MARS: Meaning-Aware Response Scoring for Uncertainty Estimation in Generative LLMs

Anonymous ACL submission

Abstract

Generative Large Language Models (LLMs) are widely utilized for their excellence in various tasks. However, their tendency to produce inaccurate or misleading outputs poses a potential risk, particularly in high-stakes environments. Therefore, estimating the correctness of generative LLM outputs is an important task for enhanced reliability. Uncertainty Estimation (UE) in generative LLMs is an evolving domain, where SOTA probability-based methods commonly employ length-normalized scoring. In this work, we propose Meaning-Aware Response Scoring (MARS) as an alternative to length-normalized scoring for UE methods. MARS is a novel scoring function that considers the semantic contribution of each token in the generated sequence in the context of the question. We demonstrate that integrating MARS into UE methods results in a universal and significant improvement in UE performance. We conduct experiments using three distinct closed-book question-answering datasets across five popular pre-trained LLMs. Lastly, we validate the efficacy of MARS on a Medical QA dataset. Code can be found here.

1 Introduction

Generative Large Language Models (LLMs) have risen in popularity due to their remarkable ability to understand, generate, and process human language at an unprecedented scale and accuracy (Ye et al., 2023; OpenAI, 2023; Touvron et al., 2023). These models have become the state-of-the-art in various fields, including machine translation, content generation, and even scientific research (Huang et al., 2023; OpenAI, 2023) due to their capability to handle diverse tasks such as text summarization, sentiment analysis, and question-answering in a few-shot or zero-shot manner.

Despite their growing popularity and success, generative LLMs are not infallible and can sometimes produce erroneous or misleading outputs, especially when dealing with complex reasoning problems or closed-book questions. This limitation becomes particularly critical in question-answering systems used in high-stakes environments. Quantifying the uncertainty of generative LLM responses in such scenarios is not just beneficial but essential for ensuring trustworthy operation. For example, in a medical advice application, accurately assessing the uncertainty of the responses provided by LLMs can prevent the provision of incorrect medical advice. This is crucial because erroneous advice may lead to devastating medical missteps or misunderstandings. Thus, understanding and quantifying uncertainty helps in reliable risk assessment and in maintaining the overall quality of the answers provided, ensuring that users can assess how much reliance they should place on LLM responses.

Uncertainty Estimation (UE) is a well-studied problem in classification scenarios, especially in the computer vision domain (Lakshminarayanan et al., 2017; Gal and Ghahramani, 2016; Shen et al., 2021). The proposed UE methods in classification tasks, which rely on the class probabilities, are not directly applicable to generative LLMs due to the auto-regressive generative structure of LLMs (Malinin and Gales, 2021), which implies that LLMs generate text sequentially by predicting each subsequent word based on the combined context of all preceding words. This process differs significantly from classification tasks, where the output is typically a single label or a set of labels assigned to an entire input, without the sequential and context-accumulating nature of generative LLMs. Recent work (Malinin and Gales, 2021), formalizes how to adapt popular UE methods developed for classification tasks to the context of generative LLMs. They propose using length-normalized scoring to estimate the likelihood of a sequence generated by the model, and the subsequent works (Kuhn et al., 2023; Lin et al., 2023; Chen and Mueller, 2023) utilize that idea of length-normalized scoring.

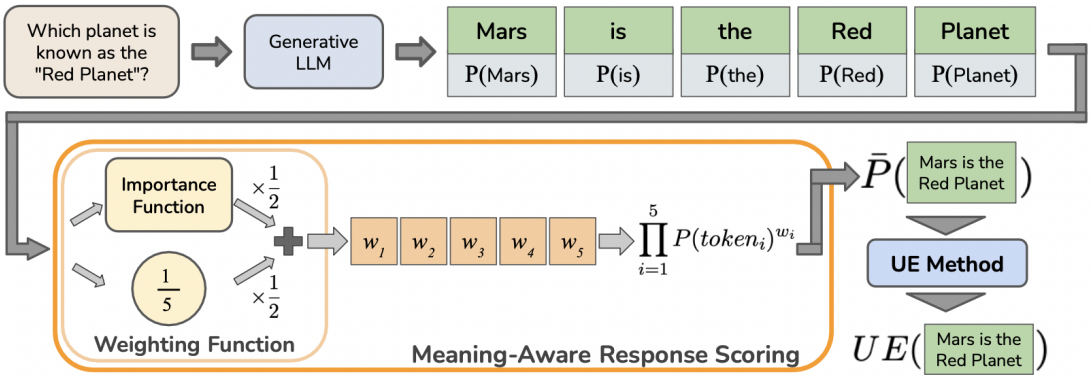

Figure 1: Overview of Meaning-Aware Response Scoring (MARS). Each token in the response of a generative LLM is assigned a weight based on its importance in the meaning. The product of the weighted probabilities of these tokens yields the response score. MARS is then used for Uncertainty Estimation (UE) methods in generative LLMs.

A downside of these existing UE techniques in the generative LLM literature is treating length-normalized scoring like the class probabilities in classification tasks. However, better ways may exist for estimating uncertainty than directly using the length-normalized score of a sequence, as it treats all tokens equally. In reality, each word's contribution to the sentence's meaning in the question context might vary. For example, given the question "Which planet is known as the Red Planet?" and with the generated response "Mars is the Red Planet", the tokens of "Mars" are the most critical ones in the response because those tokens are the ones actually answering the question. Thus, assigning more weight to semantically significant tokens in the response score calculation can improve UE methods, resulting in more accurate predictions.

Based on this word importance intuition, we propose a novel scoring function for generative LLMs called *Meaning-Aware Response Scoring (MARS)*, as outlined in Figure 1. To compute the LLM response score as an input to UE methods, we first assign an importance coefficient to each token in the generation. This importance essentially reflects the impact of masking a token in a sequence on the meaning of the generated response, where tokens with a greater influence on the meaning receive higher importance. By leveraging these meaning-aware coefficients (w_i in Figure 1), MARS returns the multiplication of the weighted probabilities of the tokens in the generated sequence.

We list our main contributions as follows:

- We propose a novel scoring function for UE in generative LLMs named Meaning-Aware Response Scoring (MARS).
- We introduce a BERT-like model, efficiently assigning meaning-aware importance weights to the tokens in a single model pass within MARS calculation.
- We evaluate probability-based UE metrics with MARS on question-answer datasets and show that MARS universally improves the UE performance for an extensive list of LLMs.

2 Background

In this section, we will go over probability-based UE methods that our work built on, for a detailed discussion on related works, refer to Appendix A.

In the literature, UE is used as a proxy for the correctness of the model output (Malinin and Gales, 2021; Gal and Ghahramani, 2016; Lakshminarayanan et al., 2017; Band et al., 2021). For generative LLMs in the question-answer context, we consider the most probable sequence as the model output and utilize UE to predict the correctness of the response following Kuhn et al. (2023). The goal of UE is to assign higher scores to incorrect responses, indicating greater uncertainty, and lower scores to correct responses, signifying less uncertainty.

2.1 Bayesian View to Estimate Uncertainty

Bayesian UE is used in machine learning to quantify uncertainty in predictions. It treats model parameters as random variables, assigning a prior probability distribution to them. Through Bayesian inference, this distribution is updated with training data, yielding a posterior distribution. Prediction uncertainty stems from this posterior distribution. Let $\{\theta_i\}_{i=1}^{M}$ be an ensemble of models sampled from approximate posterior $q(\theta) \approx p(\theta|D)$ where D is the training data.

The predictive posterior of input $x \in \mathcal{X}$ for target $y \in \mathcal{Y}$ is derived by expectation over the en-

semble:

$$P(y|x,D) = \mathbb{E}_{q(\theta)}[P(y|x,\theta)] \approx \frac{1}{M}\sum_{m=1}^{M} P(y|x,\theta_m), \quad (1)$$

where we have $\theta_m \sim q(\theta) \approx p(\theta|D)$. Using the posterior probability definition, we can define the entropy of predictive posterior as:

$$\mathcal{H}(x,D) = -\sum_{y\in\mathcal{Y}} P(y|x,D)\log P(y|x,D). \quad (2)$$

In classification tasks, commonly used tools for estimating uncertainty are the entropy of the predictive posterior and the negative predictive posterior probability of the most probable answer (Gal and Ghahramani, 2016; Lakshminarayanan et al., 2017; Malinin and Gales, 2021; Xiao et al., 2022; Chen and Mueller, 2023). However, the formulation in (1) is not applicable to generative LLMs because of their auto-regressive generative structure.

2.2 Uncertainty Estimation (UE) of Auto-Regressive Generative Models

Malinin and Gales (2021) formalizes posterior probability definition for auto-regressive generative models where the output $\mathbf{s}$ is not a single entity but a sequence of tokens $\mathbf{s} = \{s_1, s_2, ..., s_L\}$. They simply replace $P(y|x,\theta)$ in (1) with sequence probability $P(\mathbf{s}|\mathbf{x},\theta)$. The probability of a sequence $\mathbf{s}$ for a given model parametrized with θ is defined as the multiplication of probabilities of its tokens:

$$P(\mathbf{s}|\mathbf{x},\theta) = \prod_{l=1}^{L} P(s_l|s_{<l},\mathbf{x};\theta) \quad (3)$$

where $s_{<l} \triangleq s_1, s_2, .., s_{l-1}$ referring to generated tokens before the generation of s_l. Kuhn et al. (2023) simplifies the ensemble sampling in (1) by using a single model in the ensemble due to the large size of foundation models. We follow the simplified version in the rest of the paper:

$$P(\mathbf{s}|\mathbf{x},D) \approx P(\mathbf{s}|\mathbf{x},\theta) = \prod_{l=1}^{L} P(s_l|s_{<l},\mathbf{x};\theta). \quad (4)$$

2.3 Length-Normalized Scoring

One of the key issues with using sequence probability $P(\mathbf{s}|\mathbf{x},\theta)$ as a proxy for $P(y|x,\theta)$ lies in its tendency to decrease as the sequence length increases. To overcome this issue, Malinin and Gales (2021) uses a length-normalized scoring function instead of sequence probability.[1] Length-normalized scoring $\tilde{P}(\mathbf{s}|\mathbf{x},\theta)$ is defined as follows:

$$\tilde{P}(\mathbf{s}|\mathbf{x},\theta) = \prod_{l=1}^{L} P(s_l|s_{<l},\mathbf{x};\theta)^{\frac{1}{L}}, \quad (5)$$

which assigns equal weights to each token in the generation where these weights are inversely proportional to the sequence length L. Although length-normalized scoring $\tilde{P}(\mathbf{s}|\mathbf{x},\theta)$ does not correspond to an actual probability distribution, Malinin and Gales (2021) and Kuhn et al. (2023) consider $\tilde{P}(\mathbf{s}|\mathbf{x},\theta)$ as auxiliary probabilities and replace the sequence probability $P(\mathbf{s}|\mathbf{x},\theta)$ in (4) with the length-normalized scoring given in (5).

2.4 Entropy-Based UE for Generative LLMs

To obtain the entropy of the output for given input $\mathbf{x}$, Malinin and Gales (2021) uses Monte-Carlo approximation over beam-sampled generations of a single model, as going through the entire answer set is infeasible due to its exponential computation complexity. Approximated entropy is defined as:

$$\mathcal{H}(\mathbf{x},\theta) \approx -\frac{1}{B}\sum_{b=1}^{B} \ln \tilde{P}(\mathbf{s}_b|\mathbf{x},\theta), \quad (6)$$

where $\mathbf{s}_b$ is an output sampled by beam-search and B is the total number of sampled generations.

Kuhn et al. (2023) proposes an alternative entropy definition, named Semantic Entropy (SE), considering the meaning of the generations. They use the same entropy definition in (6), but cluster sampled generations based on their meaning. For example, in response to the question "What is the capital city of France?", a model might output: "Paris" with score $\tilde{p}_1$ and "It's Paris" with score $\tilde{p}_2$. While standard entropy in (6) treats these as distinct outputs, SE clusters them together as they convey the same meaning in the question context, forming a single cluster c with summed score $\tilde{p}_1 + \tilde{p}_2$. More formally, cluster scoring is defined as:

$$\tilde{P}(\mathrm{c}|\mathbf{x},\theta) = \sum_{\mathbf{s},\mathbf{x}\in\mathrm{c}} \tilde{P}(\mathbf{s}|\mathbf{x},\theta). \quad (7)$$

SE follows from this cluster scoring $\tilde{P}(\mathrm{c}|\mathbf{x},\theta)$:

$$SE(\mathbf{x},\theta) = -\frac{1}{|C|}\sum_{i=1}^{|C|} \log \tilde{P}(\mathrm{c}_i|\mathbf{x},\theta), \quad (8)$$

[1] A scoring function K takes two inputs: the predicted probability p of an event and its actual outcome o, and returns a numerical score (Gneiting and Raftery, 2007).

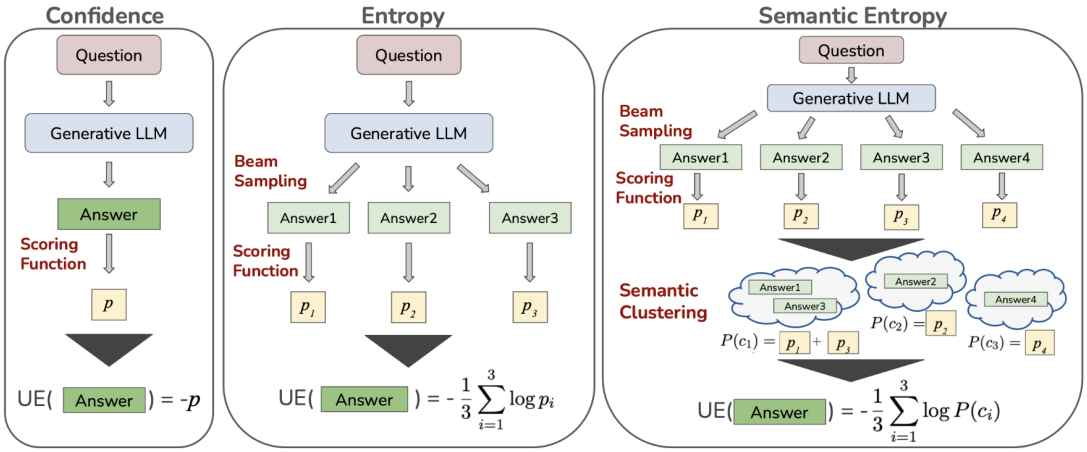

Figure 2: The most common probability-based UE methods for generative LLMs. The aim is to calculate the uncertainty of the most probable answer (shown in darker green) to the given question. Length-normalized scoring (5) is used in all these methods to obtain output scores. We propose MARS to replace it in these schemes.

where c_i refers to each semantic cluster and C is the set of all clusters. Similar to length-normalized scoring in (5), semantic entropy lacks theoretical justification, yet shows empirical success (Kuhn et al., 2023). In Appendix B, we provide a theoretical explanation for these heuristic design choices.

Negative length-normalized scoring of the most probable answer, standard sequence entropy in (6) and semantic entropy in (8) are the most common probability-based UE methods for generative LLMs (Malinin and Gales, 2021; Kuhn et al., 2023; Chen and Mueller, 2023; Lin et al., 2023) and are visualized in Figure 2. All of these methods depend on length-normalized scoring which we aim to replace with our alternative scoring, MARS.

3 Method

3.1 Key Intuition

Existing literature utilizes length-normalized scoring in UE as shown in (5), (6), and (7). Length-normalized scoring, given in (5), assigns equal importance/weight $(1/L)$ to each token in the generated sentence. The normalization aims to compare the probabilities of short and long sequences more fairly (Malinin and Gales, 2021). Such a normalization method may fall short in considering semantic contribution of tokens, even though it balances length differences across sequences.

To illustrate, consider the following example: **Question:** "Which planet is known as the Red Planet?" **Generated Answer:** "Mars is known as the Red Planet". In this answer, the word "Mars" is relatively more important as it directly addresses the question. Other words in the sentence primarily serve syntactic purposes or help achieve human-like answer. Thus, while designing a scoring function, we should give more importance/weight to the word "Mars". With this intuition, we want to replace length-normalized scoring and propose an alternative scoring function that assigns importance/weight to each word in the sentence considering both its contribution to the overall meaning in the given context and sequence length.

3.2 Meaning-Aware Response Scoring

Following our word importance intuition, we propose to replace length-normalized scoring $\tilde{P}(\mathbf{s}|\mathbf{x},\theta)$ in (5), (6), and (7) with Meaning-Aware Response Scoring (MARS). MARS is defined as:

$$\bar{P}(\mathbf{s}|\mathbf{x},\theta) = \prod_{l=1}^{L} P(s_l|s_{<l},\mathbf{x};\theta)^{w(\mathbf{s},\mathbf{x},L,l)}, \quad (9)$$

where $w(\cdot)$ is the weighting function that assigns a weight to each token regarding the generated answer, question context, and sequence length.

We design $w(\cdot)$ as a convex combination of importance coefficient and $1/L$, which enables MARS to consider both sequence length and meaning contribution of tokens. Formally, we define

$$w(\mathbf{s},\mathbf{x},L,l) \triangleq \frac{1}{2L} + \frac{u(\mathbf{s},\mathbf{x},l)}{2}, \quad (10)$$

where $u(\cdot)$ is importance function taking three arguments: generated sequence $\mathbf{s}$, contextual information $\mathbf{x}$, and the position l of a token within the sequence. The function $u(\cdot)$ assigns an importance coefficient to each token, where this coefficient ranges between 0 and 1. Additionally, it ensures

that the total sum of the importance coefficient for all tokens in a single generation $\mathbf{s}$ is 1. Next, we explain how to design the importance function $u(\cdot)$.

3.3 Importance Function Design

We design the token importance function $u(\cdot)$ by measuring the semantic impact of removing a specific token from the generated text. This evaluation of meaning is context-sensitive. In question-answer tasks, which is the focus of this work, the context is defined as the question itself. Thus, $u(\cdot)$ is designed to determine the importance of each token based on its influence on the overall meaning of the response within the context of the question.

To measure the amount of semantic change in the given context, we employ a neural network model originally developed as a question-answer evaluator by Bulian et al. (2022). This model, called BERT matching (BEM), takes three inputs: question, ground truth answer, and predicted answer, returning a probability score indicating answer correctness. For a question $\mathbf{x}$ and a generated answer $\mathbf{s} = \{s_1, s_2, \ldots, s_L\}$, we determine the importance of each token as follows: We mask token s_l in the generated answer and feed the question $\mathbf{x}$, the original answer $\mathbf{s}$, and masked response sequence $\mathbf{s} \setminus \{s_l\}$ into the BEM model. The output o, ranging from 0 to 1, indicates the impact of the masked token on answer correctness. A token s_l with substantial impact yields an output o close to 0, whereas a lesser impact results in an output closer to 1. Hence, we define $1 - o$ as the preliminary coefficient of s_l. Once we compute preliminary coefficients for all tokens, we normalize them using a softmax function with a temperature parameter τ. In our experiments, we set $\tau = 0.01$.

Addressing Token Dependency. Our initial approach for assigning importance coefficients to tokens assumes their semantic independence even though tokens often exhibit semantic interdependencies. For example, in the sentence "Hamlet is written by William Shakespeare," tokens "William" and "Shakespeare" are intrinsically linked. Treating such tokens independently ignores linguistic nuances, so we refine our methodology. Instead of masking tokens individually, we mask tokens at the phrase level (details in Appendix C.1). This approach acknowledges and preserves the inherent semantic relationships between closely related tokens, resulting in a more accurate and contextually aware assessment of token importance.

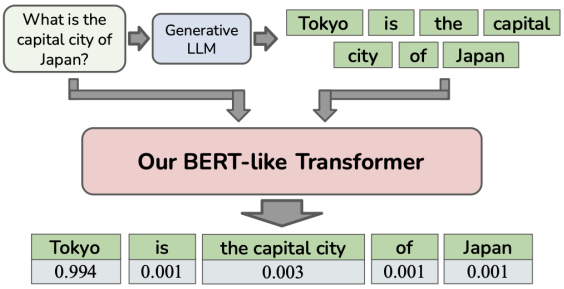

Figure 3: Our Bert-like transformer model takes the question and the generated answer as inputs, and outputs phrases in the generated answer and corresponding importance coefficients.

In particular, a response $\mathbf{s} = \{s_1, s_2, \ldots, s_L\}$ is composed of phrases $\{h_1, h_2, \ldots, h_K\}$, where each token s_l belongs to a phrase h_k. We mask phrases one by one and find the importance coefficient of each phrase with BEM model. To translate phrase-level importance coefficients into token-level coefficients, we distribute the importance score to all tokens in the phrase equally. We summarize the enhanced algorithm in Appendix C.2. Further, in Section 4.3, we show that allocating importance score only to the most uncertain token within a phrase also yields comparable results.

Reducing Computation. The necessity of performing a separate neural network pass for each phrase to determine its importance score increases the computational load of the proposed approach. Additionally, detecting phrases themselves requires another neural network pass, further increasing the computational complexity. To address these challenges, we have developed a BERT-like neural network model with 110M parameters (a significantly smaller model compared to LLMs). This model is capable of performing both tasks simultaneously for a given sequence in a single neural network pass: it identifies phrases within the generated text and their importance scores (see Figure 3). This dual-functionality substantially reduces the computational cost, making the algorithm more efficient and scalable. For detailed model architecture and performance metrics, please refer to Appendix C.

4 Experiments

4.1 Experimental Design

In the UE context, we expect that if the model is uncertain about the generated answer, then the answer should be less reliable and tend to be incorrect.

Datasets. We use three closed-book Question-Answer (QA) datasets for evaluation: Trivi-

aQA (Joshi et al., 2017), Natural Questions (Kwiatkowski et al., 2019), and WebQA (Chang et al., 2022). We give further details in Appendix D.

Models. Our evaluation consists of 5 popular open-source LLMs. First two models are Llama-7B and Llama-7B-chat, where the latter one is fine-tuned for dialogue use cases (Touvron et al., 2023). We also use Mistral-7B (Jiang et al., 2023) as well as Falcon-7B (Almazrouei et al., 2023) which is fine-tuned on a mixture of chat/instruct datasets. To extend our analysis to larger models, we include Llama-13B (Touvron et al., 2023). We do not perform any further training on these models, rather we use their pre-existing configurations. Following Kuhn et al. (2023), we abstain from assuming any ensemble of the models, considering the significant size and time requirements associated with LLMs.

Baselines. As we focus on the probability-based UE methods, we do not include heuristic-based and black-box methods. We use 3 SOTA probability-based UE methods as baselines (see Figure 2 for visualization): **1.** Negative length-normalized score (Confidence), which provides the confidence score of the most likely generation only by using its token probabilities as in (5). **2.** Entropy as in (6), which requires generating multiple answers to obtain the score for the most likely answer. **3.** Semantic Entropy (SE), which considers the meaning of the generated answer while computing entropy, as shown in (8). All 3 baselines depend on length-normalized scoring. We replace length-normalized scoring with MARS and arrive at Confidence + MARS, Entropy + MARS, SE + MARS.

Metrics. Following previous works (Malinin and Gales, 2021; Kuhn et al., 2023), we use Area Under the Receiver Operating Characteristic Curve (AUROC) score for our UE performance metric. AUROC quantifies a method's ability to distinguish between two classes by plotting the true positive rate against the false positive rate for various threshold values. AUROC score is the area under this curve, ranging from 0 to 1. Higher AUROC score indicates a superior performance, while a score of 0.5 implies a random chance. In our case, ground truth is the correctness[2] of the model response to the question and the prediction is the output of an UE method.

[2]We use GPT-3.5-turbo for evaluating the correctness of the model, as in (Lin et al., 2023; Chen and Mueller, 2023).

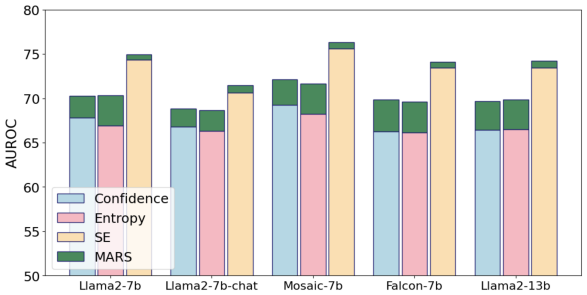

Figure 4: Average AUROC scores of UE methods over 3 different datasets for various LLMs. The improvement of MARS on top of baselines is shown in green.

4.2 Main Results

We present our detailed results in Table 1. Figure 4 illustrates the average AUROC scores for each method and model across three distinct datasets. Upon closer examination of the results, it becomes apparent that the application of MARS consistently improves all baseline methods across various datasets and models. Specifically, MARS yields improvements of up to 5.8 points for Confidence, 6.24 points for Entropy, and 1.51 points for SE.

It is crucial to mention that the choice among the baselines depends on the available computational resources. Confidence score is the least resource-intensive, requiring only a single output generation. Entropy, on the other hand, demands multiple generations (set to 5 in our experiments). SE is the most computationally demanding, needing both multiple generations and $O(n^2)$ Natural Language Inference (NLI) model passes for clustering, where n represents the number of generations.

One of the main contributions of MARS becomes evident when we compare SE with Confidence+MARS or Entropy+MARS. With our method, we are able to increase the scores of Confidence+MARS and Entropy+MARS to a level they can compete with basic SE. Consequently, given the computational overhead of SE, Confidence+MARS and Entropy+MARS emerge as more practical and desirable alternatives. Furthermore, in scenarios where sampling (i.e., multiple answer generation) is not feasible, the improvement offered by MARS to Confidence method becomes crucial with an average increase of 2.8 points. We note that the additional computational and memory demands of MARS are relatively minor, approximately 1.5% of the 7b models and 0.8% of the 13b models, because MARS's importance function is implemented with 110M Bert-like model.

		Method	Llama2-7b	Llama2-7b-chat	Mistral-7b	Falcon-7b	Llama2-13b
TriviaQA		Confidence	70.18	70.40	72.55	68.47	68.19
		Entropy	69.70	69.94	72.57	69.10	69.04
		SE	81.10	76.19	82.17	76.78	79.49
	Ours	Confidence + MARS	**75.06**	**74.23**	**77.97**	**72.95**	**73.99**
		Entropy + MARS	**75.94**	**73.82**	**78.51**	**72.87**	**74.95**
		SE + MARS	**82.22**	**77.67**	**83.63**	**77.48**	**81.00**
NaturalQA		Confidence	68.56	65.98	69.54	63.78	68.56
		Entropy	67.08	65.23	68.05	63.28	68.34
		SE	72.47	68.66	75.12	70.41	73.56
	Ours	Confidence+ MARS	**69.81**	**67.86**	**71.36**	**68.30**	**70.88**
		Entropy + MARS	**69.32**	**67.41**	**70.71**	**67.51**	**70.63**
		SE + MARS	**72.75**	**69.43**	**75.50**	**71.24**	**73.89**
WebQA		Confidence	64.76	64.06	65.66	66.56	62.60
		Entropy	64.04	63.82	64.15	65.98	62.11
		SE	69.44	67.11	69.51	73.16	67.31
	Ours	Confidence + MARS	**66.04**	**64.48**	**67.16**	**68.26**	**64.23**
		Entropy + MARS	**65.83**	**64.69**	**65.76**	**68.44**	**64.02**
		SE + MARS	**69.88**	**67.27**	**69.86**	**73.57**	**67.75**

Table 1: AUROC performance of UE methods in various datasets with different pre-trained LLMs.

	Method	Llama2-7b	Mistral-7b
Token	Confidence + MARS	72.53	75.31
	Entropy + MARS	74.46	77.58
	SE + MARS	81.55	83.25
Phrase	Confidence + MARS	**75.06**	**77.97**
	Entropy + MARS	**75.94**	**78.51**
	SE + MARS	**82.22**	**83.63**

Table 2: AUROC score of UE methods + MARS with token/phrase-level importance functions on TriviaQA.

Method	Distribution	Llama2-7b	Mistral-7b
Confidence + MARS	Min	69.92	72.20
	Max	**75.13**	77.73
	Equal	75.06	**77.97**
Entropy + MARS	Min	70.56	72.75
	Max	**77.11**	**79.22**
	Equal	75.94	78.51
SE + MARS	Min	81.67	82.33
	Max	82.07	**83.62**
	Equal	**82.22**	**83.63**

Table 3: AUROC score of UE methods + MARS with different coefficient distributions in phrases in importance function on TriviaQA.

4.3 Ablation Studies

Effect of Phrase Separation. In Section 3.3, we suggest using a phrase-level separation instead of token-level separation in designing the importance function so that tokens having strong relations are evaluated together on their semantic impact on the sequence. To validate this design, we conduct an experiment where we revert to token-level separation. The results in Table 2 demonstrate that while token-level separation outperforms other baselines, phrase-level separation consistently yields superior results, reaffirming the efficacy of our approach.

Importance Coefficient Distribution in Phrases. In Section 3.3, we state that we equally distribute the importance of phrases to each token. Alternative distribution strategies might include prioritization of the least or most uncertain token. Those strategies assign the phrase importance coefficient to the least or most uncertain token of that phrase. In Table 3, we provide AUROC performances when different distribution strategies are adopted. Notably, we find that max-uncertain distribution is nearly as effective as our adopted equally assigning approach. In contrast, the min-uncertain assigning strategy underperforms. This outcome can be contextualized with a hypothetical scenario: Consider the model's response is "Shakespeare" to the query "Who wrote Hamlet?", which is tokenized into "Shake" and "-speare". Once "Shake" is produced, the subsequent arrival of "-speare" is almost assured. The uncertainty primarily resides in the token "Shake", making the probability of "-speare" relatively uninformative. Consequently, focusing on the least uncertain (most uninformative) token in a phrase drops the performance of MARS significantly, and focusing on the most uncertain token only is still reasonable.

4.4 Effect of Sampling Hyperparameters

We explore the influence of key hyperparameters on the performance of UE methods that rely on sampling, specifically Entropy and SE. We focus on two critical hyperparameters: Temperature, which adjusts the diversity of the sampling process, and the number of sampling, which dictates how many samples are sampled in entropy calculation.

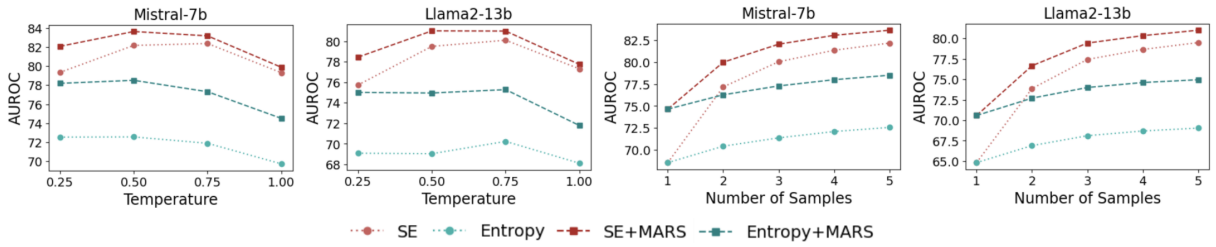

Figure 5: AUROC scores for various temperatures and sampling numbers.

Temperature. The temperature parameter determines the smoothness of the probabilities while sampling. A higher (lower) temperature value indicates more (less) diverse sampling. Figure 5 presents the AUROC scores for Entropy, SE, and their enhancements via MARS for the Llama2-13b and Mistral-7b models on the TriviaQA dataset. The improvement of MARS is consistent for all temperature values. The choice of temperature is application-dependent: higher temperatures are advisable for tasks demanding creativity, whereas lower temperatures are preferable for applications where consistency is important.

Number of Sampling. The number of sampled sequences is important for entropy and semantic entropy calculation. More sampling leads to better entropy estimation; however, the cost also increases. Beyond the sampling expense, SE incurs an additional cost from Natural Language Inference (NLI) model passes, a point elaborated in Section 4.2. In Figure 5, we provide the AUROC performance of Llama2-13b and Mistral-7b models on TriviaQA with various sampling numbers. Notably, the efficacy of MARS remains stable across diverse sampling numbers, with its advantages becoming more obvious under lower sampling numbers.

4.5 UE in Medical QA Dataset

Next, we evaluate the UE methods using a medical QA dataset. Publicly available medical QA datasets typically fall into two categories: those with multiple-choice questions (Pal et al., 2022; Kotonya and Toni, 2020; Jin et al., 2021) and those without clear ground truths (Zhu et al., 2019, 2020). To tackle this, we create a subset from the MedMCQA multiple-choice dataset (Pal et al., 2022), selecting questions that can be answered objectively without multiple choices. For this, we collaborate with medical professionals to ensure the accuracy and relevance of the selected questions, yielding a dataset of 415 samples. We use AdaptLLM's Medicine-Chat (Cheng et al., 2023), a medical-domain adapted LLaMA-2-Chat-7B model[3]. To evaluate the correctness of model-generated responses, we leverage GPT-4 (OpenAI, 2023) and assess response validity in the medical domain.

In Table 4, we provide the AUROC performance of the UE methods. Although MARS still consistently improves the performance of probability-based UE methods, AUROC scores are still low compared to Table 1. This might be because of the nature of medical questions. General knowledge questions mostly require a straight, single-sentence answer. On the other hand, although we curated closed-ended questions, medical questions still require a more complex explanation spanning multiple sentences. This difference between domains can affect the prediction performance of the probability-based methods. This observation emphasizes the necessity for further investigation across various specialized fields, including medicine and law. Customized explorations are essential to address domain-specific challenges and optimize UE methods accordingly.

	Method	Medicine-Chat-7b
	Confidence	62.41
	Entropy	59.58
	SE	62.89
Ours	Confidence + MARS	**62.89**
Ours	Entropy + MARS	**60.33**
Ours	SE + MARS	**64.48**

Table 4: AUROC score of UE methods on medical QA.

5 Conclusion

We introduce Meaning-Aware Response Scoring (MARS), a novel scoring function designed to replace length-normalized scoring in probability-based UE methods when evaluating generative LLMs. MARS consistently and significantly boosts the performance of current probability-based UE methods with minimal additional computational overhead. The efficacy of MARS is shown in three closed-book and closed-ended question-answer datasets and a medical question-answer dataset.

[3]https://huggingface.co/AdaptLLM/medicine-chat

6 Limitations

The importance function model within MARS utilizes an unsupervised methodology, leveraging pre-existing models for its formulation. Nonetheless, the performance of MARS can potentially be further enhanced by using human labelers to assign importance coefficients for training the importance function model. Besides, our analysis is limited to the closed-ended question-answering domain in English, where a question has an objective ground-truth answer(s). Extensive analysis of MARS and other probability-based UE methods on open-ended question-answering tasks and other languages are beyond the scope of the current study and are left as future work.

7 Ethics Statement

Although probability-based UE methods combined with MARS have a remarkable prediction performance on the correctness of generative LLM outputs, it is crucial to acknowledge that these methods do not achieve 100% accuracy. Besides, as LLMs may have biases against gender, ethnicity, age, etc., probability-based methods can carry those biases to UE outputs. Thus, one should be aware of these potential risk factors before employing such probabilistic UE methods in real-world systems. Ensuring fairness, transparency, and accountability in the deployment of these technologies is important in mitigating risks and fostering trust in their application.

References

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. The Falcon series of open language models.

Neil Band, Tim G. J. Rudner, Qixuan Feng, Angelos Filos, Zachary Nado, Michael W Dusenberry, Ghassen Jerfel, Dustin Tran, and Yarin Gal. 2021. Benchmarking Bayesian deep learning on diabetic retinopathy detection tasks. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Jannis Bulian, Christian Buck, Wojciech Gajewski, Benjamin Börschinger, and Tal Schuster. 2022. Tomayto, tomahto. beyond token-level answer equivalence for question answering evaluation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 291–305.

Yingshan Chang, Guihong Cao, Mridu Narang, Jianfeng Gao, Hisami Suzuki, and Yonatan Bisk. 2022. WebQA: Multihop and Multimodal QA. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16474–16483.

Jiuhai Chen and Jonas Mueller. 2023. Quantifying uncertainty in answers from any language model and enhancing their trustworthiness.

Daixuan Cheng, Shaohan Huang, and Furu Wei. 2023. Adapting large language models via reading comprehension.

Roi Cohen, May Hamri, Mor Geva, and Amir Globerson. 2023. LM vs LM: Detecting factual errors via cross examination. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12621–12640, Singapore. Association for Computational Linguistics.

Shrey Desai and Greg Durrett. 2020. Calibration of pre-trained transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 295–302, Online. Association for Computational Linguistics.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. 2020. Unsupervised quality estimation for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:539–555.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48, pages 1050–1059.

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

Mengting Hu, Zhen Zhang, Shiwan Zhao, Minlie Huang, and Bingzhe Wu. 2023. Uncertainty in natural language processing: Sources, quantification, and applications.

Qian Huang, Jian Vora, Percy Liang, and Jure Leskovec. 2023. Benchmarking large language models as AI research agents.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021. How can we know when language models know? On the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14).

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.

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. 2022. Language models (mostly) know what they know.

Neema Kotonya and Francesca Toni. 2020. Explainable automated fact-checking for public health claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7740–7754, Online. Association for Computational Linguistics.

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

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, page 6405–6416.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2023. Generating with confidence: Uncertainty quantification for black-box large language models.

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

OpenAI. 2023. GPT-4 Technical Report.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. MedMCQA: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Proceedings of the Conference on Health, Inference, and Learning*, volume 174 of *Proceedings of Machine Learning Research*, pages 248–260.

Yichen Shen, Zhilu Zhang, Mert R Sabuncu, and Lin Sun. 2021. Real-time uncertainty estimation in computer vision via uncertainty-aware distribution distillation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 707–716.

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. 2023. Llama 2: Open foundation and fine-tuned chat models.

Artem Vazhentsev, Gleb Kuzmin, Artem Shelmanov, Akim Tsvigun, Evgenii Tsymbalov, Kirill Fedyanin, Maxim Panov, Alexander Panchenko, Gleb Gusev, Mikhail Burtsev, Manvel Avetisian, and Leonid Zhukov. 2022. Uncertainty estimation of transformer predictions for misclassification detection. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long*

Papers), pages 8237–8252, Dublin, Ireland. Association for Computational Linguistics.

Tim Z. Xiao, Aidan N. Gomez, and Yarin Gal. 2020. Wat zei je? detecting out-of-distribution translations with variational transformers. *CoRR*, abs/2006.08344.

Yijun Xiao and William Yang Wang. 2019. Quantifying uncertainties in natural language processing tasks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):7322–7329.

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

Junjie Ye, Xuanting Chen, Nuo Xu, Can Zu, Zekai Shao, Shichun Liu, Yuhan Cui, Zeyang Zhou, Chao Gong, Yang Shen, Jie Zhou, Siming Chen, Tao Gui, Qi Zhang, and Xuanjing Huang. 2023. A comprehensive capability analysis of GPT-3 and GPT-3.5 series models.

Ming Zhu, Aman Ahuja, Da-Cheng Juan, Wei Wei, and Chandan K. Reddy. 2020. Question answering with long multiple-span answers. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3840–3849, Online. Association for Computational Linguistics.

Ming Zhu, Aman Ahuja, Wei Wei, and Chandan K. Reddy. 2019. A hierarchical attention retrieval model for healthcare question answering. In *The World Wide Web Conference*, WWW '19, page 2472–2482, New York, NY, USA. Association for Computing Machinery.

A Related Works

Uncertainty Estimation (UE) has emerged as a vital concept in various machine learning domains, particularly in Natural Language Processing (NLP). The study of Xiao et al. (2022) concentrates on the UE for tasks like common-sense reasoning and sentiment analysis; Jiang et al. (2021) explores model calibration for UE in the context of multiple-choice question answering; Desai and Durrett (2020) tackles the challenge of UE in specific NLP tasks such as paraphrase detection and natural language inference. These studies represent just a fraction of the UE works in the field of NLP and there is an expanding corpus of research focusing on the investigation of UE in NLP (Hu et al., 2023; Xiao and Wang, 2019; Vazhentsev et al., 2022). The vast majority of these studies only focus on classification and regression tasks, unlike our work where the goal is to study UE for generative LLMs.

Few recent works deal with UE of generative LLMs. Xiao et al. (2020) and Fomicheva et al. (2020) propose heuristic-based uncertainty metrics for generative LLMs considering machine translation. Chen and Mueller (2023), Lin et al. (2023), Cohen et al. (2023), and Kadavath et al. (2022) propose black-box UE methods for generative LLMs under the assumption that the token probabilities are not accessible. Although these works have experimental validation, they lack a mathematical foundation. Malinin and Gales (2021) is the first study adapting popular uncertainty tools in Bayesian UE literature to the generative LLMs. The main idea of Malinin and Gales (2021) is to utilize length-normalized scoring in computing the entropy of the LLM answers. A more recent approach by Kuhn et al. (2023) further improves this result by introducing the concept of semantic entropy, which considers the meaning of the generated sentences in entropy calculation in uncertainty prediction. Our work is distinct from these works as we no longer utilize length-normalized scoring. Instead, we utilize the proposed MARS in entropy computations, by also taking into consideration token importance to the answer correctness , thereby achieving an improved UE performance.

B Conceptualizing the Response Semantics in Generative LLM Probabilities

In classification tasks, the *class probability* reflects the model's confidence in assigning a specific class to an input. It is inherently tied to the *semantics* of the class. For instance, if a well-calibrated classifier gives a 75% probability to the label "cat" for a given question, it suggests a 75% likelihood that the answer of the question is indeed a cat. This output probability is not only a numerical value; it conveys a semantic understanding of the image content as a cat. However, previously proposed length-normalized scoring and semantic entropy definitions for generative LLMs (Sections 2.3 and 2.4) do not directly correspond to the semantics of the LLM generation. Moreover, they are not proper probability and entropy definitions, lacking theoretical background. Hence, we propose a new random variable that is directly related to the semantics of the output and provide a justification for

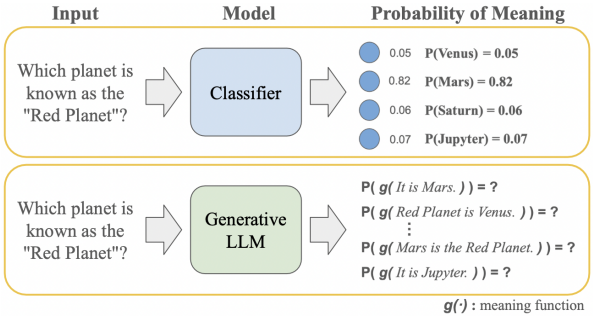

Figure 6: In classification tasks, output probabilities give the probability of the semantic meaning. In the case of generative LLMs, probabilities of semantic meaning are unknown. Thus, we propose an alternative probability distribution MARS for generative LLMs.

the heuristic decisions of the previous works (Kuhn et al., 2023; Malinin and Gales, 2021).

Let Y be a random variable with arbitrary dimension corresponding to the meaning of the sequences generated by an LLM parametrized with θ. The values of Y can be the set of all possible meanings of generated sequences in a given context. Formally, the set is $\{g(\mathbf{s}, \mathbf{x})\}_{\mathbf{s}\in\mathcal{S},\mathbf{x}\in\mathcal{X}}$, where $g(\cdot)$ is the meaning function that takes generated sentence $\mathbf{s}$ and context $\mathbf{x}$ as inputs and returns the meaning as output. By defining the properties of the meaning function $g(\cdot)$ and the distribution of Y, we can rationalize the heuristic design choices made by previous works.

Malinin and Gales (2021) considers $g(\cdot)$ as a one-to-one function which means that each unique sentence in the given context corresponds to different meanings. In this case, the distribution of Y is defined by using the length-normalized scoring of the generated sequences. More formally

$$P(Y = y|\theta) = \frac{\tilde{P}(\mathbf{s}|\mathbf{x}, \theta)}{\sum_{\mathbf{s}\in\mathcal{S},\mathbf{x}\in\mathcal{X}} \tilde{P}(\mathbf{s}|\mathbf{x}, \theta)}, \qquad (11)$$

where $y = g(\mathbf{s}, \mathbf{x})$ and $\tilde{P}(\mathbf{s}|\mathbf{x}, \theta)$ is the length-normalized scoring defined as $\prod_{l=1}^{L} P(s_l|s_{<l}, x; \theta)^{1/L}$. To make the distribution of Y a valid probability distribution, we normalize each $\tilde{P}(\mathbf{s}|\mathbf{x}, \theta)$ by the sum of all possible scores, making their summation 1. By defining Y as above, we essentially create an actual probability distribution of length-normalized scoring.

On the other hand, Kuhn et al. (2023) claims different sequences can have equal meaning. By considering $g(\cdot)$ as a many-to-one function, we can write their proposal with the new meaning random variable Y as follows

$$P(Y = y|\theta) = \frac{\sum_{\mathbf{s},\mathbf{x}\in c_y} \tilde{P}(\mathbf{s}|\mathbf{x}, \theta)}{\sum_{\mathbf{s}\in\mathcal{S},\mathbf{x}\in\mathcal{X}} \tilde{P}(\mathbf{s}|\mathbf{x}, \theta)} \qquad (12)$$

where c_y corresponds to the meaning cluster, formally written as $c_y = \{\mathbf{s}, \mathbf{x}|g(\mathbf{s}, \mathbf{x}) = y\}$. By employing this new probability definition within the standard entropy calculation in (6), we obtain the concept of semantic entropy as follows

$$SE(\mathbf{x}, \theta) = -\frac{1}{B}\sum_{b=1}^{B} \log P(Y = y_b|\theta) \qquad (13)$$

With the new random variable Y, we essentially write the semantic entropy as the standard Monte-Carlo approximated entropy over a total of B distinct meanings.

Notice that the normalization term $\sum_{\mathbf{s}\in\mathcal{S},\mathbf{x}\in\mathcal{X}} \tilde{P}(\mathbf{s}|\mathbf{x}, \theta)$ featured in both (11) and (12), acts as a constant across all $P(Y = y|\theta)$ calculations, ensuring that Y conforms to a valid probability distribution. Therefore, it only shifts the proposed UE scores which does not affect the performance of accurately predicting the correctness of the model generation. Moreover, by introducing the random variable Y, we not only provide a theoretical foundation for heuristic choices of the previous works but also create flexibility to define new distributions for Y which may potentially improve the existing UE tools.

Using the definition of Y, we can also rationalize our scoring function MARS. We replace the length-normalized scoring function with MARS as in (9). We believe that MARS is a better choice to define the probability distribution of Y. This is because MARS considers the semantic contribution of tokens and the values of Y are closely related to the semantics of the generated sentences in the context of question.

Once we do that, the new probability distribution of $P(Y = y|\theta)$ becomes the following if we consider g as a one-to-one function as the work of Malinin and Gales (2021)

$$P(Y = y|\theta) = \frac{\bar{P}(\mathbf{s}|\mathbf{x}, \theta)}{\sum_{\mathbf{s}\in\mathcal{S},\mathbf{x}\in\mathcal{X}} \bar{P}(\mathbf{s}|\mathbf{x}, \theta)}. \qquad (14)$$

If we follow Kuhn et al. (2023) and make g a many-to-one function, we reach the following distribution for $P(Y = y|\theta)$:

$$P(Y = y|\theta) = \frac{\sum_{\mathbf{s},\mathbf{x}\in C_y} \bar{P}(\mathbf{s}|\mathbf{x}, \theta)}{\sum_{\mathbf{s}\in\mathcal{S},\mathbf{x}\in\mathcal{X}} \bar{P}(\mathbf{s}|\mathbf{x}, \theta)}. \qquad (15)$$

Overall, by defining the new random variable Y and the properties of meaning function $g(\cdot)$, we build a theoretical background for the heuristic design choices of previous works (Malinin and Gales, 2021; Kuhn et al., 2023). Moreover, this structure provides a background for further studies by either changing length-normalized scoring (as we do with MARS) or by re-defining the probability distribution of Y and properties of the meaning function $g(\cdot)$.

C Training of BERT-like Model for Importance Function

As described in Section 3.3, we optimize the computational efficiency of MARS by training a single Bert-like model with 110M parameters to execute the importance function. This model is an adaptation of the pre-trained Bert-base-uncased[4], modified by removing its last layer and incorporating two independent fully-connected (FC) layers. The first FC layer focuses on phrase detection with two output logits: "Begin Phrase" (BP) and "Inside Phrase" (IP), and classifies each token as BP if it marks the start of a phrase or as IP otherwise. This setup enables sentence segmentation into phrases. The second FC layer, tasked with assigning importance coefficients, produces a single output logit for each token's importance coefficient.

For training data, we take a subset of 69192 question samples from the TriviaQA training set and questions of the whole training set of NaturalQA consisting of 87925. Then, we use these questions as input and feed them to all 7B-sized baseline models (Llama2-7b, Llama2-7b-chat, Mistral-7b, Falcon-7b) to yield the responses. This provides us with question-answer pairs. We use the Flair phrase chunking model to determine phrase labels in the answers, as described in Appendix C.1. For importance coefficient labels per token in the responses, we follow Algorithm 1.

Sample outputs of our model are provided in Table 5. Here, question and answer are inputs to the model, and the model divides the answer into phrases while assigning importance score to them.

We train the model only for 1 epoch with 5e-5 learning rate and 32 batch size. The training process involves a convex combination of two loss functions: cross-entropy for phrase chunking and negative log-likelihood for importance coefficient assignment, with equal weight assigned to both losses. Table 6 displays the training and validation losses at the end of the training, indicating that our training objectives are effectively generalizable to test sets.

C.1 Dividing a Sentence to Phrases

To divide a sentence into phrases, we use the Flair phrase chunking model[5] (Akbik et al., 2018), that uses 10 tags which are adjectival, adverbial, conjunction, interjection, list marker, noun phrase, prepositional, particle, subordinate clause and verb phrase. For example, the Flair model divides the sentence "The happy man has been eating at the dinner" as "The happy man", "has been eating", "at", "the diner".

C.2 Pseudocode of the Importance Function Algorithm

The pseudocode of the importance function algorithm is given in Algorithm 1.

Algorithm 1 Phrase-Level Importance Function

Input: Question $\mathbf{x}$, generated answer $\mathbf{s} = \{s_1, s_2, \ldots, s_L\}$, phrases $\{h_1, h_2, \ldots, h_K\}$, token probabilities $\{p_i = P(s_i|s_{<i}, \mathbf{x}; \theta)\}_{s_i \in \mathbf{s}}$, temperature τ
Output: Importance scores I
$I \leftarrow []$
for $k = 1$ to K **do**
  $\mathbf{s}_{\text{masked}} \leftarrow \mathbf{s} \setminus \{s_l\}_{s_l \in h_k}$
  $o_k \leftarrow BEM(\mathbf{x}, \mathbf{s}, \mathbf{s}_{\text{masked}})$
  for each token s_l in phrase h_k **do**
    $I[l] \leftarrow (1 - o_k)/|h_k|$
$I \leftarrow softmax(I, \tau)$
return I

D Experimental Details

Datasets. We employ the validation split of the Natural Questions dataset, comprising 3610 samples. Following Kuhn et al. (2023), a subset of 8000 QA pairs is selected from the validation split of the TriviaQA dataset. For WebQA, we combine its training and test splits to form a combined dataset of 6642 samples.

Example Samples from Datasets. We provide data samples from the datasets we used in the evaluation of UE methods in Table 7.

[4] https://huggingface.co/bert-base-uncased

[5] https://huggingface.co/flair/chunk-english

Question	Answer	Output
Which planet is known as Red Planet?	It is Mars	It (0.017) is (0.017) Mars (0.956)
What is the capital city of Japan?	Tokyo is the capital city of Japan	Tokyo (0.994) is (0.001) the capital city (0.003) of (0.001) Japan (0.001)
Which element has the chemical symbol "O"?	The chemical symbol "O" represents Oxygen	The chemical symbol (0.01) "O" (0.01) represents (0.003) Oxygen (0.976)

Table 5: Sample outputs of our BERT-like model used for importance function. Question and answer are given to the model as input, and the model divides the answer into phrases while assigning importance score.

	Classification Loss	Scoring Loss
Train	0.0275	0.1957
Validation	0.0205	0.1901

Table 6: Train and validation loss values calculated at the end of training of BERT-like importance model. Classification loss stands for cross-entropy loss for phrase chunking, and Scoring loss indicated negative log-likelihood loss for importance coefficient.

Number of Sampling and Temperature. Following previous work (Kuhn et al., 2023), we sampled 5 samples and used 0.5 as the temperature value for the results presented in Table 1.

Generation Configurations. We use the Huggingface library's generate function for model generations. We set token "." as eos_token_id which prevents model to generate long paragraphs to closed-book questions. We set num_beams = 1 which corresponds to greedy decoding.

Computational Cost. We use 40 GB Nvidia A-100 GPUs for all the experiments. The total GPU-hours for Table 1 is approximately 400. Labeling of the data used for training of BERT-like importance model takes approximately 200 GPU-hours. Fine-tuning of BERT-like model on the importance dataset takes 7 GPU-hours. Due to expensive computational demands, all presented results are the output of a single run.

Prompts. We use the same 2-shot prompt for all of the models and the datasets for answer generation:

```
Answer these questions:
Question: What is the capital city of
Australia?
Answer: The capital city of Australia is
Canberra.
Question: Who painted the famous
artwork "Starry Night"?
Answer: "Starry Night" was painted by
Vincent van Gogh.
Question: {sample['question']}?
Answer:
```

To evaluate the correctness of the generated answer, we use gpt-3.5-turbo as the evaluator. The prompt for gpt-3.5-turbo is the following:

```
You will behave as a question-answer
evaluator. I will give you a question,
the ground truth of the question
and a generated answer by a language
model. You will output "correct"
if the generated answer is correct
regarding question and ground truth.
Otherwise, output "false".
Question: {question}?,
Ground Truth: {answer},
Generated Answer: {generation}
```

	Question	Answer
TriviaQA	Which American-born Sinclair won the Nobel Prize for Literature in 1930?	Sinclair Lewis
	Which musical featured the song Thank Heaven for Little Girls?	Gigi
	What was the first movie western called?	Kit Carson
NaturalQA	When did the eagles win last super bowl?	2017
	Who was the ruler of england in 1616?	James I
	What is the hot coffee mod in san andreas?	a normally inaccessible mini-game
WebQA	what character did natalie portman play in star wars?	Padmé Amidala
	what country is the grand bahama island in?	Bahamas
	where did saki live?	United Kingdom

Table 7: Data samples from the datasets we use to evaluate UE methods: TriviaQA, NaturalQA, and WebQA.