CONSTRAINTRAG: Constraint-Violation Diagnostics for Hard-Negative Filtering in NQ RAG

UNIST CSE402 Default Project

Jeonghoon Park
Department of Computer Science & Engineering
UNIST
hoonably@unist.ac.kr

Abstract

Retrieval-augmented generation (RAG) can fail when topically relevant passages do not satisfy the constraints in a question. This work studies this failure mode in open-domain QA and introduces CONSTRAINTRAG, a constraint-aware evidence selection method for Llama-3.2-1B-Instruct RAG. Given a question, CONSTRAINTRAG decomposes it into evidence checks, labels retrieved passages by whether they satisfy or violate those checks, and uses the labels for passage selection and answer-candidate generation. On 6,515 Natural Questions examples under a shared generation budget, CONSTRAINTRAG does not dominate exact-match baselines, but reveals a trade-off between span coverage, answer alignment, and concise output quality. Direct is competitive with BM25 and scalar-relevance filtering while improving ROUGE-L, First-candidate obtains the strongest concise ROUGE-L, and Recall reaches 0.4562 subspan exact match by preserving generated and evidence-derived candidates. The results suggest that constraint diagnostics are most useful as evidence-management and candidate-generation signals, especially when combined with a final answer verifier. The project page is available at: https://hoonably.github.io/cse-archive/constraint-rag.

1 Introduction

Large language models store substantial factual knowledge in their parameters, but parameter updates are expensive and can damage previously learned behavior. Retrieval-augmented generation (RAG) addresses this limitation by retrieving external evidence and conditioning generation on that evidence (Lewis et al., 2020). This study focuses on open-domain question answering on Natural Questions (Kwiatkowski et al., 2019), where a retriever supplies Wikipedia passages and a language model produces a short answer.

A central difficulty is that retrieval relevance is not identical to answer support. A top-ranked passage can match the topic of a question while missing a crucial qualifier such as a date, role, location, or negation. Such passages are hard negatives: they look useful, but can push the model toward a nearby wrong answer. This issue is related to QUEST, where entity-seeking queries often require satisfying multiple implicit constraints rather than matching only a single entity mention (Malaviya et al., 2023).

This work asks whether a lightweight constraint-violation diagnostic stage can improve RAG answer quality beyond BM25 rank or scalar LLM relevance alone. The proposed method, CONSTRAINTRAG, inserts question decomposition and passage-level constraint diagnostics between retrieval and generation. Rather than training a new retriever or collecting new labels, it uses the instruction-following model itself to identify whether retrieved passages satisfy, miss, contradict, or ignore the constraints implied by the question. The diagnostic stage therefore acts as an evidence-management layer: BM25 provides broad lexical recall, while the added labels expose which passages satisfy the constraints needed to answer the question.

The main finding is not a win on all metrics. After aligning generation and passage-selection budgets across Llama-based baselines, zero-shot Llama RAG obtains the highest concise subspan exact match because it often emits full-sentence outputs containing the gold span. A two-example few-shot prompt improves lexical overlap over zero-shot but not exact-match accuracy, so it serves as a prompt-control baseline. In contrast, CONSTRAINTRAG variants expose a design space: Direct is competitive with passage-filtering baselines and improves ROUGE, First-candidate produces the strongest lexical alignment with gold answers, and Recall shows that constraint-guided candidate generation recovers many answer spans that single-answer decoding misses. The contribution is evidence management, not a replacement for retrieval scoring.

2 Related Work

Retrieval-augmented generation. RAG combines parametric generation with non-parametric retrieval so that a model can answer knowledge-intensive questions using external evidence (Lewis et al., 2020). The experiments here follow a decoder-only RAG setting with prompt-based generation, while the original RAG formulation used an encoder-decoder model. Retrieval is based on the DPR-derived NQ-open setting (Karpukhin et al., 2020) and Pyserini BM25 over a Wikipedia passage index (Lin et al., 2021; Robertson and Zaragoza,

2009).

Open-domain question answering. Natural Questions contains real questions submitted to Google Search and annotated with short and long answers (Kwiatkowski et al., 2019). The experiments use the DPR-preprocessed NQ-open version with 58,880 training examples, 6,515 development examples, and a Wikipedia corpus of about 21 million passages. Because retrieved contexts include positive, negative, and hard-negative evidence, the setting is well suited for studying failures caused by plausible but constraint-mismatched passages.

Constraint-sensitive retrieval. QUEST studies entity-seeking retrieval queries involving implicit set operations such as intersection, union, and difference (Malaviya et al., 2023). A key lesson is that topical similarity can hide constraint failures: a passage may satisfy one condition while failing another. CONSTRAINTRAG transfers this idea to RAG evidence selection by prompting the generator to produce lightweight constraint diagnostics over retrieved passages, rather than training a new retriever or collecting new labels.

Small Transformer implementation. A small GPT-style model is also implemented as part of the complete experimental system. The model follows the Transformer decoder family (Vaswani et al., 2017) and includes grouped-query attention (Ainslie et al., 2023), rotary positional embeddings (Su et al., 2021), RMSNorm (Zhang and Sennrich, 2019), feed-forward layers, causal masking, and task-specific heads. These components are not the novel contribution, but they provide the required small-model baselines and the pretrained-vs.-scratch comparison.

3 Method

3.1 System Components

The small GPT model is a decoder-only Transformer with a GPT-2 tokenizer, hidden size 512, four layers, 16 attention heads, four key/value heads, head dimension 32, RoPE, RMSNorm, and a feed-forward hidden size of 2048. The model has approximately 64M trainable parameters. It is pretrained on 1,048,576 text blocks and evaluated on 8,096 held-out blocks before downstream finetuning.

The RAG system uses BM25 to retrieve passages, formats the retrieved evidence into prompts, and generates answers with either the finetuned small model or Llama-3.2-1B-Instruct (Grattafiori et al., 2024). The Llama-based experiments are the primary setting for evaluating the proposed evidence-selection mechanism because the small GPT-style model has very weak downstream QA performance.

Figure 1 summarizes the inference-time pipeline of CONSTRAINTRAG.

Inference procedure. The full inference-time procedure is intentionally simple and does not require training additional modules:

1. retrieve the top five BM25 passages for the input question;
2. decompose the question into a small set of evidence checks;
3. label each retrieved passage as satisfying, missing, contradicting, or being unrelated to each check;
4. combine BM25 rank with the diagnostic labels to choose the final evidence set; and
5. decode the selected evidence using one of the output modes described below.

This procedure separates two decisions that are often conflated in vanilla RAG: whether a passage is topically relevant, and whether it satisfies the particular constraints needed to answer the question.

3.2 Constraint Diagnostics

Let a question q imply a set of evidence requirements $C(q) = \{c_1, \ldots, c_m\}$. A retrieved passage p is useful only if it supports the requirements needed to answer q. CONSTRAINTRAG approximates this judgment with a prompted diagnostic stage. For each question, BM25 first retrieves the top $k = 5$ passages. Llama-3.2-1B-Instruct then decomposes the question into at most three short checks, such as answer type, entity, relation, date, location, or role. For each retrieved passage, the model assigns one label per check:

> `satisfied`, `missing`, `contradicted`, or `unrelated`.

Robust parsers handle malformed lists, code fences, alternate key names, and label aliases.

Given a passage diagnostic, the method counts the number of satisfied, missing, and contradicted checks as $S(p)$, $M(p)$, and $C(p)$. The diagnostic score is

$$d(p) = \beta S(p) - \gamma M(p) - \delta C(p), \tag{1}$$

and the final passage score combines the BM25 score $B(p)$ with the diagnostic score:

$$s(p) = \alpha B(p) + d(p). \tag{2}$$

The final configuration uses $\alpha = 1.0$, $\beta = 0.75$, $\gamma = 0.15$, and $\delta = 0.25$. The selector always preserves the top BM25 passage as a recall anchor, then fills the remaining selected slots with diagnostically strong passages and falls back to BM25 order when needed. Unless otherwise stated, the final selected-passage budget is `final_top_k` $= 3$.

This hybrid design reflects a trade-off between recall and constraint satisfaction. BM25 is reliable for broad recall but can rank plausible distractors highly, while diagnostic labels can demote passages that miss or contradict an implicit condition. Keeping the top BM25 passage as an anchor avoids discarding high-recall evidence entirely.

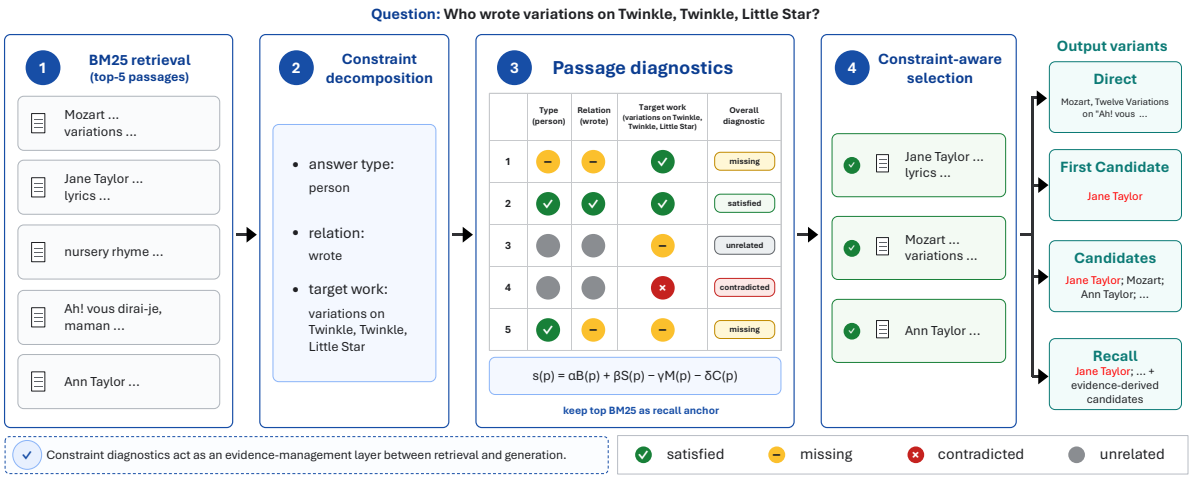

Figure 1: Overview of CONSTRAINTRAG. Given a question, the method retrieves BM25 passages, decomposes the question into evidence checks, diagnoses whether each passage satisfies or violates those checks, and selects evidence using both BM25 rank and constraint diagnostics. The selected evidence can then be decoded in different output modes. In this example, direct generation is still distracted by a plausible Mozart passage, while the candidate-oriented variants surface the gold answer span.

3.3 Output Variants

The diagnostic selector can be paired with several output formats. This distinction is important because the evaluation metric rewards different behaviors depending on output length.

Direct. This variant uses the diagnostic-selected passages but asks the generator for a direct short answer. It is the cleanest single-answer version of the method and is the most appropriate variant when a user-facing answer must be returned without an additional verifier.

First candidate. This variant prompts the model to produce a compact answer-candidate list and evaluates only the first generated candidate after post-processing. It often yields the most lexically aligned span, but it can lose exact-match accuracy when the first candidate is an alias, spelling variant, or near miss.

Candidate-set variants. The Candidates variant evaluates the generated candidate list rather than a single answer. The Recall variant preserves both generated candidates and evidence-derived candidates, such as dates and proper nouns extracted from selected passages. These variants are not concise final-answer systems; they estimate whether the diagnostic pipeline surfaces the gold answer somewhere in a candidate set. They are therefore useful for studying answer coverage and for motivating reranking or verification.

Because these variants share the same retrieval setting and diagnostic cache when possible, the ablations mainly isolate output formatting and candidate preservation rather than changes in the underlying evidence pool.

4 Experimental Setup

Data. All main RAG experiments use the full NQ development set of 6,515 examples. The small-model comparisons use the same task structure for pretrained and scratch checkpoints.

Metrics. Accuracy is normalized subspan exact match: a prediction is correct if any normalized gold answer appears as a subspan of the normalized prediction. This rewards answer coverage and can favor verbose outputs. ROUGE-1, ROUGE-2, and ROUGE-L (Lin, 2004) measure lexical overlap with the reference answer and penalize long candidate-list outputs.

Fair generation budget. All Llama-based concise RAG systems are evaluated with the same generation budget, `max_new_tokens` $= 32$. Passage-filtering baselines and comparable CONSTRAINTRAG variants use `final_top_k` $= 3$, avoiding a smaller or larger output budget than the baselines.

Baselines. The comparison includes five non-proposed systems. The finetuned small RAG baseline uses the implemented GPT-small model. Llama zero-shot RAG prompts Llama-3.2-1B-Instruct with retrieved evidence, while Llama few-shot RAG uses the same BM25 top-5 retrieval with two in-prompt QA examples. BM25FilteredRAG selects passages using only BM25 rank. RelevanceOnlyRAG uses Llama to assign scalar passage relevance, without constraint-wise diagnostics.

Implementation details. The final full-development Llama-based RAG evaluations were run on an NVIDIA H200 GPU. Runtime measurements are only approximate cost indicators because they vary with hardware, batching, and cache state.

Method	Accuracy	ROUGE-1	ROUGE-2	ROUGE-L
Finetuned small RAG	0.0009	0.0001	0.0000	0.0001
Llama zero-shot RAG	0.3727	0.1268	0.0650	0.1254
Llama few-shot RAG	0.3191	0.1912	0.1027	0.1896
BM25FilteredRAG	0.3288	0.2231	0.1259	0.2215
RelevanceOnlyRAG	0.3225	0.2199	0.1215	0.2179
CONSTRAINTRAG-DIRECT	0.3229	0.2333	0.1307	0.2321
CONSTRAINTRAG-FIRSTCANDIDATE	0.2809	**0.3212**	**0.1846**	**0.3205**
CONSTRAINTRAG-CANDIDATES	0.3863	0.1155	0.0472	0.1143
CONSTRAINTRAG-RECALL	**0.4562**	0.0296	0.0113	0.0289

Table 1: Main RAG results on the full NQ development set. The first seven rows evaluate concise or single-answer outputs, while the last two rows evaluate candidate-set outputs for answer coverage.

5 Results

5.1 Default Implementation Results

Figure 2 shows the GPT-small pretraining progress. Evaluation loss decreases and token accuracy improves throughout pretraining, indicating that the implementation learns a stable language-modeling objective. Table 2 then shows the small GPT-style downstream comparison. Pretraining gives small gains for classification and small-model RAG, but all downstream numbers are weak in absolute terms. These results support the decision to focus the proposed RAG extension on the Llama-based setting.

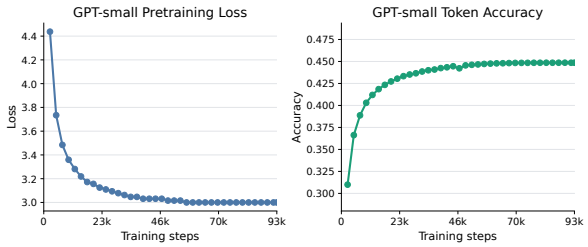

Figure 2: GPT-small pretraining progress. Evaluation loss decreases during pretraining and saturates near 3.00, while token accuracy reaches approximately 0.449.

Task	Pretrained	Scratch
Summarization ROUGE-1	0.0009	0.0013
Classification accuracy	0.0550	0.0471
Small RAG accuracy	0.0009	0.0005

Table 2: Downstream comparison with and without GPT-small pretraining.

5.2 Main RAG Comparison

Table 1 and Figure 3 present the full-development RAG results under the fair generation budget. The strongest concise subspan exact match is obtained by Llama zero-shot RAG (0.3727), which often emits full-sentence answers that contain the gold span. As a secondary prompt control, Llama few-shot RAG with two in-prompt QA examples reaches 0.3191 accuracy and improves ROUGE-L over zero-shot (0.1896 vs. 0.1254), but it does not close the exact-match gap. BM25FilteredRAG reaches 0.3288 accuracy, and RelevanceOnlyRAG reaches 0.3225.

The proposed variants behave differently. Direct reaches 0.3229 accuracy, almost identical to RelevanceOnlyRAG and slightly below BM25FilteredRAG, while improving ROUGE-L over both filtering baselines (0.2321 vs. 0.2215 and 0.2179). First-candidate has lower accuracy (0.2809), but it achieves the highest concise ROUGE-L (0.3205). This indicates that the first candidate is often lexically close to the reference answer, even when it is not always the exact normalized span required by subspan exact match.

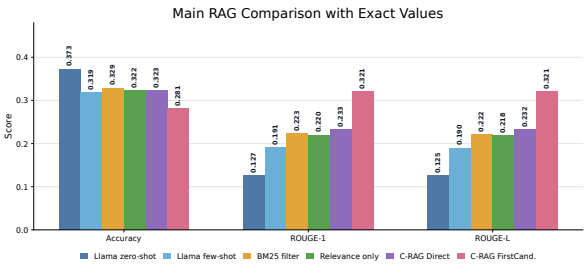

Figure 3: Main RAG comparison under the shared generation budget. Llama zero-shot RAG has the highest subspan exact-match accuracy, the two-shot prompt control improves ROUGE over zero-shot, and First-candidate has the strongest ROUGE among concise outputs.

The candidate-set variants show a separate behavior. Candidates reaches 0.3863 accuracy, and Recall reaches 0.4562 accuracy by preserving both generated and evidence-derived candidates. These scores should not be interpreted as clean short-answer performance because the outputs are longer candidate sets. Instead, they show that the diagnostic stage often surfaces the gold answer somewhere, even when the top candidate is not perfectly ranked.

Variant	Selected passages	Accuracy	ROUGE-1	ROUGE-2	ROUGE-L
Direct	3	0.3229	0.2333	0.1307	0.2321
Direct	2	0.2887	0.2222	0.1267	0.2215
First candidate	3	0.2809	0.3212	0.1846	0.3205
First candidate	2	0.2531	0.2933	0.1643	0.2924
Candidates	3	0.3863	0.1155	0.0472	0.1143
Candidates	2	0.3437	0.1047	0.0405	0.1037
Recall	3	0.4562	0.0296	0.0113	0.0289
Recall	2	0.4370	0.0285	0.0107	0.0278
Recall (w/o evidence)	3	0.3908	0.1088	0.0440	0.1075

Table 3: Output-format and selected-passage controls for CONSTRAINTRAG. Shaded rows denote the default selected-passage budget, `final_top_k=3`. Direct and First-candidate produce concise answers, Candidates preserves generated answer candidates, and Recall additionally preserves evidence-derived candidates.

5.3 Output-Format Analysis

Table 3 isolates output-format and selected-passage effects. Reducing the selected-passage budget from three to two lowers performance across the evaluated controls. For Direct, accuracy drops from 0.3229 to 0.2887 and ROUGE-L drops from 0.2321 to 0.2215. For First-candidate, accuracy drops from 0.2809 to 0.2531 and ROUGE-L drops from 0.3205 to 0.2924. The same trend appears for Candidates and Recall, indicating that the diagnostic selector benefits from retaining three passages rather than two.

The evidence-derived suffix is useful only when outputs are interpreted as candidate sets. Removing it from Recall lowers accuracy from 0.4562 to 0.3908, showing that extracted evidence candidates add coverage. Appendix B provides pairwise exact-match overlap among representative methods.

5.4 Why Accuracy and ROUGE Diverge

The main results are best understood through the difference between the evaluation metrics. The implemented accuracy metric is span-inclusive: if a long prediction contains a normalized gold answer anywhere, it is counted as correct. This benefits verbose outputs. For example, zero-shot Llama can generate full-sentence answers that include the correct span and therefore score well under subspan exact match, even when the answer is not a minimal string. The few-shot prompt moves in the opposite direction: its examples improve ROUGE by encouraging more answer-like outputs, but the shorter generations lose some span-inclusive exact-match hits.

ROUGE behaves differently. It rewards lexical overlap with the short gold answer and penalizes extra material. First-candidate often returns a compact answer string, which explains its strong ROUGE. However, the exact-match score can fall when the candidate is a semantically correct spelling variant, an alias, or a nearby date. For example, First-candidate may output "St. Lawrence River" while the gold string is "the Saint Lawrence River"; a direct answer that contains the full spelling can be counted as exact even if it is less concise. Similarly, candidate generation may include the correct answer in a later slot while the first candidate is wrong. This explains why Recall is much higher than First-candidate in accuracy but much lower in ROUGE.

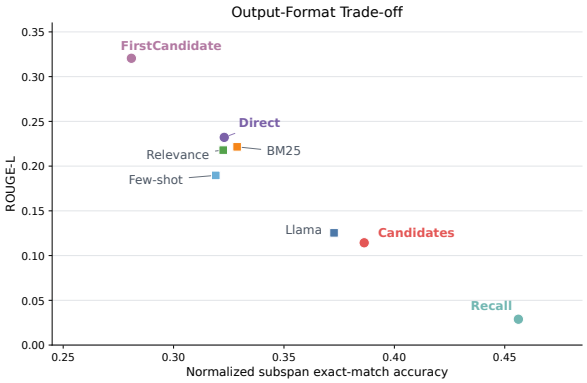

Figure 4: Trade-off between subspan exact-match accuracy and ROUGE-L across prompt baselines and output formats. Candidate-set outputs increase answer coverage, while First-candidate maximizes lexical alignment among concise outputs.

5.5 Robustness to Injected Retrieval Noise

Table 4 evaluates robustness under injected lower-ranked BM25 hits. This is a retrieval-noise stress test rather than a DPR hard-negative evaluation. RelevanceOnlyRAG obtains the highest concise accuracy under noise (0.3042). The Direct variant is close to BM25FilteredRAG in accuracy (0.2870 vs. 0.2881) and has higher ROUGE-L (0.2170 vs. 0.2002). The First-candidate noisy variant has lower accuracy (0.2407) but the strongest ROUGE-L (0.2809). Recall remains high in subspan accuracy (0.4284), showing that candidate coverage persists even when the retrieved set is noisier.

These results reinforce the same trade-off observed in the main setting. Diagnostics help preserve answer alignment and candidate coverage, but they do not guarantee that the first selected answer is the exact span preferred by the metric. A practical system should therefore combine constraint diagnostics with answer-level reranking or verification. Appendix C provides the corresponding robustness visualization.

Method (HardNegative)	Accuracy	ROUGE-1	ROUGE-2	ROUGE-L
BM25FilteredRAG	0.2881	0.2016	0.1056	0.2002
RelevanceOnlyRAG	0.3042	0.2139	0.1166	0.2125
CONSTRAINTRAG-FIRSTCANDIDATE	0.2407	**0.2817**	**0.1534**	**0.2809**
CONSTRAINTRAG-DIRECT	0.2870	0.2180	0.1177	0.2170
CONSTRAINTRAG-RECALL	**0.4284**	0.0283	0.0106	0.0276

Table 4: RAG results under injected lower-ranked BM25 retrieval noise. CONSTRAINTRAG variants include two concise outputs and the recall-oriented candidate-set output.

6 Analysis

6.1 When Constraint Diagnostics Help

Constraint diagnostics help most when a retrieved set contains several plausible entities, dates, or roles and the question requires selecting the one that satisfies a specific relation. Table 6 shows representative cases. For a temporal range question about piracy, the filtering baselines focus on a narrower range, while CONSTRAINTRAG variants recover the full gold range. For a misleading entity question about "Twinkle, Twinkle, Little Star," scalar relevance is drawn toward Mozart, whereas candidate generation surfaces Jane Taylor. For the Super Bowl question, the retrieved set contains multiple football distractors, but diagnostic selection surfaces Denver Broncos. Appendix A provides an aggregate diagnostic-label check, showing that selected passages contain more satisfied labels while rejected passages are dominated by missing checks.

6.2 Inference Cost

Constraint diagnostics add inference overhead because CONSTRAINTRAG performs question decomposition and passage-level diagnostic labeling before final answer generation. Table 5 reports approximate wall-clock runtimes for the full 6,515-example NQ development evaluation on an NVIDIA H200 GPU. The filtering baselines finish in about 7.4–7.7 minutes, while the diagnostic variants take about 9.7–11.5 minutes depending on the output mode. This shows that the diagnostic stage adds measurable cost, but remains practical for offline evaluation when diagnostics and selected passages are cached.

Method	Runtime (min.)
BM25FilteredRAG	7.4
RelevanceOnlyRAG	7.7
CONSTRAINTRAG-DIRECT	9.7
CONSTRAINTRAG candidate variants	11.5

Table 5: Approximate wall-clock runtime for the full 6,515-example NQ development evaluation on an NVIDIA H200 GPU. The CONSTRAINTRAG candidate variants row includes First-candidate, Candidates, and Recall. Runtimes depend on hardware, batching, and cache state.

6.3 Why the Best Accuracy Variant Is Not the Best Final Answer

The highest subspan exact match among proposed variants comes from Recall, not from the direct or first-candidate outputs. This is expected: Recall is a candidate-set representation, so it has more opportunities to include the gold span. It is useful when the next stage is a verifier, reranker, or abstention mechanism. It is not a clean final response because its low ROUGE indicates substantial extra material.

For user-facing short answers, Direct is the safest proposed variant because it produces a single answer and remains competitive with passage-filtering baselines. For applications where lexical exactness and concision matter more than span inclusion, First-candidate is attractive because it achieves the strongest ROUGE among concise outputs. For applications that prioritize not missing the gold answer before a downstream verifier, Recall is the appropriate variant.

6.4 Failure Modes

Three failure modes remain. First, the decomposition prompt can produce a generic answer-type check instead of the discriminative relation in the question. Second, passage diagnostics sometimes label partial evidence as sufficient when a passage contains a nearby entity or date. Third, candidate ranking is imperfect: the correct answer may appear in the generated or evidence-derived candidate set but not in the first position. A remaining example from the development outputs is the Fuller House release-date case, where the recall-oriented output contains the gold date but the concise variants select a nearby wrong date. These failures motivate replacing prompted diagnostics or candidate ordering with calibrated answer verification.

7 Limitations and Future Work

The first limitation is metric sensitivity. The provided project accuracy metric is normalized subspan exact match, so it rewards outputs that contain the gold span even when they are verbose. This makes the metric useful for measuring answer coverage, but less reliable as a standalone measure of final-answer precision. ROUGE rewards lexical alignment with the short reference answer but can penalize candidate sets that are useful for recall. The method should therefore be evaluated with both concise-answer and candidate-coverage metrics.

Output source	**Q1:** when did piracy hit its apex in the caribbean	**Q2:** who wrote variations on twinkle twinkle little star	**Q3:** who did seattle beat in the super bowl
Gold answer	1660s to 1730s	Jane Taylor	Denver Broncos
BM25FilteredRAG	1715–1725	Mozart, Twelve Variations on "Ah! vous dirai-je, maman"	the Pittsburgh Steelers... Washington Redskins ...
RelevanceOnlyRAG	between 1715 and 1725	Mozart	Denver Title: 2017 Philadelphia Eagles season ...
CONSTRAINTRAG-DIRECT	1660s to 1730s	Mozart, Twelve Variations on "Ah! vous dirai-je, maman" ...	Seattle beat the Denver Broncos ...
CONSTRAINTRAG-FIRSTCANDIDATE	1660s to 1730s	Jane Taylor	Denver Broncos

Table 6: Representative qualitative cases from the full development outputs. Long outputs are shortened with ellipses for readability while preserving the relevant answer spans. The examples illustrate three constraint-sensitive patterns: recovering the correct temporal range, separating an author entity from a plausible music distractor, and selecting the correct team from noisy football evidence.

The second limitation is inference cost. As shown in Table 5, CONSTRAINTRAG requires additional LLM calls for decomposition and passage diagnostics, which add measurable overhead compared with filtering-only baselines. An interactive deployment would need either a cheaper verifier or a trained diagnostic scorer.

The third limitation is the absence of supervised diagnostic calibration. The labels are generated by Llama and can be noisy, and the diagnostic weights in the passage score are fixed rather than learned or adapted per question. A natural next step is to train a lightweight answer verifier or reranker from cached diagnostics and retrieval-noise experiments, so that candidate-set outputs can improve coverage without being used directly as final answers. Another extension is targeted retrieval: when all retrieved passages miss the same constraint, the missing or contradicted checks could be used to formulate a second BM25 query.

8 Conclusion

CONSTRAINTRAG introduces constraint-violation diagnostics for RAG evidence selection on Natural Questions. Under a fair shared generation budget, the method does not simply beat all baselines in concise exact-match accuracy. Instead, it exposes a meaningful trade-off. Direct is competitive with passage-filtering baselines and improves ROUGE, First-candidate yields the strongest concise ROUGE, and Recall shows that constraint-guided candidate generation substantially increases answer-span coverage. Hard-negative experiments show a similar pattern: relevance-only filtering is strongest in noisy exact-match accuracy, while CONSTRAINTRAG variants preserve lexical alignment or candidate coverage. These results suggest that constraint diagnostics are most valuable as structured evidence-management signals, especially when paired with a downstream answer verifier.

References

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. 2023. GQA: Training generalized multi-query transformer models from multi-head checkpoints. *arXiv preprint arXiv:2305.13245*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

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

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*, 7:452–466.

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

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira.

2021. Pyserini: A python toolkit for reproducible information retrieval research with sparse and dense representations. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2356–2362.

Chaitanya Malaviya, Peter Shaw, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2023. QUEST: A retrieval dataset of entity-seeking queries with implicit set operations. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14032–14047. Association for Computational Linguistics.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. 2021. Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30.

Biao Zhang and Rico Sennrich. 2019. Root mean square layer normalization. In *Advances in Neural Information Processing Systems*, volume 32.

A Diagnostic Label Distribution

Figure 5 compares diagnostic-label shares between selected and rejected passages. Selected passages contain more satisfied checks, while rejected passages are dominated by missing checks. This supports the intended behavior of the selector, although the remaining missing and contradicted labels show that the diagnostic stage is not a perfect support verifier.

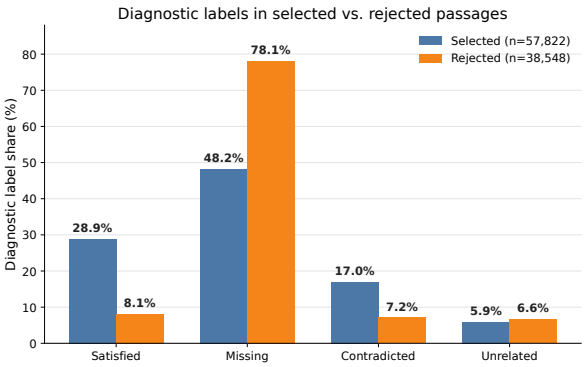

Figure 5: Diagnostic-label distribution for selected and rejected passages.

B Pairwise Exact-Match Overlap

Aggregate scores hide whether methods answer the same questions. Figure 6 visualizes representative pairwise exact-match overlap patterns. The direct and recall variants are compared with Llama zero-shot, Llama few-shot, BM25, and scalar-relevance baselines. Direct is the main single-answer version of CONSTRAINTRAG, while Recall represents candidate-set coverage. The additional few-shot rows do not change the qualitative pattern: Direct has a more balanced profile against concise baselines, whereas Recall recovers many additional answer spans that baselines miss.

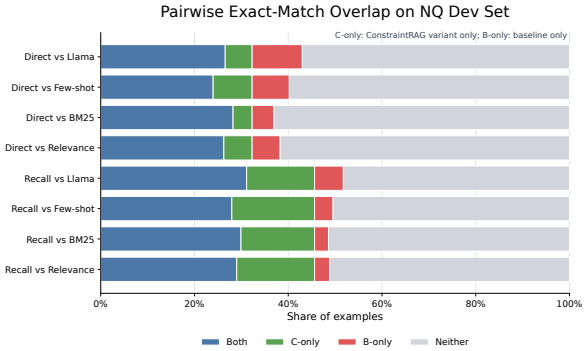

Figure 6: Representative pairwise exact-match overlap, including the two-shot Llama prompt control. The direct variant represents the single-answer setting, while the recall variant represents candidate-set coverage. C-only denotes examples correct only under the CONSTRAINTRAG variant, and B-only denotes examples correct only under the baseline.

C Robustness Visualization

Figure 7 visualizes the robustness results from Table 4. Relevance-only filtering has the highest noisy exact-match accuracy, while CONSTRAINTRAG variants retain stronger ROUGE or candidate coverage.

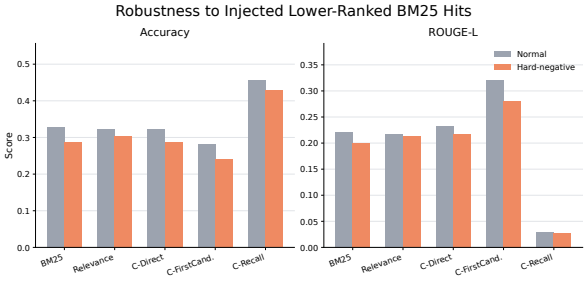

Figure 7: Robustness to injected lower-ranked BM25 hits.