Ambiguity Detection and Uncertainty Calibration for Question Answering with Large Language Models

Zhengyan Shi[1*] **Giuseppe Castellucci**[2] **Simone Filice**[3*] **Saar Kuzi**[2]
Elad Kravi[4*] **Eugene Agichtein**[2] **Oleg Rokhlenko**[2] **Shervin Malmasi**[2]
[1]University College London [2]Amazon [3]Technology Innovation Institute [4]Meta
michaelszx117,filice.simone,eladkravi@gmail.com
giusecas,eugeneag,olegro,malmasi@amazon.com

Abstract

Large Language Models (LLMs) have demonstrated excellent capabilities in Question Answering (QA) tasks, yet their ability to identify and address ambiguous questions remains underdeveloped. Ambiguities in user queries often lead to inaccurate or misleading answers, undermining user trust in these systems. Despite prior attempts using prompt-based methods, performance has largely been equivalent to random guessing, leaving a significant gap in effective ambiguity detection. To address this, we propose a novel framework for detecting ambiguous questions within LLM-based QA systems. We first prompt an LLM to generate multiple answers to a question, and then analyze them to infer the ambiguity. We propose to use a lightweight Random Forest model, trained on a bootstrapped and shuffled 6-shot examples dataset. Experimental results on ASQA, PACIFIC, and ABG-COQA datasets demonstrate the effectiveness of our approach, with accuracy up to 70.8%. Furthermore, our framework enhances the confidence calibration of LLM outputs, leading to more trustworthy QA systems that are able to handle complex questions.

1 Introduction

Recent advancements in Large Language Models (LLM) (Chung et al., 2022; Touvron et al., 2023; OpenAI, 2023) have significantly improved their capabilities in Question Answering (QA). However, users often ask under-specified questions that can have multiple interpretations (Min et al., 2020; Sun et al., 2023). Those ambiguities typically lead to inaccurate or misleading answers, which undermine the user trust in the systems (Ovalle et al., 2023). Identifying questions requiring clarification is thus a crucial task to build trustworthy NLP systems.

Recent studies (Cole et al., 2023; Deng et al., 2023) explored how LLMs can detect question ambiguity with prompting (*e.g.,* binary prompts where LLM answers with ‘Yes’ or ‘No’). These works found that the prompting strategy is ineffective and performs at random guessing levels.

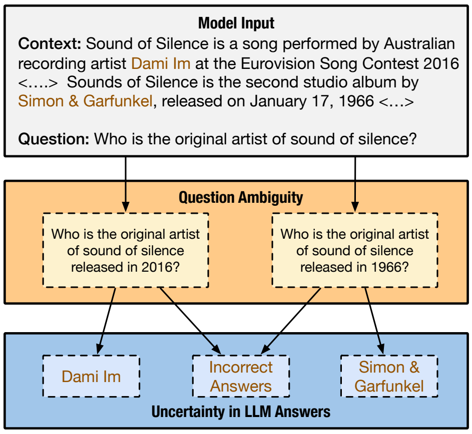

Figure 1: Ambiguity can either originate from the inherent ambiguity in the question (denotational uncertainty) or stem from the model's own indecision about potential answers (epistemic uncertainty).

In light of these findings, we propose to address the problem from a different angle by analyzing the responses of the LLM to the potentially ambiguous question. Intuitively, as illustrated in Figure 1, if an LLM provides multiple plausible responses, such as “*Dami Im*” and “*Simon & Garfunkel*” for the question “*Who is the original artist of Sound of Silence?*”, it can suggest ambiguity in the user question. Therefore, we hypothesize that understanding the variance of the LLM outputs can assist in detecting the ambiguity of questions.

A straightforward implementation would be to prompt the LLM to generate many possible answers to the question and then measure the entropy (*i.e.,* uncertainty) over the answers (Kuhn et al., 2023; Lin et al., 2023). The entropy can serve as a proxy for the question ambiguity: when the LLM insists on a single answer, the entropy will be 0 (indicating *non-ambiguity*); instead, if the LLM

[*]Work done while working at Amazon.

Proceedings of the 5th Workshop on Trustworthy NLP (TrustNLP 2025), pages 41–55
May 3, 2025 ©2025 Association for Computational Linguistics

is confident about multiple answers, the resulting entropy would increase towards 1 (thus indicating *ambiguity*). However, LLMs often produce incorrect, incomplete, or misleading answers, due to a lack of specific knowledge, hallucination, or other underlying factors (Tian et al., 2023; Bang et al., 2023). In Figure 1, such LLMs' outputs, labeled as "*incorrect answers*", amplify the measured entropy. Therefore, a more refined interpretation model is necessary to discern the question ambiguity.

In this work, we propose a novel framework to detect ambiguity in questions in LLM-based QA systems in low-resource settings. As shown in Figure 2, our framework first prompts an LLM to generate multiple answers to a question given some contextual information, *i.e.,* supporting evidence in a retrieval-augmented setting (Lewis et al., 2020); we prompt the LLM through self-consistency prompting (Wang et al., 2022). Then, we use an interpreter model to analyze the answers with various distributional features of the LLM responses to infer the ambiguity. We found that a Random Forest (RF) model, trained on a diverse range of LLM output patterns simulated through *bootstrapping* based on a very few-shot example set, is capable of accurately identifying ambiguity in questions. This approach outperforms various baselines including self-interpretation by the LLM *itself*, a ROBERTA-based classifier, and different prompting strategies. In particular, we conduct experiments on the ASQA (Stelmakh et al., 2022), PACIFIC (Deng et al., 2022), and ABG-COQA (Guo et al., 2021) datasets, and show that our proposed framework substantially improves the performance of the ambiguity detection task, with accuracy levels up to 70.8%; this is a substantial improvement over the existing prompt-based approaches, which barely surpass a random baseline. Our evaluation also shows that the prediction probabilities derived from the RF are reliable indicators of the model's accuracy, which effectively reduces the likelihood of providing incorrect or misleading answers, thus improving the trustworthiness of the resulting system. Our analysis also explores the benefits of bootstrapping few-shot examples and reveals that our approach delivers much fewer false positives, compared to the heuristic method using entropy.

In summary, the contributions of this work are: i) we introduce a novel framework for ambiguity detection in LLM-based QA systems by prompting the LLM to generate multiple answers which are then analyzed by an RF model, trained using bootstrapping; ii) experiments on the ASQA, PACIFIC, and ABG-COQA datasets show that the proposed framework considerably enhances the performance of the ambiguity detection task; iii) our study reveals that prediction probabilities generated by the RF model are reliable indicators of the model's accuracy. This aspect is crucial as it minimizes the likelihood of providing incorrect responses, improving the reliability of the resulting QA systems.

2 Related Work

Ambiguous Question Answering and Clarification. Ambiguity is an element of human language, which has led to numerous studies including in instruction following (Shi et al., 2022a), conversational search (Keyvan and Huang, 2022; Aliannejadi et al., 2019), product search (Chen et al., 2023, 2024), and question answering (Shao and Huang, 2022; Sun et al., 2023; Lee et al., 2023; Ji et al., 2024; Zhang et al., 2024; Wu et al., 2024). Previous studies (Min et al., 2020; Shi et al., 2022b; Cole et al., 2023) emphasize the importance of grounding the ambiguity detection task within a relevant context, as the definition of "ambiguous" is inherently subjective. While the ClariQ dataset (Aliannejadi et al., 2021) is one of the pioneering datasets for query ambiguity, it does not offer a grounding context, leading to some inconsistent annotations (see Appendix §B). Similarly, AmbigQA (Min et al., 2020) and WebQuestionsSP (Yih et al., 2016) do not provide annotated context. In this research, we focus on a context-enhanced setting.

Uncertainty Estimation. Estimating uncertainty/confidence is crucial for assessing the reliability of LLMs (Gal and Ghahramani, 2016; Yang et al., 2024a; Geng et al., 2024; Zhou et al., 2023a). Ideally, a perfectly calibrated confidence estimation reflects the true likelihood of the prediction being correct (Niculescu-Mizil and Caruana, 2005; Guo et al., 2017). Earlier studies (Murray and Chiang, 2018; Malinin and Gales, 2020; Jiang et al., 2021) often used the token probability from the language model to calculate the marginal probability of a sequence and use it to estimate the model confidence. Recent works have raised the question of whether post-training (Ouyang et al., 2022; Wei et al., 2022a) might negatively impact model calibration (OpenAI, 2023). Many efforts have been made to calibrate uncertainty in LLMs. Kadavath et al. (2022) estimated the LLM confidence using the likelihood of the "True" token

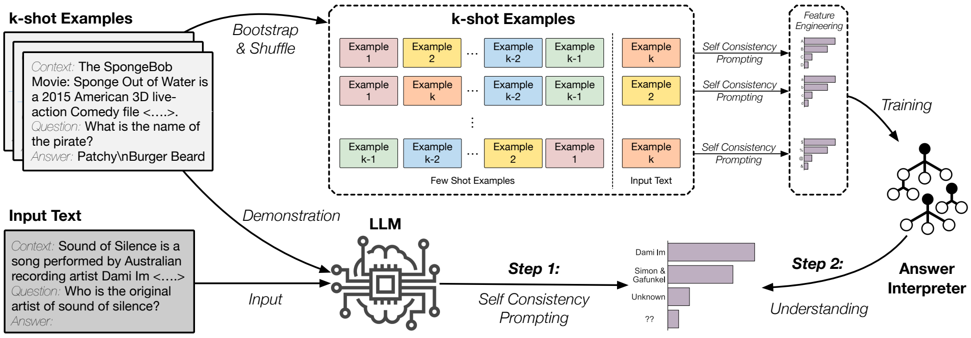

Figure 2: Overview of our framework. Given a question, we first retrieve a set of supporting pieces of evidence with a retrieval engine. Then, we perform two steps: (i) generate all feasible answers using self-consistency prompting; (ii) adopt an interpreter to infer the ambiguity in the question. The interpreter is trained with a *bootstrapping and shuffling* technique of 6 examples over distributional features from the generated answers.

when prompted to validate the correctness of its prior response. Other works prompted LLMs to generate their confidence (Mielke et al., 2022; Lin et al., 2022; Tian et al., 2023; Zhou et al., 2023b). Additionally, Si et al. (2023) considered the frequency of the answer as a proxy for confidence. Another line of work assumes that when the LLM generates a broad range of semantically varied answers, it indicates a high level of uncertainty (Lin et al., 2023; Nikitin et al., 2024; Shi et al., 2024). They measure the uncertainty via entropy over answers sampled from the model output distribution. After identifying semantically different answers a, the overall uncertainty can be represented as $H(q) = -\sum_a p(a|q) \log p(a|q)$. However, these approaches assume the existence of a single correct answer.

Answer Calibration. Understanding when to trust an LLM is essential for building safer AI systems (Amodei et al., 2016; Hendrycks et al., 2021; Zhao et al., 2020; Maynez et al., 2020; Portillo Wightman et al., 2023; Yang et al., 2024b). Selective question answering is a popular approach for addressing this problem (Chow, 1957; El-Yaniv et al., 2010; Kamath et al., 2020; Zhang et al., 2021). Specifically, the idea is to assign the confidence $s(q)$ for answering the question q. A threshold τ is used to decide whether to answer it, ask for clarification, or abstain from answering. An accurate uncertainty estimation may help reduce the risk of generating false or unfounded outputs.

3 Task Formulation

We focus on the scenario where we prompt an LLM to get an accurate answer to an unambiguous question or detect an ambiguous question under the few-shot setting. More specifically, given a user's question q, the QA system has access to some external information c relevant to the question. The contextual information is either provided with the question (e.g., a document-grounded conversation) or is retrieved from a collection of documents $\mathcal{D}$ (*i.e.,* retrieval-augmented QA). Given c and q, the goal of the QA system is to (1) find an accurate answer a to an unambiguous question; or (2) request clarification when the question is ambiguous (*i.e.,* has multiple plausible interpretations).[1]

In this paper, we focus on two tasks: *ambiguity detection* and *confidence calibration*. The goal of the ambiguity detection task is to identify whether a given question is ambiguous. As for confidence calibration, the goal is to measure the quality of the confidence estimation, which is crucial for avoiding inaccurate, incomplete, or misleading answers.

4 Our Approach

In this section, we describe our framework (see Figure 2), aiming to (i) identify ambiguous questions; and (ii) avoid providing incorrect, incomplete, or misleading answers. We first prompt the LLM to generate several answers (*answer-oriented prompting*; see §4.1) and then deduce the ambiguity by analyzing cues from the LLM outputs (see §4.2).

4.1 Self-consistency Prompting for Multiple Plausible Answers

Differently from previous work which prompted the LLM to generate a single answer using standard self-consistency prompting (Wang et al., 2022; Kuhn et al., 2023; Si et al., 2023), we prompt the

[1]In this paper we do not tackle the problem of generating a clarification question, which we leave for future research.

LLM to list all plausible answers, separated by a delimiter (*e.g.,* '\n'). Given a question q and a corresponding context c, this can be represented as:

$$P_{\text{LLM}}(\mathcal{A} \mid \text{prompt}, c, q), \quad (1)$$

where $a \in \mathcal{A}$ represents a single answer. This process repeats m times by sampling from the LLM's decoder with a temperature value t (the number of answers $|\mathcal{A}|$ can be varied across different samples). Subsequently, we group all generated answers from the m sampling outputs using exact text matching, which is sufficient as the answers are typically short phrases, and categorize them to generate a collection $\mathcal{A}_m = \{(a_1, f_1), \ldots, (a_n, f_n)\}$. Here, each element $(a_i, f_i) \in \mathcal{A}_m$ represents an individual answer and its corresponding occurrence frequency across the m LLM outputs.

4.2 LLM Outputs Analysis

The next phase is to analyze the LLM outputs. The objectives of this step are twofold: (i) recognize when the LLM is confident about a single answer (indicating *non-ambiguity*); (ii) determine when the LLM is confident about multiple answers (indicating *ambiguity*). Intuitively, when an LLM repeatedly generates the same answer, it implies a high confidence level and likelihood of correctness (Wang et al., 2022). Conversely, a variety of low-frequency occurring answers may indicate either low confidence or potential inaccuracies (Kuhn et al., 2023). Therefore, we hypothesize that by examining the frequency of the LLM answers, we can infer the ambiguity in the question. In this work, we utilize an RF model to analyze the LLM outputs. The RF input is a set of features derived from the set $\mathcal{A}_m$ of answers and their frequencies. The model is used to predict a score reflecting the probability that the question is ambiguous.

Random Forest Few-shot Training. A notable challenge in training the interpreter model is the limited amount of data available. To overcome this, we adopt a *bootstrapping and shuffling* strategy using few-shot examples to create an expanded training dataset with N examples. Specifically, as shown in Figure 2, given k examples in the demonstration data, we *bootstrap* by selecting up to $k-1$ examples from these to form a new demonstration, while using the remaining example as the input and its corresponding label (*i.e.,* ambiguous or non-ambiguous) as the ground truth. Next, we *shuffle* the examples to generate additional demonstrations. This allows us to form a diverse collection of demonstration-input pairs that are fed into the LLM to produce a set of answer-frequency $\mathcal{A}_m$ for training. Then, we construct specific features to capture the distributional patterns of the answers.

Feature Extraction. We hypothesize that frequently occurring answers are more likely to be accurate, regardless of semantic meaning. This leads us to assume that discarding less common answers, which might be incorrect, can help in better assessing the question's ambiguity. However, different models of varying sizes exhibit distinct patterns in generating erroneous answers, making it challenging to set a fixed frequency threshold. To address this, we compute the entropy over answers with different frequencies. Specifically, we calculate the entropy of answers occurring more than m times, denoted as e_m. We find that using a binary value as the feature enhances model performance. Thus we define binary features $f_{e_m,t} \triangleq \mathbf{1}_{e_m>t}(e_m)$, where t represents a threshold within the range $[0, 1]$. We then generate a feature set by choosing various values for m and t. These features and the corresponding labels are used to train the random forest model, which serves as an interpreter to analyze $\mathcal{A}_m$. The advantage of the RF model with our bootstrapping and shuffling strategy against more sophisticated models is to simulate and learn different potential answer distributions, rather than relying on the semantic content.

4.3 Calibration for Question Answering

Another focus of this study is estimating the model certainty when answering questions. Our approach uses two types of confidence estimation. First, we assess the model confidence in determining whether a question is ambiguous using the probability estimation from an interpreter model, which we denote as $c_{\text{amb}} \propto P(\text{ambiguity}|x;\Theta)$, where x is the input to the interpreter model and Θ represents its parameters. Secondly, to estimate the model confidence in a specific answer a, we use a conditional probability formula $c_{\text{a}} \propto f_a \cdot P(\neg\text{ambiguity}|x;\Theta)$, where f_a is the frequency ratio of the answer a to the total frequency of all answers. Our hypothesis is that a high probability assigned by the model to either a single or multiple correct answers could signify a greater chance of accuracy. Conversely, a probability that reflects indecision or difficulty in distinguishing between these scenarios might indicate potential inaccuracies.

Models	ASQA				PACIFIC				ABG-COQA			
	P ↑	R ↑	F_1 ↑	Acc ↑	P ↑	R ↑	F_1 ↑	Acc ↑	P ↑	R ↑	F_1 ↑	Acc ↑
**Supervised Learning and Random Baselines*												
RANDOM	$57.64_{2.3}$	$51.66_{1.3}$	$54.48_{1.7}$	$50.76_{2.3}$	$55.34_{1.2}$	$54.1_{3.7}$	$54.68_{2.4}$	$53.47_{1.5}$	$42.95_{1.9}$	$37.36_{1.7}$	$50.52_{2.4}$	$50.54_{1.7}$
ROBERTA-L (Full)	$62.08_{6.3}$	$94.54_{4.2}$	$71.81_{1.4}$	$73.12_{3.6}$	$67.16_{2.1}$	$86.76_{1.5}$	$75.69_{1.8}$	$73.33_{3.6}$	$67.54_{2.1}$	$81.93_{6.8}$	$73.81_{1.4}$	$75.12_{1.2}$
ROBERTA-L (6-shot)	$50.86_{1.5}$	$61.42_{6.3}$	$55.57_{3.4}$	$58.01_{1.6}$	$63.81_{2.2}$	$33.74_{5.6}$	$43.75_{4.5}$	$45.87_{0.8}$	$46.70_{0.8}$	$71.11_{3.4}$	$56.36_{1.6}$	$47.20_{1.1}$
**Binary Prompting (Standard Few-shot Prompting for Ambiguity)*												
FLAN-T5-XL	$62.59_{1.8}$	$62.50_{7.6}$	$62.19_{3.0}$	$57.09_{0.6}$	$25.85_{1.3}$	$39.19_{3.6}$	$31.14_{2.1}$	$36.28_{0.4}$	$32.90_{0.1}$	$29.23_{0.0}$	$30.96_{0.1}$	$32.20_{0.2}$
FLAN-T5-XXL	$59.99_{0.1}$	$81.31_{0.9}$	$69.04_{0.4}$	$58.43_{0.3}$	$14.11_{3.8}$	$11.46_{5.6}$	$12.46_{5.0}$	$43.57_{3.3}$	$38.21_{0.8}$	$30.00_{1.5}$	$33.60_{1.3}$	$38.40_{0.4}$
LLAMA-2-7B	$56.67_{1.4}$	$90.81_{8.4}$	$69.70_{3.4}$	$55.32_{3.1}$	$36.22_{1.0}$	$94.79_{8.7}$	$52.37_{2.5}$	$36.76_{0.1}$	$51.44_{32.0}$	$6.15_{6.5}$	$10.50_{10.7}$	$50.10_{2.2}$
LLAMA-2-13B	$58.32_{0.4}$	$85.36_{1.9}$	$69.28_{0.3}$	$56.85_{0.3}$	$30.83_{1.0}$	$27.86_{8.6}$	$28.70_{5.3}$	$50.67_{2.9}$	$56.18_{7.4}$	$29.42_{8.8}$	$37.07_{7.4}$	$50.10_{1.4}$
LLAMA-2-70B	$52.59_{2.6}$	$54.62_{5.3}$	$53.85_{4.9}$	$50.80_{2.7}$	$38.64_{1.0}$	$44.27_{6.0}$	$41.26_{2.9}$	$53.55_{1.2}$	$48.56_{3.8}$	$32.69_{12.8}$	$37.86_{8.3}$	$47.00_{2.1}$
**Answer-Oriented Prompting with Random Forest (Ours)*												
FLAN-T5-XL	$57.00_{0.2}$	$94.58_{2.4}$	$71.13_{0.9}$	$56.31_{0.7}$	$45.70_{1.6}$	$77.87_{3.7}$	$57.93_{3.4}$	$57.42_{0.5}$	$60.96_{3.2}$	$62.16_{3.6}$	$61.44_{2.3}$	$59.45_{2.8}$
FLAN-T5-XXL	$61.14_{0.7}$	$77.37_{6.6}$	$68.29_{2.8}$	$59.48_{1.1}$	$47.81_{1.9}$	$72.90_{7.6}$	$57.71_{6.4}$	$59.94_{3.1}$	$67.73_{4.5}$	$60.31_{14.1}$	$62.47_{7.2}$	$63.71_{2.7}$
LLAMA-2-7B	$59.84_{1.7}$	$91.64_{8.8}$	$73.18_{3.1}$	$61.39_{2.2}$	$39.13_{3.1}$	$75.82_{6.5}$	$51.46_{2.1}$	$47.19_{0.7}$	$60.75_{1.3}$	$57.18_{6.1}$	$58.80_{3.8}$	$58.60_{2.2}$
LLAMA-2-13B	$61.42_{6.8}$	$88.67_{8.9}$	$72.30_{4.2}$	$61.25_{4.7}$	$42.19_{10.8}$	$71.00_{4.1}$	$53.89_{3.0}$	$54.78_{1.2}$	$60.29_{1.9}$	$58.33_{5.4}$	$59.21_{3.5}$	$58.40_{2.4}$
LLAMA-2-70B	$64.04_{3.4}$	$88.67_{4.1}$	$73.98_{2.2}$	$64.57_{2.2}$	$58.16_{2.7}$	$72.40_{8.2}$	$58.16_{4.9}$	$61.61_{4.0}$	$73.95_{3.2}$	$67.69_{3.6}$	$70.68_{2.8}$	$70.80_{2.3}$

Table 1: Ambiguity detection task results on the development set. We report the average performance with standard deviation across 3 random seeds. The best prompting performance for each column is highlighted in blue.

Methods	ASQA	PACIFIC	ABG-COQA
**Binary Prompting*			
Standard Prompting	$48.76_{4.7}$	$53.55_{1.2}$	$47.00_{2.1}$
CoT Prompting	$56.91_{3.4}$	$44.41_{3.0}$	$48.93_{1.5}$
Self-Consistency	$53.66_{6.0}$	$52.71_{2.1}$	$45.90_{3.6}$
**Answer-oriented Prompting*			
LLM-*itself*	$44.08_{1.1}$	$45.66_{2.1}$	$46.84_{2.2}$
ROBERTA-L	$54.27_{5.6}$	$61.55_{1.9}$	$55.56_{2.1}$
Frequency Heuristic	$57.42_{1.3}$	$54.75_{2.3}$	$59.70_{3.4}$
Heuristic Method	$61.57_{2.3}$	$59.86_{4.2}$	$62.00_{2.3}$
Sampling Repetition	$51.61_{1.7}$	$54.55_{5.0}$	$51.64_{1.5}$
Sampling Diversity	$50.85_{4.6}$	$50.84_{2.1}$	$47.03_{2.6}$
Random Forest (ours)	$64.57_{2.2}$	$61.61_{4.0}$	$70.80_{2.3}$

Table 2: Ambiguity detection accuracy on the dev set (3 seeds average) with different prompting using LLAMA-2-70B. The best performance is marked in blue.

5 Experimental Setup

5.1 Datasets

We experimented with three datasets, including ASQA (Stelmakh et al., 2022) , PACIFIC (Deng et al., 2022), and ABG-COQA (Guo et al., 2021). ASQA was created based on AmbigQA (Min et al., 2020) by adding a context to each question and long-form answers. PACIFIC is a QA dataset in the financial domain, constructed based on the TAT-QA dataset (Zhu et al., 2021) where the context is in the form of tables and text. ABG-COQA, which was built on top of the CoQA dataset (Reddy et al., 2019), consists of narratives and corresponding ambiguous questions. Following prior studies (Deng et al., 2023; Cole et al., 2023; Tian et al., 2023), we use the development sets for evaluation. See more details and examples in Appendix §B.

5.2 Implementation Details

We experimented with a range of LLMs with different sizes, including encoder-decoder, i.e., Flan-T5 (3B, 11B) (Chung et al., 2022) and decoder-only, i.e., LLAMA-2 (7B, 13B, 70B) (Touvron et al., 2023); for LLAMA-2, we used the CHAT variant. We set the number of few-shot examples to 6 in all models and prompting strategies due to the limited length of the model input. We used the oracle context as the input, except for our experiments with noisy contexts over the ASQA dataset. For those experiments, we utilized evidence retrieved by a Dense Passage Retrieval (DPR) model (Karpukhin et al., 2020); the retrieval corpus is the English Wikipedia dump of 12/20/2018 and the documents are split into chunks of 100 words (Karpukhin et al., 2020). Examples of the different prompts and further implementation details can be found in Appendix §C and §D, respectively.

5.3 Baselines

Ambiguity Detection. The first set of baselines is based on *Binary Prompting* (Deng et al., 2023) where the idea is to prompt the LLM to perform binary classification to determine question ambiguity. We evaluated different prompting strategies for binary prompting, including **Standard** prompting (Brown et al., 2020), **Chain-of-Thought (CoT)** prompting (Wei et al., 2022b), and **Self-Consistency** prompting (Wang et al., 2022). The second set of baselines is based on *Answer-oriented Prompting*, where we prompt the LLM to generate multiple answers for a question and then detect ambiguity based on the analysis of these answers. In our approach, we use a Random Forest model[2] to analyze the answers. To test the effectiveness of other models, we experimented with the following baselines. (i) **Heuristic Method**: a

[2]Details on the Random Forest training are provided in Appendix C.

question is predicted as ambiguous if the entropy of the generated answers exceeds a certain threshold. (ii) **Frequency Heuristic**: a question is predicted as ambiguous if there are multiple high-frequency answers. We experiment with various thresholds to define 'high frequency'. (iii) **LLM-*itself***: prompting the model for question ambiguity binary classification based on the concatenation of all generated answers, the original context, the question, and some few-shot demonstrations. (iv) **RoBERTa-L**: we train a RoBERTa-L model with the bootstrapping dataset generated in §4 and use it for prediction based on the same inputs as in LLM-*itself*. (v) **Sampling Repetition** and **Sampling Diversity** measure the frequency of the most confident answer and count the number of unique answers among samples from the LM respectively. Following Cole et al. (2023), we report the best performance among different values of *Num Disagreements* and *Num Answers*.

Confidence Calibration. We use the following approaches as baselines: **Self-consistency Confidence** (Si et al., 2023) uses the frequency of the most frequent answer from self-consistency prompting as the confidence score. **Sampling Diversity** estimates the confidence in inverse proportion to the number of distinct samples. Specifically, the score is zero if every sample differs from the others. We also use the **Verbalized Confidence** approach (Mielke et al., 2022; Tian et al., 2023) which concatenates the most frequent answer to the original context and question, and prompts the LLM to express its confidence in the range of 0 to 100. **`P(True)`** (Kadavath et al., 2022) concatenates the most frequent answer to the original context and question, and prompts the LLM to determine whether the answer is true. Then, the confidence score is computed based on the logit probability associated with the "`True`" token. The methods described above focus on assessing the confidence of a single answer. Therefore, for a more comprehensive evaluation, we also consider approaches that estimate the model confidence based on multiple answers. For **LLM-*itself***, we prompt the LLM with all generated answers, the original context, and the question. Then, unlike the ambiguity detection task, the LLM is prompted to express its confidence towards multi-correct answers in the range of 0 to 100. For **RoBERTa-L**, the approach is similar, but it uses the logits from the RoBERTa model to quantify confidence. Finally, the **Heuristic** method uses entropy as a measure of confidence.

Methods	P ↑	R ↑	F_1 ↑	Acc ↑
Llama-2-7B	$59.84_{1.7}$	$91.64_{8.8}$	$73.18_{3.1}$	$61.39_{2.2}$
`w/ Top-3`	$57.80_{0.3}$	$94.93_{0.1}$	$71.85_{0.3}$	$57.24_{0.6}$
Llama-2-13B	$61.42_{6.8}$	$88.67_{8.9}$	$72.30_{4.2}$	$61.25_{4.7}$
`w/ Top-3`	$57.99_{0.3}$	$96.58_{2.1}$	$72.45_{0.3}$	$57.80_{0.2}$
Llama-2-70B	$64.04_{3.4}$	$88.67_{4.1}$	$73.98_{2.2}$	$64.57_{2.2}$
`w/ Top-3`	$58.62_{1.2}$	$94.66_{1.4}$	$72.39_{0.7}$	$58.60_{1.1}$

Table 3: Results on the ambiguity detection task using retrieved passages on the ASQA dataset. `w/ Top-3` represents using the top-3 retrieved documents rather than the oracle context. We report the accuracy of the development set across three random seeds. The best performance for each column is highlighted in blue.

5.4 Evaluation Metrics

For the ambiguity detection task, we use Precision, Recall, F_1, and Accuracy for evaluation. For the confidence calibration task, we report the Accuracy of whether the model provides the correct answer to unambiguous user questions or accurately identifies the question ambiguity. For the confidence calibration task, we report the Expected Calibration Error (ECE) to measure the discrepancy between the predicted accuracy (*i.e.,* confidence) and its actual performance. Specifically, the predictions are divided into M uniform bins B_m w.r.t. confidence scores. Then, we compute the average absolute difference between the confidence (`cnf`) and the actual accuracy (`acc`) for each bin over n samples:

$$ECE = \sum_{m=1}^{M} \frac{|B_m|}{n} |\mathtt{acc}(B_m) - \mathtt{cnf}(B_m)| \qquad (2)$$

Due to the limitations of ECE stemming from its bucketing approach (Si et al., 2023), we also report the Brier score (Brier, 1950). We also evaluate how the system performs when it selectively responds based on its confidence. Acc@50 indicates the accuracy of questions if the QA system only answers questions with the top 50% confidence scores.

6 Experimental Results

6.1 Ambiguity Detection

Tables 1 and 2 present experimental results on the ambiguity detection task using various LLMs, and different prompting strategies with the ground truth context. Table 3 shows the more realistic scenario results when using the context retrieved with a DPR model. In particular, we use top-3 retrieved documents instead of the ground-truth documents.

Method	ASQA				PACIFIC				ABG-COQA			
	Acc ↑	Acc@50 ↑	ECE ↓	Brier ↓	Acc ↑	Acc@50 ↑	ECE ↓	Brier ↓	Acc ↑	Acc@50 ↑	ECE ↓	Brier ↓
**Single Answer Assumption*												
Verbalization	25.51	29.48	43.10	38.40	35.89	35.14	23.26	28.92	31.60	30.51	34.78	36.24
P(True)	25.51	44.64	28.58	28.61	35.89	40.15	14.00	24.26	31.60	50.85	25.48	28.29
Self-Consistency	25.51	62.37	28.09	24.02	35.89	49.62	9.15	21.18	31.60	52.00	25.23	26.33
Sampling Diversity	25.51	62.84	26.40	23.54	35.89	48.45	9.54	21.94	31.60	54.65	22.47	25.91
**Ambiguous Question Answering*												
LLM-*itself*	40.12	43.01	21.81	25.81	31.73	32.91	12.73	25.83	41.93	43.01	25.82	28.93
ROBERTA-L	46.84	49.02	20.30	24.72	35.31	49.02	9.20	20.83	42.41	51.03	22.31	25.05
Heuristic Method	52.35	53.61	26.07	33.55	33.21	47.69	10.37	21.50	44.80	55.20	25.25	27.44
Random Forest (ours)	61.26	65.82	10.15	23.90	37.39	53.08	8.67	19.49	49.60	59.20	16.83	24.84

Table 4: Calibration results on three datasets using LLAMA-2-70B on the development set. ↑ and ↓ indicate whether higher or lower metrics are preferable, respectively. The best performance for each column is highlighted in blue.

#1. Limited Effectiveness of Binary Prompting in Ambiguity Detection. As shown in Table 1, we find that the performance of binary prompting is inconsistent across different datasets. For example, the ASQA dataset obtains a performance slightly above random guessing, while the results on the PACIFIC and ABG-COQA datasets are underwhelming. Moreover, the increased model size does not necessarily improve the performance of this strategy. For example, the LLAMA-2-70B does not perform better than LLAMA-2-7B on the ASQA dataset. These findings indicate that binary prompting might struggle to detect ambiguity consistently. In Table 2 (Top), we further evaluate the performance of different binary prompting strategies (*i.e.,* CoT and Self-consistency). We find that these strategies did not yield any performance improvement. Our findings align with the prior study (Deng et al., 2023), underscoring the difficulty of this strategy to decide if a question is ambiguous. Similarly, Cole et al. (2023) suggests that none of the prompting strategies seems particularly useful, with none surpassing the baseline precision of 53%.

#2. Improved Performance in Ambiguity Detection with Answer-Oriented Prompting and Random Forest. Table 1 presents the performance of the ambiguity detection task using our approach, which achieves the best performance across datasets and model sizes. Notably, we observe a clear trend where the effectiveness in detecting ambiguity improves with the model size. This highlights that our approach can identify cases where the LLM confidently suggests multiple answers (indicating ambiguity) versus when it leans towards a single answer (*indicating non-ambiguity*).

Table 2 (Bottom) shows the results where we explore alternative models to the Random Forest using answer-oriented prompting. We find that Random Forest emerges as the most effective technique. Moreover, we observe that LLMs lack the ability to self-interpret their outputs. This observation aligns with findings from prior studies (Valmeekam et al., 2023; Stechly et al., 2023), indicating that self-interpretation of responses remains a challenging task for the LLMs. Apart from our approach, the heuristic method based on entropy delivers the most optimal results. Please, find a detailed error analysis of these two approaches in §6.3.

#3. Noisy Contexts Experiments. Table 3 evaluates a more realistic setting, where the context is retrieved with ASQA. This experiment shows what would be the performance when the retrieved passages are noisy. The performance slightly declines when using only the retrieved context (`w/ Top-3`) across all model sizes. Still, it is within 1-2 points in the F_1 score compared to the ground truth context setting, i.e., our approach is effective in coping with noisy contexts.

#4. Low-resource Setting. Table 1 compares our approach with supervised models in low-resource settings. In fact, our model outperforms supervised models trained on the same set of 6 examples (ROBERTA-L 6-shot): these models require much more training examples to be competitive.

6.2 Confidence Calibration

#1. Our approach responds to unambiguous questions or detects ambiguity. As shown in Table 4, our approach consistently outperforms all baselines, including models like the LLM or ROBERTA. It reaches 61.26% accuracy, outperforming the closest competitor (*i.e.,* heuristic) by roughly 9% on ASQA. Similar outcomes can be observed on other datasets. Interestingly, the accuracy for *Ambiguous Question Answering* does not always outperform those with *Single Answer*

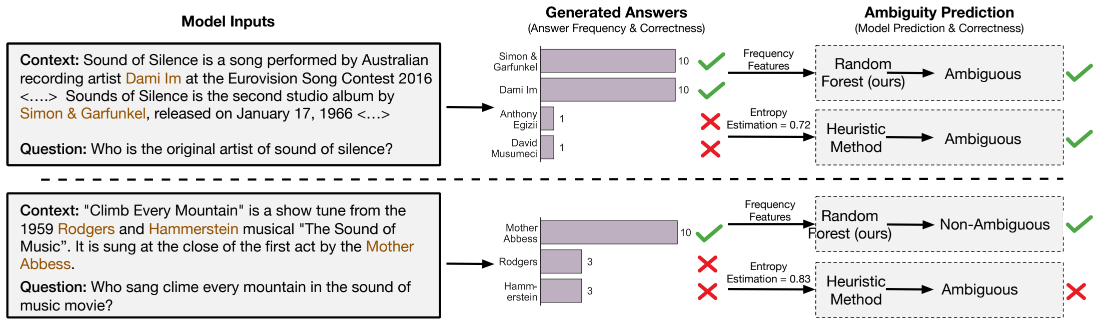

Figure 3: Our model against the entropy-based heuristic: the latter tends to have a higher entropy when the LLM produces incorrect answers. This leads to an overestimated denotational uncertainty, *i.e.,* higher false positives rate.

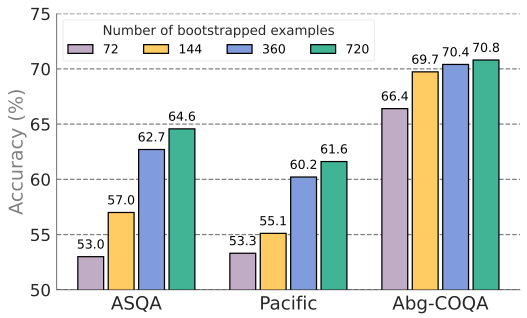

Figure 4: Impact of bootstrap size using LLAMA-2-70B. The performance increases with the bootstrap size.

Assumption on PACIFIC[3].

#2. Our approach demonstrates a superior ability to avoid incorrect, incomplete, or misleading answers. Our experiments indicate that using the Random Forest's probability, our approach generates more accurately calibrated confidence estimates. In various metrics like ECE, ACC@50, and Brier score, our method consistently outperforms other baseline methods across datasets. Our approach has thus an enhanced grasp of the trustworthiness of its answers, thereby minimizing the chances of providing incorrect information.

6.3 Further Analysis

Bootstrapping size. The main goal of the *bootstrapping and shuffling* strategy is to generate a diverse distribution of answers. Figure 4 shows the impact of the bootstrapping size on the performance. The accuracy improves with the size of the bootstrapping set: this result is impressive, given that only 6 annotated examples are initially used.

Error Analysis. Figure 3 provides case studies to compare the entropy-based heuristic and our approach on ASQA. When the LLM gives some incorrect answers, (*e.g.,* "rodgers" and "hammerstain"), the heuristic method tends to have higher entropy. In this case, the heuristic method misinterprets the source of this uncertainty to the question ambiguity, rather than its knowledge gaps or inaccuracies. This misinterpretation, often a result of the LLM's errors or 'hallucinations', leads to increased entropy values and, consequently, a higher rate of false positives. In our analysis, the heuristic method exhibits a 32.1% false positive rate and a 7.0% false negative rate. In contrast, our approach achieves a reduced false positive rate of 25.4% while obtaining a slight increase in false negatives at 10.1%.

7 Conclusion

In this work, we introduce a novel framework that enables LLMs to recognize ambiguous questions. Our approach prompts the LLM to generate multiple answers that are then analyzed through an interpreter model (i.e., Random Forest) to detect ambiguity. The Random Forest is trained with only 6 examples that are bootstrapped and shuffled to create multiple answer distributions. Our experiments on three datasets demonstrate the effectiveness of our approach in low-resource settings in identifying ambiguous questions. Furthermore, our approach has been shown to effectively refine the confidence calibration of LLM outputs: this improves the LLMs' ability to accurately interpret and respond to complex queries, contributing to more reliable and trustworthy QA systems.

[3] In PACIFIC the context documents are mainly tables with numbers; in this scenario, LLMs generally struggle, regardless of their size.

Limitations

Our research is a step forward in identifying ambiguous questions in LLM-based QA systems. However, we must recognize certain limitations, particularly regarding the dependency on model scale. The effectiveness of our method for detecting ambiguity is closely tied to the size of the LLM used. Essentially, our approach requires a robust LLM capable of accurately answering questions first, before assessing the ambiguity of these questions. If the model is smaller or prone to errors, our method may face challenges in accurately identifying ambiguities. This reliance on large-scale models brings advantages in terms of performance but also introduces scalability and resource challenges, especially in environments with limited resources. Moreover, our approach requires the LLM model to generate (possibly) all the answers to a question. This may be inefficient from a latency perspective, especially when using very large models. Finally, the current work doesn't specifically address the problem of disambiguation, which is crucial in improving trust in the NLP systems.

References

Mohammad Aliannejadi, Julia Kiseleva, Aleksandr Chuklin, Jeff Dalton, and Mikhail Burtsev. 2021. Building and evaluating open-domain dialogue corpora with clarifying questions. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4473–4484, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W. Bruce Croft. 2019. Asking clarifying questions in open-domain information-seeking conversations. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, page 475–484, New York, NY, USA. Association for Computing Machinery.

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. 2016. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023*.

Glenn W Brier. 1950. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 785–794, New York, NY, USA. Association for Computing Machinery.

Zhiyu Chen, Jason Choi, Besnik Fetahu, Oleg Rokhlenko, and Shervin Malmasi. 2023. Generate-then-retrieve: Intent-aware FAQ retrieval in product search. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, pages 763–771, Toronto, Canada. Association for Computational Linguistics.

Zhiyu Chen, Jason Ingyu Choi, Besnik Fetahu, and Shervin Malmasi. 2024. Identifying high consideration E-commerce search queries. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 563–572, Miami, Florida, US. Association for Computational Linguistics.

Chi-Keung Chow. 1957. An optimum character recognition system using decision functions. *IRE Transactions on Electronic Computers*, EC-6(4):247–254.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Jeremy Cole, Michael Zhang, Daniel Gillick, Julian Eisenschlos, Bhuwan Dhingra, and Jacob Eisenstein. 2023. Selectively answering ambiguous questions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 530–543, Singapore. Association for Computational Linguistics.

Yang Deng, Wenqiang Lei, Lizi Liao, and Tat-Seng Chua. 2023. Prompting and evaluating large language models for proactive dialogues: Clarification, target-guided, and non-collaboration. *arXiv preprint arXiv:2305.13626*.

Yang Deng, Wenqiang Lei, Wenxuan Zhang, Wai Lam, and Tat-Seng Chua. 2022. PACIFIC: Towards proactive conversational question answering over tabular and textual data in finance. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6970–6984, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ran El-Yaniv et al. 2010. On the foundations of noise-free selective classification. *Journal of Machine Learning Research*, 11(5).

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

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

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR.

Meiqi Guo, Mingda Zhang, Siva Reddy, and Malihe Alikhani. 2021. Abg-coQA: Clarifying ambiguity in conversational question answering. In *3rd Conference on Automated Knowledge Base Construction*.

Dan Hendrycks, Nicholas Carlini, John Schulman, and Jacob Steinhardt. 2021. Unsolved problems in ml safety. *arXiv preprint arXiv:2109.13916*.

Jiaming Ji, Boyuan Chen, Hantao Lou, Donghai Hong, Borong Zhang, Xuehai Pan, Tianyi Qiu, Juntao Dai, and Yaodong Yang. 2024. Aligner: Efficient alignment by learning to correct. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021. How Can We Know When Language Models Know? On the Calibration of Language Models for Question Answering. *Transactions of the Association for Computational Linguistics*, 9:962–977.

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

Amita Kamath, Robin Jia, and Percy Liang. 2020. Selective question answering under domain shift. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5684–5696, Online. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Kimiya Keyvan and Jimmy Xiangji Huang. 2022. How to approach ambiguous queries in conversational search: A survey of techniques, approaches, tools, and challenges. *ACM Comput. Surv.*, 55(6).

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*.

Dongryeol Lee, Segwang Kim, Minwoo Lee, Hwanhee Lee, Joonsuk Park, Sang-Woo Lee, and Kyomin Jung. 2023. Asking clarification questions to handle ambiguity in open-domain QA. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11526–11544, Singapore. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

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

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

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

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

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

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. AmbigQA: Answering ambiguous open-domain questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5783–5797, Online. Association for Computational Linguistics.

Kenton Murray and David Chiang. 2018. Correcting length bias in neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 212–223, Brussels, Belgium. Association for Computational Linguistics.

Alexandru Niculescu-Mizil and Rich Caruana. 2005. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML '05, page 625–632, New York, NY, USA. Association for Computing Machinery.

Alexander Nikitin, Jannik Kossen, Yarin Gal, and Pekka Marttinen. 2024. Kernel language entropy: Fine-grained uncertainty quantification for llms from semantic similarities. *arXiv preprint arXiv:2405.20003*.

OpenAI. 2023. Gpt-4 technical report. *arXiv*.

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

Anaelia Ovalle, Kai-Wei Chang, Ninareh Mehrabi, Yada Pruksachatkun, Aram Galystan, Jwala Dhamala, Apurv Verma, Trista Cao, Anoop Kumar, and Rahul Gupta, editors. 2023. *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)*. Association for Computational Linguistics, Toronto, Canada.

Gwenyth Portillo Wightman, Alexandra Delucia, and Mark Dredze. 2023. Strength in numbers: Estimating confidence of large language models by prompt agreement. In *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)*, pages 326–362, Toronto, Canada. Association for Computational Linguistics.

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.

Zhihong Shao and Minlie Huang. 2022. Answering open-domain multi-answer questions via a recall-then-verify framework. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1825–1838, Dublin, Ireland. Association for Computational Linguistics.

Zhengxiang Shi, Yue Feng, and Aldo Lipani. 2022a. Learning to execute actions or ask clarification questions. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2060–2070, Seattle, United States. Association for Computational Linguistics.

Zhengxiang Shi, Jerome Ramos, To Eun Kim, Xi Wang, Hossein A Rahmani, and Aldo Lipani. 2022b. When and what to ask through world states and text instructions: Iglu nlp challenge solution. *Neural Information Processing Systems (NeurIPS) IGLU Workshop*.

Zhengyan Shi, Sander Land, Acyr Locatelli, Matthieu Geist, and Max Bartolo. 2024. Understanding likelihood over-optimisation in direct alignment algorithms. *arXiv preprint arXiv:2410.11677*.

Chenglei Si, Zhe Gan, Zhengyuan Yang, Shuohang Wang, Jianfeng Wang, Jordan Boyd-Graber, and Lijuan Wang. 2023. Prompting gpt-3 to be reliable. In *International Conference on Learning Representations (ICLR)*.

Kaya Stechly, Matthew Marquez, and Subbarao Kambhampati. 2023. Gpt-4 doesn't know it's wrong: An analysis of iterative prompting for reasoning problems. *arXiv*.

Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. 2022. ASQA: Factoid questions meet long-form answers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8273–8288, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Weiwei Sun, Hengyi Cai, Hongshen Chen, Pengjie Ren, Zhumin Chen, Maarten de Rijke, and Zhaochun Ren. 2023. Answering ambiguous questions via iterative prompting. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7669–7683, Toronto, Canada. Association for Computational Linguistics.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D Manning. 2023. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. *arXiv preprint arXiv:2305.14975*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Karthik Valmeekam, Matthew Marquez, and Subbarao Kambhampati. 2023. Can large language models really improve by self-critiquing their own plans? *arXiv preprint arXiv:2310.08118*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022a. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022b. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Bin Wu, Zhengyan Shi, Hossein A Rahmani, Varsha Ramineni, and Emine Yilmaz. 2024. Understanding the role of user profile in the personalization of large language models. *arXiv preprint arXiv:2406.17803*.

Adam X Yang, Maxime Robeyns, Thomas Coste, Jun Wang, Haitham Bou-Ammar, and Laurence Aitchison. 2024a. Bayesian reward models for llm alignment. *arXiv preprint arXiv:2402.13210*.

Adam X Yang, Maxime Robeyns, Xi Wang, and Laurence Aitchison. 2024b. Bayesian low-rank adaptation for large language models. In *The Twelfth International Conference on Learning Representations*.

Wen-tau Yih, Matthew Richardson, Chris Meek, Ming-Wei Chang, and Jina Suh. 2016. The value of semantic parse labeling for knowledge base question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 201–206, Berlin, Germany. Association for Computational Linguistics.

Mingtian Zhang, Shawn Lan, Peter Hayes, and David Barber. 2024. Mafin: Enhancing black-box embeddings with model augmented fine-tuning. *arXiv preprint arXiv:2402.12177*.

Shujian Zhang, Chengyue Gong, and Eunsol Choi. 2021. Knowing more about questions can help: Improving calibration in question answering. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1958–1970, Online. Association for Computational Linguistics.

Zheng Zhao, Shay B. Cohen, and Bonnie Webber. 2020. Reducing quantity hallucinations in abstractive summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2237–2249, Online. Association for Computational Linguistics.

Han Zhou, Xingchen Wan, Lev Proleev, Diana Mincu, Jilin Chen, Katherine Heller, and Subhrajit Roy. 2023a. Batch calibration: Rethinking calibration for in-context learning and prompt engineering. *arXiv preprint arXiv:2309.17249*.

Kaitlyn Zhou, Dan Jurafsky, and Tatsunori Hashimoto. 2023b. Navigating the grey area: Expressions of overconfidence and uncertainty in language models. *arXiv preprint arXiv:2302.13439*.

Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. 2021. TAT-QA: A question answering benchmark on a hybrid of tabular and textual content in finance. 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 3277–3287, Online. Association for Computational Linguistics.

A Retrieval Performance

The table 5 provides a comparison of retrieval performance metrics using Dense Passage Retrieval (DPR), focusing on its effectiveness in passage/document retrieval tasks. The performance is measured using the MRECALL metric at two different recall levels: 3 and 5. These recall levels indicate the number of retrieved items (passages) considered for evaluating the method's accuracy.

Method	MRECALL@3	MRECALL@5
DPR	43.46/33.70	48.66/38.08

Table 5: Performance on passage retrieval in MRECALL. The two numbers in each cell indicate performance on all questions and on questions with more than one answer, respectively.

B Datasets

Datasets used in this work. In this section, we provide details for each dataset, along with representative examples in Table 7. Following the previous work (Si et al., 2023; Tian et al., 2023), we downsample the evaluation set to assess model performance more effectively. Specifically, we sampled 638 examples from the ASQA dataset, 521 from the PACIFIC dataset, and 250 from the ABG-COQA dataset, all taken from their respective evaluation sets.

Discussion about the ClariQ dataset. Here we also discuss some potential inconsistent annotations in the ClariQ dataset. The ambiguity annotations within ClariQ can differ significantly based on the perspective of the annotators, resulting in multiple interpretations. For instance, while the query "*Find condos in Florida*" is ambiguous, "*Tell me about hotels in New York.*" is considered unambiguous. Here we provide 10 pairs of questions (20 questions in total) with inconsistent ambiguity annotations. It is noteworthy that ClariQ only consists of roughly 200 questions across both its training and development sets. Such inconsistent annotations highlight the importance of grounding the ambiguity of a question within the context. Datasets such as ASQA, PACIFIC, and ABG-COQA address this issue by grounding questions within their context.

User Query	Ambiguity
Find condos in Florida.	Yes
Tell me about hotels in New York.	No
I want to learn about rock art.	Yes
I'd like to learn about lymphoma in dogs	No
How to change the toilet in the house	Yes
how to build a fence?	No
Tell me more about USA tax for annuity	Yes
Find me information about the sales tax in Illinois.	No
How to cook pork tenderlion	Yes
How to get organised?	No
I'm looking for information on worm	Yes
I'm looking for information about South Africa	No
Tell me about vines for shade.	Yes
Tell me more on health clubs in Arkansas	No
Tell me about source of the nile	Yes
Tell me about american military university.	No
Tell me about Barbados.	Yes
Tell me more about dnr	No
Where should I order dog clean-up bags	Yes
Where can I buy pressure washers?	No

Table 6: Analysis of ClariQ dataset. We provide 10 pairs of questions with potentially inconsistent annotations.

C Implementation Details

We randomly select 6 examples from the training set for few-shot examples in demonstrations, because (1) even if the datasets we used in our experiments contain a large number of examples, our solution targets low-resource scenarios where just a bunch of annotated data are available; and (2) we wanted to be sure the examples can easily fit into the prompt of LLMs. Thus, we sample a very low number of examples (i.e., 6 examples) and demonstrate that these are sufficient to make our method work.

We follow (Kuhn et al., 2023; Cole et al., 2023; Si et al., 2023; Tian et al., 2023) to decode $m = 10$ times. For each, we generate `10` sampled outputs (temperature=`0.3`,`0.5`,`0.7`) and use exact match (after lowercasing and removing punctuation) for comparison among outputs. We do sub-string and exact matching to group the equivalent answers. While previous works use the NLI model, it does not work. We use XGboost (Chen and Guestrin, 2016) to train the Random Forest model. We performed a grid search for the hyper-parameters of the model by searching the best configuration on a development set with respect to the max depth among 1, 2, 3, 4, 5 and the number of estimators among 20, 30, 50, 100. For the feature engineering, in our experiments, we set m to 0,1,2 and t to 0.5,0.7,0.9.

To determine the confidence levels for both single and multiple answers using the **LLM-*itself***, **ROBERTA-L**, and **Heuristic** baselines, we first calculate the confidence for multiple answers, denoted as p_m. Once p_m is established, we then derive the confidence for a single answer using $p_s = 1 - p_m$. This approach assumes that the confidence in a single answer inversely correlates with the confidence in multiple answers. For the baseline **ROBERTA-L**, we concatenate the questions with the context and train them with a few labelled examples or all examples in the train sets.

D Examples of Prompting

Table 8 provides examples of prompts used in our work, including binary prompting, binary prompting with CoT, answer-oriented prompting, verbalized confidence, and self-evaluation of LLMs towards correctness. For self-consistency prompting, we repeat the above-mentioned prompt multiple times.

<table>
<tr><th>Dataset</th><th>Example</th></tr>
<tr><td>ASQA</td><td><code>id</code>: 7089015503030534342
<code>question</code>: Who is the original artist of sound of silence?
<code>answers</code>: Simon & Garfunkel, Dami Im
<code>contexts</code>: "Sound of Silence" is a song performed by Australian recording artist Dami Im. Written by Anthony Egizii and David Musumeci of DNA Songs, it is best known as Australia's entry at the Eurovision Song Contest 2016 which was held in Stockholm, Sweden, where it finished 2nd, receiving a total of 511 points. The song also won the Marcel Bezençon Award in the composer category. The song was leaked on 10 March 2016, one day before its initial release date. It is Dami Im's fourth Australian top 20 hit and worldwide, it reached the top 40 in more than six countries after the Eurovision Song Contest 2016 Final.
<code>Ambiguity</code>: Yes</td></tr>
<tr><td>PACIFIC</td><td><code>id</code>: e4fe0666-9c0e-43c0-9f67-538dae3092b9
<code>question</code>: What is the amount of total sales?
<code>clarification question</code>: Which year are you asking about?
<code>answer to clarification question</code>: 2019
<code>contexts</code>: "Sales by Contract Type: Substantially all of our contracts are fixed-price type contracts. Sales included in Other contract types represent cost plus and time and material type contracts. On a fixed-price type contract, we agree to perform the contractual statement of work for a predetermined sales price. On a cost-plus type contract, we are paid our allowable incurred costs plus a profit which can be fixed or variable depending on the contract's fee arrangement up to predetermined funding levels determined by the customer. On a time-and-material type contract, we are paid on the basis of direct labor hours expended at specified fixed-price hourly rates (that include wages, overhead, allowable general and administrative expenses and profit) and materials at cost. The table below presents total net sales disaggregated by contract type (in millions):
<code>Table</code>:
| Years Ended September 30 |
	2019	2018	2017
Fixed Price	$ 1,452.4	$ 1,146.2	$ 1,036.9
Other	44.1	56.7	70.8
Total sales	$1,496.5	$1,202.9	$1,107.7 | <code>Ambiguity</code>: Yes</td></tr>
<tr><td>ABG-COQA</td><td><code>id</code>: 3ns0a6kxc48ribjdggweghvkamnzgll15l2
<code>question</code>: What politics did Lloyd George have?
<code>answers</code>: Liberalism
<code>contexts</code>: "Wales is a country that is part of the United Kingdom and the island of Great Britain. It is bordered by England to the east, the Irish Sea to the north and west, and the Bristol Channel to the south. It had a population in 2011 of 3,063,456 and has a total area of . Wales has over of coastline and is largely mountainous, with its higher peaks in the north and central areas, including Snowdon, its highest summit. The country lies within the north temperate zone and has a changeable, maritime climate. Welsh national identity emerged among the Celtic Britons after the Roman withdrawal from Britain in the 5th century, and Wales is regarded as one of the modern Celtic nations. Llywelyn ap Gruffudd's death in 1282 marked the completion of Edward I of England's conquest of Wales, though Owain Glyndŵr briefly restored independence to Wales in the early 15th century. The whole of Wales was annexed by England and incorporated within the English legal system under the Laws in Wales Acts 1535–1542. Distinctive Welsh politics developed in the 19th century. Welsh Liberalism, exemplified in the early 20th century by Lloyd George, was displaced by the growth of socialism and the Labour Party. Welsh national feeling grew over the century; "Plaid Cymru" was formed in 1925 and the Welsh Language Society in 1962. Established under the Government of Wales Act 1998, the National Assembly for Wales holds responsibility for a range of.
<code>Ambiguity</code>: No</td></tr>
</table>

Table 7: Examples for ASQA, PACIFIC, and ABG-COQA datasets.

<table>
<tr><th>Method</th><th>Prompt Template</th></tr>
<tr><td>Binary Prompting</td><td><code>Let's work this out in a step by step way to be sure we have the right answer. Please determine whether the question needs the further clarification, given the context. Note that only use information from the context to answer the question. Context: {CONTEXT}\nQuestion: {Question}.\nWhether a clarification question is needed:</code></td></tr>
<tr><td>Binary Prompting (CoT)</td><td><code>Let's work this out in a step by step way to be sure we have the right answer. Please determine whether the question needs the further clarification, given the context. Note that only use information from the context to answer the question. Context: {CONTEXT}\nQuestion: {Question}.\nGenerated Answers: {Answers}\nWhether a clarification question is needed:</code></td></tr>
<tr><td>Answer-oriented Prompting</td><td><code>Provide all the accurate responses to the question based on the given context. You must only use words that appear in the context to formulate your answer. Context: {CONTEXT}\nQuestion: {Question}.\nAll correct answers for the question are:</code></td></tr>
<tr><td>Verbalized Confidence</td><td><code>Let's work this out in a step by step. Please indicate your confidence level (from 0 to 100) regarding the accuracy of the provided answer, based on the given context. You must use numerical values only. Context: {CONTEXT}\nQuestion: {Question}.\nGenerated Answers: {Answers}\nAnswer: Answer.\nConfidence in accuracy:</code></td></tr>
<tr><td>LLM Self-Eval</td><td><code>Let's work this out in a step by step. Please determine whether the generated answer is correct or not. Context: {CONTEXT}\nQuestion: {Question}.\nGenerated Answers: {Answers}\nAnswer: Answer.\nWhether this answer is correct:</code></td></tr>
</table>

Table 8: Prompt templates for each method evaluated. Each example will be concatenated with several demonstration examples, which contain ground-truth labels.