LM-Polygraph: Uncertainty Estimation for Language Models

Ekaterina Fadeeva[3,5] ◇ **Roman Vashurin**[2] ◇ **Akim Tsvigun**[5,6,7] ◇ **Artem Vazhentsev**[3,4] ◇
Sergey Petrakov[3] ◇ **Kirill Fedyanin**[2] **Daniil Vasilev**[5] **Elizaveta Goncharova**[4,5,6]
Alexander Panchenko[3,4] **Maxim Panov**[1] **Timothy Baldwin**[1,8] **Artem Shelmanov**[1]
[1]MBZUAI [2]TII [3]Center for Artificial Intelligence Technology [4]AIRI
[5]HSE University [6]AI Center NUST MISiS [7]Semrush [8]The University of Melbourne
{ekaterina.fadeeva, sergey.petrakov}@skol.tech {roman.vashurin, kirill.fedyanin}@tii.ae
akim.tsvigun@semrush.com {vazhentsev, panchenko, goncharova}@airi.net
{maxim.panov, timothy.baldwin, artem.shelmanov}@mbzuai.ac.ae

Abstract

Recent advancements in the capabilities of large language models (LLMs) have paved the way for a myriad of groundbreaking applications in various fields. However, a significant challenge arises as these models often “hallucinate”, i.e., fabricate facts without providing users an apparent means to discern the veracity of their statements. Uncertainty estimation (UE) methods are one path to safer, more responsible, and more effective use of LLMs. However, to date, research on UE methods for LLMs has been focused primarily on theoretical rather than engineering contributions. In this work, we tackle this issue by introducing LM-Polygraph, a framework with implementations of a battery of state-of-the-art UE methods for LLMs in text generation tasks, with unified program interfaces in Python.[1] Additionally, it introduces an extendable benchmark for consistent evaluation of UE techniques by researchers, and a demo web application that enriches the standard chat dialog with confidence scores, empowering end-users to discern unreliable responses.[2,3] LM-Polygraph is compatible with the most recent LLMs, including BLOOMz, LLaMA-2, ChatGPT, and GPT-4, and is designed to support future releases of similarly-styled LMs.

1 Introduction

Large language models (LLMs) have demonstrated remarkable performance across a variety of text generation tasks. Instruction fine-tuning and reinforcement learning from human feedback (RLHF) have brought the zero-shot performance of these models to a new level (Ouyang et al., 2022). However, the capabilities of LLMs, despite their profound power and complexity, are inherently constrained. Limitations arise from the finite nature of the training data and the model’s intrinsic memorization and reasoning capacities. Hence, their utility is bounded by the depth and breadth of the knowledge they embed.

Due to their training objectives, even when the embedded knowledge of an LLM on a given topic is limited, it tends to be over-eager to respond to a prompt, sometimes generating misleading or entirely erroneous output. This dangerous behavior of attempting to appease the user with plausible-sounding but potentially false information is known as “hallucination” (Xiao and Wang, 2021; Dziri et al., 2022). It poses a significant challenge when deploying LLMs in practical applications.

There are several well-known approaches to censoring LLM outputs, including: filtering with stop-word lists, post-processing with classifiers (Xu et al., 2023), rewriting of toxic outputs (Logacheva et al., 2022), and longer fine-tuning with RLHF. However, these approaches cannot be relied on to completely resolve hallucinations. Since LMs are natural (if “unintentional”) liars, we propose LM-Polygraph — a program framework that, similar to a human polygraph, leverages various hidden signals to reveal when one should not trust the subject. In particular, LM-Polygraph provides a comprehensive collection of uncertainty estimation (UE) techniques for LLMs in text generation tasks.

Uncertainty estimation refers to the process of quantifying the degree of confidence in the predictions made by a machine learning model. For classification and regression tasks, there is a well-developed battery of methods (Gal, 2016). There has also been a surge of work investigating UE, particularly in text classification and regression in conjunction with encoder-only LMs such as BERT (Zhang et al., 2019; He et al., 2020; Shelmanov et al., 2021; Xin et al., 2021; Vazhentsev et al., 2022; Kotelevskii et al., 2022; Wang et al., 2022; Kuzmin et al., 2023). However, UE for sequence generation tasks, including text generation,

[1] `http://lm-polygraph.nlpresearch.group`
[2] `http://lm-polygraph-demo.nlpresearch.group`
[3] `http://lm-polygraph-video.nlpresearch.group`
◇ Equal contribution

Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 446–461
December 6-10, 2023 ©2023 Association for Computational Linguistics

is a much more complex problem. To quantify the uncertainty of the whole sequence, we have to aggregate uncertainties of many individual token predictions and deal with non-trivial sampling and pruning techniques like beam search. Contrary to classification tasks where the number of possible prediction options is finite, in text generation, the number of possible predictions is infinite or exponential in vocabulary size, complicating the estimation of probabilities and information-based scores. Finally, a natural language text is not a simple sum of its tokens; it is a nuanced interposition of context, semantics, and grammar, so two texts can have very diverse surface forms but similar meanings, which should be taken into account during the UE process.

Several recent studies have delved into developing UE methods for LMs in text generation tasks (Malinin and Gales, 2021; van der Poel et al., 2022; Kuhn et al., 2023; Ren et al., 2023; Vazhentsev et al., 2023b; Lin et al., 2023). However, the current landscape of this research is quite fragmented with many non-comparable or even concurrent studies, which makes it challenging to consolidate the findings and draw holistic conclusions.

In this work, with the development of LM-Polygraph, we strive to bridge these disparate research efforts, fostering more cohesion and synergy in the field. We envision a framework that consolidates the scattered UE techniques within unified frameworks in Python, provides an extendable evaluation benchmark, and offers tools to integrate uncertainty quantification in standard LLM pipelines seamlessly. This endeavor will not only make the journey less challenging for individual researchers and developers but also set the stage for more robust, reliable, and trustworthy LLM deployments for end-users.

Our **contributions** are as follows:

- We provide a comprehensive framework that implement state-of-the-art methods for UE of LM predictions. We also provide the ability to combine multiple uncertainty scores together as suggested by Ren et al. (2023); Vazhentsev et al. (2023a).
- We create a tool that enriches standard LLM chat capabilities with uncertainty scores for model outputs. The tool can potentially be used by end-users to determine whether the answers of language models are reliable or not, and by researchers to develop novel UE techniques for LMs in text generation tasks.
- We construct an easy-to-extend benchmark for UE methods in text generation tasks and provide reference experimental results for implemented UE techniques.

```
from lm_polygraph import estimate_uncertainty
from lm_polygraph import WhiteboxModel
from lm_polygraph.estimators import *

model = WhiteboxModel.from_pretrained(
    "bigscience/bloomz-3b",
    device="cuda:0",
)
ue_method = MeanPointwiseMutualInformation()

input_text = "Who is George Bush?"
estimate_uncertainty(model, ue_method, input_text)

# Output:
# UncertaintyOutput(
#   generation='President of the United States',
#   uncertainty=-6.858096446298684)
```

Figure 1: Code example of how LM predictions could be enriched with uncertainty scores using LM-Polygraph.

2 Python Library

LM-Polygraph implements a set of state-of-the-art UE techniques for LLMs with unified program interfaces in Python. It is compatible with models from the Huggingface library and is tested with recent public-domain LLMs such as BLOOMz (Scao et al., 2022; Yong et al., 2023), Dolly v2 (Conover et al., 2023), Alpaca (Taori et al., 2023), LLaMA-2 (Touvron et al., 2023), and Flan-T5 (Chung et al., 2022). The framework supports both conditional models with a seq2seq architecture and unconditional decoder-only LMs. Figure 1 contains a code example of LM-Polygraph with BLOOMz-3B for UE in open-domain question answering. Some methods that do not require access to the model itself or its logits could be used in conjunction with web-hosted LLMs like ChatGPT or GPT-4 through APIs. We provide a program wrapper for integration with popular online services.

3 Uncertainty Estimation Methods

Here, we summarize UE methods implemented in LM-Polygraph, as listed in Table 1.

There are two major technique types: white-box and black-box. The *white-box* methods require access to logits, internal layer outputs, or the LM itself. The *black-box* methods require access only to the generated texts, and can easily be integrated with third-party online services like OpenAI LM API. We note that the methods differ by computational requirements: some techniques pose high

Uncertainty Estimation Method	Type	Category	Compute	Memory	Need Training Data?
Maximum sequence probability	White-box	Information-based	Low	Low	No
Perplexity (Fomicheva et al., 2020)			Low	Low	No
Mean token entropy (Fomicheva et al., 2020)			Low	Low	No
Monte Carlo sequence entropy (Kuhn et al., 2023)			High	Low	No
Pointwise mutual information (PMI) (Takayama and Arase, 2019)			Medium	Low	No
Conditional PMI (van der Poel et al., 2022)			Medium	Medium	No
Semantic entropy (Kuhn et al., 2023)	White-box	Meaning diversity	High	Low	No
Sentence-level ensemble-based measures (Malinin and Gales, 2021)	White-box	Ensembling	High	High	Yes
Token-level ensemble-based measures (Malinin and Gales, 2021)			High	High	Yes
Mahalanobis distance (MD) (Lee et al., 2018)	White-box	Density-based	Low	Low	Yes
Robust density estimation (RDE) (Yoo et al., 2022)			Low	Low	Yes
Relative Mahalanobis distance (RMD) (Ren et al., 2023)			Low	Low	Yes
Hybrid Uncertainty Quantification (HUQ) (Vazhentsev et al., 2023a)			Low	Low	Yes
p(True) (Kadavath et al., 2022)	White-box	Reflexive	Medium	Low	No
Number of semantic sets (NumSets) (Lin et al., 2023)	Black-box	Meaning diversity	High	Low	No
Sum of eigenvalues of the graph Laplacian (EigV) (Lin et al., 2023)			High	Low	No
Degree matrix (Deg) (Lin et al., 2023)			High	Low	No
Eccentricity (Ecc) (Lin et al., 2023)			High	Low	No
Lexical similarity (LexSim) (Fomicheva et al., 2020)			High	Low	No

Table 1: UE methods implemented in LM-Polygraph.

computational or memory overheads, e.g., due to repeated inference, making them less suitable for practical usage. The application of some methods also can be hindered by the need for access to the model training data.

Let us consider the input sequence $\mathbf{x}$ and the output sequence $\mathbf{y} \in \mathcal{Y}$ of length L, where $\mathcal{Y}$ is a set of all possible output sequences. Then the probability of an output sequence given an input sequence for probabilistic autoregressive language models is given by:

$$P(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}) = \prod\nolimits_{l=1}^{L} P(y_l \mid \mathbf{y}_{<l}, \mathbf{x}, \boldsymbol{\theta}), \qquad (1)$$

where the distribution of each y_l is conditioned on all the previous tokens in a sequence $\mathbf{y}_{<l} = \{y_1, \dots, y_{l-1}\}$, and $\boldsymbol{\theta}$ denotes the parameters of the model.

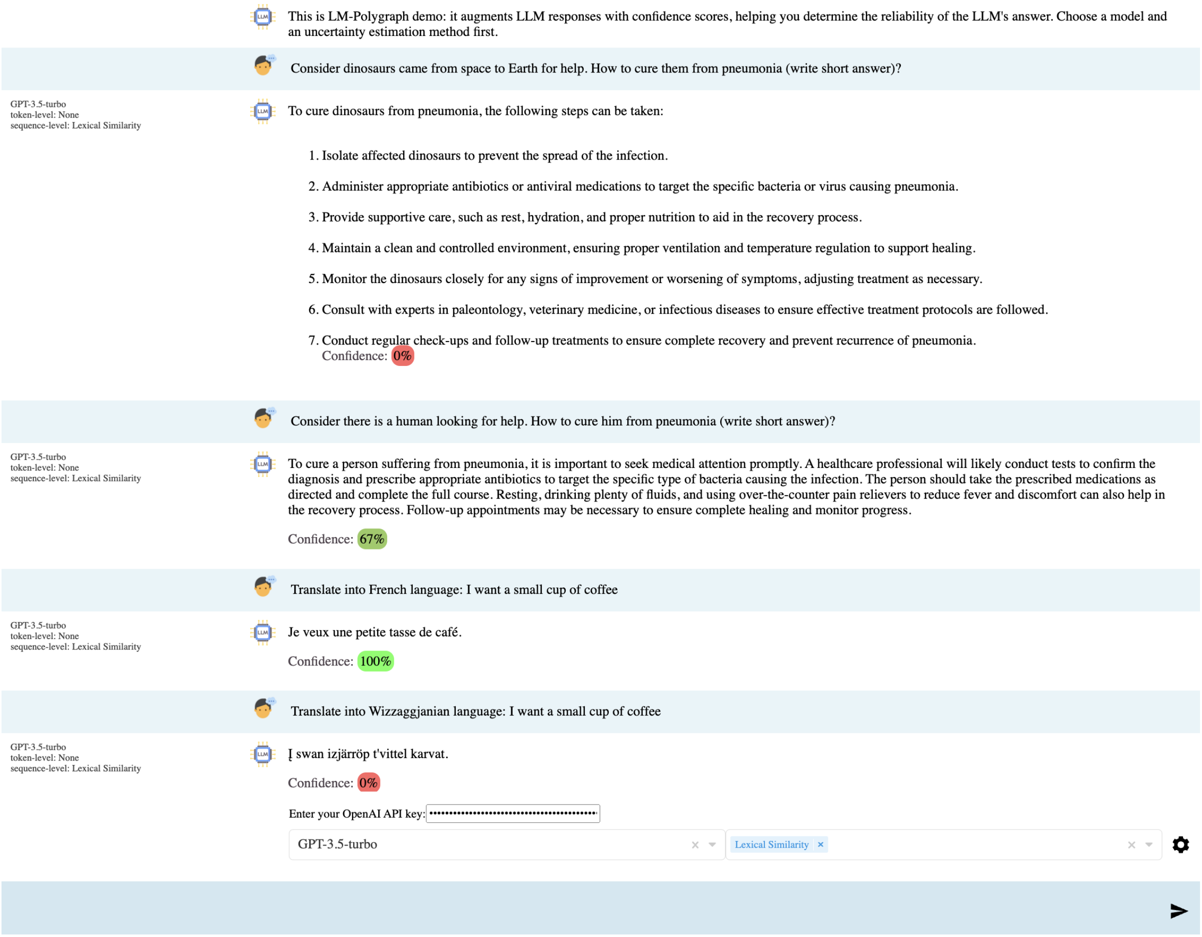
3.1 White-box Methods

We start the discussion of white-box techniques from **information-based methods**. These techniques are based on token $P(y_l \mid \mathbf{y}_{<l}, \mathbf{x}, \boldsymbol{\theta})$ and sequence $P(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta})$ probabilities obtained from a single model prediction. The notable example is *entropy*, which can be calculated on the token or sequence level. The benefits of information-based methods are that they are cheap to compute and simple to implement. However, the quality of these methods is usually relatively low, so they are considered as baselines. Some domain-specific methods were recently proposed in an attempt to improve over standard information-based approaches, such as *semantic entropy* (Kuhn et al., 2023).

The second category of white-box techniques is **ensemble-based methods**, which leverage the diversity of output predictions made by multiple slightly different versions of models under slightly different conditions. Let us assume that M models are available with parameters $\boldsymbol{\theta}_i, i = 1, \dots, M$. These parameters can be obtained via independent training of models. Then one can use token $P(y_l \mid \mathbf{y}_{<l}, \mathbf{x}, \boldsymbol{\theta}_i)$ and sequence $P(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}_i)$ probabilities to compute various metrics such as *mutual information* that measures the discrepancy between model predictions.

Density-based methods leverage latent representations of instances and construct a probability density on top of them. Usually, these methods approximate training data distribution with the help of one or multiple Gaussian distributions. They can provide a probability or an unnormalized score that determines how likely instances belong to the training data distribution. Therefore, they are good at spotting out-of-distribution (OOD) instances (Vazhentsev et al., 2023b). Several variations of these methods have been proposed in the literature (Lee et al., 2018; Yoo et al., 2022; Ren et al., 2023; Kotelevskii et al., 2022).

The primary advantage of these methods is that they are computationally efficient: they do not need much time for additional model inference, and memory overhead for storing additional parameters is minimal. The drawback is that these methods require access to the model's training data to fit auxiliary models like Gaussians (e.g., the Mahalanobis Distance method requires constructing



This is LM-Polygraph demo: it augments LLM responses with confidence scores, helping you determine the reliability of the LLM's answer. Choose a model and an uncertainty estimation method first.

Consider dinosaurs came from space to Earth for help. How to cure them from pneumonia (write short answer)?

GPT-3.5-turbo
token-level: None
sequence-level: Lexical Similarity

To cure dinosaurs from pneumonia, the following steps can be taken:

1. Isolate affected dinosaurs to prevent the spread of the infection.
2. Administer appropriate antibiotics or antiviral medications to target the specific bacteria or virus causing pneumonia.
3. Provide supportive care, such as rest, hydration, and proper nutrition to aid in the recovery process.
4. Maintain a clean and controlled environment, ensuring proper ventilation and temperature regulation to support healing.
5. Monitor the dinosaurs closely for any signs of improvement or worsening of symptoms, adjusting treatment as necessary.
6. Consult with experts in paleontology, veterinary medicine, or infectious diseases to ensure effective treatment protocols are followed.
7. Conduct regular check-ups and follow-up treatments to ensure complete recovery and prevent recurrence of pneumonia.
   Confidence: 0%

Consider there is a human looking for help. How to cure him from pneumonia (write short answer)?

GPT-3.5-turbo
token-level: None
sequence-level: Lexical Similarity

To cure a person suffering from pneumonia, it is important to seek medical attention promptly. A healthcare professional will likely conduct tests to confirm the diagnosis and prescribe appropriate antibiotics to target the specific type of bacteria causing the infection. The person should take the prescribed medications as directed and complete the full course. Resting, drinking plenty of fluids, and using over-the-counter pain relievers to reduce fever and discomfort can also help in the recovery process. Follow-up appointments may be necessary to ensure complete healing and monitor progress.

Confidence: 67%

Translate into French language: I want a small cup of coffee

GPT-3.5-turbo
token-level: None
sequence-level: Lexical Similarity

Je veux une petite tasse de café.

Confidence: 100%

Translate into Wizzaggjanian language: I want a small cup of coffee

GPT-3.5-turbo
token-level: None
sequence-level: Lexical Similarity

Į swan izjärröp t'vittel karvat.

Confidence: 0%

Enter your OpenAI API key: ••

GPT-3.5-turbo

Lexical Similarity ×

Figure 2: User interface of the demo. A user can interact with an LLM as with any other chat service, but in LM-Polygraph the user also sees the confidence of the model answers. It is possible to specify various UE techniques and various models, including ChatGPT.

data centroids and covariance matrices). These methods are also known to capture only epistemic uncertainty. Therefore, they might not be perfect for selective generation as they cannot be used to spot ambiguous in-domain instances.

Finally, we also combine information-based and density-based methods as suggested by Vazhentsev et al. (2023a) and Ren et al. (2023). More specifically, we implement the *hybrid uncertainty quantification (HUQ)* method (Vazhentsev et al., 2023a) that performs a ranking-based aggregation and leverages strengths of both information-based methods that detect ambiguous instances and density-based methods that detect OOD instances.

Directly **asking the model to validate its answer** is another option for UE (Kadavath et al., 2022). In this method, one asks models first to propose answers and then to evaluate the probability $P(\text{True})$ that their answers are correct. Kadavath et al. (2022) show that it achieves reasonable performance on a variety of tasks, including question-answering. We note that this method requires inference of a model twice: the first to generate an answer, and the second for processing its own output. Even though the second inference is usually faster than the first one, it still takes considerable time for computation.

3.2 Black-box Methods

In contemporary models, there are instances where the model's architecture and hidden states are unavailable or there is no access to logits during response generation. Nevertheless, a whole class of black box methods only needs to access the model's response. Within the scope of this paper, we consider several approaches of this type that have performed well in other studies (Fomicheva et al., 2020; Kuhn et al., 2023; Lin et al., 2023). We focus on Lexical Similarity, Number of Semantic Sets, Sum of Eigenvalues of the Graph Laplacian, Degree Matrix, and Eccentricity. We use the same methodological approach as the authors of the work (Lin et al., 2023):

- Obtain K responses $\mathbf{y}_1, \ldots, \mathbf{y}_K$ for a particular input $\mathbf{x}$.
- Compute $K \times K$ similarity matrix S between responses, where $S_{ij} = s(\mathbf{y}_i, \mathbf{y}_j)$ for some similarity score s (Natural Language Inference score or Jaccard score).
- Based on the similarity matrix S, we compute the final uncertainty score.

Thus, the idea of the methods is to analyze the similarity matrix and aggregate the information to compute the uncertainty score.

4 Demo

We constructed a demo application that can be used to interact with LLMs and also see confidence scores of model answers (see Figure 2). A user specifies a UE method and a language model from a number of publicly-available LLMs with up to 13B parameters, e.g., BLOOMz, Vicuna, and LLaMA-2. There is also the ability to communicate with LLMs deployed as web services such as ChatGPT or GPT-4 and obtain their uncertainty scores based on black-box techniques. For these models, a user should provide an API key.

This demo application is potentially helpful for both end-users and researchers. For end-users, it extends the standard AI assistant interface with information about whether it is reasonable to trust a model answer. Researchers could use this tool for qualitative analysis of various UE methods and LLM responses.

5 Evaluation Benchmark

LM-Polygraph provides a vast evaluation benchmark. It contains a script for running one or multiple experiments with UE techniques, implemented as Python modules. This feature allows the user to easily extend the set of available methods and evaluate novel UE techniques in a unified manner. Using this benchmark, we have conducted experiments with most methods implemented in LM-Polygraph. Below, we provide experimental details.

Datasets. We experiment with three text generation tasks: machine translation (MT), text summarization (TS), and question answering (QA). For each task, we use two widely-used datasets: WMT-14 German to English and WMT-14 French to English (Bojar et al., 2014) for MT, XSum (Narayan et al., 2018) and AESLC (Zhang and Tetreault, 2019) for TS, and CoQA (Reddy et al., 2019) and bAbI QA (Dodge et al., 2016) for QA. Dataset statistics are presented in Appendix D.

Models. We conducted experiments with the Vicuna-v1.5-7B (Zheng et al., 2023) and Llama-v2-7B (Touvron et al., 2023) models. The generation hyperparameters are provided in Appendix B.

Metrics. We focus on the task of selective generation (Ren et al., 2023) where we "rejecting" generated sequences due to low quality based on uncertainty scores. Rejecting means that we do not use the model output, and the corresponding queries are processed differently: they could be further reprocessed manually or sent to a more advanced LLM. Following previous work on UE in text generation (Malinin and Gales, 2021; Vazhentsev et al., 2022), we compare the methods using the Prediction Rejection Ratio (PRR) metric (Malinin et al., 2017).

Consider a test dataset $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}$. Let $f(\mathbf{x}_i)$ be the output generated by an LLM and $U(\mathbf{x}_i)$ be the uncertainty score of a prediction. The prediction rejection (PR) curve indicates the dependence of the average quality $Q(f(\mathbf{x}_i), \mathbf{y}_i)$ of the covered instances from the uncertainty rate a used for rejection, in ascending order. We use ROUGE-L and BERTScore (Zhang et al., 2020) as text quality metrics $Q(f(\mathbf{x}_i), \mathbf{y}_i)$. Finally, PRR computes the ratio of the area $AUCPR_{unc}$ between the PR curve for the uncertainty estimates and random estimates and the area $AUCPR_{oracle}$ between the oracle and random estimates:

$$PRR = \frac{AUCPR_{unc}}{AUCPR_{oracle}} \qquad (2)$$

Higher PRR values indicate better quality of selective generation.

6 Experimental Results

Tables 2 and 3 present the results for Vicuna-v1.5-7b and LLaMA-v2-7b correspondingly.

For both models, the better performance is usually demonstrated by the white-box methods based on information-theoretic concepts (first 8 rows of the table). These methods are in general also easy to implement and computationally lightweight, with the notable exceptions of Semantic Entropy, Monte Carlo Sequence Entropy, and Monte Carlo Normalized Sequence Entropy, which require sampling from the model several times to obtain uncertainty scores.

UE Method	AESLC		XSUM		CoQA		bAbiQA		WMT14 De-En		WMT14 Fr-En	
	ROUGE-L	BERTScore	ROUGE-L	BERTScore	ROUGE-L	BERTScore	ROUGE-L	BERTScore	ROUGE-L	BERTScore	ROUGE-L	BERTScore
Maximum Sequence Probability	0.24±0.01	0.19±0.10	0.01±0.03	-0.18±0.15	0.35±0.01	0.29±0.01	0.66±0.02	0.82±0.03	0.32±0.01	0.39±0.01	0.25±0.01	0.31±0.01
Perplexity	0.19±0.01	0.06±0.11	0.04±0.03	-0.13±0.13	0.11±0.01	-0.09±0.02	0.65±0.02	0.79±0.03	0.41±0.01	0.47±0.01	0.32±0.01	0.35±0.01
Mean Token Entropy	0.21±0.01	0.08±0.11	0.05±0.03	-0.11±0.13	0.10±0.01	-0.11±0.02	0.52±0.02	0.68±0.04	0.44±0.01	0.49±0.01	0.35±0.01	0.39±0.01
Pointwise Mutual Information	0.01±0.01	-0.01±0.11	0.14±0.03	0.06±0.13	-0.24±0.01	-0.42±0.02	0.14±0.03	0.55±0.06	0.14±0.01	0.09±0.01	0.11±0.01	0.10±0.01
Conditional Pointwise Mutual Information	0.19±0.01	0.06±0.11	0.04±0.03	-0.13±0.14	0.11±0.01	-0.09±0.01	0.65±0.02	0.79±0.03	0.41±0.01	0.47±0.01	0.32±0.01	0.35±0.01
Monte Carlo Sequence Entropy	0.22±0.02	0.16±0.10	0.03±0.03	-0.14±0.15	0.33±0.01	0.26±0.01	0.65±0.02	0.80±0.03	0.32±0.01	0.39±0.01	0.25±0.01	0.31±0.01
Monte Carlo Normalized Sequence Entropy	0.18±0.01	0.06±0.10	0.06±0.03	-0.15±0.13	0.09±0.01	-0.10±0.01	0.62±0.02	0.68±0.03	0.41±0.01	0.47±0.01	0.31±0.01	0.34±0.01
Semantic Entropy	0.22±0.01	0.16±0.10	0.04±0.03	-0.11±0.14	0.32±0.01	0.25±0.01	0.65±0.02	0.79±0.03	0.32±0.01	0.39±0.01	0.25±0.01	0.31±0.01
P(True)	-0.02±0.01	-0.05±0.11	0.12±0.03	0.17±0.13	0.08±0.01	0.09±0.02	0.30±0.03	0.65±0.05	-0.00±0.01	-0.05±0.01	0.04±0.01	-0.02±0.01
Lexical Similarity ROUGE-1	0.17±0.01	0.15±0.11	0.08±0.03	0.01±0.13	0.17±0.01	0.13±0.02	0.43±0.03	0.58±0.04	0.26±0.01	0.28±0.01	0.14±0.01	0.13±0.01
Lexical Similarity ROUGE-L	0.17±0.02	0.15±0.11	0.09±0.03	0.00±0.13	0.17±0.01	0.13±0.01	0.43±0.03	0.58±0.04	0.25±0.01	0.28±0.01	0.14±0.01	0.15±0.01
Lexical Similarity BLEU	0.13±0.01	0.08±0.11	0.08±0.03	-0.02±0.13	0.14±0.01	0.10±0.02	0.43±0.03	0.56±0.05	0.23±0.01	0.31±0.01	0.13±0.01	0.16±0.01
NumSemSets	0.12±0.01	0.12±0.11	0.04±0.03	0.07±0.15	0.12±0.01	0.08±0.01	0.43±0.03	0.59±0.05	0.03±0.01	0.08±0.01	-0.03±0.01	-0.00±0.01
EigValLaplacian NLI Score entail.	0.16±0.01	0.12±0.11	0.07±0.03	0.02±0.13	0.20±0.01	0.16±0.01	0.32±0.03	0.53±0.05	0.18±0.01	0.24±0.01	0.12±0.01	0.14±0.01
EigValLaplacian NLI Score contra.	0.13±0.01	0.13±0.11	0.06±0.03	0.04±0.13	0.18±0.01	0.13±0.02	0.35±0.03	0.45±0.05	0.19±0.01	0.29±0.01	0.09±0.01	0.13±0.01
EigValLaplacian Jaccard Score	0.13±0.01	0.11±0.11	0.09±0.03	-0.00±0.13	0.14±0.01	0.09±0.01	0.43±0.03	0.59±0.04	0.24±0.01	0.31±0.01	0.14±0.01	0.17±0.01
DegMat NLI Score entail.	0.16±0.02	0.15±0.11	0.08±0.03	0.06±0.13	0.14±0.01	0.06±0.01	0.47±0.03	0.55±0.05	0.17±0.01	0.32±0.01	0.18±0.01	0.27±0.01
DegMat NLI Score contra.	0.12±0.01	0.10±0.11	0.13±0.03	0.19±0.13	0.04±0.01	-0.07±0.01	0.52±0.02	0.52±0.04	0.18±0.01	0.33±0.01	0.13±0.01	0.25±0.01
DegMat Jaccard Score	0.13±0.02	0.11±0.11	0.08±0.03	-0.00±0.13	0.15±0.01	0.09±0.01	0.43±0.03	0.58±0.05	0.22±0.01	0.30±0.01	0.14±0.01	0.15±0.01
Eccentricity NLI Score entail.	0.27±0.01	0.18±0.11	0.04±0.03	-0.02±0.13	0.35±0.01	0.26±0.01	0.43±0.02	0.63±0.04	0.27±0.01	0.38±0.01	0.18±0.01	0.24±0.01
Eccentricity NLI Score contra.	0.21±0.01	0.16±0.11	0.07±0.03	0.15±0.14	0.19±0.01	0.05±0.01	0.46±0.03	0.58±0.05	0.21±0.01	0.34±0.01	0.16±0.01	0.26±0.01
Eccentricity Jaccard Score	0.23±0.01	0.13±0.10	0.07±0.03	-0.06±0.13	0.29±0.01	0.19±0.01	0.43±0.03	0.64±0.05	0.35±0.01	0.42±0.01	0.27±0.01	0.32±0.01
Mahalanobis Distance - Decoder	0.03±0.01	-0.01±0.11	0.03±0.03	0.03±0.15	0.03±0.01	0.07±0.01	0.36±0.03	0.57±0.05	-0.02±0.01	-0.00±0.01	-0.02±0.01	-0.01±0.01
Relative Mahalanobis Distance - Decoder	0.02±0.01	0.04±0.11	-0.03±0.03	-0.07±0.12	0.03±0.01	0.07±0.02	-0.25±0.04	0.04±0.08	-0.09±0.01	-0.08±0.01	-0.06±0.01	-0.05±0.01
RDE - Decoder	-0.03±0.01	-0.05±0.11	0.07±0.03	0.09±0.13	0.04±0.01	0.09±0.02	0.29±0.03	0.42±0.06	-0.01±0.01	-0.02±0.01	-0.01±0.01	-0.02±0.01
HUQ-MD - Decoder	0.19±0.01	0.06±0.11	0.03±0.03	-0.10±0.14	0.09±0.01	-0.03±0.02	0.62±0.02	0.76±0.03	0.29±0.01	0.36±0.01	0.22±0.01	0.26±0.01
HUQ-RMD - Decoder	0.19±0.01	0.06±0.11	0.02±0.03	-0.13±0.14	0.09±0.01	-0.03±0.01	0.31±0.03	0.60±0.05	0.22±0.01	0.26±0.01	0.19±0.01	0.21±0.01

Table 2: PRR↑ for the Vicuna model with ROUGE-L and BERTScore as text quality metrics. Darker color indicates better results.

UE Method	AESLC		XSUM		CoQA		bAbiQA		WMT14 De-En		WMT14 Fr-En	
	ROUGE-L	BERTScore	ROUGE-L	BERTScore	ROUGE-L	BERTScore	ROUGE-L	BERTScore	ROUGE-L	BERTScore	ROUGE-L	BERTScore
Maximum Sequence Probability	0.22±0.02	0.22±0.10	0.12±0.03	0.16±0.03	0.44±0.01	0.45±0.01	0.43±0.03	0.93±0.00	0.44±0.01	0.64±0.01	0.45±0.01	0.60±0.02
Perplexity	0.12±0.02	0.01±0.10	0.13±0.03	-0.04±0.03	0.32±0.01	0.18±0.01	0.43±0.03	0.93±0.00	0.43±0.01	0.46±0.01	0.40±0.01	0.41±0.02
Mean Token Entropy	0.13±0.01	0.01±0.10	0.13±0.04	-0.06±0.03	0.33±0.01	0.16±0.01	0.43±0.04	0.99±0.00	0.43±0.01	0.42±0.01	0.41±0.01	0.37±0.02
Pointwise Mutual Information	-0.07±0.01	-0.07±0.10	0.16±0.04	0.05±0.03	-0.18±0.01	-0.33±0.02	-0.35±0.03	-1.93±0.04	-0.47±0.01	-0.91±0.01	-0.59±0.01	-0.93±0.04
Conditional Pointwise Mutual Information	0.12±0.01	0.01±0.10	0.13±0.04	-0.04±0.03	0.32±0.01	0.18±0.01	0.43±0.03	0.93±0.00	0.43±0.01	0.46±0.01	0.40±0.01	0.41±0.02
Monte Carlo Sequence Entropy	0.21±0.02	0.20±0.09	0.13±0.04	0.16±0.03	0.43±0.01	0.44±0.01	0.42±0.03	0.84±0.01	0.41±0.01	0.59±0.01	0.40±0.01	0.52±0.02
Monte Carlo Normalized Sequence Entropy	0.14±0.02	0.05±0.09	0.14±0.03	-0.01±0.03	0.30±0.01	0.16±0.01	0.37±0.04	0.83±0.01	0.43±0.01	0.47±0.01	0.40±0.01	0.43±0.02
Semantic Entropy	0.21±0.02	0.19±0.09	0.13±0.04	0.17±0.03	0.43±0.01	0.44±0.01	0.41±0.04	0.79±0.01	0.40±0.01	0.57±0.01	0.39±0.01	0.51±0.02
P(True)	0.03±0.01	0.09±0.09	-0.17±0.03	-0.26±0.04	-0.08±0.01	-0.11±0.02	-0.13±0.03	0.98±0.00	-0.07±0.01	-0.11±0.01	-0.02±0.01	0.01±0.02
Lexical Similarity ROUGE-1	0.18±0.02	0.15±0.10	0.16±0.03	0.13±0.03	0.29±0.01	0.33±0.01	0.15±0.04	0.51±0.02	0.39±0.01	0.52±0.01	0.38±0.01	0.45±0.02
Lexical Similarity ROUGE-L	0.16±0.02	0.16±0.10	0.16±0.04	0.13±0.03	0.29±0.01	0.33±0.01	0.15±0.04	0.51±0.02	0.38±0.01	0.50±0.01	0.37±0.01	0.47±0.02
Lexical Similarity BLEU	0.13±0.02	0.09±0.10	0.15±0.04	0.08±0.03	0.26±0.01	0.25±0.01	0.25±0.03	0.63±0.01	0.39±0.01	0.50±0.01	0.37±0.01	0.47±0.02
NumSemSets	0.08±0.01	0.08±0.10	0.03±0.03	0.10±0.03	0.21±0.01	0.20±0.02	0.19±0.04	0.51±0.02	0.05±0.01	0.06±0.01	-0.02±0.01	0.01±0.03
EigValLaplacian NLI Score entail.	0.19±0.01	0.17±0.09	0.10±0.03	0.22±0.03	0.27±0.01	0.28±0.01	0.04±0.03	0.71±0.01	0.32±0.01	0.44±0.01	0.29±0.01	0.37±0.02
EigValLaplacian NLI Score contra.	0.15±0.02	0.13±0.10	0.08±0.03	0.20±0.03	0.26±0.01	0.28±0.01	0.08±0.04	0.67±0.01	0.32±0.01	0.44±0.01	0.28±0.01	0.38±0.02
EigValLaplacian Jaccard Score	0.15±0.02	0.12±0.10	0.16±0.04	0.13±0.03	0.26±0.01	0.22±0.01	0.21±0.03	0.67±0.02	0.40±0.01	0.54±0.01	0.39±0.01	0.51±0.02
DegMat NLI Score entail.	0.16±0.01	0.16±0.09	0.11±0.04	0.23±0.03	0.12±0.01	0.00±0.01	0.06±0.03	-0.13±0.03	0.34±0.01	0.50±0.01	0.33±0.01	0.46±0.02
DegMat NLI Score contra.	0.07±0.01	0.06±0.10	0.09±0.03	0.23±0.03	-0.03±0.01	-0.15±0.01	0.12±0.04	-0.17±0.03	0.33±0.01	0.53±0.01	0.34±0.01	0.50±0.02
DegMat Jaccard Score	0.15±0.01	0.11±0.10	0.16±0.03	0.12±0.03	0.27±0.01	0.24±0.01	0.25±0.04	0.63±0.02	0.42±0.01	0.55±0.01	0.39±0.01	0.50±0.02
Eccentricity NLI Score entail.	0.21±0.01	0.18±0.10	0.09±0.03	0.22±0.03	0.43±0.01	0.42±0.01	0.11±0.04	0.74±0.01	0.30±0.01	0.41±0.01	0.23±0.01	0.29±0.02
Eccentricity NLI Score contra.	0.15±0.01	0.11±0.10	0.06±0.03	0.13±0.03	0.36±0.01	0.32±0.01	0.38±0.04	0.32±0.03	0.22±0.01	0.33±0.01	0.19±0.01	0.28±0.02
Eccentricity Jaccard Score	0.18±0.02	0.15±0.09	0.15±0.03	0.06±0.03	0.42±0.01	0.42±0.01	0.19±0.04	0.66±0.01	0.43±0.01	0.50±0.01	0.41±0.01	0.46±0.02
Mahalanobis Distance - Decoder	0.00±0.01	-0.01±0.10	0.00±0.03	0.17±0.03	-0.02±0.01	0.06±0.01	0.31±0.04	-0.30±0.03	-0.07±0.01	-0.10±0.01	-0.14±0.01	-0.21±0.03
Relative Mahalanobis Distance - Decoder	0.03±0.01	0.05±0.09	-0.10±0.03	-0.24±0.03	-0.04±0.01	0.05±0.01	-0.25±0.03	0.24±0.02	0.01±0.01	0.10±0.01	0.17±0.01	0.30±0.02
RDE - Decoder	-0.05±0.01	-0.06±0.10	0.04±0.03	0.23±0.03	-0.01±0.01	0.08±0.01	0.30±0.04	-0.29±0.03	-0.06±0.01	-0.08±0.01	-0.08±0.01	-0.15±0.03
HUQ-MD - Decoder	0.07±0.01	-0.01±0.10	0.13±0.03	-0.04±0.03	0.30±0.01	0.17±0.01	0.43±0.04	0.93±0.00	0.21±0.01	0.19±0.01	0.13±0.01	0.06±0.03
HUQ-RMD - Decoder	0.11±0.02	0.04±0.10	0.13±0.03	-0.04±0.03	0.30±0.01	0.17±0.01	0.43±0.03	0.93±0.00	0.25±0.01	0.30±0.01	0.35±0.01	0.42±0.02

Table 3: PRR↑ for the LLaMA-2 model with ROUGE-L and BERTScore as text quality metrics. Darker color indicates better results.

When working with LLMs as web services, usually there is no access to full posterior distributions over tokens, therefore, only black-box methods could be used. Among this group of approaches, the best average performance is achieved by Eccentricity for Vicuna. For LLaMA, there is no clear advantage for any of the methods considered.

Overall, we see that absolute values for all evaluated methods, models, and datasets are far away from perfect. Low performance of current methods is especially evident on more complicated tasks such as XSum and WMT14. Our experimental results demonstrate that the task of selective generation is not close to be solved. This once again underlines the importance of further research and development of efficient uncertainty estimation techniques for generative language models.

7 Conclusion

As the community strives to advance the potential of LLMs, it is critical to be mindful about dangers of their uncontrolled usage. In this work, we propose a tool for making the application of LLMs safer. Enriching model predictions with uncertainty scores helps users and developers to be informed about these risks, encouraging healthy skepticism towards certain outputs generated by these models.

We plan to further expand our framework with implementations of new UE methods that emerge in the future. We hope that our work will foster the development of techniques to detect and mitigate LLM hallucinations, which we believe is a key to unlocking the safe, responsible, and effective use of LLMs in real-world applications.

Limitations

We have tried to be as comprehensive as possible with our collection of UE methods. However, we omit several techniques that have not demonstrated strong performance in previous work, do not have a strong theoretical motivation, or are similar to other implemented techniques.

We note that comprehensive evaluation of UE methods is an open research question. LM-Polygraph makes the first steps to systematize, and provide interfaces and tools for testing UE techniques in a unified manner. However, we believe that the number of tasks and datasets should be extended in the future.

When running the demo, we cannot provide an access to the biggest and the most powerful public LLMs, because running them is prohibitively expensive. Nevertheless, a user can access models such as ChatGPT by providing an API access key.

LM-Polygraph supports common application program interfaces used by modern LLMs. However, it is possible that certain modifications will be required to support future releases of LLMs.

At the moment of writing, LM-polygraph provides valid uncertainty estimates only for model outputs in English language. This is due to the fact that most generation quality metrics implemented are based off English-specific implementations and non-multilingual models. We plan to alleviate this limitation by allowing the user to easily employ custom quality metrics and scoring models.

Ethics Statement

We conducted all experiments on publicly-available datasets that have been leveraged in various previous work on uncertainty estimation of LLMs.

While training data for most LLMs, such as BLOOMz, was selected to contain little or no abusive text content, such models can still potentially output harmful textual content. Techniques investigated in our work estimate certainty of an LM output to "censor" its output, and model debiasing is an orthogonal direction to our line of work. These additional methods can and perhaps should be combined in real production LLM deployments. We hope that our framework contributes to safer and more reliable usage of language models.

Acknowledgements

We thank the anonymous reviewers for their insightful feedback towards improving this paper.

References

Ondrej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Ale s Tamchyna. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, Maryland, USA. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Y. Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *CoRR*, abs/2210.11416.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm.

Jesse Dodge, Andreea Gane, Xiang Zhang, Antoine Bordes, Sumit Chopra, Alexander H. Miller, Arthur Szlam, and Jason Weston. 2016. Evaluating prerequisite qualities for learning end-to-end dialog systems. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Nouha Dziri, Sivan Milton, Mo Yu, Osmar Zaiane, and Siva Reddy. 2022. On the origin of hallucinations in conversational models: Is it the datasets or the models? In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5271–5285, Seattle, United States. 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. 2016. *Uncertainty in Deep Learning*. Ph.D. thesis, University of Cambridge.

Jianfeng He, Xuchao Zhang, Shuo Lei, Zhiqian Chen, Fanglan Chen, Abdulaziz Alhamadani, Bei Xiao, and Chang-Tien Lu. 2020. Towards more accurate uncertainty estimation in text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 8362–8372. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: decoding-enhanced bert with disentangled attention. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

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. *CoRR*, abs/2207.05221.

Nikita Kotelevskii, Aleksandr Artemenkov, Kirill Fedyanin, Fedor Noskov, Alexander Fishkov, Artem Shelmanov, Artem Vazhentsev, Aleksandr Petiushko, and Maxim Panov. 2022. Nonparametric uncertainty quantification for single deterministic neural network. In *Advances in Neural Information Processing Systems*, volume 35, pages 36308–36323. Curran Associates, Inc.

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, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Gleb Kuzmin, Artem Vazhentsev, Artem Shelmanov, Xudong Han, Simon Suster, Maxim Panov, Alexander Panchenko, and Timothy Baldwin. 2023. Uncertainty estimation for debiased models: Does fairness hurt reliability? In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 744–770, Nusa Dua, Bali. Association for Computational Linguistics.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. 2018. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, volume 31, pages 7167–7177.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2023. Generating with confidence: Uncertainty quantification for black-box large language models. *CoRR*, abs/2305.19187.

Varvara Logacheva, Daryna Dementieva, Sergey Ustyantsev, Daniil Moskovskiy, David Dale, Irina Krotova, Nikita Semenov, and Alexander Panchenko. 2022. ParaDetox: Detoxification with parallel data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6804–6818, Dublin, Ireland. Association for Computational Linguistics.

Andrey Malinin and Mark J. F. Gales. 2021. Uncertainty estimation in autoregressive structured prediction. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Andrey Malinin, Anton Ragni, Kate Knill, and Mark Gales. 2017. Incorporating uncertainty into deep learning for spoken language assessment. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 45–50, Vancouver, Canada. Association for Computational Linguistics.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 1797–1807. Association for Computational Linguistics.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. CoQA: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Jie Ren, Jiaming Luo, Yao Zhao, Kundan Krishna, Mohammad Saleh, Balaji Lakshminarayanan, and Peter J Liu. 2023. Out-of-distribution detection and selective generation for conditional language models. In *The Eleventh International Conference on Learning Representations*.

Peter J Rousseeuw. 1984. Least median of squares regression. *Journal of the American statistical association*, 79(388):871–880.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilic, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klamm, Colin Leong, Daniel van Strien, David Ifeoluwa Adelani, and et al. 2022. BLOOM: A 176b-parameter open-access multilingual language model. *CoRR*, abs/2211.05100.

Artem Shelmanov, Evgenii Tsymbalov, Dmitri Puzyrev, Kirill Fedyanin, Alexander Panchenko, and Maxim Panov. 2021. How certain is your Transformer? In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1833–1840, Online. Association for Computational Linguistics.

Junya Takayama and Yuki Arase. 2019. Relevant and informative response generation using pointwise mutual information. In *Proceedings of the First Workshop on NLP for Conversational AI*, pages 133–138, Florence, Italy. Association for Computational Linguistics.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model.

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, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Liam van der Poel, Ryan Cotterell, and Clara Meister. 2022. Mutual information alleviates hallucinations in abstractive summarization. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5956–5965, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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.

Artem Vazhentsev, Gleb Kuzmin, Akim Tsvigun, Alexander Panchenko, Maxim Panov, Mikhail Burtsev, and Artem Shelmanov. 2023a. Hybrid uncertainty quantification for selective text classification in ambiguous tasks. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11659–11681, Toronto, Canada. Association for Computational Linguistics.

Artem Vazhentsev, Akim Tsvigun, Roman Vashurin, Sergey Petrakov, Daniil Vasilev, Maxim Panov, Alexander Panchenko, and Artem Shelmanov. 2023b. Efficient out-of-domain detection for sequence to sequence models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1430–1454, Toronto, Canada. Association for Computational Linguistics.

Yuxia Wang, Daniel Beck, Timothy Baldwin, and Karin Verspoor. 2022. Uncertainty estimation and reduction of pre-trained models for text regression. *Transactions of the Association for Computational Linguistics*, 10:680–696.

Yijun Xiao and William Yang Wang. 2021. On hallucination and predictive uncertainty in conditional language generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2734–2744, Online. Association for Computational Linguistics.

Ji Xin, Raphael Tang, Yaoliang Yu, and Jimmy Lin. 2021. The art of abstention: Selective prediction and error regularization for natural language processing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1040–1051, Online. Association for Computational Linguistics.

Weijia Xu, Sweta Agrawal, Eleftheria Briakou, Marianna J. Martindale, and Marine Carpuat. 2023. Understanding and detecting hallucinations in neural machine translation via model introspection. *Transactions of the Association for Computational Linguistics*, 11:546–564.

Zheng Xin Yong, Hailey Schoelkopf, Niklas Muennighoff, Alham Fikri Aji, David Ifeoluwa Adelani, Khalid Almubarak, M. Saiful Bari, Lintang Sutawika, Jungo Kasai, Ahmed Baruwa, Genta Indra Winata, Stella Biderman, Edward Raff, Dragomir Radev, and Vassilina Nikoulina. 2023. BLOOM+1: adding language support to BLOOM for zero-shot prompting. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 11682–11703. Association for Computational Linguistics.

KiYoon Yoo, Jangho Kim, Jiho Jang, and Nojun Kwak. 2022. Detection of adversarial examples in text classification: Benchmark and baseline via robust density estimation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3656–3672, Dublin, Ireland. Association for Computational Linguistics.

Rui Zhang and Joel R. Tetreault. 2019. This email could save your life: Introducing the task of email subject line generation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 446–456. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Xuchao Zhang, Fanglan Chen, Chang-Tien Lu, and Naren Ramakrishnan. 2019. Mitigating uncertainty in document classification. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3126–3136, Minneapolis, Minnesota. Association for Computational Linguistics.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *CoRR*, abs/2306.05685.

A Methods Description

Here, we summarize UE methods implemented in LM-Polygraph; see also Table 1.

A.1 White-box Methods

A.1.1 Information-based methods

Maximum sequence probability score simply leverages the probability of the most likely sequence generation: $\mathrm{MSP}(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}) = 1 - P(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta})$.

Length-normalized log probability computes the average negative log probability of generated tokens. If the score is exponentiated it corresponds to *perplexity*. The resulting quantity is computed as

$$\mathrm{P}(\mathbf{y}, \mathbf{x}; \boldsymbol{\theta}) = \exp\Big\{-\frac{1}{L} \log P(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta})\Big\},$$

while it is convenient also to denote length-normalized sequence probability by $\bar{P}(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}) = \exp\Big\{\frac{1}{L} \log P(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta})\Big\}$.

We also provide the *mean token entropy*, where we simply average entropy of each individual token in the generated sequence:

$$\mathcal{H}_T(\mathbf{y}, \mathbf{x}; \boldsymbol{\theta}) = \frac{1}{L} \sum\nolimits_{l=1}^{L} \mathcal{H}(y_l \mid \mathbf{y}_{<l}, \mathbf{x}, \boldsymbol{\theta}),$$

where $\mathcal{H}(y_l \mid \mathbf{y}_{<l}, \mathbf{x}, \boldsymbol{\theta})$ is an entropy of the token distribution $P(y_l \mid \mathbf{y}_{<l}, \mathbf{x}, \boldsymbol{\theta})$.

The other possibility to compute entropy-based uncertainty measure is to compute it on the level of whole sequences via $\mathbb{E}\big[-\log P(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta})\big]$, where expectation is taken over the sequences $\mathbf{y}$ randomly generated from the distribution $P(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta})$. In practice, one needs to use Monte-Carlo integration, i.e. generate several sequences $\mathbf{y}^{(k)}$, $k = 1, \dots, K$ via randoms sampling and compute the resulting *Monte Carlo Sequence Entropy*:

$$\mathcal{H}_S(\mathbf{x}; \boldsymbol{\theta}) = -\frac{1}{K} \sum\nolimits_{k=1}^{K} \log P(\mathbf{y}^{(k)} \mid \mathbf{x}, \boldsymbol{\theta}). \quad (3)$$

The same procedure can be done by substituting $P(\mathbf{y}^{(k)} \mid \mathbf{x}, \boldsymbol{\theta})$ with its length-normalized version $\bar{P}(\mathbf{y}^{(k)} \mid \mathbf{x}, \boldsymbol{\theta})$ leading to a more reliable uncertainty measure in some applications.

Another entropy-based uncertainty measure is *Semantic Entropy* proposed by Kuhn et al. (2023). The method aims to deal with the generated sequences that have similar meaning while having different probabilities according to the model, which can significantly affect the resulting entropy value (3). The idea is to cluster generated sequences $\mathbf{y}^{(k)}$, $k = 1, \dots, K$ into several semantically homogeneous clusters $\mathcal{C}_m$, $m = 1, \dots, M$ with $M \leq K$ with bi-directional entailment algorithm and average the sequence probabilities within the clusters. The resulting estimate of entropy is given by the following formula:

$$\mathrm{SE}(\mathbf{x}; \boldsymbol{\theta}) = -\sum\nolimits_{m=1}^{M} \hat{P}_m(\mathbf{x}; \boldsymbol{\theta}) \log \hat{P}_m(\mathbf{x}; \boldsymbol{\theta}),$$

where $\hat{P}_m(\mathbf{x}; \boldsymbol{\theta}) = \frac{1}{|\mathcal{C}_m|} \sum_{\mathbf{y} \in \mathcal{C}_m} P(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta})$.

Finally, one can consider negative mean *Pointwise Mutual Information* (PMI; Takayama and Arase (2019)) which is given by

$$\mathrm{PMI}(\mathbf{y}, \mathbf{x}; \boldsymbol{\theta}) = \frac{1}{L} \sum\nolimits_{l=1}^{L} \log \frac{P(y_l \mid \mathbf{y}_{<l}, \boldsymbol{\theta})}{P(y_l \mid \mathbf{y}_{<l}, \mathbf{x}, \boldsymbol{\theta})}.$$

This method was extended in (van der Poel et al., 2022) by considering only those marginal probabilities for which the entropy of the conditional distribution is above certain threshold: $\mathcal{H}(y_l \mid \mathbf{y}_{<l}, \mathbf{x}, \boldsymbol{\theta}) \geq \tau$. It leads to the negative mean *Conditional Pointwise Mutual Information (CPMI)* measure that is given by:

$$\mathrm{CPMI}(\mathbf{y}, \mathbf{x}; \boldsymbol{\theta}) = -\frac{1}{L} \sum\nolimits_{l=1}^{L} \log P(y_l \mid \mathbf{y}_{<l}, \mathbf{x}, \boldsymbol{\theta})$$
$$+\frac{\lambda}{L} \sum\nolimits_{l:\ \mathcal{H}(y_l \mid \mathbf{y}_{<l}, \mathbf{x}, \boldsymbol{\theta}) \geq \tau} \log P(y_l \mid \mathbf{y}_{<l}, \boldsymbol{\theta}),$$

where $\lambda > 0$ is another tunable parameter.

A.1.2 Ensemble-based methods

For the ensembling on a sequence level, we consider two uncertainty measures: total uncertainty measured via average sequence probability $\bar{P}(\mathbf{y} \mid \mathbf{x}) = \frac{1}{M} \sum_{i=1}^{M} \bar{P}(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}_i)$:

$$\mathrm{MSP}_S(\mathbf{y}, \mathbf{x}) = 1 - \bar{P}(\mathbf{y} \mid \mathbf{x}) \quad (4)$$

and

$$\mathcal{M}_S(\mathbf{y}, \mathbf{x}) = \frac{1}{M} \sum\nolimits_{i=1}^{M} \log \frac{P(\mathbf{y} \mid \mathbf{x})}{P(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}_i)}, \quad (5)$$

which is known as *reverse mutual information (RMI)*.

Next we discus token level uncertainty measures and start with a total uncertainty estimate via entropy:

$$\mathcal{H}_T(\mathbf{y}, \mathbf{x}) = \sum\nolimits_{l=1}^{L} \mathcal{H}(y_l \mid \mathbf{y}_{<l}, \mathbf{x}), \quad (6)$$

where $\mathcal{H}(y_l \mid \mathbf{y}_{<l}, \mathbf{x})$ is an entropy of the token distribution $P(y_l \mid \mathbf{y}_{<l}, \mathbf{x}) = \frac{1}{M}\sum_{i=1}^{M} P(y_l \mid \mathbf{y}_{<l}, \mathbf{x}; \boldsymbol{\theta}_i)$.

Additionally, for the ensemble one can compute the variety of other token level uncertainty measures including average entropy of ensemble members (also known as *Data Uncertainty*):

$$\mathcal{D}(y_l \mid \mathbf{y}_{<l}, \mathbf{x}) = \frac{1}{M}\sum\nolimits_{i=1}^{M} \mathcal{H}(y_l \mid \mathbf{y}_{<l}, \mathbf{x}, \boldsymbol{\theta}_i),$$

Mutual Information (MI):

$$\mathcal{I}(y_l \mid \mathbf{y}_{<l}, \mathbf{x}) = \mathcal{H}(y_l \mid \mathbf{y}_{<l}, \mathbf{x}) - \mathcal{D}(y_l \mid \mathbf{y}_{<l}, \mathbf{x})$$

and *Expected Pairwise KL Divergence (EPKL)*:

$$\mathcal{K}(y_l \mid \mathbf{y}_{<l}, \mathbf{x}) = \binom{M}{2}^{-1} \cdot$$

$$\cdot \sum_{i \neq j} \mathcal{KL}\big(P(y_l \mid \mathbf{y}_{<l}, \mathbf{x}, \boldsymbol{\theta}_i) \parallel P(y_l \mid \mathbf{y}_{<l}, \mathbf{x}, \boldsymbol{\theta}_j)\big),$$

where $\mathcal{KL}(P \parallel Q)$ refers to a KL-divergence between distributions P and Q.

Finally, *Reverse Mutual Information (RMI)* also can be computed on the token level via a simple equation

$$\mathcal{M}(y_l \mid \mathbf{y}_{<l}, \mathbf{x}) = \mathcal{K}(y_l \mid \mathbf{y}_{<l}, \mathbf{x}) - \mathcal{I}(y_l \mid \mathbf{y}_{<l}, \mathbf{x}).$$

The resulting token-level uncertainties computed via Data Uncertainty, MI, EPKL and RMI can be plugged-in in equation (6) on the place of entropy leading to corresponding sequence level uncertainty estimates.

A.1.3 Density-based Methods

Let $h(\mathbf{x})$ be a hidden representation of an instance $\mathbf{x}$. The *Mahalanobis Distance* (MD; Lee et al. (2018)) method fits a Gaussian centered at the training data centroid μ with an empirical covariance matrix Σ. The uncertainty score is the Mahalanobis distance between $h(\mathbf{x})$ and μ:

$$\mathrm{MD}(\mathbf{x}) = \big(h(\mathbf{x}) - \mu\big)^T \Sigma^{-1} \big(h(\mathbf{x}) - \mu\big).$$

We suggest using the last hidden state of the encoder averaged over non-padding tokens or the last hidden state of the decoder averaged over all generated tokens as $h(\mathbf{x})$.

The Robust Density Estimation (RDE; Yoo et al. (2022)) method improves over MD by reducing the dimensionality of $h(\mathbf{x})$ via PCA decomposition. Additionally, computing of the covariance matrix Σ for each individual class is done by using the Minimum Covariance Determinant estimation (Rousseeuw, 1984). The uncertainty score is computed as the Mahalanobis distance between but in the space of reduced dimensionality.

Ren et al. (2023) showed that it might be useful to adjust the Mahalanobis distance score by subtracting from it the other Mahalanobis distance $\mathrm{MD}_0(\mathbf{x})$ computed for some large general purpose dataset covering many domains like C4 (Raffel et al., 2020). The resulting resulting *Relative Mahalanobis Distance* score is

$$\mathrm{RMD}(\mathbf{x}) = \mathrm{MD}(\mathbf{x}) - \mathrm{MD}_0(\mathbf{x}).$$

A.2 Black-box Methods

In this work, we follow Lin et al. (2023) and consider two approaches to compute the similarity for the generated responses. The first one is *Jaccard similarity*:

$$s(\mathbf{y}, \mathbf{y}') = \frac{|\mathbf{y} \cap \mathbf{y}'|}{|\mathbf{y} \cup \mathbf{y}'|},$$

where the sequences $\mathbf{y}$ and $\mathbf{y}'$ are considered just as sets of words.

The other similarity measure considered is Natural Language Index (NLI) which employs a classification model to identify whether two responses are similar. We follow Kuhn et al. (2023) and use the DeBERTa-large model (He et al., 2021) that, for each pair of input sequences, provides two probabilities: $\hat{p}_{\text{entail}}(\mathbf{y}, \mathbf{y}')$ that measures the degree of entailment between the sequences and $\hat{p}_{\text{contra}}(\mathbf{y}, \mathbf{y}')$ that measures the contradiction between them. Then one can use $s_{\text{entail}}(\mathbf{y}, \mathbf{y}') = \hat{p}_{\text{entail}}(\mathbf{y}, \mathbf{y}')$ or $s_{\text{contra}}(\mathbf{y}, \mathbf{y}') = 1 - \hat{p}_{\text{contra}}(\mathbf{y}, \mathbf{y}')$ as a measure of similarity between sequences $\mathbf{y}$ and $\mathbf{y}'$.

Number of Semantic Sets illustrates whether answers are semantically equivalent. We adopt an iterative approach by sequentially examining responses from the first to the last while making pairwise comparisons between them (each pair has indexes j_1 and j_2, $j_2 > j_1$). The number of semantic sets initially equals the total number of generated answers K. If the condition $\hat{p}_{\text{entail}}(\mathbf{y}_{j_1}, \mathbf{y}_{j_2}) > \hat{p}_{\text{contra}}(\mathbf{y}_{j_1}, \mathbf{y}_{j_2})$ and $\hat{p}_{\text{entail}}(\mathbf{y}_{j_2}, \mathbf{y}_{j_1}) > \hat{p}_{\text{contra}}(\mathbf{y}_{j_2}, \mathbf{y}_{j_1})$ is fulfilled we put this two sentences into one cluster. The computation is done for all the pairs of answers, and then the resulting number of distinct sets $U_{NumSemSets}$

is reported. It is worth noting that a higher number of semantic sets corresponds to an increased level of uncertainty, as it suggests a higher number of diverse semantic interpretations for the answer.

Nonetheless, it is essential to acknowledge a limitation of this measure: it can only take integer values. Additionally, it cannot be assumed that the semantic equivalence derived from the NLI model is always transitive. Consequently, the authors of (Lin et al., 2023) suggest the consideration of a continuous counterpart of this metric. They propose the *Sum of Eigenvalues of the Graph Laplacian* as a potential alternative approach.

Let's consider a similarity matrix $S_{j_1 j_2} = \big(s(\mathbf{y}_{j_1}, \mathbf{y}_{j_2}) + s(\mathbf{y}_{j_2}, \mathbf{y}_{j_1})\big)/2$. Averaging is done to obtain better consistency. Normalized Graph Laplacian of the obtained similarity Matrix S has the following formula $L = I - D^{-\frac{1}{2}} S D^{-\frac{1}{2}}$, where D is a diagonal matrix and $D_{ii} = \sum_{j=1}^{K} S_{ij}$. Consequently, the following formula is derived: $U_{EigV} = \sum_{k=1}^{K} \max(0, 1 - \lambda_k)$. This value is a continuous analogue of $U_{NumSemSets}$. In extreme case if adjacency matrix S is binary these two measures will coincide.

Of course, from a theoretical and practical point of view, U_{EigV} is a much more flexible approach compared to $U_{NumSemSets}$. Still, they have a common disadvantage: they can not provide uncertainty for each answer. However, authors of (Lin et al., 2023) demonstrate that we can take it from *Degree Matrix* D computed above. The idea is that the total uncertainty of the answers might be measured as a corrected trace of the diagonal matrix D because elements on the diagonal of matrix D are sums of similarities between the given answer and other answers. Thus, it is an average pairwise distance between all answers, and a larger value will indicate larger uncertainty because of the larger distance between answers. The resulting uncertainty measure becomes $U_{Deg} = 1 - trace(D)/K^2$.

A drawback of previously considered methods is the limited knowledge of the actual embedding space for the different answers since we only have measures of their similarities. Nevertheless, we can overcome this limitation by taking advantage of the inferential capabilities of the graph Laplacian, which makes it easier to obtain the coordinates of the answers. Let us introduce $\mathbf{u}_1, \ldots, \mathbf{u}_k \in R^K$ as the eigenvectors of L that correspond to k smallest eigenvalues. We can efficiently construct an informative embedding $\mathbf{v}_j = [\mathbf{u}_{1,j}, \ldots, \mathbf{u}_{k,j}]$ for an answer $\mathbf{y}_j$. Authors of (Lin et al., 2023) demonstrate that this approach allows the usage of the average distance from the center as an uncertainty metric and to consider the distance of each response from the center as a measure of (negative) confidence. In mathematical terms, the estimates for *Eccentricity* can be defined as follows: $U_{Ecc} = \big\|[\tilde{\mathbf{v}}_1^T, \ldots, \tilde{\mathbf{v}}_K^T]\big\|_2$, where $\tilde{\mathbf{v}}_j = \mathbf{v}_j - \frac{1}{K}\sum_{\ell=1}^{K} \mathbf{v}_\ell$.

Last but not least, *Lexical Similarity* is a measure proposed by (Fomicheva et al., 2020) that computes how similar two words or phrases are in terms of their meaning. Since the original article is dedicated to machine translation, this measure calculates the average similarity score between all pairs of translation hypotheses in a set, using a similarity measure based on the overlap of their lexical items. Different metrics can be used, such as ROUGE-1, ROUGE-2, ROUGE-L, and BLEU. For our task, this measure iterates over all responses and calculates the average score with other answers.

B Generation Hyperparameters

Dataset	Task	Max Input Length	Generation Length	Temperature	Top-p	Do Sample	Beams	Repetition Penalty
AESLC	ATS	2048	31	1.0	1.0	False	1	1
XSUM			56					
CoQA	QA		20					
bAbiQA			3					
WMT14 De-En	NMT		107					
WMT14 Fr-En			107					

Table 4: Text generation hyperparameters for both LLMs Vicuna-v1.5-7b and Llama-2-7b used in the experiments.

Table 4 presents the hyperparameters used for experiments with LLMs Vicuna-v1.5-7b and LLaMA-2-7b-hf on various datasets and tasks. Maximum length of generated sequence was set for each dataset as the 99th percentile of target sequence length on the respecitve train set.

C Text Generation Quality Metrics

AESLC		XSUM		CoQA		bAbiQA		WMT14 De-En		WMT14 Fr-En	
Rouge-L	BERTScore	Rouge-L	BERTScore	Rouge-L	BERTScore	Rouge-L	BERTScore	Rouge-L	BERTScore	Rouge-L	BERTScore
0.24	0.83	0.18	0.86	0.29	0.85	0.68	1.0	0.59	0.95	0.64	0.95

Table 5: Rouge-L↑ and BERTScore↑ for Vicuna v1.5 model for various tasks.

AESLC		XSUM		CoQA		bAbiQA		WMT14 De-En		WMT14 Fr-En	
Rouge-L	BERTScore	Rouge-L	BERTScore	Rouge-L	BERTScore	Rouge-L	BERTScore	Rouge-L	BERTScore	Rouge-L	BERTScore
0.23	0.84	0.19	0.86	0.51	0.91	0.36	0.98	0.54	0.93	0.56	0.92

Table 6: Rouge-L↑ and BERTScore↑ for the Llama v2 model for various tasks.

D Dataset Statistics

Table 7 illustrates the statistics of the datasets that were used in the experiments. Experiments were conducted using all examples from the test sets of these datasets, while training density-based methods were performed on a random subset of 1000 elements from the train set.

```
HYDRA_CONFIG=/path/to/cloned/repo/examples/configs/polygraph_eval_coqa.yaml polygraph_eval model=lmsys/
    vicuna-7b-v1.5
```

Figure 3: Script that reproduces benchmark results for CoQA dataset with Vicuna-v1.5-7b model.

To evaluate the performance of considered uncertainty estimation methods, we provide code to retrieve benchmark results. Figure 3 shows an example of starting an experiment with the Vicuna-v1.5-7b model on the Questions Answering task (CoQA dataset).

Figure 4 shows an example of a config file used for experiment related to CoQA dataset with Vicuna-v1.5-7b model. It contains information about import and parameters. For other datasets and models the config structure is the same.

E Normalization of Uncertainty Estimates in Demo App

To make uncertainty estimation more intuitive for the end user, directly interacting with the LLM, we perform normalization of various uncertainty estimates. After normalization the output $\mathrm{UE}(\mathbf{x})$ of any uncertainty estimation approach becomes a confidence score $\mathrm{C}(\mathbf{x}) \in [0, 1] \subset \mathbb{R}$.

We experimented with several ways of achieving this normalization, including quantile-based approach and simple linear normalization on maximum value obtained from validation dataset. Eventually we

Dataset	Num. instances	Av. document len.	Av. target len.	Language
		NMT		
WMT'14	4.51M / 3000 / **3003**	19.8 / 18.3	23.0 / 21.3	German-to-English
WMT'14	40.8M / 3000 / **3003**	33.5 / 32.1	29.2 / 27.0	French-to-English
		ATS		
XSum	204045 / 11332 / **11334**	454.6	26.1	English
AESLC	14436 / 1960 / **1906**	165.5	6.7	English
		QA		
CoQA	7199 / 500 / -	271.4	2.7	English
bAbiQA	2000 / - / **200**	31.1	1.0	English

Table 7: Quantitative information regarding the datasets from experiments. It includes the count of instances available for the training, validation, and **test** sets, as well as the mean lengths of both texts and targets (answers / translations / summaries) measured in terms of tokens. In addition, the languages of the source and target texts are also specified.

```yaml
hydra:
  run:
    dir: ${cache_path}/${task}/${model}/${dataset}/${now:%Y-%m-%d}/${now:%H-%M-%S}

cache_path: ./workdir/output
save_path: '${hydra:run.dir}'

device: cpu

task: qa

dataset: coqa
text_column: questions
label_column: answers
prompt: "Answer a question given a story. Output only the answer.\nStory:\n{story}\n\nQuestion:\n{question}\
     n\nAnswer:\n"
train_split: train
eval_split: validation
max_new_tokens: 20
load_from_disk: false

train_dataset: null
train_test_split: false
test_split_size: 1

background_train_dataset: allenai/c4
background_train_dataset_text_column: text
background_train_dataset_label_column: url
background_train_dataset_data_files: en/c4-train.00000-of-01024.json.gz
background_load_from_disk: false

subsample_background_train_dataset: 1000
subsample_train_dataset: 1000
subsample_eval_dataset: -1

model: lmsys/vicuna-7b-v1.5
use_auth_token:

use_density_based_ue: true
use_seq_ue: true
use_tok_ue: false

ignore_exceptions: false

batch_size: 1
deberta_batch_size: 10

seed:
    - 1
```

Figure 4: Config Example for Question Answering on CoQA dataset.

performed normalization as a calibration procedure, where normalized confidence score represents expected value of generation quality metric of choice (i.e. RougeL) for a given uncertainty estimate. This expectation is estimated by computing sample averages of quality metric over bins of uncertainty estimates, calculated for some validation dataset. For RougeL metric, the confidence estimate $\mathrm{C}(\mathbf{x}_{input})$ thus becomes:

$$\mathrm{C}(\mathbf{x}_{input}) = \frac{1}{|\mathcal{B}|} \sum_{\mathbf{x}_i, \mathbf{y}_i \in \mathcal{B}} rougeL(\hat{\mathbf{y}}_i, \mathbf{y}_i),$$

where $\hat{\mathbf{y}}_i$ is model output for input $\mathbf{x}_i$, and

$$\mathcal{B} = \{(\mathbf{x}, \mathbf{y}) \in \mathcal{D}_{calib} \mid \mathrm{UE}(\mathbf{x}) \in [\mathrm{UE}_{\min}, \mathrm{UE}_{\max})\}$$

is the bin to which uncertainty estimate of the input belongs. The bounds of this bin are selected from the predetermined set of bin boundaries to be the neighboring pair for which condition

$$\mathrm{UE}_{\min} \leq \mathrm{UE}(\mathbf{x}_{input}) < \mathrm{UE}_{\max}$$

is satisfied.

This dataset $\mathcal{D}_{calib}$ is constructed to be representative of different modes of operation of a given model. For this purpose it is constructed as a mixture of several different datasets for different tasks, with different values of relevant statistics, such as input sequence length, typical generated output length etc.

It is obvious that quality of this normalized confidence score depends heavily on the size and diversity of the calibration dataset. In general we consider the problem of translating opaque uncertainty estimates into intuitive absolute confidence scores, that correctly represent likelihood of the generated output being correct and relevant, as an important and complicated task. We leave solving this problem in a more efficient and universal way to the future work.