Inter-Passage Verification for Multi-evidence Multi-answer QA

Bingsen Chen[1,2], Shengjie Wang[1,2], Xi Ye[3], Chen Zhao[1,2]
[1]NYU Shanghai, [2] New York University [3]Princeton Language and Intelligence
{bale.chen, sw5973, cz1285}@nyu.edu, xi.ye@princeton.edu

Abstract

Multi-answer question answering (QA), where questions can have many valid answers, presents a significant challenge for existing retrieval-augmented generation-based QA systems, as these systems struggle to retrieve and then synthesize a large number of evidence passages. To tackle these challenges, we propose a new multi-answer QA framework – Retrieval-augmented Independent Reading with Inter-passage Verification (RI^{2}VER). Our framework retrieves a large set of passages and processes each passage individually to generate an initial high-recall but noisy answer set. Then we propose a new inter-passage verification pipeline that validates every candidate answer through (1) **Verification Question Generation**, (2) **Gathering Additional Evidence**, and (3) **Verification with inter-passage synthesis**. Evaluations on the QAMPARI and RoMQA datasets demonstrate that our framework significantly outperforms existing baselines across various model sizes, achieving an average F1 score improvement of 11.17%. Further analysis validates that our inter-passage verification pipeline enables our framework to be particularly beneficial for questions requiring multi-evidence synthesis.[1]

1 Introduction

Question Answering (QA) represents a long-standing challenge in natural language processing (Voorhees and Tice, 2000; Chen et al., 2017; Min et al., 2021a), which in recent years has become an important knowledge and factuality benchmark task of large language models (LLMs) (Wei et al., 2024). While most such QA benchmark datasets operate under the assumption that each question has a single answer, this assumption often fails to reflect real-world scenarios where multiple valid answers exist (Min et al., 2020; Amouyal et al., 2023). For instance, as shown in Figure 1, a legal expert assessing AI software in the EU must consider multiple applicable laws to ensure compliance, mitigate legal risks, and make informed decisions. Thus, developing QA systems capable of providing comprehensive information is crucial.

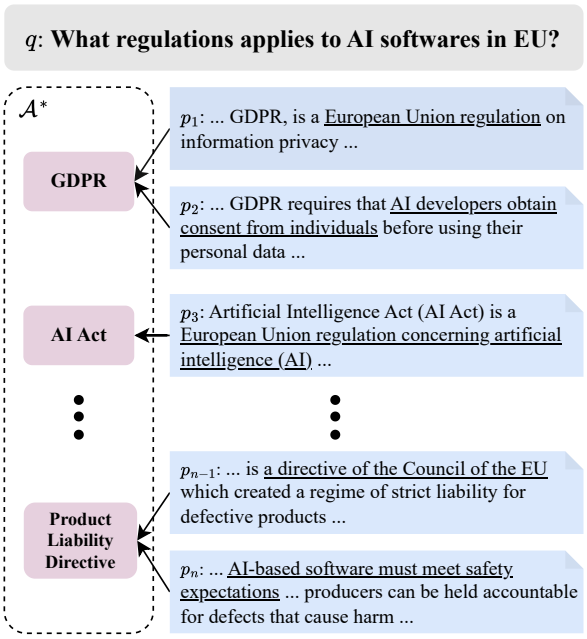

Figure 1: A representative example of a multi-evidence multi-answer question. Note that answers like "GDPR" and "Product Liability Directive" require multiple evidence passages to be supported.

Nevertheless, existing QA systems still fall short in providing multiple answers reliably (Min et al., 2020; Amouyal et al., 2023; Zhong et al., 2022). State-of-the-art LLMs, relying solely on their parametric knowledge, remain incompetent for multi-answer QA, as LLMs are prone to hallucination (Zhang et al., 2023; Ji et al., 2022) and possess limited knowledge of long-tail entities (Kandpal et al., 2022; Sun et al., 2024). A typical solution is to employ the retrieval-augmented generation (RAG) paradigm (Lewis et al., 2020; Guu

[1]Our code is available at https://github.com/BaleChen/RIVER

Findings of the Association for Computational Linguistics: ACL 2025, pages 6811–6829
July 27 - August 1, 2025

et al., 2020). In such systems, a retriever module first searches for supporting passages from an external corpus based on the query, followed by the LLM that reads the passages to produce answers. However, even with external knowledge, multi-answer QA remains challenging for LLMs: First, to find all answers, LLMs need to parse information from a large number of passages. For instance, the question "Who played midfielder for Argentina?" has 98 correct answers, which are likely scattered across dozens of passages. Moreover, finding an answer often requires reasoning across multiple passages. For instance, in Figure 1, to correctly identify "GDPR" as an answer requires the LLM to synthesize two passages—one mentioning "GDPR is an EU regulation" and another stating "GDPR requires AI developers to obtain user consent" – among all other passages irrelevant to GDPR. This multi-evidence synthesis capabilities over a large amount of contextual information have been recognized as a major limitation even for modern long-context LLMs (Liu et al., 2024; Wang et al., 2024; Bai et al., 2025; Ye et al., 2025). Hence, how to design QA systems with LLMs to solve those multi-evidence and multi-answer questions remains an interesting yet underexplored problem.

In this paper, we propose a new framework for multi-evidence multi-answer QA – **R**etrieval-augmented **I**ndependent Reading with **I**nter-passage **VER**ification (RI^{2}VER). We first adopt off-the-shelf retrievers to find a large set of passages to ensure sufficient coverage. Next, an LLM processes each passage independently to extract all potentially correct candidate answers. This process avoids the challenge of synthesizing and reasoning over a large collection of passages but instead introduces a noisy answer candidate set, achieving high coverage of correct answers. We then propose the inter-passage verification (IPV) pipeline to effectively filter the noisy answer set, which is composed of three steps: (1) **Verification Question Generation** that leverages an LLM to generate verification questions about each atomic fact involved in the input question. (2) **Gathering Additional Evidence** that collects evidence passages for each specific verification question. (3) **Verification with multi-evidence synthesis** that adopts an LLM to synthesize evidence passages and determine the correctness of the candidate answers on each verification question. Through this process, we filter out candidate answers that fail the verification steps, resulting in a refined answer set with high precision.

We evaluated our framework on QAMPARI and RoMQA, two challenging multi-answer QA datasets characterized by large answer set sizes and complex questions that require multi-evidence synthesis. RI^{2}VER significantly outperforms baseline RAG approaches with on average an 11.17% F1 score enhancement. Through ablation studies, we further validate the effectiveness of IPV, especially for the complex subsets of questions that require multi-evidence synthesis. Our manual error analysis reveals that more than half of the prediction errors stem from suboptimal annotations. Therefore, we argue that, in addition to developing better multi-answer QA systems, ensuring more accurate answer annotations is equally crucial for the community.

2 Background: Multi-Answer QA

We first formally define the multi-answer QA task. Given a question q and a text corpus of numerous passages $\mathcal{C} = \{p_i\}_{i=1}^{|C|}$, a QA system predicts a set of answers $\mathcal{A} = (a_1, a_2, \dots)$ as close as the ground truth answer set $\mathcal{A}^* = (a_1^*, a_2^*, \dots)$, where each answer a_i is a short-form answer (e.g., an entity) to q. Each ground truth answer can have one or more supporting passages in $\mathcal{C}$, with the latter case being multi-evidence multi-answer QA.

The predicted answer set $\mathcal{A}$ is evaluated by set precision, recall, and F1 with respect to $\mathcal{A}^*$. Specifically, for each question, if a predicted answer a_i exactly matches one of the ground truth answers a_j^* or one of a_j^*'s aliases, we count it as a true positive answer, or otherwise a false positive answer. If there exists an a_i^* such that neither itself nor its aliases are present in $\mathcal{A}$, we count it as a false negative error. Thus, a QA system should balance between finding all correct answers (high recall) and ensuring the predicted answers are accurate (high precision) to achieve the optimal F1 score.

3 Method

In this section, we introduce our multi-answer QA framework, RI^{2}VER, that introduces the Inter-passage Verification (IPV) stage to the vanilla RAG paradigm for multi-answer QA. At a high level, RI^{2}VER first finds a large pool of passages and predicts a set of answers grounded on each passage. This retrieve-then-independently-read process prioritizes the recall of correct answers at a cost of precision, leading to many false answers in the predicted answer set. Then we filter each answer

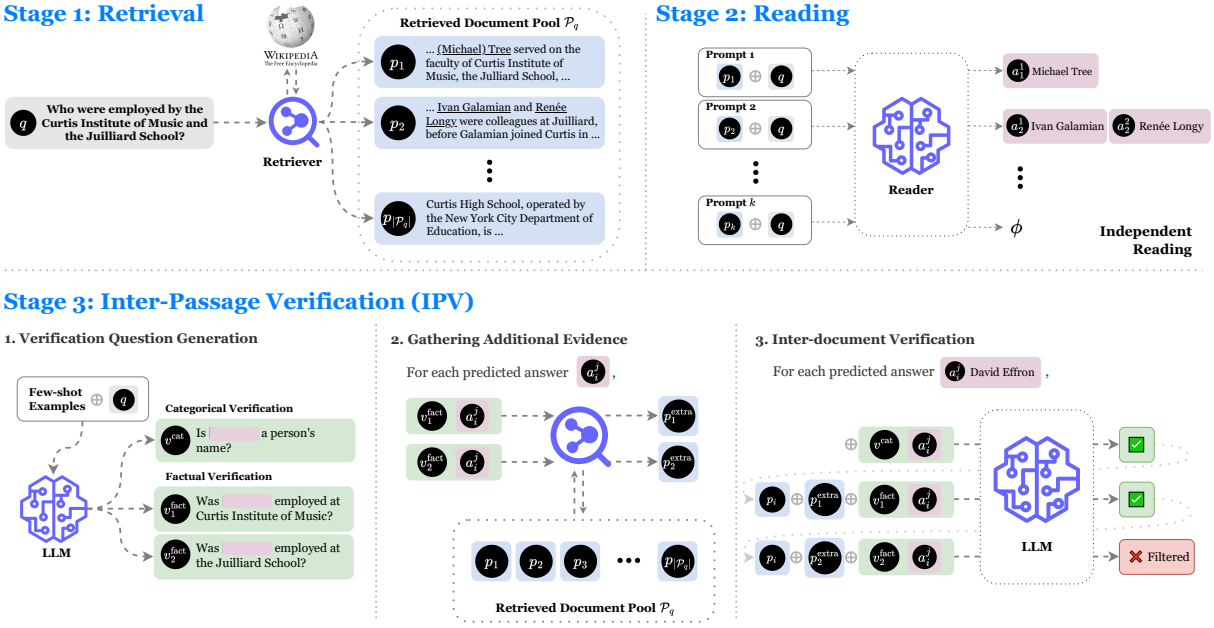

Figure 2: **Illustration of RI^{2}VER.** In Stage 1, we search broadly for a large pool of passages to cover sufficient evidence. In Stage 2, a reader model processes each top-k retrieved passage independently to generate a set of answer candidates. In Stage 3, we employ the IPV pipeline to verify each answer against a series of verification questions generated based on the original question, along with an additional retrieval step to gather evidence.

candidate to improve precision with the proposed IPV pipeline.

As illustrated in Figure 2, RI^{2}VER is composed of three stages. First, in the retrieval stage, we broadly search for a large pool of passages. Second, in the reading stage, we employ an LLM reader with the independent reading paradigm to elicit all potential correct answers (§3.1). Last, in the third stage, we design the IPV pipeline to effectively verify each answer candidate through three steps: (1) generate verification questions based on the original question (§3.2), (2) gather additional evidence for verification by reusing the initial passage pool (§3.3), and (3) verify whether each answer candidate satisfies all verification questions, filtering the false positive answers (§3.4).

3.1 Retrieval-Augmented Independent Reading

RAG has demonstrated its ability to improve the language models' factuality in knowledge-intensive tasks (Guu et al., 2020; Lewis et al., 2020). In our framework, we adopted the retrieve-and-read paradigm with independent reading, which we illustrate in Figure 2 and formally describe as follows:

Retrieval In the retrieval stage, we use a sparse (Robertson and Zaragoza, 2009) or dense retriever (Karpukhin et al., 2020) to retrieve a pool of passages $\mathcal{P}_q = \arg\text{topK}_{p\in\mathcal{C}}\,\texttt{Retriever}(q,p)$ that are relevant to each question q from a text corpus $\mathcal{C}$. $\mathcal{P}_q$ is an ordered set of passages ranked by their relevance score predicted by $\texttt{Retriever}(q,p)$. In our RI^{2}VER framework, we enlarged $|\mathcal{P}_q|$ to obtain a pool of passages that ensures sufficient coverage of relevant passages about question q. Then, only the top-k passages from the pool, denoted as $\mathcal{P}_q^{\text{top-}k} = \mathcal{P}_q[:k]$ are used for the reading step, while the rest of $\mathcal{P}_q$ will be utilized in the subsequent IPV pipeline described in §3.2-§3.4.

Reading During the reading stage, a reader language model $\mathcal{M}$ generates answers to q based on the passages in $\mathcal{P}_q^{\text{top-}k}$. In our framework, we adopt *independent reading* where $\mathcal{M}$ reads each retrieved passage $p_i \in \mathcal{P}_q^{\text{top-}k}$ independently to generate answers. The reader model is instructed to generate all potentially correct candidate answers that are fully or partially supported by each passage and abstain if the passage is irrelevant to q. Formally, the reader model predicts a set of answers $\mathcal{M}(q,p_i) = \{a_i^1, a_i^2, \dots\}$ by reading each passage p_i, and we take the union of generated answers from all passages, $\mathcal{A} = \bigcup_{p\in\mathcal{P}_q^{\text{top-}k}} \mathcal{M}(q,p)$, to form the final prediction set. We denote a_i^j as the j-th answer generated from reading passage p_i.

3.2 Verification Question Generation

We first generate a set of verification questions for each question q. The main goal of this step is to decompose complex questions into atomic factual units (Press et al., 2023; Zhou et al., 2023) and validate the answer for each unit. Particularly, we prompt an LLM to decompose the input question q into a series of factual verification questions $\{v_l^{\text{fact}}\}_{l=1}^{L}$, with each question asking about an atomic factual unit. As shown in Figure 2, we decompose the input question into two sub-questions, "Was `[ANSWER]` employed at Curtis Institute of Music?" and "Was `[ANSWER]` employed at the Juilliard School?". Each verification question isolates a distinct aspect of the original query, which not only makes the subsequent evidence retrieval easier but also ensures that the answer candidate satisfies all aspects of the original query.

We additionally generate a categorical verification question v^{cat} for each input question that checks whether the answer is in the correct category. This design is based on the observation that LLM readers sometimes generate spurious answers that violate the category of the question (e.g. answering "Field Concert Hall" to "Who were employed by the Curtis Institute of Music and the Juilliard School?"). Both factual and categorical verification questions are generated with a placeholder ("`[ANSWER]`") that will later be populated by answer candidates.

3.3 Evidence Gathering

Next, we gather evidence passages for each verification question. Recall that for independent reading introduced in §3.1, each predicted answer a_i^j is derived from a passage p_i, which can be naturally used as evidence for verification. For factual verification, we conduct an extra retrieval step to obtain additional verification question-specific evidence. This extra retrieval aims to find evidence to verify the answers that require multiple evidence. For example in Figure 2, a_2^2 = "Renée Longy" is generated as a potential answer from independently reading p_2, but verifying whether "Renée Longy" also "was employed at the Curtis Institute of Music" requires additional passages that are specific to "Renée Longy" and "Curtis Institute of Music". Thus, in this evidence-gathering step, we fill in an answer a_i^j to the answer placeholder in v_l^{fact}, and used it as queries to retrieve k_{extra} passages from $\mathcal{P}_q$ using the same retriever from the retrieval stage, denoted as $\mathcal{P}_l^{\text{extra}}$.[2] By that means, categorical verification will be grounded on p_i, and factual verification will be based on $\{p_i\} \cup \mathcal{P}_l^{\text{extra}}$.

3.4 Answer Verification

Finally, we verify each candidate answer a_i^j given a sequence of verification questions v^{cat} and $\{v_l^{\text{fact}}\}_{l=1}^{L}$ as well as the corresponding evidence passages p_i or $\{p_i\} \cup \mathcal{P}_l^{\text{extra}}$. We prompt an LLM with each verification question concatenated with its corresponding evidence passage(s) and instruct the LLM only to respond "True" or "False"[3]. We compute the verification results of a_i^j by comparing the probability of generating "True" (p^+) with "False" (p^-) immediately after the input prompt, and outputs "True" if p^+ is larger than p^- otherwise "False". We retain an answer a_i^j if the outputs for all verification questions are "True".

4 Experiments

4.1 Datasets

We tested our framework on the following two multi-evidence multi-answer QA datasets. Statistics of both datasets are in Appendix A.3.

QAMPARI (Amouyal et al., 2023) is constructed from Wikipedia's knowledge graph and tables. The dataset is split into simple questions (e.g., What films did Stephen King write?), intersectional questions (e.g., What films have Stephen King as screenwriter and Kubrick as director?), and compositional questions (Who is the director of a film that has Stephen King as screenwriter?). We note that the latter two types typically require more than one supporting passage for each answer.

RoMQA (Zhong et al., 2022) aims to test the robustness of multi-answer QA systems. It clusters entity attributes from WikiData to form complex entity-seeking queries that have many answers (Avg. # = 108) and require synthesizing multiple pieces of evidence (e.g., Which members of the Royal Society received the Order of Australia but were not employed by the University of Oxford?).

4.2 Baselines

Besides the retrieve-and-read framework with **independent reading** described in Section 3.1, we also

[2] In practice, we first do categorical verification before gathering extra evidence passages for factual verification since we can skip the extra retrieval step if the answer candidate does not pass categorical verification.

[3] We allow for minor token variations following Guan et al.'s (2023) implementation.

		QAMPARI			RoMQA		
	Reader Model	Prec.	Rec.	F1 (Δ)	Prec.	Rec.	F1 (Δ)
Llatrieval	GPT-4o-mini	32.04	17.21	20.29	20.14	15.23	15.53
Close-book	70b	27.92	20.47	21.26	18.22	11.53	10.29
	GPT-4o	33.98	23.65	24.59	22.30	12.81	12.20
FiD	T5	36.55	28.30	30.24	-	-	-
Concat.†	8b	26.60	37.84	28.10	14.27	18.32	10.74
	70b	32.83	37.23	33.57	21.33	14.95	12.24
Indep.†	T5	12.94	32.50	16.72	-	-	-
	8b	10.24	69.54	16.28	6.54	44.19	8.39
	70b	27.80	66.46	35.42	12.90	40.17	14.40
RI^{2}VER	T5	26.14	28.75	24.89 (+8.17)	-	-	-
	8b	31.41	60.17	**36.33** (+20.05)	15.43	32.08	**15.83** (+7.44)
	70b	37.23	58.91	**40.70** (+5.28)	19.55	31.53	**19.24** (+4.84)

Table 1: **Main Results.** Precision, recall, and F1 scores are macro-averaged across all test samples. The Δ values represent the F1 score difference before and after applying IPV on top of Independent Reading. † Indep. = Independent Reading; Concat. = Concatenated Reading

compare our framework to the following baselines:
Llatrieval (Li et al., 2024a) is a RAG system with LLM verifiers that outperforms many strong retrievers and rerankers on QAMPARI. They leveraged LLMs to verify the retrieved passage set, progressively select passages, and iteratively refine the query to improve retrieval performance. GPT-4o-mini is used for both retrieval verification and generation in the pipeline.
Close-book QA We also compare our methods with GPT-4 Omni (OpenAI, 2024) and Llama-3.1-70b (Team, 2024) in a setting without access to external passages.
Fusion-in-Decoder (FiD) (Izacard and Grave, 2021) represents a retrieve-and-read baseline on QA tasks that process multiple retrieved passages in parallel through independent encoders before fusing their representations in the decoder for answer generation.
Concatenated Reading As an alternative to independent reading, we can instead concatenate all passages in $\mathcal{P}_q^{\text{top-}k}$ alongside the question q as one single prompt to the reader model. The reader model is instructed to produce a list of correct answers to q based on the passages.

4.3 Experiment Settings

For the retriever model, we ensemble BM25 and NV-Embed-v2[4] (Lee et al., 2025) with Reciprocal Rank Fusion (Cormack et al., 2009) for QAMPARI. On RoMQA, we only use NV-Embed-v2 for the retriever, as ensembling negatively impacts retrieval performance. We report and discuss the retriever Answer Recall@K (ARecall@K), which calculates the macro-averaged percentage of answer strings matched in the K retrieved passages, in Appendix A.1. We use the 2021-08-01 Wikipedia dump as the retrieval corpus $\mathcal{C}$, where each Wikipedia document is chunked into 100-word passages. We initially retrieve $|\mathcal{D}_q| = 1,000$ passages for each question q, and the reader model reads the top $k = 200$ passages. For the reader model, we use the publicly available Llama-3.1-8b-instruct and Llama-3.1-70b-instruct models, which, without ambiguity, we refer to as 8b and 70b. We also used Amouyal et al.'s (2023) T5-large checkpoints for FiD and independent reading, which are finetuned on QAMPARI and Natural Questions (Kwiatkowski et al., 2019), thus only reported on QAMPARI.

For our IPV pipeline, we first prompt 8b with few-shot examples to generate both the categorical and factual verification questions. Then, in the evidence-gathering step, we retrieve $k_{\text{extra}} = 1$ extra passage from the pool $\mathcal{D}_q$ for each factual verification question and discuss the impact of k_{extra} in Section 4.5. For answer verification, we use 8b as the verifier model to compute p^+ and p^-. All prompt templates can be found in Appendix A.4.

[4]#1 on MTEB English Retrieval Leaderboard. The rank is as of February 9, 2025, and is available here.

4.4 Main Results

Our framework substantially outperforms various baselines on both datasets. As shown in the lower section of Table 1, with RI^{2}VER, 8b and 70b reader models achieve an F1 score of 36.33% and 40.70% respectively on QAMPARI, which surpasses all other methods. Notably, RI^{2}VER outperforms LLMs when applied in a closed-book setting (e.g., 24.59% F1 for GPT-4o) by large margins. Moreover, when holding the retriever module constant, IPV also outperforms baseline retrieve-and-read methods, with the 8b model surpassing both concatenated (33.57% F1) and independent reading (35.42% F1) with the 70b model. We also observe consistent trends on the RoMQA dataset.

Independent reading achieves high recall but leads to low precision. Independent reading enables both 8b and 70b to elicit nearly all valid answers in the retrieved passages, as evidenced by their high answer set recall (69.54% and 66.46% on QAMPARI) approaching the retriever's RECALL@200 of 72.54%. However, especially for the 8b model, independent reading leads to low precision (10.24% on QAMPARI and 6.54% on RoMQA). This is likely because the 8b model is more susceptible to hallucinating on irrelevant passages when processing each passage individually, generating many false positive answers (examples in Appendix 8). Although other methods like concatenated reading, FiD, and Llatrieval strike a better precision-recall balance, we argue that, in those frameworks, the performance is bottlenecked by the limited number of true positive answers. Instead, reading passages independently provides a favorable high-recall candidate set in RI^{2}VER, since the subsequent IPV pipeline is dedicated to filtering false answers to improve precision.

The IPV pipeline significantly improves answer set precision while maintaining high recall. IPV effectively addresses the low precision incurred by independent reading. For QAMPARI, the precision increases by 21.17% for 8b and 9.43% for 70b. Similarly, on RoMQA, IPV improves precision by 8.89% for 8b and from 6.65% for 70b. These consistent improvements highlight IPV's ability to enhance the precision of RI^{2}VER, ensuring more accurate answer sets.

Figure 3 further demonstrates how IPV makes our framework more robust to irrelevant passages. As the dashed line indicates, 8b generates more false positive answers when reading more lower-ranked passages, leading to steeply decreasing precision. In contrast, IPV improves precision consistently over all cases and maintains relative stability as the model reads through $\mathcal{P}_q$, with the 8b model's precision decreasing much more gradually and the 70b model stabilizing around 36-38% even after reading all 1000 passages. Notably, while recall continues to improve with reading more passages and precision stabilizes, we achieve a significantly better F1 score. This finding indicates that IPV allows RI^{2}VER to benefit from the high recall of independently reading more passages without suffering from diminishing precision due to hallucinations.

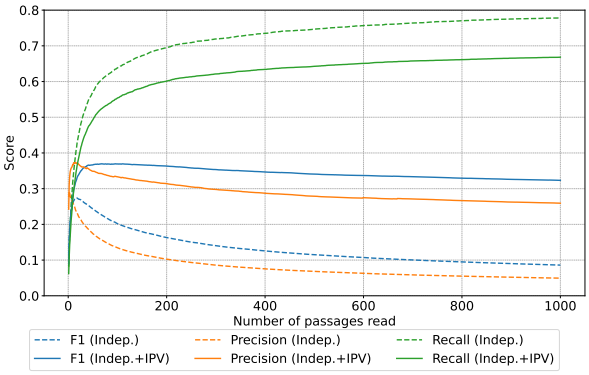

Figure 3: The prediction set precision, recall, F1 score on QAMPARI against the number of passages read by the 8b model before and after applying IPV.

	QAMPARI			RoMQA
	Simple	Inter.	Comp.	
IPV	25.13	53.92	41.96	15.83
w/o Extra retrieval	−1.64	−3.09	−3.10	−1.20
w/o Factual Veri.	−8.73	−24.12	−18.48	−5.75
w/o Category Veri.	−1.16	−0.74	−4.30	−0.63
Self-reflection	−1.21	−8.43	−6.45	−2.60

Table 2: **Ablation of IPV.** The reader model is controlled to be 8b and we ablate each component of the IPV pipeline. The F1 scores are reported as the difference from the full IPV pipeline. Inter. = Intersectional questions, Comp. = Compositional questions.

4.5 Ablation Studies

In this section, we discuss the design choices and the significance of evidence gathering and verification question generation in IPV.

Extra retrieval leads to more substantial improvements on complex questions that require evidence synthesis across passages. As shown in Table 2, we compare the F1 scores between our full IPV pipeline and a simplified version that per-

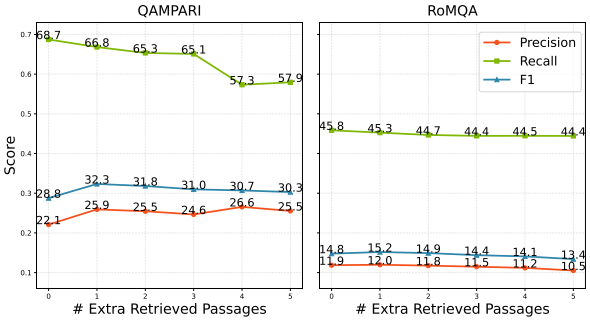

Figure 4: Precision, recall, F1 as a function of # passages retrieved during IPV's evidence gathering step.

forms verification without the additional retrieval step ("w/o Extra Retrieval") for evidence gathering. On QAMPARI, the results demonstrate that IPV's performance gains are particularly high for compositional and intersectional questions that require evidence from multiple passages to verify an answer. On RoMQA, where all questions require multiple evidence passages, IPV also shows superior results to verification with a single passage. Thus, the evidence-gathering step is especially helpful for multi-evidence, multi-answer questions.

Optimal evidence gathering requires just one additional retrieved passage. From Figure 4, we find that on both datasets, retrieving one extra passage results in better F1 scores than none or more. A potential reason is that verifying an answer might not require ≥ 3 passages, especially on QAMPARI, where the compositional and intersection questions require at most two evidence passages. Adding extra and potentially irrelevant passages distracts the model and hurts performance.

Both factual and categorical verification are necessary in IPV. We compare the contribution of factual and categorical questions in Table 2. We find removing either type of verification leads to performance degradation across all question types and datasets. Although most F1 improvement comes from factual verification questions, we find that checking the category of the answer first is still beneficial. We also ablate the whole verification question generation step and reduce the answer verification to a naive self-reflection with the original question ("Self-reflection"). This leads to a significant F1 decrease, particularly profound on the intersectional and compositional questions since those questions are more challenging to verify in one run.

	QAMPARI	RoMQA
Actual FP	38%	52%
Missing Annotation	12%	26%
Equivalent to a TP	13%	6%
Relatively Vague	5%	7%
Relatively Specific	32%	9%

Table 3: Human-annotated error types for 100 randomly sampled false positive predictions. More than half of the errors stem from suboptimal annotations.

	QAMPARI	RoMQA
Wrong Verification	12%	26%
Insufficient evidence	83%	73%
Annotation Error	5%	1%

Table 4: Human-annotated percentage of each error type for 100 randomly sampled true positives that are filtered after IPV.

5 Error Analysis

To better understand the performance of our framework RI^{2}VER, we conducted an error analysis to answer the following two questions:

What causes the low precision on both datasets? We randomly sample 100 false positive answers from the final predictions of 8b with IPV and manually label them into five categories shown in Table 3. In Appendix A.2, we show an example of each type to illustrate them better. We find that 62% on QAMPARI and 48% on RoMQA of the false positive errors are not actual false positives. There is a notable amount of missing annotations or aliases of golden answers, likely because Wikipedia may contain additional answers that are not well-annotated in Wikidata, echoing the ExtendedSet results in Amouyal et al. (2023). Furthermore, we observe many false positive errors that are mistaken due to the different granularity between prediction and ground truth answers (e.g., answering "electronics" or "iPhone 15" to "What does Apple produce?" when the annotation is "iPhone"). This issue in single-answer QA evaluation has been studied by Yona et al. (2024), and we advocate for future research on multi-answer QA benchmarks to annotate answers at multiple granularity levels. We also discuss the potential use of LLM-as-a-judge for evaluation in Appendix A.5.

Besides, we qualitatively inspected how the actual false positive answers are generated (Example

Question: Paul Heller was the producer to what film?
Passage: (Title: The Magic Sword) ...Walter R. Booth as director and Robert W. Paul as producer. Both had worked together before with the short movie of "A Railway Collision" in 1900. ...
Prediction: A Railway Collision

Table 5: A representative false positive prediction due to irrelevant context. The passage is about a different producer and film that is irrelevant to Paul Heller, while 8b hallucinates and predicts a false positive answer.

in Table 5 and more in Table 8). We find that they are commonly generated when the retrieved passage is irrelevant or partially relevant to the question, which concurs with the findings from Shi et al. (2023a). In short, a significant cause of the low precision is the difficulty in multi-answer QA annotation and evaluation of varying answer granularity; yet, we acknowledge that there is still headroom for future works on developing multi-answer QA systems that are more robust to irrelevant contexts.

Why does IPV lead to decreased recall? We also randomly sample 100 true positive answers that are filtered during the IPV process on both datasets. We categorize each of them into three causes: wrong verification, insufficient evidence, and annotation error. We find that most reduction in recall comes from insufficient evidence, which means the LLM reader model produces the answer based on either partial evidence in the retrieved passage or the LLM's internal knowledge. Since there is insufficient evidence, those answers are filtered during the IPV process, reducing the effectiveness of internal knowledge while also combating hallucinations. We hypothesize that IPV can further benefit from a more powerful retriever that can obtain more relevant passages both during initial retrieval and evidence gathering in IPV.

6 Related Works

6.1 Multi-answer QA

Many existing works on multi-answer QA focus on ambiguous questions where each interpretation can have a correct answer (Min et al., 2020; Stelmakh et al., 2022; Lee et al., 2024). For instance, several works focus on disambiguating questions and elicit more distinct and relevant answers (Gao et al., 2021; Sun et al., 2023b; Kim et al., 2023), while others tried retrieving a more diverse set of passages to cover more answers (Min et al., 2021b; In et al., 2025; Nandigam et al., 2022). Sun et al. (2023a) generates a database of disambiguated question-answer pairs to answer questions via retrieving from this database. RECTIFY (Shao and Huang, 2021) adopted a post-hoc verification pipeline, but they are limited in the scope of ambiguous questions and naive self-reflection for verification. Instead, our paper focuses on general entity-seeking questions in QAMPARI and RoMQA, which is also more challenging due to the larger answer set size and requirement for multi-passage synthesis. To our knowledge, there have not been previous works on such problems.

6.2 Retrieval Augmented Generation (RAG)

RAG is widely adopted in the era of LLMs since it can be directly applied to off-the-shelf LLMs or with minimal fine-tuning to reduce hallucination and augment the model's knowledge (Lewis et al., 2020; Guu et al., 2020; Ram et al., 2023; Shi et al., 2023b; Gao et al., 2023; Ye et al., 2024). Recent works have also explored how LLMs can in return enhance the performance of an RAG system, including refining queries (Ma et al., 2023; Chan et al., 2024), dealing with irrelevant context (Yoran et al., 2024; Asai et al., 2023), or extending to a more agentic framework that interleaves retrieval and generation (Jiang et al., 2023; Guan et al., 2025; Singh et al., 2025; Li et al., 2025). In this work, we extended the RAG framework with the IPV pipeline to solve the multi-answer QA task, significantly improving precision while maintaining high recall.

6.3 LLM for Fact Verification

Fact verification is a well-established task in natural language understanding (Thorne et al., 2018; Wadden et al., 2020; Honovich et al., 2022). With the rise of LLMs, several works have proposed to leverage LLMs' strong context understanding ability to verify factuality (Zeng and Gao, 2023; Guan et al., 2023; Tang et al., 2024). Notably, Guan et al. (2023) found that relatively small LLMs are surprisingly good at verifying facts despite being prone to hallucinating in a generation. Besides, to verify complex generation beyond a single claim in an open setting, previous works have explored decomposing generation into sub-claims and then using a retriever to gather evidence from reliable sources (Min et al., 2023; Zhang and Gao, 2023; Song et al., 2024). Our IPV pipeline borrows ideas from this line of work and effectively verifies a

large set of answer candidates with relatively small-scale LLMs.

7 Conclusion

In this paper, we propose a new multi-answer QA framework, RI^{2}VER, in which we retrieve broadly for relevant passages, extract all potential answers from each passage independently, and then refine the prediction set with the IPV pipeline that gathers inter-passage evidence to verify each answer candidate with an LLM. Our framework consistently outperforms various baselines across different model sizes and benchmarks. Through further analysis, we demonstrate that the IPV pipeline is beneficial in answering questions that require synthesizing multiple evidence passages. It also enables the reader model to benefit from the high recall from independently reading more passages without suffering from the increase in irrelevant information. We believe this work offers valuable insights and paves the way for developing more robust and reliable multi-answer QA systems.

Limitations

First, we did not study how the model's internal knowledge would affect the multi-answer QA performance, and instructed the model to only use evidence in context. As modern LLMs are increasingly knowledgeable, studying how to harmonize the parametric and retrieved knowledge (Cheng et al., 2024) in the context of multi-answer QA is an interesting future direction. Second, RI^{2}VER mainly improves the answer generation and verification part, leaving the retriever module constant. As we discussed in §5, we believe developing better retrievers in the context of multi-answer QA, potentially in the direction of diverse retrieval (Chen et al., 2022; Li et al., 2024b), is beneficial for our and future multi-answer QA systems. Third, the IPV pipeline involves an extra retrieval step and a verification step, which introduce inference delay. We further discuss the inference latency in Appendix A.7. Future work can explore efficient verification approaches, such as distilled retriever or verifier (Zhang et al., 2025; Hsieh et al., 2023). Lastly, the scope of our experiments is limited to Wikidata-based QA datasets and the Wikipedia corpus. Experimenting with other text corpora and developing domain-specific multi-answer QA frameworks are both promising avenues.

Acknowledgements

Bingsen Chen, Shengjie Wang and Chen Zhao were supported by Shanghai Frontiers Science Center of Artificial Intelligence and Deep Learning, NYU Shanghai. This work was supported in part through the NYU IT High Performance Computing resources, services, and staff expertise.

References

Samuel Amouyal, Tomer Wolfson, Ohad Rubin, Ori Yoran, Jonathan Herzig, and Jonathan Berant. 2023. QAMPARI: A benchmark for open-domain questions with many answers. In *Proceedings of the Third Workshop on Natural Language Generation, Evaluation, and Metrics*.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *Proceedings of the International Conference on Learning Representations*.

Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2025. Longbench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. *Preprint*, arXiv:2412.15204.

Chi-Min Chan, Chunpu Xu, Ruibin Yuan, Hongyin Luo, Wei Xue, Yike Guo, and Jie Fu. 2024. RQ-RAG: Learning to refine queries for retrieval augmented generation. In *Proceedings of the Conference on Language Modeling*.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proceedings of the Association for Computational Linguistics*.

Sihao Chen, Siyi Liu, Xander Uyttendaele, Yi Zhang, William Bruno, and Dan Roth. 2022. Design challenges for a multi-perspective search engine. In *Findings of the Association for Computational Linguistics*.

Sitao Cheng, Liangming Pan, Xunjian Yin, Xinyi Wang, and William Yang Wang. 2024. Understanding the interplay between parametric and contextual knowledge for large language models. *Preprint*, arXiv:2410.08414.

Gordon V. Cormack, Charles L A Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré,

Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The faiss library. *Preprint*, arXiv:2401.08281.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023. Enabling large language models to generate text with citations. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Yifan Gao, Henghui Zhu, Patrick Ng, Cicero Nogueira dos Santos, Zhiguo Wang, Feng Nan, Dejiao Zhang, Ramesh Nallapati, Andrew O. Arnold, and Bing Xiang. 2021. Answering ambiguous questions through generative evidence fusion and round-trip prediction. In *Proceedings of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing*.

Jian Guan, Jesse Dodge, David Wadden, Minlie Huang, and Hao Peng. 2023. Language models hallucinate, but may excel at fact verification. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*.

Xinyan Guan, Jiali Zeng, Fandong Meng, Chunlei Xin, Yaojie Lu, Hongyu Lin, Xianpei Han, Le Sun, and Jie Zhou. 2025. Deeprag: Thinking to retrieval step by step for large language models. *Preprint*, arXiv:2502.01142.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *Proceedings of the International Conference on Machine Learning*.

Or Honovich, Roee Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. 2022. TRUE: Re-evaluating factual consistency evaluation. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics*.

Yeonjun In, Sungchul Kim, Ryan A. Rossi, Mehrab Tanjim, Tong Yu, Ritwik Sinha, and Chanyoung Park. 2025. Diversify-verify-adapt: Efficient and robust retrieval-augmented ambiguous question answering. In *Proceedings of the Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics*.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Delong Chen, Wenliang Dai, Andrea Madotto, and Pascale Fung. 2022. Survey of hallucination in natural language generation. *ACM Computing Surveys*.

Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. *Preprint*, arXiv:2305.06983.

Nikhil Kandpal, H. Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2022. Large language models struggle to learn long-tail knowledge. In *Proceedings of the International Conference on Machine Learning*.

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 Conference on Empirical Methods in Natural Language Processing*.

Gangwoo Kim, Sungdong Kim, Byeongguk Jeon, Joonsuk Park, and Jaewoo Kang. 2023. Tree of clarifications: Answering ambiguous questions with retrieval-augmented large language models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

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

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM Symposium on Operating Systems Principles*.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2025. Nv-embed: Improved techniques for training llms as generalist embedding models. *Preprint*, arXiv:2405.17428.

Yoonsang Lee, Xi Ye, and Eunsol Choi. 2024. Ambigdocs: Reasoning across documents on different entities under the same name. In *Proceedings of the Conference on Language Modeling*.

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 *Proceedings of the Advances in Neural Information Processing Systems*.

Xiaonan Li, Changtai Zhu, Linyang Li, Zhangyue Yin, Tianxiang Sun, and Xipeng Qiu. 2024a. LLatrieval: LLM-verified retrieval for verifiable generation. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*.

Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. 2025. Search-o1: Agentic search-enhanced large reasoning models. *Preprint*, arXiv:2501.05366.

Zhicong Li, Jiahao Wang, Zhishu Jiang, Hangyu Mao, Zhongxia Chen, Jiazhen Du, Yuanxing Zhang, Fuzheng Zhang, Di Zhang, and Yong Liu. 2024b. Dmqr-rag: Diverse multi-query rewriting for rag. *Preprint*, arXiv:2411.13154.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting in retrieval-augmented large language models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Sewon Min, Jordan Boyd-Graber, Chris Alberti, Danqi Chen, Eunsol Choi, Michael Collins, Kelvin Guu, Hannaneh Hajishirzi, Kenton Lee, Jennimaria Palomaki, Colin Raffel, Adam Roberts, Tom Kwiatkowski, Patrick Lewis, Yuxiang Wu, Heinrich Küttler, Linqing Liu, Pasquale Minervini, Pontus Stenetorp, Sebastian Riedel, Sohee Yang, Minjoon Seo, Gautier Izacard, Fabio Petroni, Lucas Hosseini, Nicola De Cao, Edouard Grave, Ikuya Yamada, Sonse Shimaoka, Masatoshi Suzuki, Shumpei Miyawaki, Shun Sato, Ryo Takahashi, Jun Suzuki, Martin Fajcik, Martin Docekal, Karel Ondrej, Pavel Smrz, Hao Cheng, Yelong Shen, Xiaodong Liu, Pengcheng He, Weizhu Chen, Jianfeng Gao, Barlas Oguz, Xilun Chen, Vladimir Karpukhin, Stan Peshterliev, Dmytro Okhonko, Michael Schlichtkrull, Sonal Gupta, Yashar Mehdad, and Wen-tau Yih. 2021a. Neurips 2020 efficientqa competition: Systems, analyses and lessons learned. In *Proceedings of the NeurIPS 2020 Competition and Demonstration Track*.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Sewon Min, Kenton Lee, Ming-Wei Chang, Kristina Toutanova, and Hannaneh Hajishirzi. 2021b. Joint passage ranking for diverse multi-answer retrieval. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. AmbigQA: Answering ambiguous open-domain questions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Poojitha Nandigam, Nikhil Rayaprolu, and Manish Shrivastava. 2022. Diverse multi-answer retrieval with determinantal point processes. In *Proceedings of the 29th International Conference on Computational Linguistics*.

OpenAI. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics*.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends in Information Retrieval*.

Zhihong Shao and Minlie Huang. 2021. Answering open-domain multi-answer questions via a recall-then-verify framework. In *Proceedings of the Association for Computational Linguistics*.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H. Chi, Nathanael Scharli, and Denny Zhou. 2023a. Large language models can be easily distracted by irrelevant context. In *Proceedings of the International Conference on Machine Learning*.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen tau Yih. 2023b. Replug: Retrieval-augmented black-box language models. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*.

Aditi Singh, Abul Ehtesham, Saket Kumar, and Tala Talaei Khoei. 2025. Agentic retrieval-augmented generation: A survey on agentic rag.

Yixiao Song, Yekyung Kim, and Mohit Iyyer. 2024. VeriScore: Evaluating the factuality of verifiable claims in long-form text generation. In *Findings of the Association for Computational Linguistics*.

Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. 2022. ASQA: Factoid questions meet long-form answers. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Haitian Sun, William W. Cohen, and Ruslan Salakhutdinov. 2023a. Answering ambiguous questions with a database of questions, answers, and revisions. *Preprint*, arXiv:2308.08661.

Kai Sun, Yifan Xu, Hanwen Zha, Yue Liu, and Xin Luna Dong. 2024. Head-to-tail: How knowledgeable are large language models (LLMs)? A.K.A. will LLMs replace knowledge graphs? In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*.

Weiwei Sun, Hengyi Cai, Hongshen Chen, Pengjie Ren, Zhumin Chen, Maarten de Rijke, and Zhaochun Ren. 2023b. Answering ambiguous questions via iterative prompting. In *Proceedings of the Association for Computational Linguistics*.

Liyan Tang, Philippe Laban, and Greg Durrett. 2024. Minicheck: Efficient fact-checking of llms on grounding documents. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Meta Llama Team. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018. The fact extraction and VERification (FEVER) shared task. In *Proceedings of the First Workshop on Fact Extraction and VERification*.

Ellen M. Voorhees and Dawn M. Tice. 2000. Building a question answering test collection. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*.

David Wadden, Kyle Lo, Lucy Lu Wang, Shanchuan Lin, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Minzheng Wang, Longze Chen, Fu Cheng, Shengyi Liao, Xinghua Zhang, Bingli Wu, Haiyang Yu, Nan Xu, Lei Zhang, Run Luo, Yunshui Li, Min Yang, Fei Huang, and Yongbin Li. 2024. Leave no document behind: Benchmarking long-context LLMs with extended multi-doc QA. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. 2024. Measuring short-form factuality in large language models. *Preprint*, arXiv:2411.04368.

Xi Ye, Ruoxi Sun, Sercan Arik, and Tomas Pfister. 2024. Effective large language model adaptation for improved grounding and citation generation. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*.

Xi Ye, Fangcong Yin, Yinghui He, Joie Zhang, Howard Yen, Tianyu Gao, Greg Durrett, and Danqi Chen. 2025. Longproc: Benchmarking long-context language models on long procedural generation. *Preprint*, arXiv:2501.05414.

Gal Yona, Roee Aharoni, and Mor Geva. 2024. Narrowing the knowledge evaluation gap: Open-domain question answering with multi-granularity answers. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. 2024. Making retrieval-augmented language models robust to irrelevant context. In *Proceedings of the International Conference on Learning Representations*.

Fengzhu Zeng and Wei Gao. 2023. Prompt to be consistent is better than self-consistent? few-shot and zero-shot fact verification with pre-trained language models. In *Proceedings of the Association for Computational Linguistics*.

Dun Zhang, Jiacheng Li, Ziyang Zeng, and Fulong Wang. 2025. Jasper and stella: distillation of sota embedding models. *Preprint*, arXiv:2412.19048.

Xuan Zhang and Wei Gao. 2023. Towards llm-based fact verification on news claims with a hierarchical step-by-step prompting method. In *Proceedings of the International Joint Conference on Natural Language Processing*.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2023. Siren's song in the ai ocean: A survey on hallucination in large language models. *Preprint*, arXiv:2309.01219.

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

Victor Zhong, Weijia Shi, Wen tau Yih, and Luke Zettlemoyer. 2022. Romqa: A benchmark for robust, multi-evidence, multi-answer question answering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V Le, and Ed H. Chi. 2023. Least-to-most prompting enables complex reasoning in large language models. In *Proceedings of the International Conference on Learning Representations*.

A Appendix

A.1 Retrieval Performance

We evaluate the retriever performance using the ANSWER RECALL@K (ARECALL@K), which calculated the percentage of answer strings covered in the K retrieved passages macro-averaged across all test samples. In Figure 5, we observe that on QAMPARI, ensembling the sparse and dense retriever with reciprocal rank fusion produces significantly better retrieval performance, reaching 81.23% ARECALL@K. Although Amouyal et al. (2023) showed that BM25 is hard to beat on QAMPARI, we found that the state-of-the-art embedding model NV-Embed-v2 (Lee et al., 2025) has watched up and slightly outperformed BM25 when the number of retrieved passages is large ($|\mathcal{D}_q| > 400$). On RoMQA, however, we found fusing the two retrievers underperforms using NV-Embed-v2 by itself, likely because BM25 is lagging significantly behind NV-Embed-v2. Zhong et al. (2022) mentioned that they found dense retrievers perform significantly more poorly than BM25 on RoMQA, but we observe that modern embedding models have outperformed BM25 by a large margin. Comparing the two datasets, we also found that RoMQA is much harder for retrievers, potentially because the questions have many more answers than QAMPARI and because the questions contain more complex requirements (e.g. "Which members of the Royal Society received the Order of Australia, but were not employed by the University of Oxford?").

A.2 Examples for error analysis

For the false positive manual annotation, we sampled 100 false positive answers along with the corresponding question, retrieved passage, and ground truth answers for both datasets. A human annotator will verify whether this answer is indeed a false positive answer by checking whether it belongs to any of the 5 error types in Table 7. The annotator has full access to the Internet. For each answer, the annotator will spend at most 5 minutes gathering information anywhere to make a decision of the 5 categories. Usually, the decision can be made very quickly with just the retrieved passage, or a simple Google search. There are a few times when the annotator cannot find enough evidence to categorize a false positive answer, even with 5 minutes, we count those as actual FPs in the statistics.

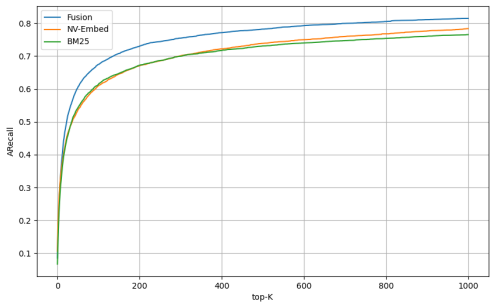

Figure 5: ARECALL@K of BM25, NV-Embed-v2, and the fusion of BM25 and NV-Embed-v2 on QAMPARI.

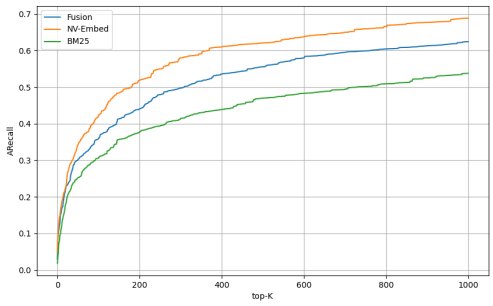

Figure 6: ARECALL@K of BM25, NV-Embed-v2, and the fusion of BM25 and NV-Embed-v2 on QAMPARI.

	QAMPARI				RoMQA
	Simple	Inter.	Comp.	All	
# Questions	500	200	300	1000	7000
Avg. # Answers	20	9	10	13	108

Table 6: Data statistics of QAMPARI and RoMQA.

A.3 Dataset statistics

Here we report the dataset statistics of QAMPARI and RoMQA in Table 6. For QAMPARI, we used their test split. As for RoMQA, we evaluated the development set.

A.4 Prompt Templates

We list all the prompt templates on the following pages. For concatenated and independent reading, the instruction part is almost the same except for some plural / single forms of words, so we show them together.

For verification question generation, we show the instructions as well as few-shot examples. Those exemplars are handpicked from the training set of both datasets. We formulate the few-shot prompting process as a multi-turn dialogue between the user and the assistant model. The role of

Error Type	Question	False Positive Prediction	Ground Truth Answers	Explanation
Actual FP	What work did Mel Brooks both write for and direct?	The Duchess and the Dirtwater Fox	Robin Hood: Men in Tights High Anxiety, History of the World, Part I, ...	The Duchess and the Dirtwater Fox is directed by Melvin Frank.
Missing Annotation	Who was a flute player who didn't die in Munich	Sam Most	Monty Waters, Johann Baptist Wendling, Theobald Boehm, ...	"Heavy Metal in Baghdad" is indeed a heavy metal film that is about a rock band. It does not exist in the ground truth set.
Equivalent to a TP	What items did Giovanola produce?	Mésoscaphe	Auguste Piccard, Goliath, ...	Mésoscaphe is a missing alias of Auguste Piccard, which refers to the same product that Giovanola produced.
Relatively Vague	What did Annibale de Gasparis discover?	Asteroids	24 Themis, 11 Parthenope, 16 Psyche, ...	Annibale de Gasparis did discover asteroids, but the annotators expect the names of specific asteroids
Relatively Specific	Which examination is administered by the College Board?	AP Statistics Exam	Advanced Placement exams, SAT, ...	AP statistics is one of the Advanced Placement exams, which is more granular than the ground-truth answers

Table 7: Examples for each type of false positive error annotation.

each message is denoted `<User>` or `<Assistant>` as prefixes. On RoMQA, the instruction is slightly different because some questions might have negative constraints (e.g., What highway system located in Tottori Prefecture is ***not maintained by Tottori Prefecture***). We instruct the model to generate the verification in positive form (e.g,. Is [answer] maintained by Tottori Prefecture?) because it is empirically beneficial for the extra retrieval step and verification performance. We also make the model generate a "tag" (NEGATION) so that when filtering the prediction set, we filter the answer candidates that appear true to this verification question.

A.5 Discussion on LLM-as-a-Judge for Evaluation

Due to the rigidity of exact match evaluation, many correct predictions are mistaken as incorrect, as stated in §5 and Appendix A.2. Some studies have started to use LLM-as-a-Judge (Zheng et al., 2023) to evaluate the correctness of LLM-predicted answers, allowing for format mismatch and some vocabulary variations. Since there have not been previous works on multi-answer QA that use LLM-as-a-Judge for evaluation metrics, we present a pilot implementation and analysis of our RI^{2}VER framework.

Implementation We prompted GPT-4o-mini to judge the predictions on QAMPARI from concatenated reading, independent reading, and RI^{2}VER, using Llama-3.1-8b-instruct as the reader model consistently. To judge each prediction, GPT-4o-mini is instructed to output the ground-truth answer index if the prediction semantically matches a ground truth, or output "None" otherwise. We then compute the set precision, recall, and F1 as usual and report in Table 9.

Analysis The LLM-as-a-judge evaluation confirms the robustness of our approach, with RI^{2}VER maintaining substantial performance gains over both baseline methods. Compared to exact match evaluation, all methods demonstrate improved precision and recall under LLM-based judgment, indicating that this evaluation paradigm could better capture semantic equivalence and handle format variations. Yet, further studies are needed to verify the reliability of LLM-as-a-Judge for multi-answer QA evaluation.

While this approach effectively addresses com-

Question	Retrieved Doc.	Prediction	Note
Who are Judge Memorial Catholic High School graduates?	(Title: **P. Thomas Thornbrugh**) Voters retained him again in 2016 with 61.04 percent approving. Personal. **Judge Thornbrugh** is married to Dr. Jean Thornbrugh, who is Dean of **the College for Working Adults at St. Gregory's University**. The couple have five adult children. Honors. **Thornbrugh was one of four graduates named as 2013 Distinguished Alumni by Emporia State University.**	P. Thomas Thornbrugh	This retrieved passage is totally irrelevant to Judge Memorial Catholic High School, but there is some lexical overlap ("Judge") and semantic overlap (graduating from a school).
The Arsenal F.C. won which year's FA Cup Final	(Title: **FA Cup Final**) **This occurred on 14 occasions, the last being in 1993 between Arsenal and Sheffield Wednesday.** In September 1998, the Football Association decided that all future finals would be decided "on the day", meaning that a penalty shootout would decide the winner if the score was level after normal and extra time. Two finals since have been decided by a penalty shootout, those of 2005 (Arsenal defeating Manchester United) and 2006 (Liverpool defeating West Ham United). Stan Mortensen's hat-trick for Blackpool in 1953 is the only hat trick ever scored at Wembley in the competition's final.	1993	The passage never mentioned Arsenal winning the 1993 FA Cup Final. It only said that Arsenal participated in it against Sheffield Wednesday.
A Tamil film starring Kamal Haasan.	(Title: Aruthu) **Aruthu is a 1976 Indian Malayalam-language film**, directed by Ravi. **The film stars Kamal Haasan**, Sumithra, M. G. Soman and Kaviyoor Ponnamma in the lead roles. The film has musical score by G. Devarajan. The film was a remake of the Hindi film K̈hel Khel Mein. Production. Äruthuf̈ilm produced by Som Prakash under production banner Sun Flower Productions. This film was shot in black-and-white. It was given an Ü(Unrestricted) certificate by the Central Board of Film Certification. The final length of the film was. Soundtrack. The music was composed by G. Devarajan and the lyrics were written by Yusufali Kechery.	Aruthu	The passage did mention that Aruthu features Kamal Haasan, but it is not a Tamil film.

Table 8: Examples for actual false positive predictions with the passage from which they are generated. Relevant information is highlighted in bold font.

	Exact Match			LLM-as-a-Judge		
	Prec.	Rec.	F1	Prec.	Rec.	F1
Concat.	26.60	37.84	28.10	30.49	46.25	31.98
Indep.	10.24	69.54	16.28	12.75	76.48	18.53
RI^{2}VER	31.41	60.17	**36.33**	36.74	68.14	**42.73**

Table 9: Performance comparison using Exact Match and LLM-as-a-Judge for answer correctness evaluation.

mon issues such as alias handling and format mismatches (e.g., "Equivalent to a TP" cases in Table 7), it cannot resolve the fundamental problem of incomplete ground-truth annotations present in both datasets. Moreover, the evaluation of answers with varying granularity levels presents ongoing challenges. For instance, when the ground truth is "iPhone" but the system predicts either "iPhone 15" (more specific) or "electronics" (more general), establishing clear correctness criteria for LLM judgment remains non-trivial.

We identify the development of refined evaluation for multi-answer QA systems as a promising direction for future research, particularly in establishing principled approaches for handling semantic granularity and annotation completeness.

A.6 Computational Resources

All our experiments are conducted on high-performance computing clusters. We used 1 NVIDIA A100 GPU for Llama-3.1-8b-instruct and 4 NVIDIA A100 GPUs for Llama-3.1-70b-instruct, using the vLLM (Kwon et al., 2023) framework to speed up inference in our experiments. For retrieval, we used FAISS (Douze et al., 2024) for efficient vector search on the large RAM (512G) CPU nodes.

A.7 Inference Latency Analysis

While our RI^{2}VER framework leads to notable performance improvements, it also brings about

additional inference latency. On the same computation node with 1 NVIDIA A100 GPU, we report the averaged inference wall-clock time (seconds) per question on QAMPARI with Llama-3.1-8b-instruct.

Method	Inference Time (s)
Concatenated Reading	9.39
Independent Reading	6.80
RI^{2}VER	32.90

Table 10: Inference time comparison of baselines and RI^{2}VER.

The inference latency is mainly caused by the additional evidence gathering and the additional forward passes through the LLM to verify each answer. The former requires encoding k very short verification questions and a $(1000, d) \times (d, k)$ matrix multiplication, where the number of verification questions k is as small as 1 or 2, and the embedding size d is 4096 in our setting. The latter step requires multiple forward passes, as many as the number of answer candidates multiplied by the number of factual verification questions. To further mitigate such latency, we discuss two potential strategies below:

Retrieval Future studies can investigate using a more light-weight embedding model for the retrieval, both in the sense of a smaller embedding dimension and fewer model parameters. As research on embedding models advances, more lightweight yet performant models will improve the inference latency of our framework. Besides, embedding quantization can further squeeze the inference latency of the matrix multiplication for evidence gathering.

Verification A more lightweight model, such as a distilled fact verification model, can potentially speed up the subsequent verification step significantly. Since the most significant bottleneck of IPV is the LLM forward passes to verify each answer candidate, a natural way to speed it up is to use a smaller model for such work. As MiniCheck (Tang et al., 2024) has shown, a similar capability can be distilled into a 770M-sized model, which means it is a viable and promising direction to accelerate the IPV pipeline with a smaller answer verifier.

A.8 Model Details

We here provide the specific model checkpoints used in our paper for reproduction purposes:

1. Llama-3.1-8b-Instruct: https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct
2. Llama-3.1-70b-Instruct: https://huggingface.co/meta-llama/Llama-3.1-70B-Instruct
3. NV-Embed-v2: https://huggingface.co/nvidia/NV-Embed-v2
4. T5: https://github.com/samsam3232/qampari/tree/master/models
5. GPT-4o: gpt-4o-2024-08-06
6. GPT-4o-mini: gpt-4o-mini-2024-07-18

Concatenated / Independent Reading

```
Read the following snippet(s) from Wikipedia documents carefully to find all
the correct answers to the question. There could be multiple answers, one
answer, or no answer. Do NOT generate additional descriptions/aliases/
explanations! Generate the answers in the form of an unordered list as
required below.

Start each answer with an asterisk and end with a new line. Your response
should be in the following format if there are correct answers supported by
the document snippets:

* Answer 1
* Answer 2
...

Or, if there is no correct answer, respond literally "There is no answer."
without generating an unordered list.

REMINDERS:

* Please answer the question PURELY based on the documents provided. DO NOT
use any additional knowledge not present in the documents.
* If the document is irrelevant to the question, do NOT use your own knowledge
 to answer it and just say "There is no answer".
* You should either give a list of answers as required above, or say "There is
 no answer". No other forms of response are accepted.

Document Snippet(s):
{document(s)}

Question: {question}
```

Closebook

```
Given a multi-answer web search question, generate ALL the answers that you
know to this question. Do NOT generate additional descriptions/aliases/
explanations! Please generate the answers in the form of an unordered list as
required below:

Start each answer with an asterisk and end with a new line. Your response
should look like this:
* Answer 1
* Answer 2
...

Now answer this question: {question}
```

Verification Question Generation (QAMPARI)

```
<User>: I am trying to answer questions that have many correct answers. I now
have a set of answers for each question but some of them are incorrect and
noisy. So, based on my web search questions that have many correct answers,
generate 2-3 verification questions so that I can use those questions to
filter my answer sets. Verification questions should always be true-or-false
questions. I will retain the answer item if an answer item appears true to the
 question and filter out the answer items otherwise. Here are some
requirements for your response:
* You should always first ask an easy category-checking question to verify
whether the answer entity belongs to the correct category.
* The second or third question should purely be based on my original web
```

```
search query. NO questions shall use inferred facts or external knowledge.
* Start by thinking about what questions are suitable to ask. Begin your
thought with "Thought: "
* After your thought, begin asking questions after "Verification Questions: ".
 Start each question with a new line and an asterisk. Do NOT generate anything
 else after the questions.
* In the verification question, when you refer to the answer item, please
leave it as [answer] and enclose it with double quotes so that I can later
fill it in with items in my answer set.

<User>: My web search question: Which album has John Reuben as performer?
<Assistant>:
Thought: To verify if an answer item is a correct album with John Reuben as
the performer, the first question should confirm whether the item belongs to
the category of music albums. The subsequent question should directly relate
to the web search query by asking if John Reuben is credited as the performer
on the album.
Verification Questions:
* Is "[answer]" a music album?
* Is John Reuben credited as a performer on the music album "[answer]"?

<User>: My web search question: What film was directed by Radha Mohan and
produced by Prakash Raj?
<Assistant>:
Thought: To filter answers effectively, the first question should confirm that
 the answer is a film. The second question will verify if the film was
directed by Radha Mohan. The third question will confirm if the film was
produced by Prakash Raj.
Verification Questions:
* Is "[answer]" a film?
* Was the film "[answer]" directed by Radha Mohan?
* Was the film "[answer]" produced by Prakash Raj?

<User>: My web search question: Who was director of a movie penned by
Christina Hodson?
<Assistant>:
Thought: To filter answers effectively, the first question should confirm that
 the answer is a person. The second question will verify if there is a movie
written by Christina Hodson and directed by this person.
Verification Questions:
* Is "[answer]" a director's name?
* Is there a movie that is directed by "[answer]" and written by Christina
Hodson?

User: My web search question: {question}
```

Verification Question Generation (RoMQA)

```
<User> I am trying to ...

(omitted because it's the same as the instruction on QAMPARI)

... so that I can later fill it in with items in my answer set.
* When you try to verify whether an answer does NOT have certain properties or
 features, ask the question in positive form and label the question with a [
NEGATION] tag at the end of the question. I will retain the answers that
appear false to the tagged question.

<User> My web search question: What highway system located in Tottori
Prefecture is not maintained by Tottori Prefecture
<Assistant> Thought: To verify if an answer item is a correct highway system
located in Tottori Prefecture but not maintained by Tottori Prefecture, the
first question should confirm whether the item belongs to the category of
highway system. The subsequent questions should check whether this highway
system is located in the Tottori Prefecture and meanwhile not maintained by
Tottori Prefecture. I will use the [NEGATION] tag to filter out the highway
```

```
systems maintained by the Tottori Prefecture.
Verification Questions:
* Is "[answer]" a highway system?
* Is the highway system "[answer]" located in Tottori Prefecture?
* Is the highway system "[answer]" maintained by Tottori Prefecture? [NEGATION
]

<User> My web search question: Which Sunni Islam figures weren't Sufist?
<Assistant> Thought: To filter answers effectively, the first question should
confirm that the answer is a Sunni Islam figure. Then, the second question
will verify whether this Sunni Islam figure was Sufist, which I will tag it
with [NEGATION].
Verification Questions:
* Is "[answer]" a Sunni Islam figure?
* Was the Sunni Islam figure "[answer]" Sufist? [NEGATION]

<User>My web search question: People who played for the Sheffield Wednesday F.
C. and the Lincoln City F.C.
<Assistant> Thought: To filter answers effectively, the first question should
confirm that the answer is a person's name. The second question will verify if
 this person played for Sheffield Wednesday F.C. The third question will check
 if this person also played for Lincoln City F.C.
Verification Questions:
* Is "[answer]" a person's name?
* Has "[answer]" ever played for the Sheffield Wednesday F.C.?
* Has "[answer]" ever played for the Lincoln City F.C.?

<User> My web search question: {question}
```

Verification

```
Read the following document(s) carefully to answer the true-or-false question
below. If the document provided is irrelevant or insufficient, then answer "
False". Answer "True" if there is sufficient evidence in the document. Do not
add additional description or explanation, and the answer can only be "True"
or "False". Begin your response with "Answer: ...".

{documents}

Question: {question}

Answer:
```