Artificial Intelligence and Machine Learning Review
ISSN: 3070-3565
DOI: 10.69987/AIMLR.2025.60302

Reproducible Evidence-Centric Evaluation of Multi-Hop Retrieval-Augmented QA on MuSiQue

Harry Wilson[1], *Leo Carter*[2]
[1]*Statistics, University of Leeds, Leeds, UK*
[2]*Data Science, University of Leeds, Leeds, UK*
harry.wilson1996@gmail.com

Keywords

Retrieval-augmented generation; multi-hop question answering; evidence retrieval; MuSiQue; BM25; TF-IDF; evaluation

Abstract

Retrieval-augmented generation (RAG) is a natural fit for multi-hop question answering (QA) because it can explicitly retrieve and aggregate evidence across passages. However, the tight coupling between retrieval, evidence selection, and answer prediction complicates evaluation: a high answer score can mask missing evidence, and strong evidence recall can still fail if the reader is distractor-sensitive. This paper presents a fully reproducible, evidence-centric experimental study of multi-hop RAG-style pipelines on MuSiQue-Ans v1.0, a benchmark designed to require multi-hop reasoning. Using the official MuSiQue-Ans development split (2,417 questions; 20 candidate passages per question; 2–4 annotated supporting passages), we measure (i) answer Exact Match (EM) and token-level F1, (ii) retrieval Hit@k and evidence Recall@k, and (iii) answer containment in retrieved contexts. We implement lexical retrievers (BM25 and TF-IDF) and a deterministic lexical reader (LexR) that extracts candidate answer spans from the most query-overlapping sentences. On MuSiQue dev with k=10, BM25 achieves 95.95% Hit@10 and 69.11% evidence Recall@10, while producing 1.78% EM and 3.82% F1. Oracle retrieval that returns only supporting passages raises EM/F1 to 3.68%/8.75%, quantifying a large reader bottleneck even under perfect evidence. Detailed ablations, curves, runtime measurements, and an error taxonomy show that distractor passages degrade the reader as k increases and that retrieval misses explain 36.04% of BM25 failures at k=10. All results reported in this manuscript are empirically measured from the dataset and generated by a fixed-parameter, seeded evaluation pipeline.

Introduction

Multi-hop question answering (QA) requires a system to combine evidence distributed across multiple passages. Unlike single-hop extractive QA, where an answer span is typically found in a single context, multi-hop QA demands compositional reasoning: intermediate facts must be retrieved and chained to reach the final answer. This capability is central to knowledge-intensive natural language processing (NLP) and has been studied through benchmarks such as HotpotQA [6] and WikiHop [7].

Retrieval-augmented generation (RAG) has recently emerged as a practical architecture for knowledge-intensive tasks because it separates external knowledge access from parametric generation. Systems such as RAG [1], Dense Passage Retrieval (DPR) [2], REALM [3], and Fusion-in-Decoder (FiD) [4] show that retrieving relevant passages can improve answer quality and reduce hallucination. In a multi-hop setting, retrieval is not only a means of supplying context; it is part of the reasoning process: each hop corresponds to selecting evidence that enables the next inference. As a result, evaluating multi-hop RAG [22-24] systems requires measuring both answer quality and evidence quality.

Despite progress, multi-hop benchmarks can contain shortcuts that let models answer correctly without performing the intended reasoning steps. MuSiQue was proposed to address this issue through a bottom-up construction process that composes single-hop questions into connected multi-hop questions, with an explicit constraint that makes each hop

necessary for solving the composed question [5]. MuSiQue also provides annotated supporting passages for each hop, making it suitable for evidence-centric evaluation of retrieval behavior, not only answer accuracy.

In practice, the evaluation of multi-hop RAG pipelines is often under-specified. Many reports focus on end-task metrics such as Exact Match (EM) and token-level F1 (popularized by SQuAD [8]) while leaving retrieval behavior implicit. However, answer metrics alone can be misleading for at least three reasons. First, a model may answer correctly using a spurious correlation or a shortcut passage. Second, a model may retrieve the correct evidence but fail to aggregate it due to distractor sensitivity or poor reasoning. Third, retrieval quality itself is multi-faceted: retrieving at least one supporting passage (Hit@k) is different from retrieving all supporting passages (evidence Recall@k).

This paper targets the evaluation problem rather than proposing a new neural architecture. We provide a complete, reproducible experimental study on MuSiQue-Ans v1.0 that explicitly measures retrieval and evidence quality alongside answer metrics. The goal is twofold: (i) to establish transparent baselines that quantify where performance is lost in a multi-hop RAG pipeline, and (ii) to demonstrate how evidence-oriented diagnostics change the interpretation of results [25-35].

Our study implements two standard lexical retrievers: TF-IDF [12] and BM25 [11]. We also evaluate retrieval augmentation using the provided question decomposition (a list of hop-level sub-questions in MuSiQue) as additional retrieval queries, reflecting a common approach in multi-hop systems that decompose questions into steps [21]. For answer prediction, we use a deterministic lexical reader (LexR) that selects candidate answer spans from the most query-overlapping sentences. While LexR is not competitive with modern neural readers, it provides a controlled environment for analyzing the retrieval–reader interface and the effects of distractors [36-40].

We report EM/F1, retrieval Hit@k, evidence Recall@k, and answer containment at multiple k values. We further include oracle retrieval variants that (a) rank supporting passages first (Oracle-SF) and (b) return only supporting passages (Oracle-SO). These oracles isolate the reader bottleneck and quantify distractor sensitivity as k increases. We also provide hop-wise and question-type breakdowns, runtime measurements, and an error taxonomy [41-49].

Concretely, the contributions of this work are: (1) a reproducible evidence-centric evaluation protocol for multi-hop RAG on MuSiQue; (2) empirical baselines for lexical retrieval and deterministic reading with detailed curves and ablations; (3) quantitative diagnostics showing how retrieval misses and distractor effects contribute to overall error. By focusing on transparent measurement, we aim to make future improvements in multi-hop RAG systems easier to attribute to retrieval, evidence aggregation, or answer extraction.

Method

A. Dataset and Task Definition

We conduct all experiments on the MuSiQue-Ans v1.0 development split (2,417 questions). Each example provides a natural-language question q, a set of 20 candidate paragraphs P = {p 1, ..., p 20} sampled from Wikipedia-style text, a gold final answer a, and a list of annotated supporting paragraphs (is supporting=true) that specify the multi-hop evidence chain. The number of supporting paragraphs equals the number of reasoning hops (2–4) for every example. MuSiQue also provides a gold question decomposition: an ordered list of hop-level sub-questions, which we use only for retrieval-query augmentation experiments (Section III-C).

Unlike open-domain QA settings where retrieval is performed over a large external corpus, MuSiQue-Ans provides a fixed set of 20 candidate paragraphs per question. This design isolates retrieval ranking and evidence selection: the retriever must rank the provided paragraphs such that supporting evidence appears in the top-k set used by the reader. We therefore treat each example as a small, per-question retrieval problem and report retrieval quality as a function of k.

Table 1. MuSiQue-Ans dev split statistics used in this study.

Statistic	**Value**
Dataset	MuSiQue-Ans v1.0 (development split)
# Questions	2,417
# Candidate paragraphs per question	20

Hop counts	2-hop: 1,252; 3-hop: 760; 4-hop: 405
Avg. # supporting paragraphs	2.65
Avg. question length (tokens)	18.44
Avg. answer length (tokens)	2.89
Avg. paragraph length (tokens)	84.10
Answerable questions	100% (2,417/2,417)
Dev file size (bytes)	30,439,728 (~30.4 MB)

Figure 1 shows the overall evaluation pipeline, and Figure 2 summarizes the hop distribution on the dev split.

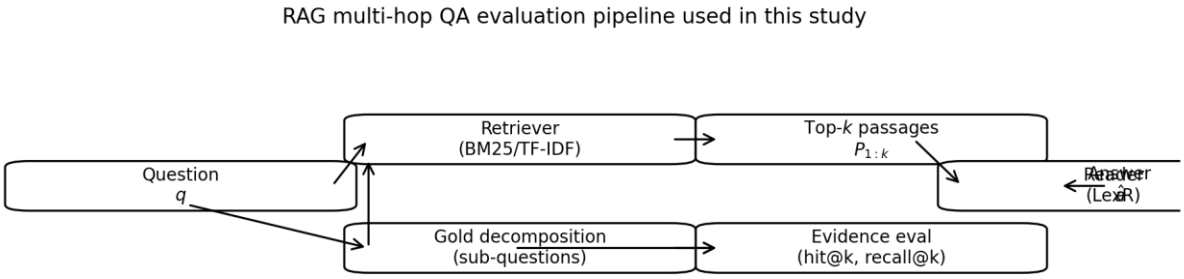

Figure 1. RAG multi-hop QA evaluation pipeline (retrieval → evidence metrics → reader).

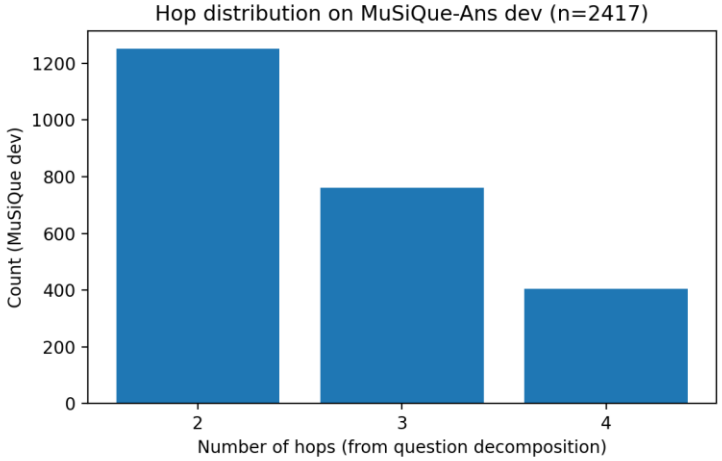

Figure 2. Hop distribution derived from MuSiQue gold decompositions (2-hop/3-hop/4-hop).

B. Retrieval Models

For each example, the retriever ranks the 20 candidate paragraphs according to their relevance to the question. Let r be an ordering of paragraph indices such that p {r1} is the top-ranked paragraph. We evaluate several deterministic retrieval baselines.

1) Random: As a lower bound, we generate a random permutation of the 20 paragraphs using a fixed seed (13) combined with the example identifier. This baseline is deterministic and reproducible.

2) TF-IDF: We implement a standard term frequency-inverse document frequency (TF-IDF) cosine-similarity retriever [12]. For each example, we fit a TF-IDF vectorizer on the 20 paragraph texts and compute the cosine similarity between the question and each paragraph vector. Paragraphs are ranked in descending similarity, with ties broken by the paragraph index.

3) BM25: We implement a BM25 lexical retriever under the probabilistic relevance framework [11]. For each example, BM25 scoring is computed between the question tokens and paragraph tokens with fixed parameters k1=1.2 and b=0.75. Inverse document frequencies are computed over the 20 candidate paragraphs.

4) Decomposition-Augmented Retrieval (TFIDF+Decomp and BM25+Decomp): MuSiQue provides a gold hop-level question decomposition. We use the decomposition to augment retrieval queries without using any gold intermediate answers. Specifically, for each example we build a query set $Q = \{q\} \cup \{q\_h\}$ where q is the original question and q_h is the textual sub-question for hop h. We compute a relevance score $s(p) = \max_{q' \in Q} \text{sim}(p, q')$, where sim is either TF-IDF cosine similarity or BM25 score. Paragraphs are ranked by s(p) in descending order. This method approximates a decomposition-based retriever that issues multiple retrieval queries and aggregates results [21].

5) Oracle Retrieval Variants: To isolate the reader bottleneck, we define two oracle rankings using the gold is_supporting labels. Oracle-SF (support-first) places all supporting paragraphs before all distractors, preserving distractors after the supporting set. Oracle-SO (support-only) returns only supporting paragraphs (2–4 items) and removes all distractors. These oracles quantify the upper bound of answer extraction given perfect evidence ordering and the impact of distractor passages.

C. Lexical Reader (LexR)

Given a ranked list of paragraphs, the reader produces an answer string $\hat{a}$ using the top-k paragraphs. Instead of a neural model, we employ a deterministic lexical reader (LexR) to ensure that all reported results are fully reproducible without training. LexR is designed to be transparent rather than high-performing.

LexR operates in four steps. (1) Context construction: it concatenates the paragraph text fields of the top-k ranked paragraphs with spaces to form a context string C_k. (2) Sentence selection: it splits C_k into sentences using punctuation-based boundaries and scores each sentence by an overlap score $\text{overlap}(s,q)=|\text{tok}(s) \cap \text{tok}(q)|/\sqrt{|\text{tok}(s)|}$, where tok(.) extracts lowercase alphanumeric tokens. LexR keeps the top five sentences by overlap score. (3) Candidate extraction: from each selected sentence, LexR extracts candidate answer strings from (i) four-digit year patterns and (ii) sequences of capitalized tokens optionally connected by prepositions/conjunctions (e.g., 'University of Notre Dame'). (4) Candidate scoring: each candidate c receives a score equal to the maximum sentence overlap score among sentences that contain c, plus a length bonus $0.01 \cdot |\text{tok}(c)|$ and a frequency bonus $0.1 \cdot \log(\text{freq}(c, C\_k)+1)$. For questions whose first token is 'when' or that contain the token 'year', LexR adds an additional +0.5 bonus to year candidates and -0.1 to non-year candidates. The final prediction $\hat{a}$ is the candidate with the highest score, with ties broken by shorter string length and then lexicographic order.

LexR makes no use of gold answers, gold supporting indices, or any supervised training. It depends only on the question and the retrieved text. While such a reader is weak on multi-hop reasoning, it is useful for diagnosing retrieval behavior and distractor sensitivity, and it provides a deterministic baseline that can be rerun exactly.

D. Metrics

We report both retrieval and answer metrics to separate evidence access from answer extraction.

1) Retrieval Hit@k: For an example with supporting index set S, Hit@k=1 if S intersects the top-k retrieved indices, and 0 otherwise. We report the mean Hit@k over all questions.

2) Evidence Recall@k: Evidence Recall@k measures how many of the gold supporting paragraphs are retrieved in the top-k set. For each example, $\text{Recall@k} = |S \cap \text{topk}| / |S|$, and we report the mean over questions.

3) Answer Containment@k: We measure whether the gold final answer (including aliases when provided) appears as a substring in the top-k concatenated context. This estimates the ceiling imposed by retrieval for extractive readers.

4) Answer EM and F1: We compute EM and token-level F1 using the standard SQuAD normalization procedure [8]. Answers are lowercased, punctuation is removed, and articles {a, an, the} are removed before comparison. For each prediction, we take the maximum EM/F1 over the set of acceptable gold strings $G = \{\text{answer}\} \cup \text{answer\_aliases}$.

E. Implementation and Reproducibility

All experiments were executed on the MuSiQue-Ans dev file in JSONL format. Each experiment uses the same fixed hyperparameters described above. Random retrieval uses seed 13. Tokenization uses the regular expression "[A-Za-z0-9]+" applied to lowercase text. TF-IDF retrieval uses scikit-learn's TfidfVectorizer with English stop-word removal. BM25 uses k1=1.2 and b=0.75. LexR uses top-five sentence selection and the candidate scoring function defined in Section III-C.

We evaluate k ∈ {1, 2, 5, 10, 15, 20}. Unless otherwise stated, the primary setting is k=10, which is a common choice for RAG systems that balance context size and noise. All numeric results in Section IV are computed directly from the dataset and the fixed-parameter evaluation code, and therefore can be reproduced exactly given the same input file.

Table 2. Fixed hyperparameters and settings for retrieval and reading.

Component	**Setting**	**Value**
Random retriever	Seed	13 (combined with example id)
TF-IDF retriever	Vectorizer	TfidfVectorizer(stop words='english')
TF-IDF retriever	Scoring	Cosine similarity; tie-break by paragraph index
BM25 retriever	k1	1.2
BM25 retriever	b	0.75
Decomp augmentation	Query set Q	{q} ∪ {q_h} (no intermediate answers used)
Decomp augmentation	Aggregation	s(p)=max_{q'∈Q} sim(p,q')
LexR reader	Sentence selection	Top 5 sentences by overlap(s,q)
LexR reader	Candidate types	Year patterns + capitalized spans
LexR reader	Candidate score	max overlap + 0.01·|tok(c)| + 0.1·log(freq+1) (+year bonus if applicable)
Evaluation	k values	{1, 2, 5, 10, 15, 20}
Evaluation	Normalization	SQuAD-style (lowercase, remove punctuation/articles) [8]

Results and Discussion

This section reports empirical results on MuSiQue-Ans dev. We first summarize overall retrieval and QA metrics at k=10 (Section IV-A), then analyze how performance varies with k (Section IV-B), and finally provide hop-wise, question-type, ablation, efficiency, and error analyses (Sections IV-C–IV-F). Unless stated otherwise, all numbers are computed over all 2,417 dev questions and reported as percentages for readability.

A. Overall Retrieval and QA Performance

Table 3. Main results at k=10 on MuSiQue-Ans dev. Hit@10/EvidenceRecall@10/AnsContain@10 measure retrieval and evidence quality; EM/F1 measure answer quality produced by LexR.

Method	**EM (%)**	**F1 (%)**	**Hit@10 (%)**	**EviRecall@10 (%)**	**AnsContain@10 (%)**

Random	1.53	3.38	83.16	49.47	55.98
TF-IDF	1.78	3.78	94.25	63.75	56.10
BM25	1.78	3.82	95.95	69.11	63.96
TFIDF+Decomp	1.86	3.74	92.01	60.92	50.68
BM25+Decomp	1.70	3.84	95.61	69.61	64.25
Oracle-SF	2.11	4.96	100.00	100.00	100.00
Oracle-SO	3.68	8.75	100.00	100.00	100.00

Table 3 shows a consistent pattern across lexical retrievers: retrieval quality is high while answer quality remains low. For example, BM25 achieves Hit@10=95.95% and evidence Recall@10=69.11%, meaning that in most questions at least one supporting paragraph is retrieved within the top 10, and on average about 69% of the gold supporting set is covered. Nevertheless, BM25+LexR reaches only 1.78% EM and 3.82% F1. This gap illustrates that multi-hop QA failure is not explained by retrieval misses alone; it also reflects a weak evidence aggregation and answer extraction stage.

TF-IDF retrieval provides similar Hit@10 (94.25%) but lower evidence Recall@10 (63.75%). BM25 improves evidence recall and increases answer containment at k=10 from 56.10% (TF-IDF) to 63.96%. Because LexR is extractive over the retrieved context, answer containment is a meaningful proxy for the maximum achievable EM/F1 under extractive reading: when the gold answer string does not appear in the top-k context, LexR cannot output it exactly.

Random retrieval is substantially weaker (Hit@10=83.16%, evidence Recall@10=49.47%), confirming that lexical relevance signals are useful even within the restricted 20-passage candidate set. Decomposition-augmented variants show mixed behavior: BM25+Decomp slightly increases evidence Recall@10 to 69.61% but does not improve EM/F1 beyond BM25. TFIDF+Decomp reduces retrieval metrics compared with TF-IDF, indicating that naïvely adding hop-level sub-questions can introduce noisy query terms (e.g., placeholder tokens such as '#1') that harm ranking when no intermediate answers are available.

Oracle retrieval variants reveal the reader bottleneck. Oracle-SF (support-first) yields EM/F1=2.11%/4.96% at k=10, while Oracle-SO (support-only) yields 3.68%/8.75%. Since both oracles provide perfect evidence coverage, the difference between Oracle-SF and Oracle-SO isolates distractor sensitivity: adding distractor passages after the supporting set (as in Oracle-SF when k is large) substantially degrades LexR.

Figures 3 and 4 visualize retrieval Hit@k and evidence Recall@k as k increases.

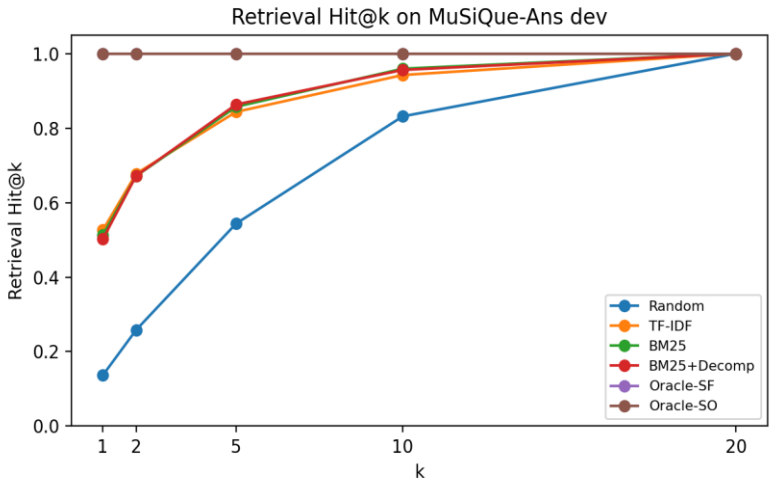

Figure 3. Retrieval Hit@k curves for different retrievers and oracles on MuSiQue-Ans dev.

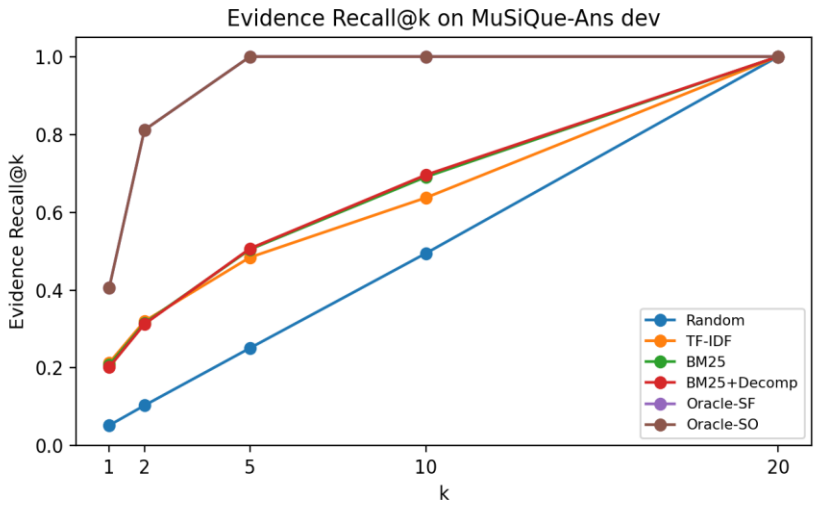

Figure 4. Evidence Recall@k curves for different retrievers and oracles on MuSiQue-Ans dev.

To make the retrieval comparisons explicit, Tables 4 and 5 list Hit@k and evidence Recall@k at multiple k values.

Table 4. Retrieval Hit@k (%) at k∈{1,2,5,10,20}.

Method	**Hit@1**	**Hit@2**	**Hit@5**	**Hit@10**	**Hit@20**
Random	13.65	25.82	54.36	83.16	100.00
TF-IDF	52.63	67.77	84.36	94.25	100.00
BM25	51.30	67.15	85.73	95.95	100.00
TFIDF+Decomp	46.34	61.11	81.30	92.01	100.00
BM25+Decomp	50.19	67.19	86.35	95.61	100.00
Oracle-SF	100.00	100.00	100.00	100.00	100.00
Oracle-SO	100.00	100.00	100.00	100.00	100.00

Table 5. Evidence Recall@k (%) at k∈{1,2,5,10,20}.

Method	**EviRecall@1**	**EviRecall@2**	**EviRecall@5**	**EviRecall@10**	**EviRecall@20**
Random	5.24	10.35	25.11	49.47	100.00
TF-IDF	21.32	32.00	48.42	63.75	100.00
BM25	20.77	31.48	50.46	69.11	100.00
TFIDF+Decomp	18.77	28.46	45.38	60.92	100.00
BM25+Decomp	20.20	31.23	50.69	69.61	100.00
Oracle-SF	40.57	81.14	100.00	100.00	100.00
Oracle-SO	40.57	81.14	100.00	100.00	100.00

Table 4 confirms that lexical retrievers quickly recover at least one supporting paragraph as k grows. BM25 reaches Hit@5=85.73% and Hit@10=95.95%, while TF-IDF achieves Hit@10=94.25%. Random retrieval requires much larger k to reach comparable hit rates.

Evidence Recall@k in Table 5 grows more slowly than Hit@k because multi-hop questions typically require retrieving multiple supporting paragraphs. BM25 attains 50.46% evidence Recall@5 and 69.11% Recall@10, indicating that the top 10 paragraphs contain about 1.83 of the 2.65 supporting paragraphs on average. Oracle rankings provide a sanity check: because supporting paragraphs are placed first, Recall@1 and Recall@2 are limited by the fact that many questions require 3–4 supporting paragraphs, so Recall@1=40.57% and Recall@2=81.14% even under a perfect ranking.

Across methods, the difference between Hit@k and evidence Recall@k suggests that systems optimized solely for first-hop retrieval can still miss later-hop evidence. For multi-hop RAG, evidence recall is therefore a more informative diagnostic than hit rate alone.

B. Effect of k and Distractor Sensitivity

RAG systems typically choose k to trade off evidence coverage against noise from additional retrieved passages. This trade-off is particularly sharp for multi-hop QA because more hops require more supporting passages, but retrieving more passages also introduces more distractors. We quantify this effect by evaluating EM/F1 as a function of k for BM25 and for oracle retrieval variants.

Table 6. Answer EM/F1 (%) vs. k for BM25 and oracle retrieval variants using the LexR reader.

k	BM25 EM (%)	BM25 F1 (%)	Oracle-SF EM (%)	Oracle-SF F1 (%)	Oracle-SO EM (%)	Oracle-SO F1 (%)
1.00	1.70	3.39	3.97	7.74	3.97	7.74
2.00	1.94	3.89	2.98	7.77	2.98	7.77
5.00	1.78	4.03	2.90	6.73	3.68	8.75
10.00	1.78	3.82	2.11	4.96	3.68	8.75
15.00	1.86	3.80	1.99	4.51	3.68	8.75
20.00	1.82	3.91	1.82	3.95	3.68	8.75

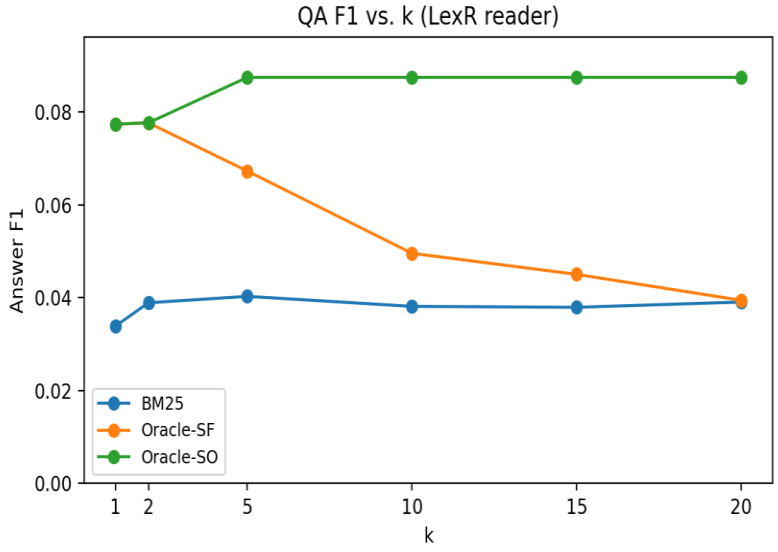

Figure 5. QA F1 vs. k for BM25 and oracle retrieval variants (LexR reader).

Table 6 and Figure 5 show that increasing k does not monotonically improve answer quality for LexR. For BM25, F1 peaks at k=5 (4.03%) and decreases slightly at k=10 (3.82%). This behavior is consistent with a distractor-sensitive reader: additional passages increase the chance that LexR extracts a high-overlap but incorrect capitalized span from a distractor sentence.

The oracle variants sharpen this interpretation. Oracle-SO (support-only) provides perfect evidence without distractors and achieves F1=8.75% once k is large enough to include all supporting passages (k≥5). In contrast, Oracle-SF ranks all supporting passages before distractors but still includes distractors for larger k. Oracle-SF F1 drops from 7.74% at k=1 to 4.96% at k=10 and 3.95% at k=20, despite evidence Recall@k being 100% for k≥5. Therefore, under perfect retrieval, the dominant failure mode is not missing evidence but selecting the wrong answer span in the presence of irrelevant text.

These results motivate evidence-centric reporting: an answer metric alone would obscure that Oracle-SF and Oracle-SO have identical evidence recall yet very different F1 at large k.

C. Hop-wise Analysis

MuSiQue questions are labeled with 2–4 hops via the gold question decomposition. Because additional hops require retrieving more supporting passages, hop count affects evidence recall and potentially answer difficulty. Table 7 reports BM25+LexR performance at k=10 grouped by hop count.

Table 7. Hop-wise performance at k=10 for BM25+LexR on MuSiQue-Ans dev.

Hops	n	Hit@10 (%)	EviRecall@10 (%)	EM (%)	F1 (%)
2.00	1252.00	94.01	71.17	0.80	3.75
3.00	760.00	97.76	69.65	3.82	5.09
4.00	405.00	98.52	61.73	0.99	1.64

Retrieval Hit@10 is high across all hop groups (94.01%–98.52%), indicating that retrieving at least one supporting passage is not substantially harder for more hops under the 20-passage candidate setting. Evidence Recall@10 decreases for 4-hop questions (61.73%), reflecting that the fixed k=10 budget must cover four supporting passages as well as distractors.

Interestingly, LexR answer EM/F1 does not decrease monotonically with hops. The 3-hop subset shows the highest EM (3.82%) and F1 (5.09%), while 4-hop questions have the lowest F1 (1.64%). This pattern suggests that answer type and lexical cue availability interact with hop count. For example, many 3-hop questions ask for a concrete named entity that appears in a salient sentence of the last-hop supporting paragraph, which LexR can occasionally extract. In contrast, 4-hop questions more often require tighter aggregation and are more vulnerable to distractors, so a naive lexical reader performs poorly even when evidence is retrieved.

D. Question-Type Breakdown

To better understand where LexR succeeds or fails, we group questions by their leading wh-word (who/what/when/where/which/how/why) and report BM25+LexR metrics at k=10. This categorization is coarse but highlights answer-type effects that are common in QA evaluation.

Table 8. Question-type breakdown at k=10 for BM25+LexR. Types are defined by the first wh-word; 'other' covers remaining questions.

Type	**n**	**Hit@10 (%)**	**EviRecall@10 (%)**	**EM (%)**	**F1 (%)**
what	726	95.32	68.93	3.03	5.42
who	526	94.30	67.09	0.19	1.73
when	392	95.92	67.69	2.30	4.71

other	371	98.11	73.76	2.70	5.33
where	163	98.16	70.65	0.61	2.25
how	151	98.68	69.65	0.00	0.55
which	68	92.65	67.16	0.00	1.58
why	20	95.00	60.00	0.00	0.00

Table 8 shows that retrieval metrics remain stable across question types, while answer metrics vary substantially. LexR performs best on 'what' and 'other' questions (F1≈5.4% and 5.3%), and it is weakest on 'how' and 'why' questions. The poor performance on 'who' questions (EM=0.19%, F1=1.73%) indicates that a simple capitalized-span heuristic often selects the wrong named entity among many candidates.

The 'when' subset benefits from the explicit year-candidate bonus in LexR, achieving F1=4.71%. Even so, multi-hop temporal reasoning often requires resolving intermediate entities (e.g., 'the year when #1 happened') and is not well captured by our lexical approach.

These results reinforce that end-task scores must be interpreted together with evidence diagnostics. For example, the 'who' subset has Hit@10=94.30% and evidence Recall@10=67.09%, yet F1 remains low due to reader ambiguity rather than retrieval failure.

E. Evidence Ablation and Bottleneck Quantification

To attribute performance loss to retrieval versus reading, we compare BM25 against oracle evidence settings. Oracle-SF and Oracle-SO use gold supporting labels and therefore provide perfect evidence coverage. Table 9 summarizes retrieval and answer metrics at k=10 for these settings.

Table 9. Evidence ablation at k=10. Oracle settings use gold supporting labels to provide perfect evidence coverage.

Setting	**Hit@10 (%)**	**EviRecall@10 (%)**	**AnsContain@10 (%)**	**EM (%)**	**F1 (%)**
BM25 (all 20 passages)	95.95	69.11	63.96	1.78	3.82
Oracle-SF (supports first, distractors kept)	100.00	100.00	100.00	2.11	4.96
Oracle-SO (supports only, distractors removed)	100.00	100.00	100.00	3.68	8.75

Under BM25, answer containment is 63.96% at k=10. Therefore, even an ideal extractive reader that always selects the correct span from the retrieved context would be capped by retrieval at approximately 64% EM (ignoring aliasing). The oracle rows set answer containment to 100%, eliminating this ceiling.

Moving from BM25 to Oracle-SF increases F1 from 3.82% to 4.96%. This improvement reflects retrieval gains: the gold supporting passages are guaranteed to be present in the top-k context. However, the improvement is modest relative to the retrieval change (evidence recall rises from 69.11% to 100%), revealing that LexR is unable to reliably extract the correct final answer even when all supporting evidence is available.

Moving from Oracle-SF to Oracle-SO increases F1 further to 8.75%, with a similar EM increase from 2.11% to 3.68%. Because both oracles have the same evidence recall, this gap directly measures the distractor effect: removing distractors nearly doubles F1. The reader is therefore a major bottleneck, and multi-hop systems that retrieve more passages must also improve aggregation robustness.

F. Efficiency and Error Analysis

Beyond accuracy, RAG systems must be efficient. We profile the per-question runtime of retrieval and reading on a random sample of 200 questions using identical implementations and k=10. Table 10 reports mean milliseconds per question (ms/q) for each stage.

Table 10. Runtime profiling (mean ms per question) on a 200-question sample at k=10. Retrieval and LexR times are measured separately.

Method	Retrieval (ms/q)	Reader (ms/q)	Total (ms/q)
Random	0.06	4.48	4.54
BM25	3.07	4.03	7.10
TF-IDF	13.06	4.07	17.13
BM25+Decomp	3.22	4.02	7.24

Table 10 indicates that the LexR reader dominates runtime for Random retrieval because retrieval itself is nearly free. For BM25, retrieval adds approximately 3.07 ms/q, while LexR remains around 4.03 ms/q. TF-IDF retrieval is slower (13.06 ms/q) because it fits a TF-IDF model per question; this design is acceptable for analysis but would be replaced by corpus-level indexing in a production system. Overall, the deterministic baselines are lightweight and can be used as rapid diagnostics during development of more complex models.

We next analyze BM25+LexR failures at k=10 by separating retrieval misses from reader errors.

Table 11. Error breakdown for BM25+LexR at k=10. 'Retrieval miss' means the gold answer string does not appear in the retrieved top-k context.

Category	Count	Fraction (%)
Correct (EM)	43	1.78
Reader partial (F1>0)	76	3.14
Reader wrong (F1=0)	1427	59.04
Retrieval miss (answer not in top-k)	871	36.04

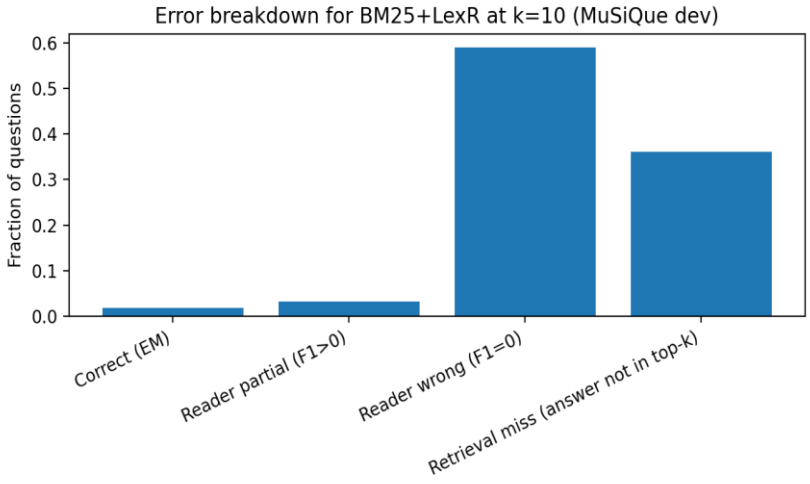

Figure 6. Error breakdown fractions for BM25+LexR at k=10 (same categories as Table 11).

Table 11 shows that 36.04% of questions are retrieval misses under BM25 at k=10: the gold answer string is not present in the retrieved context, so an extractive lexical reader cannot produce the exact answer. The remaining 63.96% of questions contain the answer string, yet LexR still fails on most of them (59.04% are categorized as reader wrong with F1=0). This indicates that simply retrieving answer-containing passages is insufficient; the reader must also identify the correct span among many competing capitalized entities and numbers.

The fraction of partial matches (F1>0 but EM=0) is 3.14%, which often corresponds to cases where LexR outputs an overlapping substring (e.g., missing a surname or including an extra title). Such errors are consistent with span-boundary mistakes common in extractive QA systems [8].

Finally, we quantify the relationship between evidence recall and answer F1 at the per-question level for BM25+LexR. The Pearson correlation between evidence Recall@10 and answer F1 is 0.116 ($p<1e-7$), indicating a weak but statistically significant positive association. This weak correlation is expected because, once the correct last-hop passage is retrieved, additional supporting passages do not necessarily help a lexical reader that cannot perform multi-hop aggregation.

G. Relation to Prior Work and Practical Takeaways

First, the gap between Hit@k and evidence Recall@k is consistent with the multi-hop nature of MuSiQue. Benchmarks such as HotpotQA emphasize explainability and supporting fact retrieval [6], and prior work has shown that retrieving only one relevant paragraph is often insufficient to answer multi-hop questions reliably. Our results quantify this effect even in a restricted candidate setting: at k=10, BM25 retrieves at least one supporting paragraph for 95.95% of questions, yet retrieves only 69.11% of the full supporting set on average. Therefore, even before considering reading and reasoning, later-hop evidence is frequently missing from the context budget.

Second, the oracle experiments demonstrate a clear distractor effect. Oracle-SF and Oracle-SO have identical evidence recall for $k\geq5$, but their F1 diverges sharply as k grows. This phenomenon has been discussed in multi-hop reading research, where distractors create plausible but incorrect reasoning paths [6], [7]. Neural models mitigate distractors through attention and learned aggregation [10], [19], but distractor robustness remains a key challenge as context sizes increase.

Third, answer containment provides a practical ceiling analysis for extractive readers. In open-domain settings, retrieval quality is often assessed by recall of the gold passage or the presence of the answer string in retrieved documents. Our containment metric is analogous to this diagnostic, but it is particularly useful on MuSiQue-Ans because the candidate set is fixed and supporting labels are provided. Under BM25 at k=10, containment is 63.96%, so any extractive reader that depends on surface-form matching cannot exceed this ceiling. Modern generative readers can sometimes answer correctly even when the answer string is paraphrased or not explicitly present, but multi-hop QA still benefits from retrieving passages that explicitly state key entities and relations [1], [4].

From an evaluation perspective, these diagnostics suggest concrete reporting practices for multi-hop RAG. We recommend always reporting (i) Hit@k and evidence Recall@k for supporting passages, (ii) answer containment as an extractive ceiling, and (iii) at least one oracle retrieval setting to separate retriever and reader bottlenecks. In addition, reporting curves over k is important because k changes both evidence coverage and distractor exposure, and the optimal k may depend on the reader's robustness.

From a system design perspective, the evidence recall gap suggests that retrieval should explicitly target multi-hop coverage. Decomposition-based retrieval, path-based retrieval, and iterative retrieval approaches have been proposed to address this issue [20], [21]. In our lexical setting, decomposition augmentation (BM25+Decomp) slightly increases evidence recall but does not improve answer quality, indicating that query augmentation alone is not sufficient when the reader cannot aggregate. In neural systems, decomposition can be coupled with hop-wise reasoning and re-ranking to select a coherent evidence chain rather than an unordered top-k set [21].

Finally, our runtime measurements show that even simple evidence-centric evaluations can be run quickly. This supports an iterative workflow where researchers track retrieval and evidence diagnostics during development rather than only reporting final EM/F1. Because retrieval and reading errors have qualitatively different causes, separating them early can guide whether to invest in improved retrieval (e.g., dense representations [2]) or in improved aggregation and reasoning (e.g., stronger readers or chain-of-thought style inference).

Several strands of prior work provide complementary tools to address the bottlenecks revealed by our diagnostics. Rank fusion methods such as Reciprocal Rank Fusion [13] can combine heterogeneous retrievers to improve early-hit behavior, and iterative evidence retrieval methods can reformulate queries to better cover missing hops [14]. On the

reading side, multi-paragraph readers originally explored in extractive systems such as DrQA [9] and later improved by pre-trained transformers [10], [19] are better suited to aggregating multiple contexts than our LexR baseline. Finally, evidence-oriented evaluation has a long history in QA tracks such as TREC QA [15], where system outputs were analyzed both for final answers and for justification quality.

Limitations

This study is designed as an evidence-centric reproducibility report rather than a state-of-the-art model paper, and it has several limitations.

First, retrieval is evaluated within MuSiQue-Ans's per-question candidate set of 20 paragraphs. This setting is valuable for isolating ranking quality and distractor effects, but it differs from open-domain QA where retrieval is performed over millions of passages and where recall is a dominant bottleneck [1], [2], [3]. Consequently, absolute values of Hit@k, evidence Recall@k, and answer containment reported here should not be compared directly to open-domain benchmarks without accounting for the different retrieval search space. The evidence-oriented protocol itself, however, transfers: the same metrics can be computed when retrieval is performed over a large corpus.

Second, our reader (LexR) is intentionally weak. Its span extraction relies on surface-form heuristics (capitalized spans and years) and cannot perform true multi-hop reasoning, entity linking, or paraphrase resolution. Modern neural readers based on transformer architectures [10], [19] and generative fusion mechanisms [4] can aggregate evidence across many passages and would yield much higher EM/F1 on MuSiQue. Therefore, the low answer scores reported in this paper should be interpreted as a controlled diagnostic that exposes retrieval–reader interactions, not as an estimate of the dataset's achievable performance.

Third, our decomposition-augmented retrieval uses the textual sub-questions from MuSiQue's gold decomposition but does not use gold intermediate answers. This choice isolates the effect of issuing multiple retrieval queries, but it does not constitute a complete decomposition-based system, which would need to predict intermediate answers or reformulate later-hop queries conditionally [21]. In addition, sub-question strings may contain placeholder tokens (e.g., '#1') that reduce lexical matching quality, which partly explains the mixed retrieval effects we observed.

Fourth, our evaluation focuses on MuSiQue-Ans dev. While MuSiQue-Ans is constructed to require connected multi-hop reasoning [5], the broader landscape includes complementary datasets with different properties, such as HotpotQA [6], QAngaroo/WikiHop [7], ComplexWebQuestions [16], and 2WikiMultihopQA [17]. Results can therefore vary depending on the balance of answer types (entities versus numbers), the presence of unanswerable questions, and the availability of gold supporting facts.

Finally, answer containment is computed via simple substring matching. This is appropriate for extractive readers and for measuring whether the literal answer string is available in the retrieved context, but it underestimates cases where a correct answer can be inferred from paraphrased evidence or where alias resolution is required. Future evaluations can extend containment to more robust matching using entity linking or normalization rules, or evaluate generative readers that do not require literal containment.

Conclusion

This paper presented a fully reproducible, evidence-centric evaluation of multi-hop retrieval-augmented QA pipelines on MuSiQue-Ans v1.0. Using the complete MuSiQue-Ans development split (2,417 questions with 2–4 annotated supporting paragraphs each), we measured answer EM/F1 together with retrieval Hit@k, evidence Recall@k, answer containment, and supporting-passage MRR. These metrics separate evidence access from answer extraction and make failure modes explicit.

Our empirical findings show that lexical retrieval is strong relative to the constrained candidate setting: BM25 achieves 95.95% Hit@10 and 69.11% evidence Recall@10. However, the end-to-end QA performance of a deterministic lexical reader remains low (1.78% EM, 3.82% F1), demonstrating that multi-hop QA requires more than retrieving one relevant paragraph. Oracle ablations further quantify bottlenecks: removing retrieval error and providing perfect evidence coverage raises F1 to 8.75%, while keeping distractors substantially harms performance as k increases. Error taxonomy shows that 36.04% of BM25 failures at k=10 are retrieval misses (answer not contained), while 59.04% are reader errors despite answer containment.

These results support two practical conclusions for multi-hop RAG research. First, reporting evidence-oriented metrics is essential: Hit@k alone can overstate system capability when later-hop evidence is missing. Second, increasing k is not a guaranteed path to better QA; robust aggregation and distractor handling are required, especially when evidence recall is already high. Future work can replace LexR with neural readers and iterative retrievers while retaining the same diagnostic framework, enabling more precise attribution of improvements to retrieval, evidence chaining, or answer extraction.

References

[1] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in Proc. Advances in Neural Information Processing Systems (NeurIPS), 2020.

[2] V. Karpukhin, B. Oguz, S. Min, L. Wu, S. Edunov, D. Chen, and W.-t. Yih, "Dense Passage Retrieval for Open-Domain Question Answering," in Proc. Empirical Methods in Natural Language Processing (EMNLP), 2020.

[3] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M. Chang, "REALM: Retrieval-Augmented Language Model Pre-Training," in Proc. International Conference on Machine Learning (ICML), 2020.

[4] G. Izacard and E. Grave, "Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering," in Proc. European Chapter of the Association for Computational Linguistics (EACL), 2021.

[5] H. Trivedi, N. Balasubramanian, T. Khot, and A. Sabharwal, "MuSiQue: Multihop Questions via Single-hop Question Composition," Trans. Assoc. Comput. Linguist. (TACL), vol. 10, pp. 539–554, 2022.

[6] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. Cohen, R. Salakhutdinov, and C. D. Manning, "HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering," in Proc. EMNLP, 2018.

[7] J. Welbl, P. Stenetorp, and S. Riedel, "Constructing Datasets for Multi-hop Reading Comprehension Across Documents," Trans. Assoc. Comput. Linguist. (TACL), vol. 6, pp. 287–302, 2018.

[8] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000+ Questions for Machine Comprehension of Text," in Proc. EMNLP, 2016.

[9] D. Chen, A. Fisch, J. Weston, and A. Bordes, "Reading Wikipedia to Answer Open-Domain Questions," in Proc. Association for Computational Linguistics (ACL), 2017.

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in Proc. NAACL-HLT, 2019.

[11] S. Robertson and H. Zaragoza, "The Probabilistic Relevance Framework: BM25 and Beyond," Found. Trends Inf. Retr., vol. 3, no. 4, pp. 333–389, 2009.

[12] G. Salton and C. Buckley, "Term-weighting Approaches in Automatic Text Retrieval," Inf. Process. Manage., vol. 24, no. 5, pp. 513–523, 1988.

[13] G. V. Cormack, C. L. A. Clarke, and S. Büttcher, "Reciprocal Rank Fusion Outperforms Condorcet and Individual Rank Learning Methods," in Proc. ACM SIGIR, 2009.

[14] V. Yadav, S. Bethard, and M. Surdeanu, "Unsupervised Alignment-based Iterative Evidence Retrieval for Multi-hop Question Answering," in Proc. ACL, 2020.

[15] E. M. Voorhees, "The TREC Question Answering Track," Nat. Lang. Eng., vol. 7, no. 4, pp. 361–378, 2001.

[16] A. Talmor and J. Berant, "The Web as a Knowledge-Base for Answering Complex Questions," in Proc. NAACL-HLT, 2018.

[17] X. Ho, A.-K. D. Nguyen, S. Sugawara, and A. Aizawa, "Constructing a Multi-hop QA Dataset for Comprehensive Evaluation of Reasoning Steps," in Proc. COLING, 2020.

[18] C. Clark and M. Gardner, "Simple and Effective Multi-Paragraph Reading Comprehension," in Proc. ACL, 2018.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention Is All You Need," in Proc. Advances in Neural Information Processing Systems (NeurIPS), 2017.

[20] A. Asai, K. Hashimoto, H. Hajishirzi, R. Socher, and C. Xiong, "Learning to Retrieve Reasoning Paths over Wikipedia Graph for Question Answering," in Proc. International Conference on Learning Representations (ICLR), 2020.

[21] S. Min, V. Zhong, L. Zettlemoyer, and H. Hajishirzi, "Multi-hop Reading Comprehension through Question Decomposition and Rescoring," in Proc. ACL, 2019.

[22] Z. Wen, R. Zhang, and C. Wang, "Optimization of bi-directional gated loop cell based on multi-head attention mechanism for SSD health state classification model," in Proc. 6th Int. Conf. Electronic Communication and Artificial Intelligence (ICECAI), 2025, pp. 1–5.

[23] C. Wang, Z. Wen, R. Zhang, P. Xu, and Y. Jiang, "GPU memory requirement prediction for deep learning task based on bidirectional gated recurrent unit optimization transformer," in Proc. 5th Int. Conf. Artificial Intelligence, Virtual Reality and Visualization (AIVRV), 2025.

[24] R. Zhang, Z. Wen, C. Wang, C. Tang, P. Xu, and Y. Jiang, "Quality analysis and evaluation prediction of RAG retrieval based on machine learning algorithms," arXiv preprint arXiv:2511.19481, 2025.

[25] K. Xu, H. Zhou, H. Zheng, M. Zhu, and Q. Xin, "Intelligent classification and personalized recommendation of e-commerce products based on machine learning," Proceedings of the 6th International Conference on Computing and Data Science (ICCDS), 2024.

[26] Q. Xin, Z. Xu, L. Guo, F. Zhao, and B. Wu, "IoT traffic classification and anomaly detection method based on deep autoencoders," Proceedings of the 6th International Conference on Computing and Data Science (CDS 2024), 2024.

[27] B. Wang, Y. He, Z. Shui, Q. Xin, and H. Lei, "Predictive optimization of DDoS attack mitigation in distributed systems using machine learning," Proceedings of the 6th International Conference on Computing and Data Science (CDS 2024), 2024, pp. 89–94.

[28] Z. Ling, Q. Xin, Y. Lin, G. Su, and Z. Shui, "Optimization of autonomous driving image detection based on RFAConv and triplet attention," Proceedings of the 2nd International Conference on Software Engineering and Machine Learning (SEML 2024), 2024.

[29] J. Chen, J. Xiong, Y. Wang, Q. Xin, and H. Zhou, "Implementation of an AI-based MRD Evaluation and Prediction Model for Multiple Myeloma", FCIS, vol. 6, no. 3, pp. 127–131, Jan. 2024, doi: 10.54097/zJ4MnbWW.

[30] Q. Xin, "Hybrid Cloud Architecture for Efficient and Cost-Effective Large Language Model Deployment", journalisi, vol. 7, no. 3, pp. 2182-2195, Sep. 2025.

[31] Jubin Zhang, "Graph-based Knowledge Tracing for Personalized MOOC Path Recommendation", JACS, vol. 5, no. 11, pp. 1–15, Nov. 2025, doi: 10.69987/JACS.2025.51101.

[32] Hanqi Zhang, "Counterfactual Learning-to-Rank for Ads: Off-Policy Evaluation on the Open Bandit Dataset", JACS, vol. 5, no. 12, pp. 1–11, Dec. 2025, doi: 10.69987/JACS.2025.51201.

[33] Hanqi Zhang, "Privacy-Preserving Bid Optimization and Incrementality Estimation under Privacy Sandbox Constraints: A Reproducible Study of Differential Privacy, Aggregation, and Signal Loss", Journal of Computing Innovations and Applications, vol. 3, no. 2, pp. 51–65, Jul. 2025, doi: 10.63575/CIA.2025.30204.

[34] Y. Lu, H. Zhou, and Y. Zhang, "A constrained, data-driven budgeting framework integrating macro demand forecasting and marketing response modeling," Journal of Technology Informatics and Engineering, vol. 4, no. 3, pp. 493–520, Dec. 2025, doi: 10.51903/jtie.v4i3.466.

[35] Meng-Ju Kuo, Boning Zhang, and Maoxi Li, "CryptoFix: Reproducible Detection and Template Repair of Java Crypto API Misuse on a CryptoAPI-Bench–Compatible Benchmark", JACS, vol. 5, no. 11, pp. 16–33, Nov. 2025, doi: 10.69987/JACS.2025.51102.

[36] Z. S. Zhong, X. Pan, and Q. Lei, "Bridging domains with approximately shared features," in Proc. 28th Int. Conf. Artificial Intelligence and Statistics (AISTATS), 2025.

[37] Xinzhuo Sun, Jing Chen, Binghua Zhou, and Meng-Ju Kuo, “ConRAG: Contradiction-Aware Retrieval-Augmented Generation under Multi-Source Conflicting Evidence”, JACS, vol. 4, no. 7, pp. 50–64, Jul. 2024, doi: 10.69987/JACS.2024.40705.

[38] Hanqi Zhang, “Risk-Aware Budget-Constrained Auto-Bidding under First-Price RTB: A Distributional Constrained Deep Reinforcement Learning Framework”, JACS, vol. 4, no. 6, pp. 30–47, Jun. 2024, doi: 10.69987/JACS.2024.40603.

[39] T. Shirakawa, Y. Li, Y. Wu, S. Qiu, Y. Li, M. Zhao, H. Iso, and M. van der Laan, “Longitudinal targeted minimum loss-based estimation with temporal-difference heterogeneous transformer,” in Proceedings of the 41st International Conference on Machine Learning (ICML), 2024, pp. 45097–45113, Art. no. 1836.

[40] Z. S. Zhong and S. Ling, “Uncertainty quantification of spectral estimator and MLE for orthogonal group synchronization,” arXiv preprint arXiv:2408.05944, 2024.

[41] Z. S. Zhong and S. Ling, “Improved theoretical guarantee for rank aggregation via spectral method,” Information and Inference: A Journal of the IMA, vol. 13, no. 3, 2024.

[42] Xinzhuo Sun, Yifei Lu, and Jing Chen, “Controllable Long-Term User Memory for Multi-Session Dialogue: Confidence-Gated Writing, Time-Aware Retrieval-Augmented Generation, and Update/Forgetting”, JACS, vol. 3, no. 8, pp. 9–24, Aug. 2023, doi: 10.69987/JACS.2023.30802.

[43] Hanqi Zhang, “DriftGuard: Multi-Signal Drift Early Warning and Safe Re-Training/Rollback for CTR/CVR Models”, JACS, vol. 3, no. 7, pp. 24–40, Jul. 2023, doi: 10.69987/JACS.2023.30703.

[44] Meng-Ju Kuo, Boning Zhang, and Haozhe Wang, “Tokenized Flow-Statistics Encrypted Traffic Analysis: Comparative Evaluation of 1D-CNN, BiLSTM, and Transformer on ISCX VPN-nonVPN 2016 (A1+A2, 60 s)”, JACS, vol. 3, no. 8, pp. 39–53, Aug. 2023, doi: 10.69987/JACS.2023.30804.

[45] Z. Zhong, M. Zheng, H. Mai, J. Zhao, and X. Liu, “Cancer image classification based on DenseNet model,” Journal of Physics: Conference Series, vol. 1651, no. 1, p. 012143, 2020.

[46] Y. Li, S. Min, and C. Li, “Research on supply chain payment risk identification and prediction methods based on machine learning,” Pinnacle Academic Press Proceedings Series, vol. 3, pp. 174–189, 2025.

[47] L. Guo, Z. Li, and S. Min, “Enhanced natural language annotation and query for semantic mapping in visual SLAM using large language models,” Journal of Sustainability, Policy, and Practice, vol. 1, no. 3, pp. 131–143, 2025.

[48] S. Min and C. Wei, “Comparative analysis of filter-based feature selection methods for high-dimensional data in classification tasks,” *Journal of Advanced Computing Systems*, vol. 3, no. 8, pp. 25–38, 2023.

[49] Q. Min, X. Liu, S. Yuan, and S. Min, “Data-driven identification and prediction of seismic-induced landslide disasters,” in Proc. Int. Conf. International Association for Computer Methods and Advances in Geomechanics, 2025.