LARGE LANGUAGE MODEL CONFIDENCE ESTIMATION VIA BLACK-BOX ACCESS

Anonymous authors
Paper under double-blind review

ABSTRACT

Estimating uncertainty or confidence in the responses of a model can be significant in evaluating trust not only in the responses, but also in the model as a whole. In this paper, we explore the problem of estimating confidence for responses of large language models (LLMs) with simply black-box or query access to them. We propose a simple and extensible framework where, we engineer novel features and train a (interpretable) model (viz. logistic regression) on these features to estimate the confidence. We empirically demonstrate that our simple framework is effective in estimating confidence of Flan-ul2, Llama-13b and Mistral-7b on four benchmark Q&A tasks as well as of Pegasus-large and BART-large on two benchmark summarization tasks with it surpassing baselines by even over 10% (on AUROC) in some cases. Additionally, our interpretable approach provides insight into features that are predictive of confidence, leading to the interesting and useful discovery that our confidence models built for one LLM generalize zero-shot across others on a given dataset.

1 INTRODUCTION

Given the proliferation of deep learning over the last decade or so (Goodfellow et al., 2016), uncertainty or confidence estimation of these models has been an active research area (Gawlikowski et al., 2023). Predicting accurate confidences in the generations produced by a large language model (LLM) are crucial for eliciting trust in the model and is also helpful for benchmarking and ranking competing models (Ye et al., 2024). Moreover, LLM hallucination detection and mitigation, which is one of the most pressing problems in artificial intelligence research today (Tonmoy et al., 2024), can also benefit significantly from accurate confidence estimation as it would serve as a strong indicator of the faithfulness of a LLM response. This applies to even settings where strategies such as retrieval augmented generation (RAG) are used (Gao et al., 2023) to mitigate hallucinations. Methods for confidence estimation in LLMs assuming just black-box or query access have been explored only recently (Kuhn et al., 2023; Lin et al., 2024) and this area of research is still largely in its infancy. However, effective solutions here could have significant impact given their low requirement (i.e. just query access) and consequently wide applicability.

There exists a slight difference in what is considered as uncertainty versus confidence in literature (Lin et al., 2024) and so to be clear we now formally state the exact problem we are solving. Let (x, y) denote an input-output pair, where x is the input prompt and y the expected ground truth response. Let $f(.)$ denote an LLM that takes the input x and produces a response $f(x)$. Let $\lambda(., .)$ denote a similarity metric (viz. rouge, bertscore, etc.) that can compare two pieces of text and output a value in $[0, 1]$, where 0 implies the texts are very different while 1 implies they are exactly the same. Then given some threshold $\theta \in [0, 1]$, we want to estimate the following probability for an input text x:

$$\text{Probability of correct } = P\left(\lambda(y, f(x)) \geq \theta | x\right) \tag{1}$$

In other words, we want to estimate the probability that the response outputted by the LLM for some input is correct. Unlike for classification or regression where the responses can be compared exactly, text allows for variation in response where even if they do not match exactly they might be semantically the same. Hence, we introduce the threshold θ which will typically be tuned based on the metric, the dataset and the LLM.

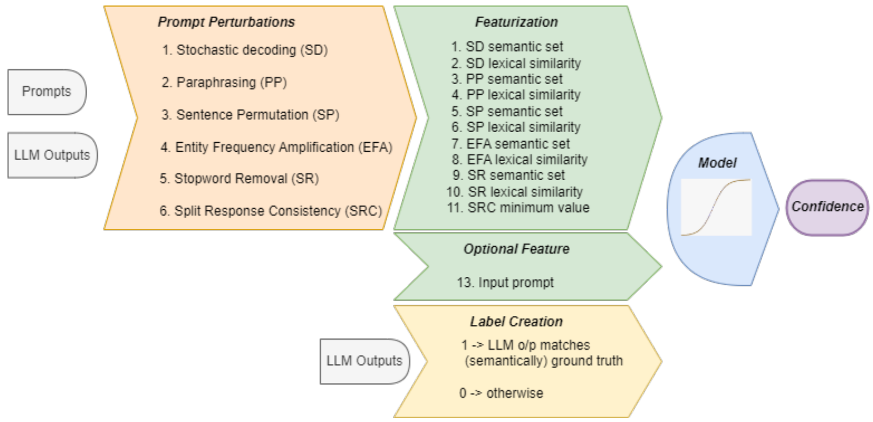

Figure 1: Above we see our (extensible) framework to estimate confidence of LLM responses. We propose six prompt perturbations which then can be converted to features based on semantic diversity in the responses and lexical similarity. The input (tokenized) prompt can optionally be also passed as a feature. The output labels for each (input) prompt are created by checking if the LLM output is correct or not. A (interpretable) logistic regression model is then trained on these features and outputs so that for any new input prompt and LLM response we can estimate the confidence of it being correct based on our model. Moreover, we can also ascertain the features important in estimating these confidences.

Having black-box access to an LLM limits the strategies one could leverage to ascertain confidence, but if the proposed strategies are effective they could be widely applied. Previous approaches (Kuhn et al., 2023; Lin et al., 2024; Jiang et al., 2023b) predominantly exploit the variability in the outputs for a given input prompt or based on an ensemble of prompts computing different estimators. Our approach enhances this idea where we design different ways of manipulating the input prompt and based on the variability of the answers produce values for each such manipulation. *We aver to these values as features.* Based on these features computed for different inputs we train a model to predict if the response was correct or incorrect. The probability of each such prediction is then the confidence that we output. Since, the models we use to produce such predictions are simple (viz. logistic regression) the confidence estimates are typically well calibrated (Morrison, 2012). Moreover, being interpretable we can also see which features were more crucial in the estimation. This general framework and the features we engineer are shown in Figure 1. The framework is extensible, since more features or prompt perturbations can be easily added to this framework.

We observe in the experiments that we outperform state-of-art baselines for black-box LLM confidence estimation on standard metrics such as Area Under the Receiver Operator Characteristic (AUROC) and Area Under Accuracy-Rejection Curve (AUARC), where improvements in AUROC are over 10% in some cases. The confidence model being interpretable we also analyze which features are important for different LLM and dataset combinations. We interestingly find that for a given dataset the important features are shared across LLMs. Intrigued by this finding we apply confidence models built for one LLM to the responses of another and further find that they generalize well across LLMs. This opens up the possibility of simply building a single (universal) confidence model for some chosen LLM and zero shot applying it to other LLMs on a dataset.

2 Related Work

The literature studying approaches for estimating the uncertainty in a machine learning model's prediction is large. One organization of this body of work involves dichotomizing it into *post-hoc* and *ab initio* approaches. Post-hoc methods attempt to calibrate outputs of a pre-trained model such that the estimate uncertainties correlate well with the accuracy of the model. These include histogram binning Zadrozny & Elkan (2001); Naeini et al. (2015), isotonic regression Zadrozny & Elkan (2002), and parametric mapping approaches, including matrix, vector, and temperature scaling Platt et al. (1999); Guo et al. (2017); Kull et al. (2019). While variants of these approaches Shen et al. (2024); Desai & Durrett (2020) have been adopted for LLMs they assume a white-box setting where access to the LLM's representations are available. In contrast, our approach quantifies a LLM's uncertainties without requiring access to the internals of the LLM. Ab initio approaches, including, training with mix-up augmentations Zhang et al. (2017), confidence penalties Pereyra et al. (2017), focal loss Mukhoti et al. (2020), label-smoothing Szegedy et al. (2016), (approximate) Bayesian procedures Izmailov et al. (2021), or those that involve ensembling over multiple models arrived at by retraining from different random initializations Lakshminarayanan et al. (2017) require substantial changes to the training process or severely increase computational burden, making them difficult to use with LLMs.

For LLMs in particular, recent works Jiang et al. (2021); Xiao et al. (2022); Chen et al. (2022) have empirically found evidence of miscalibration and had varying degrees of success in better calibrating smaller LLMs using mixup Park & Caragea (2022), temperature scaling and label smoothing Desai & Durrett (2020). Others Lin et al. (2022) have employed supervised fine-tuning to produce verbalized uncertainties to be better calibrated on certain tasks. However, this additionally requires the ability to compute gradients of the LLM's parameters. Our black-box approach has no such requirement. Another body of work Kadavath et al. (2022); Mielke et al. (2022); Zhang et al. (2021), learns an auxiliary model for predicting whether a LLM's generation is incorrect. We also employ an auxiliary model, but rely on only the prompts to the LLM and the generations produced by the LLM to train it.

Similar to us, other recent works have also explored black-box approaches. For instance, in Kuhn et al. (2023), multiple completions from an LLM are generated, grouped based on semantic content, and uncertainty is quantified across these semantic groups. Lin et al. (2024) exploit insights from spectral clustering to further finesse this process. In Tian et al. (2023); Xiong et al. (2024) the authors use carefully crafted prompts for certain more capable LLMs to express better-calibrated uncertainties. However, this approach is less effective for smaller and open-sourced LLMs Shen et al. (2024). Others Jiang et al. (2023b) have relied on ensembles of prompts created using templates or reordering of examples in few shot settings to quantify confidences. We on the other hand propose dynamic variations of the prompt applicable (even) in the zero-shot setting, where for certain of our features we only analyze the response without any variation in the prompt.

3 METHODOLOGY

We now describe our methodology to estimate confidences for individual LLM outputs.

3.1 ELICITATION OF VARIABLE LLM BEHAVIOR

We first propose six black-box strategies that can elicit variable behavior in an LLM indicative of how trustworthy its output is likely to be. Based on this variability we construct features for our confidence model in the next subsection. Note that all strategies may not be relevant in all cases. For instance, some of the strategies require a context in the prompt, while others such as SRC require longer responses (two or more sentences). For all the perturbations but for Stochastic Decoding and Split Response Consistency, the perturbations are applied to the context if available or to the question of the input.

Stochastic Decoding (SD): This is the simplest strategy which is also done in previous works. Here the prompt is not varied, but rather using various decoding strategies comprising of greedy , beam search and nucleus sampling (Holtzman et al., 2020) multiple outputs are sampled. As seen in Table 1 first row after sampling one could have four different outputs, which could be indicative of the LLM not being confident in its response. Specifically in the experiments, we obtain one generation using greedy and beam search decoding technique and 3 generations using nucleus sampling.

Paraphrasing (PP): In this strategy we paraphrase the context in the prompt and observe how that changes the output. An example of this is shown in Table 1. For paraphrasing, we use back translation, where we convert the original prompt into another language and translate it back into English. We use machine translation models from Helsinki-NLP on huggingface and translate the text from English to French and then back to English. This new prompt then can be used to query the LLM. Changes to the output could indicate brittleness in the LLMs original response. One could also prompt an LLM to paraphrase the responses, however, in our initial experiments, we observed that when context is involved, the model does not paraphrase the entire context and parts of it were omitted.

Table 1: Below we see examples of different prompt perturbations for a prompt from the SQuAD dataset. The color blue and strike outs indicate changes to the input prompt. i) SD does not change the prompt (hence empty cell), but using a stochastic decoding scheme samples multiple responses (four example samplings shown). PP paraphrases the prompt. SP randomly reorders some of the sentences. EFA repeats certain sentences with entities in them. SR removes stopwords. SRC checks for consistency in reasonable size random splits of the LLM response (again prompt is not perturbed). The splitting of the two sentences indicates inconsistency as depicted in red. Thus, the perturbations test an LLM in complementary ways.

Input Prompt

context: The Normans (Norman : Nourmands ; French : Normands ; Latin : Normanni) are the people who, in the 10th and 11th centuries, gave their name to Normandy, a region of France. They descended from the Normands ("Norman" comes from "Norseman") of the raiders and pirates of Denmark, Iceland and Norway who, under their leader Rollo, agreed to swear allegiance to King Charles III of France of the West. During generations of assimilation and mixing with the native French and Roman-Gaulese populations, their descendants would gradually merge with the Carolingian cultures of West France. The distinct cultural and ethnic identity of the Normans originally emerged in the first half of the 10th century, and it continued to evolve over the centuries that followed.

question: In what country is Normandy located?

Prompt Pert.	Perturbed Prompt	Output
SD		France, Denmark, Iceland, Norway
PP	context: Normandy, a region in France came to bear because of Normans in the 10th and 11th centuries. They descended from the Normands ~~("Norman" comes from "Norseman")~~ of the raiders and pirates of Denmark, Iceland and Norway who, under their leader Rollo, agreed to swear allegiance to King Charles III of France of the West. There was generations of mixing with the Roman-Gaulese populations and native French. The distinct cultural and ethnic identity of the Normans originally emerged in the first half of the 10th century, and it continued to evolve over the centuries that followed. question: In what country is Normandy located?	Iceland
SP	context: The Normans (Norman : Nourmands ; French : Normands ; Latin : Normanni) are the people who, in the 10th and 11th centuries, gave their name to Normandy, a region of France. The distinct cultural and ethnic identity of the Normans originally emerged in the first half of the 10th century, and it continued to evolve over the centuries that followed. They descended from the Normands ("Norman" comes from "Norseman") of the raiders and pirates of Denmark, Iceland and Norway who, under their leader Rollo, agreed to swear allegiance to King Charles III of France of the West. During generations of assimilation and mixing with the native French and Roman-Gaulese populations, their descendants would gradually merge with the Carolingian cultures of West France. question: In what country is Normandy located?	Denmark
EFA	context: The Normans (Norman : Nourmands ; French : Normands ; Latin : Normanni) are the people who, in the 10th and 11th centuries, gave their name to Normandy, a region of France. The Normans (Norman : Nourmands ; French : Normands ; Latin : Normanni) are the people who, in the 10th and 11th centuries, gave their name to Normandy, a region of France. They descended from the Normands ("Norman" comes from "Norseman") of the raiders and pirates of Denmark, Iceland and Norway who, under their leader Rollo, agreed to swear allegiance to King Charles III of France of the West. During generations of assimilation and mixing with the native French and Roman-Gaulese populations, their descendants would gradually merge with the Carolingian cultures of West France. The distinct cultural and ethnic identity of the Normans originally emerged in the first half of the 10th century, and it continued to evolve over the centuries that followed. question: In what country is Normandy located?	France
SR	context: The Normans (Norman : Nourmands ; French : Normands ; Latin : Normanni) ~~are~~ the people ~~who, in the~~ 10th and 11th centuries, gave their name ~~to~~ Normandy, a region of France. They descended ~~from the~~ Normands ("Norman" comes from "Norseman") ~~of the~~ raiders and pirates of Denmark, Iceland and Norway ~~who,~~ under their leader Rollo, agreed ~~to~~ swear allegiance ~~to~~ King Charles III ~~of~~ France of the West. During generations ~~of~~ assimilation and mixing ~~with the~~ native French and Roman-Gaulese populations, ~~their~~ descendants would gradually merge ~~with the~~ Carolingian cultures ~~of~~ West France. ~~The~~ distinct cultural and ethnic identity ~~of the~~ Normans originally emerged ~~in the~~ first half ~~of the~~ 10th century, and it continued ~~to~~ evolve ~~over the~~ centuries that followed. question: In what country is Normandy located?	Norway
SRC		Normandy is located in Denmark. Normandy is located in Iceland.

Sentence Permutation (SP): If the input has several named entities, we noticed that when the order of the named entities is changed without changing the meaning of the sentence, the output of the LLM also varied. We first use named entity detector to identify the named entities and then randomly reorder certain number of these sentences. An example of this is seen in Table 1 third row, where the last sentence in the prompt is now the second sentence. As such, if the number of sentences with named entities is less than five, we reorder all of them. If it is greater than five, then we randomly select five and reorder them. Most such reorderings do not affect the LLM output if it is confident.

Entity Frequency Amplification (EFA): Similar to above, repeating sentences with named entities could also throw off the model's outputs. We sample a sentence from all the sentences with named entities and repeat it three times. Again, here too the output of the LLM should be maintained if the LLM is confident. An example of this is seen in Table 1 fourth row, where the first sentence is repeated twice.

Stopword Removal (SR): We remove stopwords from the context as specified by the NLTK library. Stopwords are commonly occurring words (viz. "the", "are", "to", etc.) that are assumed to have limited context specific information. Removal of such words should ideally not alter the response of an LLM if the LLM is certain of the answer. An example of this is seen in Table 1 fifth row, where the stopwords are striked out. We ensured that the negative words were not removed as they would change the meaning of the sentence.

Split Response Consistency (SRC): In this case like in the SD case the prompt is not perturbed. Rather the output is analyzed where it is randomly split such that each part is at least a single sentence. Semantic inconsistency between the two parts is measured using an NLI models contradiction probability, where one part is taken as the premise and the other the hypothesis. An example of this is seen in Table 1 last row, where the two sentences are clearly at odds with each other. This strategy though requires that the response is at least a couple of sentences long.

As seen in Table 7, the four perturbations above (PP, SP, EFA and SR) that alter the original prompt still maintain the semantics as intended in almost all cases.

3.2 Featurization

Now based on the above strategies we can construct features to train our confidence model. For each of the first five strategies above we create two types of features: i) based on semantics of the outputs and ii) based on lexical overlap. For the SRC these are not relevant so we create a different feature as seen below.

Semantic Set: Based on the responses of the first strategies (run multiple times) we create semantically equivalent sets for each. A semantically equivalent set consists of outputs that are semantically the same. If a response entails another response and vice-versa, then they both are grouped under the same semantic set. The number of such sets is a feature for our model. As such, more the number of sets lower the confidence estimate. For example, if from five paraphrasings we get responses excellent, great, bad, subpar and fantastic, then the number of semantic sets would be two as excellent, great and fantastic would form one semantic set, while bad and subpar would form the other.

Lexical Similarity: We compute the average lexical similarity for outputs of each of the first five strategies (run multiple times). The similarity can be measured using standard NLP metrics such as rouge, blue score etc. The higher the lexical similarity higher the estimated confidence. We use rouge score to quantify the lexical similarity. Considering the same five paraphrasings example described above we would compute the average rouge score considering pairs of the responses and use it as a feature.

SRC Minimum Value: As mentioned above, semantic inconsistency between the two parts is measured using an NLI models contradiction probability, where one part is taken as the premise and the other the hypothesis. The highest contradiction probability amongst multiple such partitions is the feature value for this strategy. In Table 1 last row, there are only two sentences so only one split would be done and since the sentences contradict each other the NLI contradiction probability would be high or consistency would be low.

Note that optionally one can also pass the entire prompt as a feature in addition to the above. In the experiments, we saw minimal improvement with such an addition. Semantic set and lexical similarity were first used by (Kuhn et al., 2023) where they applied it only for SD perturbation discussed in the previous section.

3.3 Label Creation and Confidence Estimation

Once we have the input features to our confidence model we now need to determine labels for these inputs. For training the model we compute labels by matching the LLM output to the ground truth response in the dataset, where a match corresponds to the label 1, while a mismatch corresponds to

Table 2: AUROCs on four Q&A and two summarization datasets (CNN, XSUM) using a total of five LLMs (Llama, Flan-ul2, Mistral, Pegasus, BART). Higher values are better. Best results **bolded**.

Dataset(LLM)	# of SS	Lexical Similarity	EigenValue	Eccentricity	Degree	SE	AVC	Ours
TriviaQA(Llama)	0.73	0.76	0.77	0.76	0.77	0.75	0.79	**0.88**
TriviaQA(Flan-ul2)	0.83	0.8	0.86	0.86	0.87	0.85	0.81	**0.95**
TriviaQA(Mistral)	0.65	0.72	0.76	0.75	0.75	0.68	0.73	**0.81** $\pm.003$
SQuAD(Llama)	0.65	0.72	0.74	0.58	0.72	0.61	0.61	**0.83** $\pm.004$
SQuAD(Flan-ul2)	0.6	0.7	0.67	0.65	0.67	0.63	0.66	**0.8** $\pm.007$
SQuAD(Mistral)	0.59	0.7	0.67	0.65	0.67	0.62	0.64	**0.84** $\pm.003$
CoQA(Llama)	0.61	0.74	0.76	0.76	0.77	0.64	0.78	**0.92**
CoQA(Flan-ul2)	0.61	0.76	0.78	0.78	0.79	0.63	0.76	**0.87** $\pm.001$
CoQA(Mistral)	0.56	0.74	0.79	0.77	0.79	0.59	0.75	**0.81** $\pm.002$
NQ(Llama)	0.65	0.75	0.75	0.73	0.74	0.68	0.74	**0.85** $\pm.003$
NQ(Flan-ul2)	0.76	0.76	0.86	0.86	0.86	0.81	0.84	**0.93** $\pm.002$
NQ(Mistral)	0.66	0.73	0.77	0.77	0.78	0.68	0.75	**0.83** $\pm.003$
CNN (Pegasus)	0.51	0.67	0.73	0.72	0.72	0.55	0.73	**0.77**
CNN (BART)	0.51	**0.60**	0.52	0.48	0.54	0.53	0.5	0.57
XSUM (Pegasus)	0.51	0.58	0.69	0.70	0.71	0.54	0.71	**0.73**
XSUM (BART)	0.51	**0.59**	0.53	0.51	0.52	0.52	0.53	0.57

a label 0. In particular, we use the rouge score to compute the similarity between the output and the ground truth and if the score is greater than a threshold of 0.3, it corresponds to label 1, otherwise it is deemed incorrect and is labeled 0 similar to previous works (Lin et al., 2024). With the described features and their labels we train a logistic regression model and use it for predicting confidence scores for out-of-sample outputs.

Given that logistic regression is also an interpretable model we can also study which of our features turn out to be most beneficial and if our model trained on one LLM is transferable to other LLMs for the same dataset. Transfer across datasets can be more challenging as some datasets have contexts (viz. SQuAD), while others do not (viz. NQ) amongst other factors such as difference in domains.

Table 3: AUARCs on four Q&A and two summarization datasets (CNN, XSUM) using a total of five LLMs (Llama, Flan-ul2, Mistral, Pegasus, BART). Higher values are better. Best results **bolded**.

Dataset(LLM)	# of SS	Lexical Similarity	EigenValue	Eccentricity	Degree	SE	AVC	Ours
TriviaQA(Llama)	0.77	0.8	0.8	0.8	0.8	0.79	0.8	**0.83** $\pm.01$
TriviaQA(Flan-ul2)	0.69	0.72	0.73	0.73	0.73	0.71	0.72	**0.74** $\pm.002$
TriviaQA(Mistral)	0.55	0.63	**0.64**	**0.64**	**0.64**	0.58	0.63	**0.64** $\pm.006$
SQuAD(Llama)	0.3	0.36	0.37	0.28	0.36	0.36	0.31	**0.68** $\pm.004$
SQuAD(Flan-ul2)	0.73	0.95	0.83	0.82	0.83	0.78	0.83	**0.96** $\pm.003$
SQuAD(Mistral)	0.72	0.93	0.82	0.82	0.82	0.76	0.83	**0.96** $\pm.004$
CoQA(Llama)	0.56	0.67	0.67	0.67	0.67	0.61	0.66	**0.71** $\pm.002$
CoQA(Flan-ul2)	0.7	0.79	**0.8**	0.79	0.79	0.73	0.77	**0.8** $\pm.005$
CoQA(Mistral)	0.46	0.62	0.64	0.63	**0.64**	0.51	0.62	0.61 $\pm.003$
NQ(Llama)	0.37	0.41	0.42	0.41	0.41	0.39	0.42	**0.45** $\pm.006$
NQ(Flan-ul2)	0.41	0.44	**0.47**	0.46	0.45	0.44	0.45	**0.47** $\pm.007$
NQ(Mistral)	0.32	0.38	0.40	0.40	0.39	0.36	0.39	**0.42** $\pm.007$
CNN (Pegasus)	0.45	0.51	0.53	0.43	0.52	0.48	0.47	**0.74** $\pm.004$
CNN (BART)	0.21	0.22	0.21	0.21	0.21	0.23	0.23	**0.34**
XSUM (Pegasus)	0.16	0.17	0.19	0.17	0.17	0.21	0.19	**0.27**
XSUM (BART)	0.21	0.22	0.20	0.21	0.22	0.23	0.22	**0.35**

4 EXPERIMENTS

We demonstrate the efficacy of our method on question answering and summarization tasks. For summarization, we used BART-large (Lewis et al., 2019) and Pegasus-large (Zhang et al., 2019) and for question answering, we used Mistral-7B-Instruct-v0.2 (Jiang et al., 2023a), Llama-2-13b chat version (Touvron et al., 2023), and Flan-ul2 models (Tay et al., 2023). For question answering we elicited responses from these models on four datasets, namely, CoQA (Reddy et al., 2019), SQuAD (Rajpurkar et al., 2016), TriviaQA (Joshi et al., 2017) and Natural Questions (NQ) (Kwiatkowski et al., 2019). CoQA and SQuAD provide the context and expect the model to respond to the question based on the context, while TriviaQA and NQ do not have a context and require the model to tap into its learnt knowledge. For our experiments, we use the validation splits for all the datasets as done previously (Lin et al., 2024). CoQA has 7983 datapoints, TriviaQA has 9960 datapoints, SQuAD has 10,600 datapoints and NQ has 7830 datapoints. For summarization, we used CNN Daily Mail (See et al., 2017) and (Hermann et al., 2015) and XSUM (Narayan et al., 2018) datasets. We use a subset of the validation splits of both the datasets comprising of 4000 datapoints. For detecting entailment, we use deberta-large-nli model which is specialized for NLI tasks (He et al., 2021).

Table 4: Up to four important features (absolute coefficient value $> 1e^{-4}$) ranked based on our logistic regression model for the different dataset and LLM combinations. Rank 1 indicates the most important feature, while Rank 4 is the least important amongst the four.

Dataset(LLM)	Rank 1	Rank 2	Rank 3	Rank 4
TriviaQA(Llama)	SD lexical similarity	SD semantic set	SR lexical similarity	PP lexical similarity
TriviaQA(Flan-ul2)	SD lexical similarity	SD semantic set	PP semantic set	PP lexical similarity
TriviaQA(Mistral)	SD lexical similarity	PP lexical similarity	SP semantic set	SD semantic set
SQuAD(Llama)	SP lexical similarity	EFA semantic set	-	-
SQuAD(Flan-ul2)	SP lexical similarity	-	-	-
SQuAD(Mistral)	SP lexical similarity	EFA semantic set	-	-
CoQA(Llama)	SD lexical similarity	EFA semantic set	SD semantic set	SR lexical similarity
CoQA(Flan-ul2)	SD lexical similarity	EFA semantic set	SD semantic set	SP lexical similarity
CoQA(Mistral)	SD lexical similarity	SD semantic set	EFA semantic set	EFA lexical similarity
NQ(Llama)	PP lexical similarity	SD semantic set	SD lexical similarity	SP lexical similarity
NQ(Flan-ul2)	SR semantic set	SD lexical similarity	SP lexical similarity	PP lexical similarity
NQ(Mistral)	PP lexical similarity	SD semantic set	SD lexical similarity	SP lexical similarity
CNN(Pegasus)	SD lexical similarity	EFA lexical similarity	SR lexical similarity	SP lexical similarity
CNN(BART)	SR lexical similarity	SP lexical similarity	EFA lexical similarity	SP semantic set
XSUM(Pegasus)	SD lexical similarity	EFA semantic set	PP lexical similarity	SD semantic set
XSUM(BART)	SR lexical similarity	SP lexical similarity	EFA lexical similarity	SP semantic set

We follow previous works (Lin et al., 2024), which used 1000 datapoints for hyperparameter tuning, to train our Logistic Regression Classifier and the rest of them were used for evaluation. As such, in Table 8 in the appendix, we show that our method is quite performant even with fewer training datapoints. For each of the prompt perturbations specified above, we use five generations for each perturbation for more robust evaluation. All results are averaged over five runs and we report standard deviations rounded to three decimal places for our method. We use zero-shot prompting for the datasets with context. For TriviaQA, Flan-ul2 and Mistral-7B-Instruct-v0.2 also worked well with zero shot prompting while Llama-2-13b chat was performant with a two-shot prompt. For NQ, we used a five shot prompt. The details about the prompts used are provided in the Appendix A. We used internally hosted models to generate the responses. Thus, we used V100s GPUs for the feature extraction step once the responses were generated. The logistic regression model was trained on an intel core CPU.

We consider methods proposed in recent works (Kuhn et al., 2023; Lin et al., 2024; Xiong et al., 2024) which are state-of-the-art as the baselines. (Kuhn et al., 2023) proposed computing the number of semantic sets, semantic entropy and lexical similarity metrics from the generated outputs. (Lin et al., 2024) use eigen value, eccentricity and degree metrics inspired from spectral clustering to estimate the uncertainty of the model. While (Xiong et al., 2024) used aggregated verbalized confidence scores. We use average verbalized confidence (AVC) as that performed the best in the previous work. To be consistent with our method we average over five estimates. We use the open source code provided by the authors of (Lin et al., 2024) for comparing with the baselines [1].

Table 5: AUROC of the logistic confidence model for one LLM applied to another on a given dataset. As can be seen our confidence models transfer quite well based on AUROC.

Dataset	Source LLM	AUROC Self	Target LLM 1 AUROC	Target LLM 2 AUROC
TriviaQA	Llama	0.88	0.94 (Flan-ul2)	0.80 (Mistral)
	Flan-ul2	0.94	0.87 (Llama)	0.80 (Mistral)
	Mistral	0.81	0.84 (Llama)	0.91 (Flan-ul2)
SQuAD	Llama	0.83	0.81 (Flan-ul2)	0.80 (Mistral)
	Flan-ul2	0.8	0.79 (Llama)	0.78 (Mistral)
	Mistral	0.84	0.82 (Llama)	0.83 (Flan-ul2)
CoQA	Llama	0.92	0.79 (Flan-ul2)	0.78 (Mistral)
	Flan-ul2	0.87	0.87 (Llama)	0.81 (Mistral)
	Mistral	0.81	0.88 (Llama)	0.86 (Flan-ul2)
NQ	Llama	0.85	0.91 (Flan-ul2)	0.83 (Mistral)
	Flan-ul2	0.93	0.83 (Llama)	0.82 (Mistral)
	Mistral	0.83	0.85 (Llama)	0.90 (Flan-ul2)
CNN	Pegasus	0.77	0.57 (BART)	-
	BART	0.57	0.77 (Pegasus)	-
XSUM	Pegasus	0.73	0.58 (BART)	-
	BART	0.57	0.71 (Pegasus)	-

4.1 Confidence Estimation

We use three metrics to evaluate effectiveness of the models: i) Area under the receiver operating characteristic (AUROC) curve which computes the model's discrimination ability for various thresholds. The curve is plotted by varying the thresholds of the prediction probabilities of the model and the false positive rate and the true positive rate form the X and the Y axes. The area under this curve is called the AUROC. ii) An accuracy rejection curve can also be plotted by increasing the rejection threshold gradually and plotting the model's average accuracy at that threshold. The area under this curve is called AUARC (Lin et al., 2024). iii) Expected calibration error (ECE) is also reported in Table 18 in the appendix which measures the discrepancy between accuracy and confidences.

In Table 2, we see that our method quite consistently outperforms all baselines on AUROC. This is also seen for for ECE in Table 18. For estimating the confidence of Llama's responses on TriviaQA, our model is better than the best baseline by 11 percentage points. We are also able to estimate the confidence on the SQuAD dataset using Mistral by 14 percentage points better than the closest competitor. Qualitatively similar results are seen for the SQuAD dataset using Flan-ul2 (better by 10 percentage points) and for the CoQA and NQ datasets using Llama (better by 15 and 10 percentage points respectively). Our results on the summarization datasets using LLMs that excel at summarization (viz. Pegasus and BART) we see again that we are either better or at least competitive.

Our performance is also superior to the baselines in most cases on the AUARC metric in Table 3. Our performance on Llama's generations based on the SQuAD dataset exceeds the best baseline's performance by 31 percentage points. In the case of Mistral's performance on TriviaQA and Flan-ul2's generations on CoQA, we are as good as the baseline. We are worse than the baseline on

[1]https://github.com/zlin7/UQ-NLG/ The results are different in some cases from those reported in their paper possibly because of different random splits and different LLMs used, since we did run the provided code.

Mistral's generations of CoQA, where our AUROC was also minimally better than the best baselines. In all other instances, our performance is better than others by 1 to 4 percentage points.

We believe these improvements can be attributed to our constructed features and our framework in general. Hence, in the next section we try to ascertain which features for which datasets and LLMs played an important role in predicting the confidences accurately. Note that such an analysis with high confidence is possible because our trained model is interpretable. We also tried to pass the tokenized input prompt as additional features (maximum length 256) to our logistic model, however, the improvements were minimal at best and in some cases the performance even dropped possibly because of the model overfitting given that there were now 100s of features. Hence, we do not report these results, although passing the input prompt is still a possibility in general.

Table 6: AUARC of the logistic confidence model for one LLM applied to another on a given dataset. As can be seen our confidence models transfer quite well based on AUARC as well.

Dataset	Source LLM	AUARC Self	Target LLM 1 AUARC	Target LLM 2 AUARC
TriviaQA	Llama	0.83	0.74 (Flan-ul2)	0.64 (Mistral)
	Flan-ul2	0.74	0.83 (Llama)	0.64 (Mistral)
	Mistral	0.64	0.83 (Llama)	0.73 (Flan-ul2)
SQuAD	Llama	0.68	0.62 (Flan-ul2)	0.63 (Mistral)
	Flan-ul2	0.96	0.89 (Llama)	0.91 (Mistral)
	Mistral	0.96	0.90 (Llama)	0.91 (Flan-ul2)
CoQA	Llama	0.71	0.79 (Flan-ul2)	0.61 (Mistral)
	Flan-ul2	0.80	0.70 (Llama)	0.61 (Mistral)
	Mistral	0.61	0.69 (Llama)	0.79 (Flan-ul2)
NQ	Llama	0.45	0.46 (Flan-ul2)	0.42 (Mistral)
	Flan-ul2	0.47	0.45 (Llama)	0.42 (Mistral)
	Mistral	0.42	0.45 (Llama)	0.46 (Flan-ul2)
CNN	Pegasus	0.74	0.34 (BART)	-
	BART	0.34	0.74 (Pegasus)	-
XSUM	Pegasus	0.27	0.34 (BART)	-
	BART	0.35	0.25 (Pegasus)	-

4.2 Confidence Model Interpretability and Transferability

Interpretability: We now study which features in our logistic model were instrumental for accurate confidence estimation. In Table 4, we see the top four features for each dataset-LLM combination. Blanks indicate that there were no features at that rank or lower where their logistic coefficient was greater than $1e^{-4}$. As can be seen the simplest feature SD plays a role in many cases. This indicates that variability of output for the same input prompt is a strong indicator of response correctness. Moreover, other features such as SP and EFA are also crucial in ascertaining confidence as seen in particular for the SQuAD dataset as well as the summarization datasets. This points to order bias when looking at contexts and brittleness to redundant information being also strong indicators of response accuracy. PP and SR also play a role in some cases, where they are more crucial for datasets with no contexts such as TriviaQA and NQ. This makes sense as the specific question is more important here in the absence of context and hence the absence of also other features such as SP and EFA. Both the lexical similarity and semantic set featurizations seem to be important in estimating confidence.

Looking across the datasets and LLMs we see an interesting trend. It seems that for a given dataset different LLMs have similar features that appear to be important. For instance, SP lexical similarity is the top feature for all three LLMs on SQuAD, while EFA based feature also appears for Llama and Mistral. For TriviaQA, SD and PP appear for all three models. For CoQA, SD and EFA appear. While for NQ, PP and SD appear as important for all the models. This trend points towards an interesting prospect of applying a confidence estimator of one LLM to other LLMs on a given dataset. As such, we could have a universal confidence estimator just built for one of the LLMs that we could apply across others with reasonable assurance. We explore this exciting possibility in the next part.

Transferability: Given the commonality between the important features across LLMs for a dataset we now try to test how well does our logistic confidence model for one LLM perform in estimating confidences of another LLM. As seen in Tables 5 and 6 our confidence models are actually quite transferable as they perform comparably or even sometimes better on the other LLMs than the LLM they were built for. This is particularly true for Mistral where, its confidence model performs better for the other two LLMs than itself even coming close in performance to their own confidence models in many cases.

This suggests that we could apply our approach to one LLM and then use the same confidence model to evaluate responses of other LLMs without having to build individual models for them. It would be interesting to further stress test this hypothesis in the future with more LLMs and datasets. Nonetheless, even in the current setup – of five LLMs and six datasets – this observation is interesting and useful.

5 Discussion

In summary, we have provided an extensible framework for black-box confidence estimation of LLM responses by proposing novel features that are indicative of response correctness. By building an interpretable logistic regression model based on these features we were able to obtain state-of-the-art performance in estimating confidence on six benchmark datasets (CoQA, SQuAD, NQ, TriviaQA, CNN Daily and XSUM) and using five powerful open source LLMs (Llama-2-13b-chat, Mistral-7B-Instruct-v0.2, Flan-ul2, Pegasus-large and BART-large). The interpretability of our confidence model aided in identifying features (viz. SD, SP, EFA,PP) that were instrumental in driving its performance for different LLM-dataset combinations. This led to the interesting realization that many of the features crucial for performance were shared across the confidence models of different LLMs for a dataset. We thus tested if the confidence models generalized across LLMs for a dataset and found that it indeed was the case leading to the interesting possibility of having an *universal* confidence model trained on just a single LLMs responses, but applied across many others.

Owing to the supervised nature of training the confidence model, one limitation of our approach is that at least some of the model's generations must be close to the ground truth for us to obtain a reasonable confidence estimator. Another limitation is that the results and insights were obtained based on datasets in English, but these insights might vary when looking at datasets in other languages. More varied tasks and models could be tested upon in the future. We used rouge to test accuracy of generations consistent with previous works, however, rouge, like also other NLP metrics, can be error prone. In terms of broader impact, our approach can be widely applied as it is simple and works with just black-box access to the LLM. Access to logits or internals of the model are not required. However, our estimates although accurate can be imperfect and this should be taken into account when using our approach in high stakes applications involving LLMs. One should also be cognizant of adversaries aware of our features trying to induce misplaced trust in LLMs they create or prefer.

Given the extensibility of our framework, in the future, it would be interesting to add more features as LLMs evolve. One class of such features might be those where the correctness of a response is checked through creating questions that are (causally) related to the original question and context, and seeing how the response varies by asking this question by itself as opposed to in conjunction with the original question and response. Such and other strategies may help in generalizing these confidence estimators also across datasets something that has been seen when we have additional access to logits of LLMs. Moreover, ideas from selective classification (Bartlett & Wegkamp, 2008; Geifman & El-Yaniv, 2017) could also be adapted for learning a better confidence model.

References

Bartlett and Wegkamp. Classification with a reject option using a hinge loss. *Journal of Machine Learning Research*, 2008.

Yangyi Chen, Lifan Yuan, Ganqu Cui, Zhiyuan Liu, and Heng Ji. A close look into the calibration of pre-trained language models. *arXiv preprint arXiv:2211.00151*, 2022.

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

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *arXiv:2312.10997*, 2023.

Jakob Gawlikowski, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, Muhammad Shahzad, Wen Yang, Richard Bamler, and Xiao Xiang Zhu. A survey of uncertainty in deep neural networks. *Artificial Intelligence Review*, 56(1):93, 2023.

Geifman and El-Yaniv. Selective classification for deep neural networks. *Advances of Neural Inf. Proc. Systems*, 2017.

Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT Press, 2016.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pp. 1321–1330. PMLR, 2017.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=XPZIaotutsD`.

Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In *NIPS*, pp. 1693–1701, 2015. URL `http://papers.nips.cc/paper/5945-teaching-machines-to-read-and-comprehend`.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL `https://openreview.net/forum?id=rygGQyrFvH`.

Pavel Izmailov, Sharad Vikram, Matthew D Hoffman, and Andrew Gordon Gordon Wilson. What are Bayesian neural network posteriors really like? In *International conference on machine learning*, pp. 4629–4640. PMLR, 2021.

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. Mistral 7b. *CoRR*, abs/2310.06825, 2023a. doi: 10.48550/ARXIV.2310.06825. URL `https://doi.org/10.48550/arXiv.2310.06825`.

Mingjian Jiang, Yangjun Ruan, Sicong Huang, Saifei Liao, Silviu Pitis, Roger Baker Grosse, and Jimmy Ba. Calibrating language models via augmented prompt ensembles. *ICML Workshop on Challenges in Deployable Generative AI*, 2023b.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 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, 2021.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In Regina Barzilay and Min-Yen Kan (eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pp. 1601–1611. Association for Computational Linguistics, 2017. doi: 10.18653/V1/P17-1147. URL `https://doi.org/10.18653/v1/P17-1147`.

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

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=VD-AYtP0dve.

Meelis Kull, Miquel Perello Nieto, Markus Kängsepp, Telmo Silva Filho, Hao Song, and Peter Flach. Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration. *Advances in Neural Information Processing Systems*, 32, 2019.

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

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in Neural Information Processing Systems*, 30, 2017.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461, 2019. URL http://arxiv.org/abs/1910.13461.

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

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. Generating with confidence: Uncertainty quantification for black-box large language models. *Transactions on Machine Learning Research*, 2024.

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

Geoffrey Stewart Morrison. Tutorial on logistic-regression calibration and fusion: Converting a score to a likelihood ratio. *Australian Journal of Forensic Sciences*, 2012.

Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, Stuart Golodetz, Philip Torr, and Puneet Dokania. Calibrating deep neural networks using focal loss. *Advances in Neural Information Processing Systems*, 33:15288–15299, 2020.

Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using Bayesian binning. In *Proceedings of AAAI*, volume 29, 2015.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *ArXiv*, abs/1808.08745, 2018.

Seo Yeon Park and Cornelia Caragea. On the calibration of pre-trained language models using mixup guided by area under the margin and saliency. *arXiv preprint arXiv:2203.07559*, 2022.

Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. Regularizing neural networks by penalizing confident output distributions. *arXiv preprint arXiv:1701.06548*, 2017.

John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100, 000+ questions for machine comprehension of text. In Jian Su, Xavier Carreras, and Kevin Duh (eds.), *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pp. 2383–2392. The Association for Computational Linguistics, 2016. doi: 10.18653/V1/D16-1264. URL https://doi.org/10.18653/v1/d16-1264.

Siva Reddy, Danqi Chen, and Christopher D. Manning. Coqa: A conversational question answering challenge. *Trans. Assoc. Comput. Linguistics*, 7:249–266, 2019. doi: 10.1162/TACL_A_00266. URL https://doi.org/10.1162/tacl_a_00266.

Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1073–1083, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1099. URL https://www.aclweb.org/anthology/P17-1099.

Maohao Shen, Subhro Das, Kristjan Greenewald, Prasanna Sattigeri, Gregory Wornell, and Soumya Ghosh. Thermometer: Towards universal calibration for large language models. In *ICML*, 2024.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

Yi Tay, Mostafa Dehghani, Vinh Q. Tran, Xavier Garcia, Jason Wei, Xuezhi Wang, Hyung Won Chung, Dara Bahri, Tal Schuster, Huaixiu Steven Zheng, Denny Zhou, Neil Houlsby, and Donald Metzler. UL2: unifying language learning paradigms. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/pdf?id=6ruVLB727MC.

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

S.M Towhidul Islam Tonmoy, S M Mehedi Zaman, Vinija Jain, Anku Rani, Vipula Rawte, Aman Chadha, and Amitava Das. A comprehensive survey of hallucination mitigation techniques in large language models. *arXiv:2401.01313*, 2024.

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. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288, 2023. doi: 10.48550/ARXIV.2307.09288. URL https://doi.org/10.48550/arXiv.2307.09288.

Yuxin Xiao, Paul Pu Liang, Umang Bhatt, Willie Neiswanger, Ruslan Salakhutdinov, and Louis-Philippe Morency. Uncertainty quantification with pre-trained language models: A large-scale empirical analysis. *arXiv preprint arXiv:2210.04714*, 2022.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, YIFEI LI, Jie Fu, Junxian He, and Bryan Hooi. Can LLMs express their uncertainty? an empirical evaluation of confidence elicitation in LLMs. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=gjeQKFxFpZ.

Fanghua Ye, Mingming Yang, Jianhui Pang, Derek F. Wong Longyue Wang, Emine Yilmaz, Shuming Shi, and Zhaopeng Tu. Benchmarking llms via uncertainty quantification. *arXiv:2401.12794*, 2024.

Bianca Zadrozny and Charles Elkan. Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. In *International Conference on Machine Learning*, volume 1, pp. 609–616, 2001.

Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 694–699. ACM, 2002.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization, 2019.

Shujian Zhang, Chengyue Gong, and Eunsol Choi. Knowing more about questions can help: Improving calibration in question answering. *arXiv preprint arXiv:2106.01494*, 2021.

A Prompt Design

Prompts for TriviaQA:

- **Flan-ul2 model** and **GPT-4**: Answer the following question in less than 5 words
  Q: {question}
  A:
- **Llama-2-13b-chat model** Answer these following question as succinctly as possible in less than 5 words
  Q: In Scotland a bothy/bothie is a?
  A: House
  Q: Who is Posh Spice in the spice girls pop band?
  A: Victoria Beckham
  Q: {question}
  A:
- **Mistral-7B-Instruct-v0.2 model** ¡s¿[INST] Answer the following question as succinctly as possible in plain text and in less than 5 words. question [/INST]

Prompts for CoQA

- **Flan-ul2 model**, **Llama-2-13b-chat model** and **GPT-4**: Provide an answer in less than 5 words for the following question based on the context below: context: {context} Question: {question} Answer:
- **Mistral-7B-Instruct-v0.2 model** ¡s¿[INST] Provide an answer in less than 5 words for the following question based on the context below:
  context: {context}
  Question: {question}
  Answer: [/INST]

Prompts for SQuAD

- **Flan-ul2 model**, **Llama-2-13b-chat model** and **GPT-4**: Provide an answer for the following question based on the context below, in less than 5 words:
- **Mistral-7B-Instruct-v0.2 model** ¡s¿[INST] Provide an answer for the following question based on the context below, in less than 5 words:
  context: {context}
  Question: {question}
  Answer: [/INST]

Prompts for NQ: For all the models we used the following prompt:
Here are 5 Example Question Answer pairs:
Question: who makes up the state council in russia
Answer: governors and presidents
Question: when does real time with bill maher come back
Answer: November 9, 2018
Question: where did the phrase american dream come from
Answer: the mystique regarding frontier life
Question: what do you call a group of eels
Answer: bed
Question: who wrote the score for mission impossible fallout
Answer: Lorne Balfe
Now answer the following Question succinctly, similar to the above examples:
Question: {question}
Answer:

Prompt for GPT-4 as-a-judge: Please provide a score between 0 and 1 of how similar the summaries are. 1 indicating very similar and 0 indicating very different.

Table 7: Percentage of prompt perturbations entailed by the original prompt for the SQuAD dataset. This dataset also has context unlike some of the other Q&A datasets and hence, is a more challenging case of our features to maintain semantics. As can be seen our perturbations produce the intended effect of maintaining the semantics of the original prompt in most cases.

Paraphrasing	Sentence Permutation	Entity Frequency Amplification	Stopword Removal
99.81%	99.23%	99.66%	99.12%

Table 8: Below we see how the AUROC, AUARC values vary with different number of samples used to train our logistic regression model for some of our datasets. As can be seen our uncertainty estimation procedure is performant even with fewer samples for training.

Dataset	LLM	250 samples	500 samples	1000 samples (results in main paper)
TriviaQA	Llama	0.83, 0.80	0.86, 0.81	0.88, 0.83
	Flan-ul2	0.95, 0.73	0.95, 0.74	0.95, 0.74
	Mistral	0.80, 0.63	0.80, 0.63	0.81, 0.64
SQuAD	Llama	0.8, 0.65	0.81, 0.66	0.83, 0.68
	Flan-ul2	0.76, 0.91	0.78, 0.94	0.8, 0.96
	Mistral	0.79, 0.90	0.81, 0.93	0.84, 0.96
CoQA	Llama	0.91, 0.70	0.92, 0.71	0.92, 0.71
	Flan-ul2	0.86, 0.79	0.87, 0.80	0.87, 0.80
	Mistral	0.80, 0.60	0.81, 0.61	0.81, 0.61
NQ	Llama	0.81, 0.4	0.82, 0.41	0.85, 0.45
	Flan-ul2	0.86, 0.43	0.87, 0.45	0.93, 0.47
	Mistral	0.80, 0.37	0.81, 0.39	0.83, 0.42

Table 9: ECEs on four Q&A and two summarization datasets (CNN, XSUM) using a total of five LLMs (Llama, Flan-ul2, Mistral, Pegasus, BART). Lower values are better. Best results **bolded**.

Dataset(LLM)	# of SS	Lexical Similarity	EigenValue	Eccentricity	Degree	SE	AVC	Ours
TriviaQA(Llama)	0.13	0.12	0.11	0.11	0.1	0.12	0.09	**0.04**
TriviaQA(Flan-ul2)	0.06	0.07	0.05	0.05	0.05	0.07	0.06	**0.01**
TriviaQA(Mistral)	0.17	0.12	0.1	0.1	0.11	0.16	0.11	**0.05**
SQuAD(Llama)	0.15	0.12	0.1	0.24	0.13	0.18	0.18	**0.04**
SQuAD(Flan-ul2)	0.17	0.09	0.13	0.14	0.14	0.17	0.16	**0.06**
SQuAD(Mistral)	0.2	0.12	0.14	0.15	0.14	0.17	0.15	**0.04**
CoQA(Llama)	0.16	0.1	0.08	0.09	0.09	0.18	0.09	**0.02**
CoQA(Flan-ul2)	0.15	0.11	0.09	0.09	0.09	0.17	0.08	**0.03**
CoQA(Mistral)	0.18	0.1	0.07	0.09	0.07	0.21	0.09	**0.05**
NQ(Llama)	0.13	0.08	0.08	0.09	0.09	0.12	0.08	**0.04**
NQ(Flan-ul2)	0.1	0.09	0.06	0.06	0.06	0.06	0.05	**0.02**
NQ(Mistral)	0.15	0.09	0.11	0.1	0.09	0.12	0.09	**0.05**
CNN (Pegasus)	0.19	0.16	0.11	0.12	0.12	0.19	0.09	**0.07**
CNN (BART)	0.51	**0.19**	0.26	0.29	0.25	0.26	0.24	**0.19**
XSUM (Pegasus)	0.21	0.2	0.15	0.13	0.11	0.21	0.11	**0.09**
XSUM (BART)	0.26	0.22	0.24	0.27	0.26	0.25	0.23	**0.2**

Table 10: AUROCs on four Q&A datasets using GPT-4. Higher values are better. Best results **bolded**.

Dataset(LLM)	# of SS	Lexical Similarity	EigenValue	Eccentricity	Degree	SE	AVC	Ours
TriviaQA(GPT-4)	0.89	0.91	0.91	0.92	0.91	0.92	0.94	**0.96**$\pm.007$
SQuAD(GPT-4)	0.79	0.82	0.84	0.79	0.83	0.81	0.86	**0.91**$\pm.004$
CoQA(GPT-4)	0.81	0.86	0.88	0.87	0.88	0.89	0.91	**0.95**$\pm.005$
NQ(GPT-4)	0.81	0.85	0.85	0.85	0.88	0.89	0.9	**0.93**$\pm.003$

Table 11: AUARCs on four Q&A datasets using GPT-4. Higher values are better. Best results **bolded**.

Dataset(LLM)	# of SS	Lexical Similarity	EigenValue	Eccentricity	Degree	SE	AVC	Ours
TriviaQA(GPT-4)	0.8	0.84	0.84	0.84	0.82	0.84	0.85	**0.89**$\pm.004$
SQuAD(GPT-4)	0.7	0.72	0.72	0.63	0.66	0.69	0.71	**0.83**$\pm.006$
CoQA(GPT-4)	0.68	0.73	0.72	0.73	0.74	0.72	0.76	**0.86**$\pm.011$
NQ(GPT-4)	0.69	0.73	0.74	0.74	0.74	0.73	0.72	**0.79**$\pm.007$

Table 12: ECEs on four Q&A datasets using GPT-4. Lower values are better. Best results **bolded**.

Dataset(LLM)	# of SS	Lexical Similarity	EigenValue	Eccentricity	Degree	SE	AVC	Ours
TriviaQA(GPT-4)	0.07	0.08	0.09	0.09	0.08	0.09	0.03	**0.01**
SQuAD(GPT-4)	0.11	0.09	0.08	0.19	0.07	0.1	0.11	**0.02**
CoQA(GPT-4)	0.11	0.09	0.08	0.08	0.08	0.06	0.05	**0.02**
NQ(GPT-4)	0.1	0.05	0.05	0.06	0.06	0.09	0.06	**0.02**

Table 13: AUROCs on four Q&A and two summarization datasets (CNN, XSUM) using a total of five LLMs (Llama, Flan-ul2, Mistral, Pegasus, BART), where the number of queries to the LLMs is the same for the baselines and our method. Higher values are better. Best results **bolded**.

Dataset(LLM)	# of SS	Lexical Similarity	EigenValue	Eccentricity	Degree	SE	AVC	Ours
TriviaQA(Llama)	0.74	0.76	0.76	0.77	0.77	0.76	0.79	**0.88**
TriviaQA(Flan-ul2)	0.82	0.81	0.87	0.86	0.86	0.85	0.81	**0.95**
TriviaQA(Mistral)	0.65	0.72	0.76	0.75	0.75	0.68	0.73	**0.81**
SQuAD(Llama)	0.65	0.72	0.74	0.58	0.72	0.61	0.61	**0.83**
SQuAD(Flan-ul2)	0.6	0.7	0.67	0.65	0.67	0.63	0.66	**0.8**
SQuAD(Mistral)	0.59	0.7	0.67	0.65	0.67	0.62	0.64	**0.84**
CoQA(Llama)	0.61	0.74	0.76	0.76	0.77	0.64	0.78	**0.92**
CoQA(Flan-ul2)	0.61	0.76	0.78	0.78	0.79	0.63	0.76	**0.87**
CoQA(Mistral)	0.56	0.74	0.79	0.77	0.79	0.59	0.75	**0.81**
NQ(Llama)	0.65	0.75	0.75	0.73	0.74	0.68	0.74	**0.85**
NQ(Flan-ul2)	0.76	0.76	0.86	0.86	0.86	0.81	0.84	**0.93**
NQ(Mistral)	0.66	0.73	0.77	0.77	0.78	0.68	0.75	**0.83**
CNN (Pegasus)	0.51	0.67	0.73	0.72	0.72	0.55	0.73	**0.77**
CNN (BART)	0.51	**0.59**	0.52	0.48	0.54	0.53	0.5	0.57
XSUM (Pegasus)	0.51	0.58	0.69	0.70	0.71	0.54	0.71	**0.73**
XSUM (BART)	0.51	**0.59**	0.54	0.52	0.52	0.52	0.53	0.57

Table 14: AUARCs on four Q&A and two summarization datasets (CNN, XSUM) using a total of five LLMs (Llama, Flan-ul2, Mistral, Pegasus, BART), where the number of queries to the LLMs is the same for the baselines and our method. Higher values are better. Best results **bolded**.

Dataset(LLM)	# of SS	Lexical Similarity	EigenValue	Eccentricity	Degree	SE	AVC	Ours
TriviaQA(Llama)	0.76	0.8	0.81	0.8	0.8	0.79	0.8	**0.83**
TriviaQA(Flan-ul2)	0.7	0.72	0.73	0.73	0.73	0.71	0.72	**0.74**
TriviaQA(Mistral)	0.55	0.63	**0.64**	**0.64**	**0.64**	0.58	0.63	**0.64**
SQuAD(Llama)	0.3	0.36	0.37	0.28	0.36	0.36	0.31	**0.68**
SQuAD(Flan-ul2)	0.73	0.95	0.83	0.82	0.83	0.78	0.83	**0.96**
SQuAD(Mistral)	0.72	0.93	0.82	0.82	0.82	0.76	0.83	**0.96**
CoQA(Llama)	0.56	0.67	0.67	0.67	0.67	0.61	0.66	**0.71**
CoQA(Flan-ul2)	0.7	0.79	**0.8**	0.79	0.79	0.73	0.77	**0.8**
CoQA(Mistral)	0.46	0.62	0.64	0.63	**0.64**	0.51	0.62	0.61
NQ(Llama)	0.37	0.41	0.42	0.41	0.41	0.39	0.42	**0.45**
NQ(Flan-ul2)	0.41	0.44	**0.47**	0.46	0.45	0.44	0.45	**0.47**
NQ(Mistral)	0.32	0.38	0.40	0.40	0.39	0.36	0.39	**0.42**
CNN (Pegasus)	0.45	0.51	0.53	0.43	0.52	0.48	0.47	**0.74**
CNN (BART)	0.21	0.22	0.21	0.21	0.21	0.23	0.23	**0.34**
XSUM (Pegasus)	0.16	0.17	0.19	0.17	0.17	0.21	0.19	**0.27**
XSUM (BART)	0.21	0.22	0.20	0.21	0.22	0.23	0.22	**0.35**

Table 15: Results with different number of decodings (for each of the features) using our method. Five decodings correspond to results in the paper. As can be seen reducing to three decodings our approach still maintains performance.

Dataset(LLM)	Our AUROC 5 decodings	Our AUROC 3 decodings	Our AUARC 5 decodings	Our AUARC 3 decodings	Our ECE 5 decodings	Our ECE 3 decodings
TriviaQA(Llama)	0.88	0.86	0.83	0.81	0.04	0.05
TriviaQA(Flan-ul2)	0.95	0.94	0.74	0.72	0.01	0.02
TriviaQA(Mistral)	0.81	0.81	0.64	0.65	0.05	0.05
SQuAD(Llama)	0.83	0.81	0.68	0.65	0.04	0.06
SQuAD(Flan-ul2)	0.8	0.8	0.96	0.94	0.06	0.08
SQuAD(Mistral)	0.84	0.82	0.96	0.93	0.04	0.05
CoQA(Llama)	0.92	0.91	0.71	0.69	0.02	0.03
CoQA(Flan-ul2)	0.87	0.85	0.8	0.78	0.03	0.05
CoQA(Mistral)	0.81	0.8	0.61	0.6	0.05	0.06
NQ(Llama)	0.85	0.83	0.45	0.44	0.04	0.06
NQ(Flan-ul2)	0.93	0.91	0.47	0.45	0.02	0.03
NQ(Mistral)	0.83	0.81	0.42	0.4	0.05	0.06
CNN (Pegasus)	0.77	0.75	0.74	0.72	0.07	0.09
CNN (BART)	0.57	0.55	0.34	0.33	0.19	0.21
XSUM (Pegasus)	0.73	0.71	0.27	0.25	0.09	0.11
XSUM (BART)	0.57	0.55	0.35	0.33	0.2	0.22

Table 16: AUROCs on two summarization datasets (CNN, XSUM) with GPT-4 as a judge. Higher values are better. Best results **bolded**.

Dataset(LLM)	# of SS	Lexical Similarity	EigenValue	Eccentricity	Degree	SE	AVC	Ours
CNN (Pegasus)	0.54	0.65	0.76	0.77	0.75	0.61	0.75	**0.81**
CNN (BART)	0.55	**0.64**	0.55	0.52	0.58	0.56	0.54	**0.64**
XSUM (Pegasus)	0.56	0.62	0.72	0.74	0.73	0.6	0.75	**0.79**
XSUM (BART)	0.55	**0.63**	0.56	0.54	0.55	0.56	0.59	0.61

Table 17: AUARCs two summarization datasets (CNN, XSUM) with GPT-4 as a judge. Higher values are better. Best results **bolded**.

Dataset(LLM)	# of SS	Lexical Similarity	EigenValue	Eccentricity	Degree	SE	AVC	Ours
CNN (Pegasus)	0.49	0.55	0.58	0.49	0.57	0.52	0.53	**0.77**
CNN (BART)	0.25	0.26	0.27	0.26	0.26	0.27	0.29	**0.35**
XSUM (Pegasus)	0.19	0.22	0.23	0.2	0.21	0.23	0.21	**0.29**
XSUM (BART)	0.26	0.26	0.25	0.27	0.27	0.27	0.26	**0.37**

Table 18: ECEs two summarization datasets (CNN, XSUM) with GPT-4 as a judge. Lower values are better. Best results **bolded**.

Dataset(LLM)	# of SS	Lexical Similarity	EigenValue	Eccentricity	Degree	SE	AVC	Ours
CNN (Pegasus)	0.18	0.14	0.11	0.1	0.09	0.15	0.07	**0.05**
CNN (BART)	0.48	0.17	0.24	0.25	0.22	0.22	0.22	**0.14**
XSUM (Pegasus)	0.18	0.18	0.13	0.11	0.09	0.17	0.1	**0.06**
XSUM (BART)	0.23	0.19	0.21	0.23	0.23	0.22	0.2	**0.16**