TRAQ: Trustworthy Retrieval Augmented Question Answering via Conformal Prediction

Shuo Li
University of Pennsylvania
lishuo1@seas.upenn.edu

Sangdon Park
Pohang University of Science and Technology
sangdon@postech.ac.kr

Insup Lee
University of Pennsylvania
lee@cis.upenn.edu

Osbert Bastani
University of Pennsylvania
obastani@seas.upenn.edu

Abstract

When applied to open-domain question answering, large language models (LLMs) frequently generate incorrect responses based on made-up facts, which are called hallucinations. Retrieval augmented generation (RAG) is a promising strategy to avoid hallucinations, but it does not provide guarantees on its correctness. To address this challenge, we propose the Trustworthy Retrieval Augmented Question Answering, or TRAQ, which provides the first end-to-end statistical correctness guarantee for RAG. TRAQ uses conformal prediction, a statistical technique for constructing prediction sets that are guaranteed to contain the semantically correct response with high probability. Additionally, TRAQ leverages Bayesian optimization to minimize the size of the constructed sets. In an extensive experimental evaluation, we demonstrate that TRAQ provides the desired correctness guarantee while reducing prediction set size by 16.2% on average compared to an ablation. The implementation is available: https://github.com/shuoli90/TRAQ.

1 Introduction

Large Language Models (LLMs) have achieved State-Of-The-Art (SOTA) results on many question answering (QA) tasks (OpenAI, 2023; Touvron et al., 2023a,b). However, in open-domain QA tasks where candidate answers are not provided, LLMs have also been shown to confidently generate incorrect responses, called hallucinations (Ouyang et al., 2022; Kuhn et al., 2023). Hallucinations have already led to real-world consequences when end users rely on the correctness of the generated text. As a consequence, there is an urgent need for techniques to reduce hallucinations.

We propose a novel framework, Trustworthy Retrieval Augmented Question Answering (TRAQ), summarized in Figure 1, that combines Retrieval Augmented Generation (RAG) (Guu et al., 2020; Lewis et al., 2021) with conformal prediction (Vovk et al., 2005; Shafer and Vovk, 2007; Park et al., 2020; Angelopoulos and Bates, 2022) to provide theoretical guarantees on question answering performance.

RAG reduces hallucinations by retrieving passages from a knowledge base such as Wikipedia and then using an LLM to answer the question. If the retrieved passages are relevant to the question, the LLM can use this information to generate correct answers. However, RAG can fail for two reasons: either the retrieved passage is not relevant to the question, or the LLM generates the incorrect answer despite being given a relevant passage.

To avoid these issues, TRAQ uses conformal prediction, an uncertainty quantification technique...
that modifies the underlying model to predict sets of outputs rather than a single output. These prediction sets are guaranteed to contain the true output at a user-specified rate, e.g., at least 90% of the time. In particular, TRAQ applies conformal prediction separately to the retrieval model (to obtain sets of retrieved passages guaranteed to contain the relevant passage with high probability) and the generator (to obtain sets of answers that contain the true answer with high probability, assuming the relevant passage is given). Then, TRAQ aggregates the two sets for the RAG task, as demonstrated in Figure 2a. By a union bound, retriever sets contain relevant passages, and generator prediction sets contain true answers with high probability, establishing that the aggregated set by TRAQ contains the ground truth answer with high probability.

A major challenge to this basic pipeline is that there may be many different ways of expressing the correct answer in natural language. For example, the responses deep learning is a subset of machine learning and machine learning is a subset of deep learning are different ways of expressing the same meaning (Kuhn et al., 2023; Lin and Demner-Fushman, 2007). This diversity of possible responses also makes prediction probabilities less reliable since if an answer can be expressed in many different but equivalent ways, then the probabilities may be divided across these different responses, making them all smaller even if the model is confident it knows the correct answer.

TRAQ addresses this challenge by modifying the notion of ground-truth coverage in conformal prediction to focus on semantic notions of uncertainty. In particular, TRAQ aggregates semantically equivalent answers across a large number of samples from the LLM and uses the number of clusters of non-equivalent answers as a measure of uncertainty. This measure is used as a nonconformity measure to construct prediction sets. Finally, the prediction sets are over clusters of equivalent answers rather than individual answers. This strategy also enables TRAQ to work on black-box APIs such as GPT-3.5-Turbo, where the predicted probabilities for individual tokens are not available.

A second challenge is that the prediction sets can become very large since we are aggregating uncertainty across multiple components. This complexity introduces hyperparameters into TRAQ; while TRAQ guarantees correctness regardless of the choice of these hyperparameters, they can affect the performance of TRAQ in terms of the average prediction set size. To address this challenge, TRAQ uses Bayesian optimization to minimize the average size of the prediction sets it generates.

We evaluate TRAQ in conjunction with several generative LLMs, including both GPT-3.5-Turbo-0613 (Ouyang et al., 2022) and Llama-2-7B (Touvron et al., 2023b); and on four datasets, including a biomedical question answering dataset. Our experiments demonstrate that TRAQ empirically satisfies the coverage guarantee (i.e. the prediction sets outputs contain semantically correct answers with the desired probability), while reducing the average prediction set size compared to an ablation by 16.2%. Thus, TRAQ is an effective strategy for avoiding hallucinations in applications of LLMs to open domain question answering.

Contributions. We offer the first conformal prediction guarantees for retrieval augmented generation (RAG) targeted question answering. Our framework, TRAQ, introduces a novel nonconformity measure that estimates the uncertainty for each semantically distinct meaning and obtains a coverage guarantee at the semantic level. Furthermore, TRAQ leverages Bayesian optimization to minimize the average size of the generated prediction sets. Finally, our experiments demonstrate that TRAQ is effective at avoiding hallucinations in open-domain question answering.

2 Background

Retrieval for Open-Domain QA. A two-stage approach is often used for open-domain question answering (QA): first, a retriever is used to obtain informative passages; and second, a generator produces answers based on the retrieved passages. A popular choice for the retriever is the Dense Passage Retriever (DPR) (Karpukhin et al., 2020b), which measures similarity by taking the inner product of the BERT (Devlin et al., 2019; Reimers and Gurevych, 2019) embeddings of the question and passage (Devlin et al., 2019; Reimers and Gurevych, 2019). Other works (Lin and Lin, 2022; Salemi et al., 2023; Lin et al., 2022; Zhang et al., 2021) have improved the performance of DPR and extended it to more diverse settings. Retrieval Augmented Generation (RAG) (Lewis et al., 2021) proposes to jointly fine-tune the retriever and the generator for QA tasks.

Conformal Prediction. Conformal prediction (Vovk et al., 2005; Papadopoulos, 2008) is a
general distribution-free approach to quantifying uncertainty for machine learning (ML) models. Let $\mathcal{X}$ be the input space, and $\mathcal{Y}$ be the output space. Conformal prediction first assumes that a nonconformity measure (e.g., negative probabilities predicted by an ML model) $s : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ is given. Lower values of $s(x,y)$ indicate better agreement between $x$ and $y$. Given a held-out calibration set $\mathcal{B} = \{(x_i, y_i)\}_{i=1}^{N}$ sampled i.i.d. from the data distribution $\mathcal{D}$, as well as a user-specified error level $\alpha$, conformal prediction constructs a prediction set for a testing data point $x_{test}$ by

$$C(x_{test}) = \{y \in \mathcal{Y} \mid s(x_{test}, y) \leq \tau\},$$

where $\tau$ is the $\frac{[(1-\alpha)(N+1)]}{N}$-th smallest score in $\{s(x_i, y_i)\}_{i=1}^{N}$. Conformal prediction guarantees that the true labels are contained in the constructed prediction sets with probability at least $1 - \alpha$:

**Theorem 1.** Conformal Prediction Guarantee. Given the current calibration set, where promising error budgets $\alpha_{\text{Ret}}$ and $\alpha_{\text{LLM}}$ are sampled from a Bayesian model. Using these budgets, the prediction sets are constructed on the optimization set and evaluated for their performance. This process is repeated $T$ times, and the final output is the error budget $\alpha_{\text{Ret}}$ and $\alpha_{\text{LLM}}$ with the highest performance. The chosen hyperparameters are used to construct prediction sets as in the first step using a separate held-out calibration set. The TRAQ framework is summarized in Figure 2.

**3 The TRAQ Framework**

TRAQ is composed of two steps. The first is the **Prediction Set Construction** step, where a question $q$ is used to create a retrieval set $C_{\text{Ret}}(q)$ for the retriever and a LLM set $C_{\text{LLM}}(q,p)$ for each pair (question $q$, passage $p$). These sets are aggregated into an Aggregation Set $C_{\text{Agg}}(q)$. The second step is the **Performance Improvement** step, where promising error budgets $\alpha_{\text{Ret}}$ and $\alpha_{\text{LLM}}$ are sampled from a Bayesian model. Using these budgets, the prediction sets are constructed on the optimization set and evaluated for their performance. This process is repeated $T$ times, and the final output is the error budget $\alpha_{\text{Ret}}$ and $\alpha_{\text{LLM}}$ with the highest performance. The chosen hyperparameters are used to construct prediction sets as in the first step using a separate held-out calibration set. The TRAQ framework is summarized in Figure 2.

**3.1 Assumptions**

To construct provable prediction sets, we first make three necessary assumptions:

**Assumption I.I.D.** For both the retrieval and LLM tasks, the examples are drawn independently and identically from the data distribution $\mathcal{D}$.

**Assumption Retriever Correctness.** Given a question $q$, the underlying retriever is able to retrieve
the most relevant passage $p^*$ within the top-$K$ retrieved passages.

**Assumption LLM Correctness.** Given a question $q$ and its most relevant passage $p^*$, the LLM is able to generate a semantically correct response within the top-$M$ samples.

Assumption I.I.D is a standard assumption from the conformal prediction literature and is needed to apply conformal prediction algorithms (it can be slightly relaxed to exchangeable distributions, but we make the i.i.d. assumption for simplicity).

Assumptions Retriever Correctness and LLM Correctness are needed to ensure that the most relevant passages and semantically correct answers can be contained in the prediction sets if the prediction sets are sufficiently large. In principle, we can use very large values of $K$ and $M$ to satisfy this assumption, though there are computational and cost limitations in practice. We discuss ways to remove these assumptions in Limitations.

### 3.2 Prediction Set Construction

**Retriever Set:** To construct the retriever sets $C_{\text{Ret}}$, we use the negative inner product between the question $q$ and the annotated most relevant passage $p^*$, denoted as $-R_{q,p^*}$, as the nonconformity measures (NCMs). Given $N$ such NCMs $\{s_1, \ldots, s_N\}$ in the calibration set and the error budget $\alpha_{\text{Ret}}$ for the retriever set, we construct the retriever set by

$$C_{\text{Ret}}(q) = \{ p \mid -R_{q,p} \leq \tau_{\text{Ret}} \}, \quad (3)$$

where

$$\tau_{\text{Ret}} = \text{Quantile} \left( \left\{ s_n \right\}_{n=1}^N; \frac{[N + 1)(1 - \alpha_{\text{Ret}})]}{N} \right)$$

Given this construction and Assumptions I.I.D and Assumption Retriever Correctness, the retriever sets are guaranteed to contain the most relevant passage with probability at least $1 - \alpha_{\text{Ret}}$.

**Lemma 2.1.** Suppose the questions $q$ and their corresponding most relevant passage $p^*$ are sampled from the distribution $D_{\text{Passage}}$. Given the error budget $\alpha_{\text{Ret}}$, the retriever sets satisfy

$$\Pr_{(q,p^*) \sim D_{\text{Passage}}}(p^* \in C_{\text{Ret}}(q)) \geq 1 - \alpha_{\text{Ret}}.$$ 

This result follows straightforwardly from Theorem 1 and Assumptions I.I.D & Retriever Correctness. We give a proof in Appendix B.

**LLM Set:** We utilize Monte Carlo sampling to approximate confidences for different semantic meanings; then, we use the negative approximated confidences as the NCMs to construct LLM sets. Specifically, for each $(question, passage)$ pair, we ask the LLM to generate $M$ responses ($M = 30$ in our experiments). Given two responses $r$ and $r'$, we cluster them together if they have high similarity, which is measured by Rouge score (Lin, 2004) or Natural Language Inference (NLI) model (Kuhn et al., 2023; He et al., 2021). We consider the two responses to be semantically similar if they have a Rouge score greater than 0.7 or are deemed to entail each other by the NLI model. After clustering, for each cluster $i$, let $N_i$ be the number of responses in the cluster; we approximate the confidence of a response $r$ by $N_i/M$ if $r$ belongs to the $i$-th cluster. Finally, given the error budget for LLM $\alpha_{\text{LLM}}$, we can utilize a similar process to that in (3) to construct LLM sets. The constructed sets satisfy the following:
Lemma 2.2. Suppose the questions \( q \), their corresponding most relevant passage \( p^* \), and semantically correct responses \( r^* \) are sampled from distribution \( D_{\text{Response}} \). Given error budget \( \alpha_{\text{LLM}} \), if Assumptions I.I.D & LLM Correctness hold, the LLM sets satisfy

\[
\Pr_{(q,p^*,r^*) \sim D_{\text{Response}}}(r^* \in C_{\text{LLM}}(q,p^*)) \geq 1 - \alpha_{\text{LLM}}.
\]

The proof of Lemma 2.2 is similar to that of Lemma 2.1; we give it in Appendix B.

Note that since the uncertainty score can be arbitrary in conformal prediction, the lemma 2.2 holds regardless of the chosen heuristic measures (e.g., Rouge score or BERT embedding). If the chosen heuristic underperforms, conformal prediction will simply construct large prediction sets to compensate. We validate this claim in Section 4.

Aggregated Set: To obtain an overall correctness guarantee, we construct an aggregated set \( C_{\text{Agg}} \) by constructing an LLM set \( C_{\text{LLM}} \) for each passage \( q \) contained in the retriever set; and take the union of the \( C_{\text{LLM}} \)’s, i.e.

\[
C_{\text{Agg}}(q) = \bigcup_{p \in C_{\text{Ret}}(q)} C_{\text{LLM}}(q,p).
\]

Then, the resulting Aggregated set \( C_{\text{Agg}} \) satisfies the following:

Theorem 3. Suppose the questions \( q \) and semantically correct responses \( r^* \) are sampled from the distribution \( D \), and a user-specified error level \( \alpha \) is given. By aggregating retriever sets with error budget \( \alpha_{\text{Ret}} \) by (4) with LLM sets with error budget \( \alpha_{\text{LLM}} \), with \( \alpha = \alpha_{\text{Ret}} + \alpha_{\text{LLM}} \), the aggregated sets satisfy

\[
\Pr_{(q,r^*) \sim D}(r^* \in C_{\text{Agg}}(q)) \geq 1 - \alpha.
\]

We give a proof in Appendix B. After taking the union, we remove duplicated responses and recluster semantic meanings. Given that this post-processing phase solely eliminates duplicate responses, it will not remove correct semantic meanings, and Theorem 3 remains valid.

Note that this aggregation process is actually a global hypothesis testing method called the Bonferroni correction. Lemmas 2.1 & 2.2 and Theorem 3 can be straightforwardly extended to the probably approximately correct (PAC) guarantee by constructing PAC prediction sets; see Appendix B.1 for details.

Algorithm 1 Prediction Set Optimization

<table>
<thead>
<tr>
<th>Input:</th>
<th>Optimization set ( B_{\text{Opt}} ), performance metric ( f ), error level ( \alpha )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:</td>
<td>Initialize Gaussian process ( G )</td>
</tr>
<tr>
<td>2:</td>
<td>for ( t \in {1, \ldots, T} ) do</td>
</tr>
<tr>
<td>3:</td>
<td>Sample ( \alpha_{\text{Ret}} ) and ( \alpha_{\text{LLM}} ) basing on ( G )</td>
</tr>
<tr>
<td>4:</td>
<td>Normalize ( \alpha_{\text{Ret}} ) and ( \alpha_{\text{LLM}} ) so that ( \alpha_{\text{Ret}}, \alpha_{\text{LLM}} \in (0, 1) ), and ( \alpha_{\text{Ret}} + \alpha_{\text{LLM}} = \alpha )</td>
</tr>
<tr>
<td>5:</td>
<td>Compute ( \tau_{\text{Ret}} ) and ( \tau_{\text{LLM}} ) on ( B_{\text{Opt}} )</td>
</tr>
<tr>
<td>6:</td>
<td>Construct ( C_{\text{Agg}} ) on ( B_{\text{Opt}} )</td>
</tr>
<tr>
<td>7:</td>
<td>Evaluate performance of the ( C_{\text{Agg}} ) using ( f )</td>
</tr>
<tr>
<td>8:</td>
<td>Update ( G ) using the evaluation results</td>
</tr>
<tr>
<td>9:</td>
<td>end for</td>
</tr>
<tr>
<td>10:</td>
<td>return: the best error budgets ( \alpha_{\text{Ret}} ) and ( \alpha_{\text{LLM}} )</td>
</tr>
</tbody>
</table>

3.3 Performance Improvement

By Theorem 3, we can guarantee that semantically correct responses are included in the aggregated set with a probability of at least \( 1 - \alpha \), assuming \( \alpha = \alpha_{\text{Ret}} + \alpha_{\text{LLM}} \). This theorem is valid for any combination of the two error budgets. However, the predictive performance of the aggregation sets is influenced by the specific choice of the error budgets. This issue has been discussed in the Bonferroni correction and the global testing literature (Neuwald and Green, 1994; Wilson, 2019; Poole et al., 2015).

Therefore, we optimize the error budgets using Bayesian optimization, a sampling-based global optimization technique suitable for non-convex, non-closed-form problems; see Appendix A.4 for details. In TRAQ, Bayesian optimization first models the underlying performance landscape using a Gaussian process; then, it samples error budgets (i.e., \( \alpha_{\text{Ret}} \) and \( \alpha_{\text{LLM}} \) based on the Gaussian process, and identifies \( \tau_{\text{Ret}} \) and \( \tau_{\text{LLM}} \) on a held-out optimization set \( B_{\text{Opt}} \). After assessing the performance of the sampled error budgets on \( B_{\text{Opt}} \), the Gaussian process is modified to more accurately reflect the performance landscape. This process is repeated \( T \) times. The pseudocode for this procedure is shown in Algorithm 1.

4 Experiments

Experiment Setup. We evaluate TRAQ on four datasets, including three standard QA datasets (Natural Question (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), SQuAD-1 (Rajpurkar et al., 2016)), and one biomedical QA dataset (BioASQ (Tsatsaronis et al., 2012)). On each dataset, we collect 1,000 samples that met the crite-
ria of Assumptions Retriever Correctness & LLM Correctness. We divide each dataset into calibration, optimization, and testing sets, with 300, 300, and 400 data points, respectively.

We employ two fine-tuned DPR models, one (Karpukhin et al., 2020a) trained on the Natural Question, TriviaQA, and SQuAD-1 datasets, and the other fine-tuned on BioASQ (see Appendix D.2 for training details). Furthermore, we use two generative large language models (LLMs): GPT-3.5-Turbo-0613 (GPT-3.5), whose internal embedding and prediction probabilities are not accessible, and Llama-2-7B (Llama-2). We separately fine-tune Llama-2 on Natural Question, TriviaQA, and SQuAD-1, with hyperparameters given in Appendix D.1.

For each question, we retrieve the top-20 passages; for each (question, passage) pair, we sample 30 responses, with a temperature of 1.0.

We evaluate using coverage levels 50%, 60%, 70%, 80%, and 90%. For the PAC guarantee, we use confidence level 90%. We use five random seeds for each experiment. To investigate the influence of prompt design, we design two prompts, one zero-shot and one few-shot prompt; the few-shot prompt includes two demonstrations. The prompt templates are provided in Appendix D.3. Unless otherwise specified, the zero-shot prompt is used for both GPT-3.5 and Llama-2.

We evaluate the performance of our approach using two metrics. The first metric is coverage rate, which is the rate at which the correct responses are contained in the constructed sets. We consider the responses to be correct if their Rouge-1 (Lin, 2004) scores with the annotated answers are greater than 0.3. The coverage rate is expected to be no less than the desired level on average across different random seeds. The second metric is the average prediction set size. Specifically, we consider two size measures: (i) the average number of semantic clusters and (ii) the average number of unique answers. Lower values indicate better performance.

We compare our approaches, TRAQ and TRAQ-P (the PAC version), to several baselines, including Vanilla, Bonf, and Bonf-P. Vanilla is a baseline that does not construct prediction sets and only uses the top retrieved passage and generated answers. Bonf and Bonf-P are ablations that omit Bayesian optimization. In all plots, we also show the Reference line indicating the desired coverage level.

We report both quantitative and qualitative results. Our quantitative experiments aim to answer the following.

(Q1) Do the coverage guarantees hold for the retriever and the generator?
(Q2) Does the overall coverage guarantee hold?
(Q3) How do Bayesian optimization and the coverage level affect prediction set sizes?
(Q4) Does TRAQ work for different semantic clustering methods and performance metrics?
(Q5) How does prompt affect results?

Q1: Do the coverage guarantees hold for the retriever and generator? To validate the coverage guarantees of the retriever and generator, we consider the coverage rates of retriever and LLM sets (named Ret and LLM), and with the PAC guarantee (named Ret-P and LLM-P). We report results on BioASQ using GPT-3.5 in Figure 3: Results for other datasets and different LLMs are reported in Figure 10, and are qualitatively similar. As shown in Figure 3, the empirical coverage levels of the retrieval and QA prediction sets are close to the desired coverage levels. Thus, the coverage guarantees hold for individual components, as desired.

We also report empirical coverage rates with 20 random seeds in Figure 11. Compared to results with 5 random seeds, empirical coverage with more random seeds become closer to the desired level. Furthermore, when using the PAC prediction sets, the empirical coverage levels were almost always above the expected coverage levels across all random seeds, as desired.

Q2: Does the end-to-end coverage guarantee hold? To verify the end-to-end guarantees from TRAQ, we report two rates. The first is the rate at which the correct responses are covered consider-
Q3: How do Bayesian optimization and the coverage level affect prediction set sizes? To demonstrate the advantages of incorporating Bayesian optimization, we evaluate the average prediction set sizes (in terms of the number of semantic clusters) across different approaches. We first show results across different coverage levels and random seeds using different methods on BioASQ dataset Figure 6. It shows that TRAQ and TRAQ-P are able to construct smaller prediction sets than their counterparts without Bayesian optimization (Bonf and Bonf-P). Furthermore, we report the average semantic counts on different datasets and coverage levels using GPT-3.5 in Table 1 and using Llama-2 in Table 2. As can be seen, Bayesian optimization is especially effective in reducing prediction set size when higher coverage rates are desired (80% and 90%). In these cases, both TRAQ and TRAQ-P are able to construct significantly smaller prediction sets, reducing their size by 16.2% on average (18.1% in Table 1 and 14.2% in Table 2). Importantly, even though the prediction sets are smaller, the desired overall coverage guarantees still hold. These tables also show that higher coverage levels tend to result in larger prediction set sizes; this trade-off is expected since stronger statistical guarantees require more conservative prediction sets.

Q4: Does TRAQ work with different semantic clustering methods? We evaluate whether TRAQ remains effective with different semantic clustering methods and performance metrics. We use the semantic clustering method proposed by Kuhn et al. (2023), which is based on BERT (Devlin et al., 2019; Reimers and Gurevych, 2019), and specified the performance metric as the average number of unique answers in the aggregated prediction sets. We evaluate this setup on the SQuAD-1 dataset using GPT-3.5. The results, shown in

Figure 4: End-to-end guarantee considering only the most relevant passage on BioASQ Dataset.

Figure 5: End-to-end coverage guarantee considering all passages on the BioASQ dataset.

Figure 6: Prediction set sizes according to the average number of semantic clusters.
<table>
<thead>
<tr>
<th>Task</th>
<th>Cov(%)</th>
<th>TRAQ</th>
<th>Bonf</th>
<th>TRAQ-P</th>
<th>Bonf-P</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIO</td>
<td>50</td>
<td>2.50</td>
<td>2.40</td>
<td>2.90</td>
<td>2.90</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>2.90</td>
<td>2.90</td>
<td>3.40</td>
<td>3.40</td>
</tr>
<tr>
<td></td>
<td>70</td>
<td>3.50</td>
<td>3.50</td>
<td>4.00</td>
<td>4.00</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>4.40</td>
<td>5.00</td>
<td>5.80</td>
<td>6.70</td>
</tr>
<tr>
<td></td>
<td>90</td>
<td>8.90</td>
<td>10.3</td>
<td>16.3</td>
<td>21.3</td>
</tr>
<tr>
<td>NQ</td>
<td>50</td>
<td>3.00</td>
<td>3.20</td>
<td>3.60</td>
<td>3.60</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>3.70</td>
<td>3.70</td>
<td>4.50</td>
<td>4.50</td>
</tr>
<tr>
<td></td>
<td>70</td>
<td>4.60</td>
<td>4.60</td>
<td>5.70</td>
<td>5.70</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>6.10</td>
<td>6.40</td>
<td>7.30</td>
<td>9.30</td>
</tr>
<tr>
<td></td>
<td>90</td>
<td>10.3</td>
<td>12.2</td>
<td>16.7</td>
<td>23.6</td>
</tr>
<tr>
<td>Trivia</td>
<td>50</td>
<td>2.00</td>
<td>2.00</td>
<td>2.40</td>
<td>2.40</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>2.90</td>
<td>2.90</td>
<td>2.70</td>
<td>2.70</td>
</tr>
<tr>
<td></td>
<td>70</td>
<td>3.00</td>
<td>2.90</td>
<td>3.50</td>
<td>3.40</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>3.70</td>
<td>3.80</td>
<td>4.60</td>
<td>4.60</td>
</tr>
<tr>
<td></td>
<td>90</td>
<td>5.90</td>
<td>5.80</td>
<td>7.20</td>
<td>7.80</td>
</tr>
<tr>
<td>SQuAD1</td>
<td>50</td>
<td>3.00</td>
<td>3.50</td>
<td>4.10</td>
<td>4.00</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>4.10</td>
<td>4.10</td>
<td>4.60</td>
<td>5.00</td>
</tr>
<tr>
<td></td>
<td>70</td>
<td>4.80</td>
<td>5.20</td>
<td>5.50</td>
<td>7.40</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>6.20</td>
<td>8.20</td>
<td>8.91</td>
<td>11.0</td>
</tr>
<tr>
<td></td>
<td>90</td>
<td>12.6</td>
<td>14.1</td>
<td>21.3</td>
<td>25.9</td>
</tr>
</tbody>
</table>

Table 1: Average semantic counts using GPT-3.5.

Figure 7: Coverage rate using BERT embeddings on SQuAD-1 dataset.

Figures 7 & 8 demonstrate that TRAQ remains successful. Specifically, Figure 7 shows that the overall coverage guarantee holds, and Figure 8 demonstrates that TRAQ and TRAQ-P reduce prediction set sizes compared to their ablations Bonf and Bonf-P, respectively.

**Q5: How does prompt engineering affect results?** We investigate how prompt engineering affects TRAQ performance using a few-shot prompt with two demonstrations. The prompt template is provided in Appendix D.3. We evaluate TRAQ on Natural Question using GPT-3.5. The end-to-end coverage rates and prediction set sizes using different methods are shown in Figure 16. TRAQ with a few shot prompt achieves the desired coverage rate on average and reduces prediction set size compared to its ablation. In Figure 9, we also compare the zero-shot and few-shot prompts in terms of performance. Interestingly, zero-shot prompting mostly yields better efficiencies. This could be because zero-shot prompting generated more diverse answers and had lower confidence in wrong answers. An example of the comparison between responses using different prompts is given in the Appendix D.3.

**Qualitative Analysis.** By constructing prediction sets, TRAQ guarantees that it includes correct responses with high probability. For example, we consider the following question: **Who played in the movie a star is born with Judy Garland?**, where James Mason is a correct answer. The responses of different methods are shown below. While standard RAG fails to return the correct answer, TRAQ and Bonf output sets containing the correct answers; and TRAQ obtains a smaller set.

Question: who played in the movie a star is born with judy garland

Figure 8: Prediction set size according to average number of unique answers.

![Figure 9: Comparison between zero-shot and few-shot prompts on prediction set size.](image-url)

In Figure 9, we compare the zero-shot and few-shot prompts in terms of performance. Interestingly, zero-shot prompting mostly yields better efficiencies. This could be because zero-shot prompting generated more diverse answers and had lower confidence in wrong answers. An example of the comparison between responses using different prompts is given in the Appendix D.3.

**Qualitative Analysis.** By constructing prediction sets, TRAQ guarantees that it includes correct responses with high probability. For example, we consider the following question: **Who played in the movie a star is born with Judy Garland?**, where James Mason is a correct answer. The responses of different methods are shown below. While standard RAG fails to return the correct answer, TRAQ and Bonf output sets containing the correct answers; and TRAQ obtains a smaller set.

Figure 9: Comparison between zero-shot and few-shot prompts on prediction set size.
We propose an algorithm, called Trustworthy Retrieval Augmented Question Answering (TRAQ), which applies conformal prediction to construct prediction sets for Retrieval Augmented Generation (RAG). TRAQ first constructs prediction sets for the retriever and generator and then aggregates these sets. TRAQ guarantees that for each question, a semantically correct answer is included in the prediction set it outputs with high probability. To the best of our knowledge, this guarantee is the first conformal guarantee for retrieval augmented generation. Additionally, to minimize prediction set size, TRAQ leverages Bayesian optimization to identify optimal hyperparameters. In our comprehensive experiments, we demonstrate that TRAQ provides an overall semantic level coverage guarantee across different tasks, and that Bayesian optimization effectively reduces prediction set size.

6 Broader Impacts

The need for trustworthy AI algorithms has recently become paramount due to the risks of spreading misleading information (Biden, 2023; Commission, 2023). We propose TRAQ, a framework that aims to address the hallucination problem by using conformal prediction to provide probabilistic guarantees for retrieval augmented generation (RAG). In addition, TRAQ leverages novel techniques to improve performance that may be useful more broadly in conformal prediction.

7 Limitations

TRAQ makes three assumptions: that the data is independent and identically distributed (I.I.D), that the retriever has good performance (Retriever Correctness), and that the language model can generate a response to the input question (LLM Correctness). Our experiments have verified I.I.D, but Retriever Correctness and LLM Correctness may not be valid if the underlying retriever and language model do not perform well. To relax Retriever Correctness, we can select more passages than the top-20 used in our experiments. To remove LLM Correctness, we propose providing a guarantee of including I do not know in the aggregation set if the language model cannot answer the input question. We describe how TRAQ can be modified to provide such guarantees in Appendix E.

TRAQ is a post-hoc method, so its prediction sets may be larger than necessary if the underlying models, such as the retriever and large language model, do not work properly. Additionally, if the semantic clustering techniques (Rouge score based or BERT-based) are invalid, then some semantically unrelated answers may be aggregated.

Finally, TRAQ can reduce inference speed due to the need for multiple retrievals, each of which needs to be embedded separately by the LLM. In our current setup, the computational complexity of the retrieval phase increases linearly with the number of retrievals (typically around 15). Avoiding this overhead is a key direction for future research.

Acknowledgement

This work was generously supported by NSF Award CCF-1917852, ARO Award W911NF20-1-0080, and Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2019-0-01906, Artificial Intelligence Graduate School Program (POSTECH)).

References


Steven Bird, Ewan Klein, and Edward Loper. 2009. Natural language processing with Python: analyzing text
with the natural language toolkit. " O'Reilly Media, Inc.”.


3808


Glenn Shafer and Vladimir Vovk. 2007. *A tutorial on conformal prediction*.


A Conformal Prediction and PAC Guarantees

A.1 Conformal Prediction and Hypothesis Testing

Conformal prediction is a distribution-free uncertainty quantification technique that constructs provable prediction sets for black-box models. Specifically, let $\mathcal{X}$ and $\mathcal{Y}$ be the input and label spaces, respectively, and $(x, y)$ be an input-label pair. Conformal prediction assumes given a calibration set $B = \{x_i, y_i\}_{i=1}^N$ with $N$ input-label pairs, along with a nonconformity measure $s : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ that measures how different a pair $(x, y)$ is from the examples sampled from the distribution $D$. Given a new input $x_{\text{test}}$, conformal prediction constructs a prediction set $C(x_{\text{test}}) \subseteq \mathcal{Y}$ using Algorithm 2. Intuitively, for each label $y \in \mathcal{Y}$, this algorithm checks whether $(x_{\text{test}}, y)$ is similar to the examples in $B$ according to the nonconformity measure $s(x_{\text{test}}, y)$. If $s(x, y)$ is low enough, then $y$ is included in the prediction set $C(x_{N+1});$ otherwise, $y$ is excluded from $C(x_{N+1})$.

Algorithm 2 The Conformal Algorithm

**Input:** Nonconformity measure $s$, significance level $\alpha$, calibration set $B = \{x_n, y_n\}_{n=1}^N$, a new input $x_{\text{test}}$, label space $\mathcal{Y}$

1. Compute the threshold $\tau$ as the $\frac{(1-\alpha)(N+1)}{N}$-th smallest score in $\{s(x_i, y_i)\}_{i=1}^N$.
2. Construct prediction set for $x_{\text{test}}$ by
   \[ C(x_{\text{test}}) = \{y \mid s(x_{\text{test}}, y), y \in \mathcal{Y} \} \]

**Return:** $C(x_{\text{test}})$.

A.2 PAC Prediction Set

PAC prediction sets (Vovk, 2012; Park et al., 2021) are a variant of conformal prediction approach that satisfies stronger PAC-style guarantees. Let $D$ be the distribution of samples, and $B = \{x_n, y_n\}_{n=1}^N$ be a held-out calibration set of i.i.d. data points from $D$ of size $N$. We denote the joint distribution on $N$ samples by $D^N$. The goal is to find a set of a small size satisfying the PAC property, that is, given $\alpha, \delta \in (0, 1)$,

\[ \Pr_{Z \sim D^n}[L_D(C) \leq \alpha] \geq 1 - \delta, \]

where the $\Pr_{Z \sim D^n}$ refers to the chances of calibration succeeding. In this case, we say $C$ is $(\alpha, \delta)$-probably approximately correct (PAC). To construct $(\alpha, \delta)$-PAC sets, the PAC prediction set considers the following one-dimensional parameterization of the prediction sets:

\[ C_\tau(x) = \{y \in \mathcal{Y} \mid g(x, y) \geq \tau \}, \]

where $\tau \geq 0$ and $g : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$ is any given scoring function (e.g., the label probabilities output by a deep neural network). The threshold $\tau$ is computed by solving the following optimization problem:

\[ \hat{\tau} = \arg \max_{\tau \geq 0} \tau \text{ subj. to } \sum_{(x, y) \in Z} \mathbb{I}[y \notin C_\tau(x)] \leq k^*, \]

(5)

where

\[ k^* = \arg \max_{k \in \mathbb{N} \cup \{0\}} k \text{ subj. to } F(k; N, \alpha) \leq \delta, \]

where $F(k; N, \alpha)$ is the cumulative distribution function of the binomial random variable $\text{Binomial}(N, \alpha)$ with $N$ trials and success probability $\alpha$. Maximizing $\tau$ corresponds to minimizing the prediction set size.

We have the following theorem:

**Theorem 4** ((Vovk, 2012; Park et al., 2021)). $C_{\hat{\tau}}$ is $(\alpha, \delta)$-correct for $\hat{\tau}$ as in (5).
A.3 Conformal Prediction and PAC Prediction Set Comparison

Conformal Prediction Guarantee Formally, we can write the conformal prediction guarantee as

\[ \Pr_{(X,Y) \sim D}(Y \in C(X)) \geq 1 - \alpha. \]

In other words, the prediction sets constructed by conformal prediction guarantee that over the whole distribution \( D \), the probability that the true label is contained in the set is at least \( 1 - \alpha \). Note that this coverage probability is marginalized over all possible calibration sets. On the other hand, for a specific calibration set \( B \), this guarantee might not hold. For example, the guarantee will not hold if the samples in \( B \) are concentrated in a small region of the joint distribution and therefore are not representative of the joint distribution \( D \).

PAC Prediction Set Guarantee Formally, we can write the guarantee of the PAC prediction set guarantee as

\[ \Pr_{B \sim D^N}(\Pr_{(X,Y) \sim D}(Y \in C(X)) \geq 1 - \alpha) \geq 1 - \delta. \]

Compared to the conformal prediction guarantee, the difference is the outer probability, which is on the given calibration set \( B \). Intuitively, the guarantee of the PAC prediction set says that conditioning on the given calibration set \( B \), we can say with high confidence (at least \( 1 - \delta \)) that the true label is contained in the constructed set \( C(X) \) with high probability \( (1 - \alpha) \). As a result, the PAC prediction set guarantee is stronger than the conformal prediction guarantee, as the PAC prediction set guarantee is over an individual calibration set, while the conformal prediction guarantee is marginalized over all possible calibration sets.

A.4 Bayesian Optimization

Bayesian optimization (BO) is a technique to find the global optimum of a potentially nonconvex, nonlinear, or nonclosed-form objective function \( f \) with decision variables \( \{b_1, \ldots, b_M\} \). It builds a probabilistic model of the objective function and then selects parameters that could maximize it. The model is then refined using the chosen parameters. This process is repeated until an iteration budget \( T \) is reached, as shown in Algorithm 3 (Frazier, 2018). Our implementation of Bayesian optimization is based on scikit-optimization (Head et al., 2021).

Algorithm 3 Bayesian Optimization

1: Place a Gaussian process prior \( g \) on \( f \).
2: Observe \( f \) at \( t_0 \) points according to an initial space-filling experimental design. Set \( t = t_0 \).
3: while \( t \leq T \) do
4: Update the posterior probability distribution on \( g \) using all available data.
5: Let \( b_t \) be a maximizer of the acquisition function over \( b \), where the acquisition function is computed using the current posterior distribution.
6: Observe \( f(b_t) \).
7: Increment \( t \).
8: end while
9: Return: either the point evaluated with the smallest \( f(b) \) or the point with the smallest posterior.

B Proofs

Proof of Lemma 2.1. First, based on Assumption I.I.D, samples collected for the construction of the retrieval prediction set are i.i.d. with unobserved samples, satisfying the i.i.d. (exchangeability) assumption required by conformal prediction (PAC prediction set).

Second, based on Assumption Retriever Correctness, for each input question \( q \), since its relevant passage can be retrieved, the prediction set can contain the relevant passage if the threshold \( \tau_{\text{Ret}} \) is appropriately set. (Otherwise, the prediction set cannot contain the relevant passage even if all retrieved passages are included.)
Third, since we construct the retriever set following conformal prediction with the error level being $\alpha_{\text{Ret}}$, the resulting retriever sets satisfy:

$$\Pr_{(q,p^*) \sim D_{\text{Passage}}} (p^* \in C_{\text{Ret}}(q)) \geq 1 - \alpha_{\text{Ret}}.$$

Proof of Lemma 2.2. First, based on Assumption I.I.D, samples collected for the construction of the LLM prediction set are i.i.d. with unobserved samples, satisfying the i.i.d. (exchangeability) assumption required by conformal prediction (PAC prediction set).

Second, based on Assumption LLM Correctness, for every input question and its most relevant passage $q^*$, since its semantically correct responses can be retrieved, the prediction set can contain correct responses if the threshold $\eta_{\text{LLM}}$ is appropriately set. (Otherwise, the prediction set cannot contain correct responses even if all responses are included.)

Third, since we construct the LLM prediction set following conformal prediction with the error level being $\alpha_{\text{LLM}}$, the resulting retriever sets satisfy:

$$\Pr_{(q,p^*,r^*) \sim D_{\text{Response}}} (r^* \in C_{\text{LLM}}(q,p^*)) \geq 1 - \alpha_{\text{LLM}}.$$

Proof of Theorem 3. We prove this theorem by union bound. Specifically, given two event $A$ and $B$, we have the following inequality:

$$\Pr(A \cup B) = \Pr(A) + \Pr(B) - \Pr(A \cap B) \leq \Pr(A) + \Pr(B).$$

In TRAQ, let event $A$ be

$$\{p^* \notin C_{\text{Ret}}(q)\};$$

and event $B$ be

$$\{r^* \notin C_{\text{LLM}}(q,p^*)\}.$$ 

By Lemma 2.1 and 2.2, we have

$$\Pr(p^* \notin C_{\text{Ret}}(q)) = 1 - \Pr(p^* \in C_{\text{Ret}}(q)) \leq \alpha_{\text{Ret}}$$

$$\Pr(p^* \notin C_{\text{LLM}}(q,p^*)) = 1 - \Pr(r^* \in C_{\text{LLM}}(q,p^*)) \leq \alpha_{\text{LLM}}.$$ 

Then, we have the following inequalities

$$\Pr(r^* \notin C_{\text{Agg}}(q))$$

$$= \Pr(r^* \notin \cup_{p \in C_{\text{Ret}}(q)} C_{\text{LLM}}(q,p))$$

$$= \Pr(r^* \notin \cup_{p \in C_{\text{Ret}}(q)} C_{\text{LLM}}(q,p), A) + \Pr(r^* \notin \cup_{p \in C_{\text{Ret}}(q)} C_{\text{LLM}}(q,p), A^C)$$

$$= \Pr(r^* \notin \cup_{p \in C_{\text{Ret}}(q)} C_{\text{LLM}}(q,p) | A) \Pr(A) + \Pr(r^* \notin \cup_{p \in C_{\text{Ret}}(q)} C_{\text{LLM}}(q,p) | A^C) \Pr(A^C)$$

$$\leq \Pr(A) + \Pr(r^* \notin \cup_{p \in C_{\text{Ret}}(q)} C_{\text{LLM}}(q,p) | A^C) \Pr(A^C)$$

$$\leq \Pr(A) + \Pr(r^* \notin C_{\text{LLM}}(q,p^*))$$

$$\leq \alpha_{\text{Ret}} + \alpha_{\text{LLM}} = \alpha.$$ 

\[ \square \]
B.1 PAC Prediction Set Construction

To construct prediction sets with probably approximately correct (PAC) guarantees, we use the same nonconformity measures states in 3.2 for retrieval and LLM tasks, respectively. Also, we will assign the error budgets $\alpha_{\text{Ret}}$ and $\alpha_{\text{LLM}}$ with $\alpha_{\text{Ret}} + \alpha_{\text{LLM}} = \alpha$. Additionally, we need to specify confidence levels for PAC prediction set. In our work, we specify $1 - \frac{\delta}{2}$ to the retriever and LLM PAC prediction set. Then, we have the following Corollaries:

**Lemma 4.1.** Suppose the questions and their corresponding most relevant passage $p^*$’s are subject to the distribution $D_{\text{passage}}$. Given the error budget $\alpha_{\text{Ret}}$ and confidence level $1 - \frac{\delta}{2}$, the constructed retriever sets satisfy the following inequality:

$$
\Pr_{B \sim D_{\text{passage}}} \left[ \Pr_{(q,p^*) \sim D_{\text{passage}}} (p^* \in C_{\text{Ret}}(q)) \geq 1 - \alpha_{\text{Ret}} \right] \geq 1 - \frac{\delta}{2}.
$$

(6)

**Lemma 4.2.** Suppose the questions, their corresponding most relevant passage $p^*$’s, and semantically correct responses $r^*$ are subject to the distribution $D_{\text{Response}}$. Given the error budget $\alpha_{\text{LLM}}$ and confidence level $1 - \frac{\delta}{2}$, if Assumption I.I.D and Assumption LLM Correctness hold, the LLM sets using PAC prediction set satisfy the following inequality:

$$
\Pr_{B \sim D_{\text{Response}}} \left[ \Pr_{(q,p^*,r^*) \sim D_{\text{Response}}} (r^* \in C_{\text{LLM}}(q,p^*)) \geq 1 - \alpha_{\text{LLM}} \right] \geq 1 - \frac{\delta}{2}.
$$

(7)

**Theorem 5.** Suppose the questions $q$’s, and semantically correct responses $r^*$’s are subject to the distribution $D$; a user-specified error level $\alpha$ is given. By aggregating retriever sets with error budget $\alpha_{\text{Ret}}$ with LLM sets with error budget $\alpha_{\text{LLM}}$ and confidence levels $1 - \delta/2$, with $\alpha = \alpha_{\text{Ret}} + \alpha_{\text{LLM}}$, the aggregation sets satisfy the following inequality:

$$
\Pr_{B \sim D} \left[ \Pr_{(q,r^*) \sim D} (r^* \in C_{\text{Agg}}(q)) \geq 1 - \alpha \right] \geq 1 - \delta.
$$

**Proof of Theorem 5.** Given Lemmas 4.1 & 4.2 and $\alpha_{\text{Ret}} + \alpha_{\text{LLM}} = \alpha$, we can prove the end-to-end guarantee in the following way: the $1 - \alpha$ coverage guarantee can be proved as the proof of Theorem 3. The confidence bound holds $(1 - \delta)$ by taking a union bound over the outer probabilities of Equation (6) and (7). □
C Additional Results

C.1 Individual Coverage

Figure 10: Individual coverages on all datasets using GPT-3.5 (first row) and Llama-2 (second row).

C.2 Individual Coverage with More Random Seeds

Figure 11: Individual coverages on all Datasets using GPT-3.5 (first row) and Llama-2 (second row) with 20 random seeds.
C.3 End-to-end Coverages

Figure 12: End-to-end coverage considering only the most relevant passage on all datasets using GPT-3.5 (first row) and Llama-2 (second row).

C.4 End-to-end Coverages

Figure 13: End-to-end coverage considering all passages on all datasets using GPT-3.5 (first row) and Llama-2 (second row).
C.5 Performance

Most of the results are similar to those in Figure 6. The results on TriviaQA using Llama-2 have a relatively large prediction set size. This could be explained by the fact that the true scores on this task have a large variance. Therefore, the identified threshold $\tau_{\text{LLM}}$ was relatively low (as in Figure 15a compared to other tasks (as in Figure 15b).

![Graphs showing average prediction set sizes on all datasets using GPT-3.5 (first row) and Llama-2 (second row).](image1)

(a) Natural Question  
(b) TriviaQA  
(c) SQuAD-1  
(d) Natural Question  
(e) TriviaQA  
(f) SQuAD-1

Figure 14: Average prediction set sizes on all datasets using GPT-3.5 (first row) and Llama-2 (second row).

![Histograms showing true scores collected on TriviaQA and Natural Question using Llama-2.](image2)

(a) True Scores on TriviaQA using Llama-2  
(b) True Scores on Natural Question using Llama-2

Figure 15: True scores collected on TriviaQA and Natural Question using Llama-2.

C.6 Additional Qualitative Results

C.6.1 All Covered

As shown in the example below, when the first retrieved passage is sufficiently informative, the LLM can probably generate correct responses for the question. In this case, TRAQ and Bonf can also include semantically correct responses in the aggregated sets. Again, TRAQ included less semantic meanings than Bonf did.

Query: who plays zack and cody in the suite life
We use 4-bit QLoRA (Dettmers et al., 2023) to fine-tune the Llama-2 (Touvron et al., 2023b) models on Natural Question, TriviaQA, and SQuAD-1 datasets separately. The hyperparameters used for QLoRA are listed in Table 3; and the fine-tuning parameters are listed in Table 4.

Table 2: Average semantic counts using Llama-2.

<table>
<thead>
<tr>
<th>Task</th>
<th>Cov(%)</th>
<th>TRAQ</th>
<th>Bonf</th>
<th>TRAQ-P</th>
<th>Bonf-P</th>
</tr>
</thead>
<tbody>
<tr>
<td>NQ</td>
<td>50</td>
<td>4.8</td>
<td>5.0</td>
<td>6.5</td>
<td>6.6</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>6.1</td>
<td>6.1</td>
<td>8.3</td>
<td>8.5</td>
</tr>
<tr>
<td></td>
<td>70</td>
<td>8.0</td>
<td>7.9</td>
<td>10.6</td>
<td>10.7</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>10.6</td>
<td>10.7</td>
<td>13.5</td>
<td>14.7</td>
</tr>
<tr>
<td></td>
<td>90</td>
<td>14.2</td>
<td>15.6</td>
<td>21.5</td>
<td>25.0</td>
</tr>
<tr>
<td>Trivia</td>
<td>50</td>
<td>4.3</td>
<td>4.5</td>
<td>5.7</td>
<td>6.5</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>5.8</td>
<td>6.8</td>
<td>7.7</td>
<td>9.2</td>
</tr>
<tr>
<td></td>
<td>70</td>
<td>8.6</td>
<td>10.2</td>
<td>13.4</td>
<td>18.5</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>15.1</td>
<td>19.1</td>
<td>29.3</td>
<td>71.3</td>
</tr>
<tr>
<td></td>
<td>90</td>
<td>117</td>
<td>122</td>
<td>145</td>
<td>157</td>
</tr>
<tr>
<td>SQuAD1</td>
<td>50</td>
<td>4.5</td>
<td>5.1</td>
<td>5.2</td>
<td>6.4</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>5.7</td>
<td>6.5</td>
<td>6.9</td>
<td>7.7</td>
</tr>
<tr>
<td></td>
<td>70</td>
<td>7.6</td>
<td>8.1</td>
<td>8.6</td>
<td>10.2</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>9.5</td>
<td>11.4</td>
<td>11.8</td>
<td>14.4</td>
</tr>
<tr>
<td></td>
<td>90</td>
<td>15.1</td>
<td>18.0</td>
<td>21.9</td>
<td>23.7</td>
</tr>
</tbody>
</table>

True answer: ['Dylan and Cole Sprouse']

Standard: ('Dylan and Cole Sprouse', 'Dylan and Cole Sprouse."

TRAQ: {'Dylan and Cole Sprouse', 'Dylan Sprouse', 'Phill Lewis'}

Bonf: {'Dylan and Cole Sprouse', 'Cole Sprouse', 'Dylan Sprouse'}

C.7 Miscovered

If the first retrieved passage lacks information, the standard RAG pipeline may struggle to provide the correct answer. However, in such scenarios, TRAQ and Bonf can construct prediction sets that contain the correct response with high probability, with TRAQ constructing smaller prediction sets.

Query: who sang i love rock and roll original

True Answer: ['Alan Merrill']

Standard: {'Joan Jett'}


Bonf: {'Joan Jett', 'Elvis Presley', 'The Velvet Underground', 'Lou Reed', 'Joan Jett & the Blackhearts', 'Alan Merrill', 'Chuck Berry', 'Donna Summer', 'Bobby Vee', 'Buddy Holly', 'Kevin Johnson', 'Mac Davis', 'The original version of "I Love Rock and Roll" was sung by The Arrows.', 'The Runaways', 'The answer to the question is not provided in the given context.', 'The Runaways sang the original version of "I Love Rock and Roll".', 'Joan Jett and The Arrows'}

D Implementation Details

D.1 Llama-2 Fine-tune Hyperparameters

We use 4-bit QLoRA (Dettmers et al., 2023) to fine-tune the Llama-2 (Touvron et al., 2023b) models on Natural Question, TriviaQA, and SQuAD-1 datasets separately. The hyperparameters used for QLoRA are listed in Table 3; and the fine-tuning parameters are listed in Table 4.
Table 3: QLoRA hyperparameters.

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>r</td>
<td>64</td>
<td>alpha</td>
<td>16</td>
</tr>
<tr>
<td>dropout</td>
<td>0.1</td>
<td>precision</td>
<td>4bit</td>
</tr>
</tbody>
</table>

Table 4: Fine-tuning hyperparameters.

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>batch_size</td>
<td>16</td>
<td>learning rate</td>
<td>2e-4</td>
</tr>
<tr>
<td>weight_decay</td>
<td>0.001</td>
<td>lr scheduler</td>
<td>constant</td>
</tr>
<tr>
<td>warmup ratio</td>
<td>0.03</td>
<td>epoch</td>
<td>3</td>
</tr>
</tbody>
</table>

D.2 Fine-tune Dense Passage Retriever (DPR) on the Biomedical Dataset (BioASQ)

We collect our dataset for DPR fine-tuning by using the collection of all the passages mentioned in BioASQ as our knowledge corpus, resulting in 56,795 passages. Following the method in (Karpukhin et al., 2020a), we create negative contexts for each sample in BioASQ by first retrieving the top-20 passages; and labeling contexts that did not contain the golden answers as the negative passages. We then divide the original BioASQ dataset into training, validation, and testing sets, with 3,775, 471, and 469 data points, respectively.

We fine-tune the DPR model (Karpukhin et al., 2020a) using the Haystack framework (Haystack), adjusting key hyperparameters to epochs=5 and batch size=16. Other hyperparameters are left at their default values. To evaluate the performance of the fine-tuned DPR, we use hit rate, which is the rate of relevant passages included in the top k retrieved passages. With k set to 20, the fine-tuned DPR achieves hit rates of 77.2% on the training set, 72.8% on the validation set, and 75.7% on the testing set.

D.3 Different Prompts

**Zero-shot Prompt**

Answer the following question based on the given context; Answer the question shortly.

Question: {question}
Context: {context}
Answer:

**Few-shot Prompt**

Answer the following question based on the given context; Answer the question shortly.

Question: {question 1}
Context: {context 1}
Answer: {answer 1}

Question: {question 2}
Context: {context 2}
Answer: {answer 2}

Question: {question}
Context: {context}
Answer:
The Great Lakes do not meet the ocean.',
'The Great Lakes meet the ocean at the Saint Lawrence River.',
'The Great Lakes meet the ocean through the Saint Lawrence River.',
'The Great Lakes do not meet the ocean.',
'The Great Lakes do not directly meet the ocean.

There is no specific answer given in the provided context about where the Great Lakes meet the ocean.

Atlantic Ocean',
'Saint Lawrence River',
'The Great Lakes do not meet the ocean. They are primarily connected to the Atlantic Ocean through the Saint Lawrence River.',
'The Great Lakes do not meet the ocean. They connect to the Atlantic Ocean through the Saint Lawrence River.',
'The Great Lakes meet the ocean through the Saint Lawrence River.',
'They do not meet the ocean."

(a) End-to-end guarantee considering only the most relevant passage
(b) Overall coverage guarantee considering all passages
(c) Average prediction set sizes

Figure 16: Results using a few-shot prompt on Natural Question using GPT-3.5.

D.4 Main Packages

<table>
<thead>
<tr>
<th>Package</th>
<th>Version</th>
<th>Package</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>transformer</td>
<td>4.32.1</td>
<td>nltk</td>
<td>3.8.1</td>
</tr>
<tr>
<td>(Wolf et al., 2020)</td>
<td></td>
<td>spacy</td>
<td>3.6.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Honnibal and Montani, 2017)</td>
<td>0.1.2</td>
</tr>
<tr>
<td>rouge-score</td>
<td></td>
<td>scikit-optimize</td>
<td>0.9.0</td>
</tr>
<tr>
<td>(Lin, 2004)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

D.5 Artifact License and Terms

Our implementation is based on haystack, transformers and DPR (Karpukhin et al., 2020a). The first two are licensed under Apache License 2.0, the third is licensed under Attribution-NonCommercial 4.0 International. We used four datasets, namely BioASQ, Natural Question, TriviaQA, and SQuAD-1. BioASQ is licensed under the CC BY 2.5 license, Natural Question is under CC BY-SA 3.0 license, TriviaQA is under the Apache License 2.0, and SQuAD-1 is under the CC BY-SA 4.0 license. We used two LLMs, namely GPT-3.5 and Llama-2. GPT-3.5 usage is subject to OpenAI’s Sharing & Publication Policy and Usage Policies. Llama-2 is licensed under the Llama-2 Community License (Meta, 2023). Our implementation and the data collected are under the MIT License.

Our use of the existing artifacts is consistent with their original intended use. Our created artifacts intend to verify our proposed method in our submission, which is consistent with original access conditions.

E Removing Assumption LLM Correctness

In certain scenarios, even if the most pertinent passage is identified and given to the language understanding model (LLM), the LLM is still unable to answer the question with accurate answers. This could be due to a variety of reasons, such as the passage not being sufficiently specific or the LLM not being able to
extract enough information from the passage. If the LLM is unable to generate correct responses even when the most pertinent passage is provided, our guarantee regarding the LLM and end-to-end pipeline may not hold. This problem can be alleviated by annotating better passages or using more powerful LLMs.

To address the issue with existing datasets and language models, we offer the guarantee of claiming I do not know if the language model is unable to generate a correct response to a question and its most relevant passage. We collect questions and their most relevant passages, and also labels that indicate whether GPT-3.5 could generate a correct response. We then divided the dataset into training, validation, and testing sets, with 6,899, 1,725, and 1,725 data points, respectively. We label True if the language model could generate a correct response and False otherwise. We then train a BERT-based text classifier, which takes in the questions and their most relevant passages, and predicts whether GPT-3.5 can generate a correct response. We name the trained classifier Conf-Classifier. Surprisingly, the Conf-Classifier achieves an accuracy of 95% on the testing set. To provide guarantees, we apply conformal prediction to the outputs of the Conf-Classifier. We include I do not know in the LLM set if the constructed prediction set contained False.

To construct the calibration set, we collect estimated confidences on not being able to answer the question on input questions in which the LLM fails to generate the correct response. We denote these estimated confidences as \( \{s_1, \ldots, s_N\} \). Given a user-specified coverage level, we then use conformal prediction to identify the \( \frac{N+1}{N} (1-\alpha) \) quantile as the threshold \( \tau_{\text{Ign}} \) to construct the set. Given an input question \( q \), we then include I do not know in the aggregation set \( C_{\text{Agg}}(q) \) if the estimated confidence \( n_{K+1} \) is above \( \tau_{\text{Ign}} \). Then we can guarantee the following:

**Lemma 5.1.** Given an input question \( q \) that the LLM cannot correctly answer and a user-specified error level \( \alpha \), if \( \alpha_{\text{Ign}} \) is used to decide whether to include I do not know, the aggregation set satisfies the following property:

\[
\Pr_{q \sim D} [I \text{ do not know} \in C_{\text{Agg}}(q)]
\]

This result follows straightforwardly from Theorem I.I.D.

We validate our guarantee using five distinct random seeds and five different coverage levels. The results are shown in Figure 17. As the figure illustrates, our method can include I do not know at various required coverage levels. By combining this with our guarantee on the LLM, we can guarantee all questions.

**Theorem 6.** Given a user-specified error level \( \alpha \), if aggregation is constructed with error level \( \alpha \), the resulting prediction sets contain true answers (i.e. semantically correct responses if the input question is answerable; or I do not know if the input question is unanswerable) with probability at least \( 1 - \alpha \), i.e.

\[
\Pr_{q \sim D} [\text{True answer} \in C_{\text{Agg}}(q)] \geq 1 - \alpha.
\]

**Proof.** Suppose we construct the aggregation set and ignorance set both with coverage level \( 1 - \alpha \); then we have the following inequalities:

\[
\Pr_{q \sim D} [\text{True answer in the resulting set}]
= \Pr_{q \sim D} [\text{Correct response} \in C_{\text{Agg}}(q)] \times \Pr[q \text{ is answerable}]
+ \Pr_{q \sim D} [I \text{ do not know} \in C_{\text{Agg}}(q)] \times \Pr[q \text{ is unanswerable}]
\leq (1 - \alpha) \times \Pr[q \text{ is answerable}] + (1 - \alpha) \times \Pr[q \text{ is unanswerable}]
= 1 - \alpha.
\]

\( \square \)

### F AI Assistant Usage

We used Copilot to assist our coding.
Figure 17: Results on identifying whether a given prompt is answerable or not.

(a) Coverage Rate on *I do not know*.

(b) False Positive Rates (claiming *I do not know* but actually being able to answer).

(c) The distribution of confidence on claiming *I do not know* using the training classifier.