathan Berant, and Amir Globerson. Crawling the internal knowledge-base of language models. In Andreas Vlachos and Isabelle Augenstein (eds.), *Findings of the Association for Computational Linguistics: EACL 2023*, pp. 1856–1869, Dubrovnik, Croatia, May 2023a. Association for Computational Linguistics.

Roi Cohen, May Hamri, Mor Geva, and Amir Globerson. LM vs LM: Detecting factual errors via cross examination. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12621–12640, Singapore, December 2023b. Association for Computational Linguistics.

Jinhao Duan, Hao Cheng, Shiqi Wang, Chenan Wang, Alex Zavalny, Renjing Xu, Bhavya Kailkhura, and Kaidi Xu. Shifting attention to relevance: Towards the uncertainty estimation of large language models. *arXiv*, 2307.01379, 2023.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, and Aurelien Rodriguez et al. The llama 3 herd of models. 2024.

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

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

Deep Ganguli, Amanda Askell, Nicholas Schiefer, Thomas I. Liao, Kamilė Lukošiūtė, Anna Chen, Anna Goldie, Azalia Mirhoseini, Catherine Olsson, Danny Hernandez, Dawn Drain, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jackson Kernion, Jamie Kerr, Jared Mueller, Joshua Landau, Kamal Ndousse, Karina Nguyen, Liane Lovitt, Michael Sellitto, Nelson Elhage, Noemi Mercado, Nova DasSarma, Oliver Rausch, Robert Lasenby, Robin Larson, Sam Ringer, Sandipan Kundu, Saurav Kadavath, Scott Johnston, Shauna Kravec, Sheer El Showk, Tamera Lanham, Timothy Telleen-Lawton, Tom Henighan, Tristan Hume, Yuntao Bai, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, Christopher Olah, Jack Clark, Samuel R. Bowman, and Jared Kaplan. The capacity for moral self-correction in large language models. *arXiv*, 2302.07459, 2023.

Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.

Sebastian Gruber and Florian Buettner. Uncertainty estimates of predictions via a general bias-variance decomposition. In Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent (eds.), *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 11331–11354. PMLR, 25–27 Apr 2023.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces, 2024.

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

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations*, 2020.

Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *arXiv*, 1112.5745, 2011.

Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning*, 110:457–506, 2021.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Vancouver, Canada, July 2017. Association for Computational Linguistics.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, 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. *arXiv*, 2207.05221, 2022.

Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

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

Nikita Kotelevskii and Maxim Panov. Predictive uncertainty quantification via risk decompositions for strictly proper scoring rules. *arXiv*, 2402.10727, 2024.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations*, 2023.

Meelis Kull and Peter Flach. Novel decompositions of proper scoring rules for classification: Score adjustment as precursor to calibration. In Annalisa Appice, Pedro Pereira Rodrigues, Vítor Santos Costa, Carlos Soares, João Gama, and Alípio Jorge (eds.), *Machine Learning and Knowledge Discovery in Databases*, pp. 68–85, Cham, 2015. Springer International Publishing.

Salem Lahlou, Moksh Jain, Hadi Nekoei, Victor I Butoi, Paul Bertin, Jarrid Rector-Brooks, Maksym Korablyov, and Yoshua Bengio. DEUP: Direct epistemic uncertainty prediction. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. Latent retrieval for weakly supervised open domain question answering. In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 6086–6096, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1612.

Stephanie Lin, Jacob Hilton, and Owain Evans. Teaching models to express their uncertainty in words. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856.

Andrey Malinin. *Uncertainty estimation in deep learning with application to spoken language assessment*. PhD thesis, University of Cambridge, 2019.

Andrey Malinin and Mark Gales. Uncertainty estimation in autoregressive structured prediction. In *International Conference on Learning Representations*, 2021.

Potsawee Manakul, Adian Liusie, and Mark Gales. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 9004–9017, Singapore, December 2023. Association for Computational Linguistics.

Sabrina J. Mielke, Arthur Szlam, Emily Dinan, and Y-Lan Boureau. Reducing conversational agents' overconfidence through linguistic calibration. *Transactions of the Association for Computational Linguistics*, 10:857–872, 2022.

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

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. Are NLP models really able to solve simple math word problems? In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2080–2094, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.168.

Xin Qiu and Risto Miikkulainen. Semantic density: Uncertainty quantification in semantic space for large language models. 2024.

Victor Quach, Adam Fisch, Tal Schuster, Adam Yala, Jae Ho Sohn, Tommi S. Jaakkola, and Regina Barzilay. Conformal language modeling. *arXiv*, 2306.10193, 2023.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2018.

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.

Jie Ren, Yao Zhao, Tu Vu, Peter J. Liu, and Balaji Lakshminarayanan. Self-evaluation improves selective generation in large language models. *arXiv*, 2312.09300, 2023.

Kajetan Schweighofer, Lukas Aichberger, Mykyta Ielanskyi, Günter Klambauer, and Sepp Hochreiter. Quantification of uncertainty with adversarial models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 19446–19484. Curran Associates, Inc., 2023.

Kajetan Schweighofer, Lukas Aichberger, Mykyta Ielanskyi, and Sepp Hochreiter. On information-theoretic measures of predictive uncertainty, 2024.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. *arXiv*, 1409.3215, 2014.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher Manning. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 5433–5442, Singapore, December 2023. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *arXiv*, 2307.09288, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30, 2017.

Yuxin Xiao, Paul Pu Liang, Umang Bhatt, Willie Neiswanger, Ruslan Salakhutdinov, and Louis-Philippe Morency. Uncertainty quantification with pre-trained language models: A large-scale empirical analysis. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 7273–7284, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. Opt: Open pre-trained transformer language models. *arXiv*, 2205.01068, 2022.

Kaitlyn Zhou, Dan Jurafsky, and Tatsunori Hashimoto. Navigating the grey area: How expressions of uncertainty and overconfidence affect language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 5506–5524, Singapore, December 2023. Association for Computational Linguistics.

Jingwei Zuo, Maksim Velikanov, Dhia Eddine Rhaiem, Ilyas Chahed, Younes Belkada, Guillaume Kunsch, and Hakim Hacid. Falcon mamba: The first competitive attention-free 7b language model. 2024.

A Comparision of Estimators

In this section we want to empirically investigate the performance of estimators for the predictive entropy $\mathrm{H}(p(\boldsymbol{y} \mid \boldsymbol{x}))$ (Eq. (5)) and the maximum likelihood $1 - \max_{\boldsymbol{y}} p(\boldsymbol{y} \mid \boldsymbol{x})$ (Eq. (8)). Therefore, we consider a synthetic experiment with the following setup. We are given a space of possible outcomes $\mathcal{V}$ with $|\mathcal{V}| = \{10, 100\}$. The task is to predict a sequence $\boldsymbol{y} = (y_1, ...y_T) \in \mathcal{Y}_T$ where $y \in \mathcal{V}$ and T is 2, 3, or 4. Predictive distributions $p(\boldsymbol{y} \mid \boldsymbol{x}) = \prod_{t=1}^{T} p(\boldsymbol{y}_t \mid \boldsymbol{y}_{<t}, \boldsymbol{x})$ are not represented by a neural network, but randomly sampled (but fixed per run) according to a Dirichlet distribution $\mathrm{Dir}(\{\alpha_1, ..., \alpha_{|\mathcal{V}|}\})$. The alpha parameters of the Dirichlet distribution are specified to yield typical predictive distributions as encountered in language models that follow a power law. For $|\mathcal{V}| = 10$ we have $\alpha_{1,2} = 10$ and $\alpha_{3-10} = 0.2$. For $|\mathcal{V}| = 100$ we have $\alpha_{1,2} = 10$, $\alpha_{3-6} = 1$ and $\alpha_{7-100} = 0.2$. Note that the order of alpha values is randomly shuffled before drawing each predictive distribution. Representative predictive distributions sampled from this Dirichlet distribution are shown in the leftmost subfigures in Fig. 2 and Fig. 3.

The experiments investigate the quality of the estimators depending on the number of samples $\{\boldsymbol{y}_n\}_{n=1}^{N}$. This is possible because it is possible to calculate the ground truth values for both the entropy and the maximum likelihood sequence for this small synthetic example by exhaustive enumeration. We average over 1,000 runs, meaning that the predictive distributions are redrawn according to the respective Dirichlet distribution. This corresponds to evaluating uncertainty for different input sequences $\boldsymbol{x}$ for language models.

Entropy estimation. The results are shown in Fig. 2. We observe that the variance of estimators increases for larger vocabulary sizes $|\mathcal{V}|$ and sequence lengths T. Furthermore, lower temperatures decrease the variance of the estimator at the cost of introducing bias.

Maximum Likelihood. The results are shown in Fig. 3. We observe that low-temperature multinomial sampling and beam search find the maximum log-likelihood with a low number of samples with high probability. Greedy decoding (beam size = 1) finds the maximum for all settings except the hardest ($\||\mathcal{V}| = 100, T = 4$), where it takes a beam size of 2 to find it.

B Detailed Results

In this section, we provide detailed results to complement the main results presented in Tab. 1.

The main results used greedy decoding (beam search of size 1) to estimate the maximum likelihood (zero-one score based measure) and 10 samples to estimate entropies (logarithmic score based measures). For each dataset, we performed a hyperparameter search on held-out instances to determine the best performing temperature $t \in \{0.5, 1.0, 1.5\}$ for sampling output sequences used for the logarithmic score based measures.

We look into how much the maximum likelihood benefits from additional samples by increasing the beam with to 5. The results are given in Tab. 2, showing that our measure continues to improve for a larger number of beams, thus better estimates of the maximum likelihood sequence. Furthermore, we provide detailed results for individual datasets in Tab. 3, complimenting the results presented in the main paper (c.f. Tab. 1).

The AUROC is considered as a primary performance measure throughout the paper. We additionally consider the average rejection accuracy, i.e. the accuracy of model predictions when allowing to reject a certain budget of predictions based on the uncertainty estimate. Thus, predictions are only evaluated for the 80% most certain predictions. Results are given in Tab. 4, again with greedy decoding for our measure based on the zero-one score. The results show, that our measure is very competitive across all settings, despite its simplicity and efficiency.

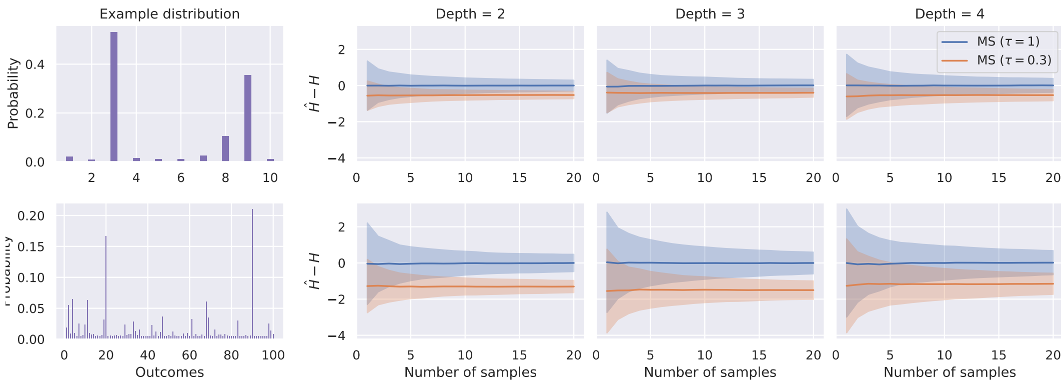

Figure 2: **Estimator of Predictive Entropy.** Results for different vocabulary sizes (rows) and sequence lengths (columns). The two leftmost subfigures show exemplary predictive distributions $p(\boldsymbol{y}_t \mid \boldsymbol{y}_{<t}, \boldsymbol{x})$. We estimate the entropy using N samples by means of Eq. (5). Lines denote the average over runs, while shades denote one standard deviation. We compare multinomial sampling (MS) for two commonly used temperatures. The experiments show that temperature decreases variance but introduces bias.

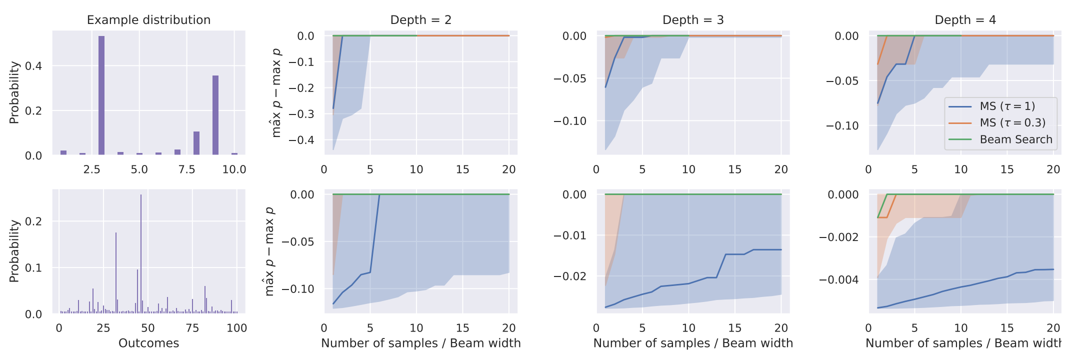

Figure 3: **Estimator of maximum likelihood.** Results for different vocabulary sizes (rows) and sequence lengths (columns). The two leftmost subfigures show exemplary predictive distributions $p(\boldsymbol{y}_t \mid \boldsymbol{y}_{<t}, \boldsymbol{x})$. We estimate the maximum likelihood using the maximum over N sampled obtained by beam search or multinomial sampling (MS) with different temperatures. Lines denote the median, shades signify the possible values between the 5 and 95 percent quantile. Even with a very low number of samples, low-temperature multinomial sampling (MS) and beam search are able to find the maximum with high probability.

Table 2: **Average AUROC** across TriviaQA, SVAMP, and NQ datasets, using uncertainty estimates of different measures as a score to distinguish between correct and incorrect answers. Varying model architectures (*transformer*, *state-space*), model sizes (*7B*, *8B*, *70B*), and model stages (*PT*, *IT*) are considered for generating answers. The reference answer is generated using **beam search with 5 beams**, either as a whole sentence (*long*) or a short phrase (*short*). The correctness of the reference answer is assessed by checking if the F1 score of the commonly used SQuAD metric exceeds 0.5 (*F1*) or the Llama-3.1-70B model considers it as correct (*LLM*). Predictive Entropy (*PE*), length-normalized Predictive Entropy (*LN-PE*), Semantic Entropy (*SE*), length-normalized Semantic Entropy (*LN-SE*), and discrete Semantic Entropy (*D-SE*) use 10 output sequences to assign an uncertainty estimate, each generated via multinomial sampling. NLL solely uses the reference answer to assign an uncertainty estimate.

Uncertainty measure based score					*Logarithmic*					*Zero-One*
Model			**Gen.**	**Metric**	**PE**	**LN-PE**	**SE**	**LN-SE**	**D-SE**	**NLL**
Transformer	8B	PT	short	F1	.775	.791	.765	.787	.799	**.822**
Transformer	8B	PT	short	LLM	.700	.712	.686	.704	.713	**.726**
Transformer	8B	PT	long	LLM	.556	.540	.493	.520	.578	**.591**
Transformer	8B	IT	short	F1	.778	.808	.805	.819	.811	**.845**
Transformer	8B	IT	short	LLM	.682	.704	.706	.713	.698	**.729**
Transformer	8B	IT	long	LLM	.535	.520	.584	.585	**.586**	.559
Transformer	70B	PT	short	F1	.788	.799	.796	.812	.798	**.833**
Transformer	70B	PT	short	LLM	.700	.717	.719	**.727**	.718	.725
Transformer	70B	PT	long	LLM	.540	.552	.489	.531	.552	**.608**
Transformer	70B	IT	short	F1	.756	.786	.796	**.806**	.788	.800
Transformer	70B	IT	short	LLM	.680	.697	.701	**.707**	.695	**.707**
Transformer	70B	IT	long	LLM	.534	.533	.544	.569	**.574**	.534
State-Space	7b	PT	short	F1	.814	.818	.806	.823	.825	**.846**
State-Space	7b	PT	short	LLM	.703	.709	.699	.711	.712	**.719**
State-Space	7b	PT	long	LLM	.570	.595	.550	**.609**	.602	.563
State-Space	7b	IT	short	F1	.799	.815	.794	.817	.828	**.845**
State-Space	7b	IT	short	LLM	.699	.713	.694	.709	.720	**.730**
State-Space	7b	IT	long	LLM	.574	.575	.582	**.621**	.607	.577

Table 3: **Average AUROC** of individual datasets, using uncertainty estimates of different measures as a score to distinguish between correct and incorrect answers.

Uncertainty measure based score						*Logarithmic*					*Zero-One*
$\mathcal{D}$	**Model**			**Gen.**	**Metric**	**PE**	**LN-PE**	**SE**	**LN-SE**	**D-SE**	**NLL**
TriviaQA	**Transformer**	**8B**	PT	short	F1	.758	.778	.788	.798	.787	**.810**
				short	LLM	.675	.694	.703	.704	.682	**.722**
				long	LLM	.592	.604	.640	.631	.650	**.704**
			IT	short	F1	.735	.768	.790	.800	.777	**.809**
				short	LLM	.660	.684	.708	.710	.680	**.716**
				long	LLM	.603	.627	**.678**	.672	.670	.670
		70B	PT	short	F1	.707	.730	.741	.743	.702	**.744**
				short	LLM	.650	.660	.696	.695	.656	**.698**
				long	LLM	.538	.533	.625	.574	.563	**.692**
			IT	short	F1	.698	.714	.722	**.726**	.688	.722
				short	LLM	.663	.675	.685	.679	.633	**.701**
				long	LLM	.530	.553	.564	**.571**	.564	.543
	State-Space	**7B**	PT	short	F1	.786	.793	.812	.818	.810	**.832**
				short	LLM	.687	.697	.712	.714	.695	**.724**
				long	LLM	.597	.653	.675	.680	.689	**.705**
			PT	short	F1	.780	.799	.810	.819	.811	**.827**
				short	LLM	.696	.701	.714	.717	.703	**.730**
				long	LLM	.645	.654	.688	**.698**	.692	.694
SVAMP	**Transformer**	**8B**	PT	short	F1	.847	.867	.865	.870	.868	**.885**
				short	LLM	.779	.788	.753	.772	**.791**	.772
				long	LLM	.575	.563	.519	.534	.601	**.669**
			IT	short	F1	.879	.903	.914	.912	.887	**.931**
				short	LLM	.706	.725	.736	.731	.701	**.753**
				long	LLM	.556	.524	.590	.608	.631	**.662**
		70B	PT	short	F1	.892	.906	.925	.929	.923	**.936**
				short	LLM	.794	.817	.814	.815	**.819**	.799
				long	LLM	.578	.554	.553	.579	.571	**.665**
			IT	short	F1	.830	.895	.915	**.922**	.915	.909
				short	LLM	.703	.744	.734	.748	**.762**	.713
				long	LLM	.601	.577	.613	.649	**.663**	.597
	State-Space	**7B**	PT	short	F1	.882	.893	.874	.883	.889	**.914**
				short	LLM	.752	.757	.730	.738	.757	**.776**
				long	LLM	.536	.585	.534	.602	**.612**	.579
			IT	short	F1	.843	.891	.854	.876	.892	**.905**
				short	LLM	.706	.730	.704	.709	.737	**.744**
				long	LLM	.577	.586	.578	.616	**.639**	.613
NQ	**Transformer**	**8B**	PT	short	F1	.725	.739	.673	.710	.758	**.776**
				short	LLM	.639	.661	.615	.641	**.683**	**.683**
				long	LLM	.517	.498	.478	.495	.550	**.573**
			IT	short	F1	.702	.732	.711	.731	.756	**.774**
				short	LLM	.662	.682	.669	.685	**.700**	.697
				long	LLM	.494	.491	**.530**	.524	.527	.514
		70B	PT	short	F1	.727	.733	.711	.737	.748	**.779**
				short	LLM	.634	.649	.642	.657	.671	**.672**
				long	LLM	.538	.514	.494	.553	.580	**.589**
			IT	short	F1	.718	.734	.734	**.748**	.746	.743
				short	LLM	.676	.674	.689	.698	**.702**	.681
				long	LLM	.535	.540	.526	.566	**.574**	.545
	State-Space	**7B**	PT	short	F1	.766	.758	.741	.765	**.785**	.782
				short	LLM	.675	.680	.661	.681	**.697**	.683
				long	LLM	.567	.553	.512	.551	**.572**	.554
			IT	short	F1	.755	.751	.727	.754	**.783**	.781
				short	LLM	.669	.672	.648	.671	**.692**	.683
				long	LLM	.541	.521	.526	.541	**.554**	.537

Table 4: **Average Rejection Accuracy (80%)** across TriviaQA, SVAMP and NQ datasets, using uncertainty estimates of different measures as a score to distinguish between correct and incorrect answers. The reference answer is generated using greedy decoding, with the correctness being assessed by checking if the F1 score of the commonly used SQuAD metric exceeds 0.5 (*F1*), the pre-trained Llama-3.1-70B model considers it as correct (*LLM*), or the instruction-tuned Llama-3.1-70B-Instruct model considers it as correct (*LLM-Instruct*).

Uncertainty measure based score					*Logarithmic*					*Zero-One*
Model		**Gen.**		**Metric**	**PE**	**LN-PE**	**SE**	**LN-SE**	**D-SE**	**NLL**
Transformer	8b	PT	short	F1	.661	.672	.651	.643	.655	**.681**
				LLM	.774	**.782**	.767	.766	.765	.778
				LLM-Instruct	.704	.721	.693	.688	.702	**.723**
			long	LLM	.596	.590	.598	.592	.590	**.619**
				LLM-Instruct	.667	.684	.632	.643	.644	**.686**
		IT	short	F1	.668	.684	.680	.673	.687	**.702**
				LLM	.775	.781	.779	.775	.778	**.788**
				LLM-Instruct	.723	.742	.732	.726	.743	**.751**
			long	LLM	.628	.630	.651	.644	**.653**	.652
				LLM-Instruct	.713	.724	.705	.713	.727	**.734**
	70b	PT	short	F1	.818	.827	.822	.827	.829	**.836**
				LLM	.844	.852	.846	.847	.851	**.855**
				LLM-Instruct	.867	.875	.876	.881	**.885**	.881
			long	LLM	.704	.699	.719	.707	.705	**.724**
				LLM-Instruct	.789	.795	.776	.781	.788	**.812**
		IT	short	F1	.795	.813	.814	.809	.819	**.823**
				LLM	.836	.842	.842	.837	.844	**.845**
				LLM-Instruct	.850	.867	.866	.865	**.874**	.870
			long	LLM	.706	.706	.712	.715	**.721**	.715
				LLM-Instruct	.855	.850	.827	.842	**.861**	.851
State-Space	7b	PT	short	F1	.598	.596	.585	.579	.583	**.612**
				LLM	.729	.737	.723	.721	.733	**.742**
				LLM-Instruct	.638	.640	.626	.621	.632	**.651**
			long	LLM	.613	**.627**	.612	.624	.620	.623
				LLM-Instruct	.606	.611	.601	.611	.618	**.633**
		IT	short	F1	.592	.603	.588	.581	.589	**.615**
				LLM	.737	.742	.730	.726	.740	**.744**
				LLM-Instruct	.632	.646	.625	.619	.637	**.653**
			long	LLM	.611	.617	.618	.612	**.625**	**.625**
				LLM-Instruct	.643	.652	.628	.628	.654	**.658**