A Multi-Stage Hybrid Retrieval Framework for Knowledge-Augmented Question Answering

Veningston K.[1], Selvan C.[2], Ronalda M.[1], Bidyut Baran Chaudhuri[3], and Anurag Tatwal[1]

[1] Department of Computer Science and Engineering,
National Institute of Technology Srinagar, J&K 190006, India
veningston@nitsri.ac.in
[2] School of Computer Science and Engineering, REVA University, Bengaluru,
Karnataka 560064, India.
selvan.c@reva.edu.in
[3] Techno India University, West Bengal, 700091, India
bbcisical@gmail.com

Abstract. Context retrieval is essential for effective question answering (QA). A language model cannot generate a correct response without receiving relevant evidence. Traditional keyword-based retrieval methods only match exact words and fail to find passages that convey the same meaning using different terms. This limitation reduces the quality of the answers. We introduce Effective Context Retrieval for Large Language Model (ECR-LLM), which is a five-stage hybrid pipeline designed to overcome this issue. First, candidate passages are selected using term-frequency-based lexical retrieval. Next, they are reranked through a sentence-embedding model that measures semantic similarity beyond simple word matching. A pre-trained extractive QA model then identifies the most likely answer spans from the reranked passages. Meanwhile, a specially designed Knowledge Graph (KG) provides a structured, fact-based answer, which is polished for fluency by a lightweight generative language model. The candidate answers are collected and presented for each query. Evaluated on 86,769 question–answer pairs from the Stanford Question Answering Dataset (SQuAD), the system reaches about 72% end-to-end accuracy. We measure retrieval quality using Mean Average Precision (MAP), Mean Reciprocal Rank (MRR), Normalised Discounted Cumulative Gain (nDCG), Precision@k, and Recall@k. We assess answer quality using Exact Match (EM), F1 score, and Recall-Oriented Understudy for Gisting Evaluation (ROUGE-1/2/L). Integrating the knowledge graph boosts the Exact Match score from 0.465 to 0.725, showing that combining lexical retrieval, semantic reranking, extractive span prediction, and structured knowledge provides notably more accurate and contextually relevant answers than traditional methods.

Keywords: Question answering · Hybrid retrieval · Semantic reranking · Knowledge graph · MiniLM · Retrieval-augmented generation

1 Introduction

Question Answering (QA) has become a foundational capability in modern natural-language processing (NLP), strengthening search engines, virtual assistants, customer-service chatbots, and domain-specific consultation systems in medicine, law, and agriculture [1]. A standard open-domain QA pipeline comprises two phases: (i) a *retrieval stage* that narrows a large document corpus to a small candidate set, and (ii) a *reading or generation stage* that produces The final answer from those candidates. The retrieval stage is the critical bottleneck that prevents a model from producing a correct answer from a passage it has never seen. Sparse retrievers such as Term Frequency-Inverse Document Frequency (TF-IDF) and Best Matching 25 (BM25) are fast and interpretable but operate on lexical overlap alone, failing whenever query and passage express the same meaning in different words. Dense Passage Retrieval (DPR) [2], Contriever [7], MiniLM embed queries and passages in a shared vector space and generalise across paraphrases, but they add GPU overhead and can underperform sparse baselines on datasets with high lexical self-similarity. Retrieval-augmented generation (RAG) systems [6] integrate these retrievers with generative readers, but their KG support is minimal, and their hardware requirements are substantial. None of the reviewed approaches offers a lightweight, multi-source answering pipeline that combines sparse retrieval, semantic reranking, extractive span prediction, structured Knowledge Graph (KG) inference, and language-fluency correction in a single deployable system. This paper presents ECR-LLM, which is a five-stage pipeline that addresses these limitations. Our contributions are:

1. A principled cascade of TF-IDF lexical retrieval and MiniLM-v6 semantic reranking that jointly achieves 86.31 % retrieval accuracy while limiting candidates to 50 passages, which is 0.26 % of the corpus.
2. Integration of a pre-built Knowledge Graph with Flan-T5 fluency correction as a structured answer source.
3. A comprehensive evaluation on 86 769 SQuAD question-answer pairs across retrieval metrics (MAP, MRR, nDCG, Precision@k, Recall@k) and generation metrics (ROUGE-1/2/L, F1, EM) on a single GPU.

2 Related Work

Open-domain QA systems universally adopt a two-stage design: (i) a *retrieval stage* narrows a large corpus to k candidate passages, and (ii) a *reading or generation stage* extracts or produces the answer from those passages. Table 1 catalogues the principal techniques at each stage.

Sparse retrieval systems such as TF-IDF and BM25 rank documents by term-frequency statistics. They are fast and interpretable but operate on lexical overlap alone, failing when query and passage are paraphrases. BM25 extends TF-IDF by accounting for term saturation and document length, providing more precise keyword-based ranking at negligible additional cost.

Table 1. Stages and techniques in information retrieval and answer generation.

Stage	Goal	Representative Techniques
Retrieval	Narrow a massive corpus to k candidate passages.	Sparse: TF-IDF, BM25 Dense: DPR, Contriever, MiniLM, E5 Hybrid: sparse first-stage + dense reranking
Reading / Generation	Extract a span or generate an answer from the k passages.	Extractive: BERT, RoBERTa Generative: T5, BART, RAG Augmented: RAG, REALM, KG-enhanced

Dense Passage Retrieval (DPR) encodes queries and passages with independent Bidirectional Encoder Representations from Transformers (BERT)-based bi-encoders and retrieves by maximum inner product. Contriever [7] trains a bi-encoder with contrastive objectives on unlabelled web documents, enabling competitive zero-shot retrieval. MiniLM is a distilled, lightweight transformer that provides competitive embedding quality at a fraction of BERT's computational cost. EmbEddings from bidirEctional Encoder rEpresentations (E5) [8] adopts multi-task, prompt-aware pre-training that improves embedding alignment across diverse query types.

Hybrid reranking systems use a sparse model for broad first-stage recall and a dense model for precise second-stage reranking. This combination is increasingly preferred because it retains the speed of keyword search while gaining semantic generalisation, which is the key insight underlying the proposed ECR-LLM design.

QA approaches have been adapted to diverse domains where retrieval precision is critical. *Medical QA* systems such as BioBERT-QA [15], MedQuAD [16], and MedExQA [14] fine-tune biomedical language models on PubMed and clinical notes, motivating the combination of dense retrieval with ontology-based KGs that we adopt. *Legal QA* systems process specialised structured documents (rulings, statutes, affidavits) where semantic comprehension is essential, while *agricultural QA* systems such as AGRO-QA [17] combine domain corpora with ontology-based knowledge representation, demonstrating the broad applicability of KG-augmented retrieval across verticals. *Chatbot QA* systems integrate RAG with conversational agents to search knowledge bases dynamically and maintain multi-turn context [1]. Early open-domain QA systems pair a sparse retriever with an extractive reader. The Document reader QA (DrQA) [2] introduced TF-IDF retrieval over Wikipedia coupled with a BiLSTM span predictor, achieving EM 69.5 / F1 78.8 on SQuAD v1.1. BERT [3] improved span-extraction accuracy (EM 80.8, F1 88.5) with transformer-based encoding but requires the correct passage to be pre-supplied. RoBERTa [4] further improved upon BERT via dynamic masking and longer pre-training (EM 88.9, F1 94.6). Table 2 summarises these systems.

Retrieval-augmented systems close the open-domain QA limitations by combining a retriever with a generative or extractive reader. Retrieval-augmented language model pre-training (REALM) [5] jointly trains retriever and reader via

Table 2. Comparison of extractive QA systems on SQuAD v1.1.

System	Peak Result (SQuAD v1.1)	Open-Domain?	Key Limitation
DrQA [2]	EM 69.5 / F1 78.8	✓	No semantic understanding
BERT-Base [3]	EM 80.8 / F1 88.5	×	Gold paragraph required
RoBERTa-Large [4]	EM 88.9 / F1 94.6	×	355M params; gold paragraph

Table 3. Retrieval-augmented QA models on SQuAD-Open.

System	Retrieval	Evaluation Metric	EM	F1
REALM [5]	Latent	—	40.7	44.4
RAG [6]	DPR	Recall@5-78%	45.5	49.0
Contriever [7]	Dense	Recall@5-83%	48.6	52.2
E5-Base [8]	Dense	Recall@10-88%	51.1	55.7
ECR-LLM (ours)	Hybrid	Recall@300-88%	**72.5***	**72.2***

*KG-augmented result on 2 000-sample evaluation set.

latent document selection. RAG [6] couples DPR retrieval with BART generation using marginalised document scores (EM 45.5, Recall@5 as 78 %). Contriever [7] achieves Recall@5 as 83 % through contrastive pre-training on unlabelled web text (EM 48.6). E5-Base [8] adopts multi-task prompt-aware encoding, reaching Recall@10 as 88 % and EM 51.1. Table 3 benchmarks these models on SQuAD-Open. Conventional systems do not incorporate a KG as a structured answer fallback, and none evaluates the combination of a lightweight semantic reranker with a purpose-built KG on the same corpus or demonstrates sub-second latency on commodity hardware.

A direct comparison on SQuAD-Open is difficult because most RAG and dense-retrieval frameworks no longer report SQuAD-Open as a primary benchmark. The field shifted toward Natural Questions (NQ) and TriviaQA, partly because of the lexical-overlap artefact. Table 4 therefore benchmarks recent systems on NQ and TriviaQA, while retaining the SQuAD-Open comparison in Table 3 for systems evaluated on it.

The reason why TF-IDF remains competitive on SQuAD is that the original DPR study [9] itself reports that BM25 *outperforms* their dense retriever specifically on SQuAD dataset, among the five different datasets they evaluated. Their reported Top-20 retrieval accuracy is 68.8 % for BM25 versus 63.2 % for DPR. It is due to two properties observed in SQuAD, (i) its questions were written by annotators *with the passage already visible* by inducing strong lexical overlap not present in NQ or TriviaQA, and (ii) its passages are drawn from only ∼500 Wikipedia articles, a narrow and non-representative slice of the corpus that dense retrievers trained on broader web-scale distributions struggle to match. This is the same effect we independently observe in Section 4.3 and Table 7 where we report MiniLM-v6 and Contriever collapse at full corpus scale, while TF-IDF remains stable.

Table 4. Exact Match (EM) of representative RAG and dense-retrieval frameworks on Natural Questions (NQ) and TriviaQA benchmarks.

System	NQ	TriviaQA	Retriever
REALM [5]	40.4	—	Latent (learned)
DPR [9]	41.5	56.8	Dense (dual-encoder)
RAG [6]	44.5	56.1	Dense (DPR)
Fusion-in-Decoder (FiD) [10]	51.4	67.6	Dense (DPR) + fusion

3 Dataset and Methodology

3.1 Dataset

We use the Stanford Question Answering Dataset (SQuAD) [13], comprising crowd-sourced question-answer pairs over Wikipedia articles where every answer is a verbatim text span. After deduplication, the corpus contains 18 881 unique context passages and 86 769 question-answer pairs. Table 5 reports key statistics. SQuAD's high annotation quality and consistent structure make it well-suited for evaluating both retrieval and extractive QA.

3.2 ECR-LLM Pipeline Overview

Fig 1 illustrates the end-to-end ECR-LLM architecture. Five sequential stages transform a raw query into up to ranked answers, combining sparse retrieval, semantic precision, deep-learning span extraction, structured knowledge, and neural text generation.

3.3 TF-IDF Lexical Retrieval (Stage 1)

For each query q, every passage d in the corpus is scored using the standard TF-IDF formulation using Eq.(1).

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \times \text{IDF}(t), \tag{1}$$

Table 5. SQuAD dataset statistics after deduplication.

Attribute	Value
Articles	442
Total contexts	19 035
Unique contexts	18 881
Total questions	86 769
Avg. context length	116 words
Avg. question length	11 words
Min / max context words	20 / 653
Answer type	Span-based

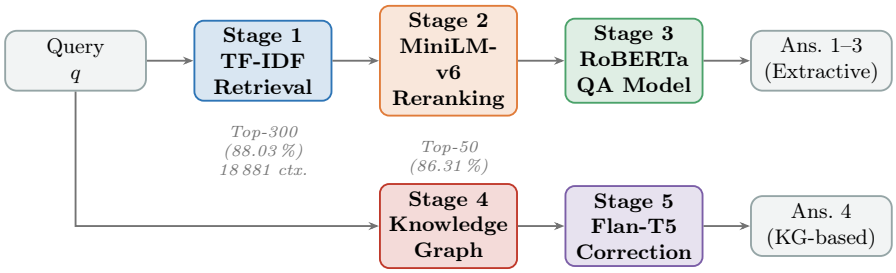

Fig. 1. ECR-LLM five-stage pipeline. The upper branch (Stages 1–3) performs cascaded lexical and semantic retrieval followed by extractive span prediction, yielding three ranked answers. The lower branch (Stages 4–5) queries a pre-built Knowledge Graph and corrects its output for fluency, yielding a structured answer per query.

where $\mathrm{TF}(t,d) = f_{t,d} / \sum_{t' \in d} f_{t',d}$, with $f_{t,d}$ the raw count of term t in document d, and $\mathrm{IDF}(t) = \log(N/\mathrm{DF}(t))$, with N the total corpus size and $\mathrm{DF}(t)$ the number of documents containing t. Documents are ranked by the aggregate TF-IDF score over query terms and the top-k are retained. We set $k = 300$ (1.59 % of the corpus) based on the experiments reported in Section 4, which yields 88.03 % retrieval accuracy (answer-containing passage present in top-300).

3.4 Semantic Reranking with MiniLM-v6 (Stage 2)

The 300 TF-IDF candidates are re-scored using cosine similarity of MiniLM-v6 sentence embeddings using Eq.(2).

$$\mathrm{score}(q,d) = \cos(\boldsymbol{E}_q, \boldsymbol{E}_d) = \frac{\boldsymbol{E}_q \cdot \boldsymbol{E}_d}{\|\boldsymbol{E}_q\| \, \|\boldsymbol{E}_d\|}, \tag{2}$$

where $\boldsymbol{E}_q$ and $\boldsymbol{E}_d$ are the query and document embeddings produced by the bi-encoder. MiniLM is trained with a contrastive loss using Eq.(3).

$$\mathcal{L} = -\log \frac{\sigma(q,d^+)}{\sum_{d' \in \mathcal{D}} \sigma(q,d')}, \quad \sigma(q,d) = \exp\big(\mathrm{score}(q,d)/\tau\big), \tag{3}$$

where d^+ is a relevant passage and τ is a temperature hyperparameter. The top-$m = 50$ passages (0.26 % of corpus) are forwarded to the reader. Post-reranking accuracy is 86.31 %; MRR rises from 0.349 (TF-IDF alone) to 0.626, reflecting a large improvement in relevant-passage rank.

3.5 Extractive QA with RoBERTa (Stage 3)

Each of the top-50 passages is concatenated with the query as [CLS] q [SEP] c [SEP] and fed to deepset/roberta-base-squad2. The model produces start and end logits s_i, e_i over passage tokens using softmax in Eq.(4).

$$P_{\mathrm{start}}(i) = \frac{e^{s_i}}{\sum_j e^{s_j}}, \quad P_{\mathrm{end}}(j) = \frac{e^{e_j}}{\sum_j e^{e_j}}. \tag{4}$$

Fig. 2. Simplified Knowledge Graph constructed from SQuAD passages. Nodes are colour-coded by type: Context (yellow-green), Question (orange), and Answer (green). Edges represent semantic relations extracted via NER and RE, forming the (s, p, o) triple store queried in Stage 4.

The span (i^*, j^*) maximising $P_{\text{start}}(i) \cdot P_{\text{end}}(j)$ subject to $j - i \leq L$ is returned with a confidence score. The three highest-confidence spans across all 50 passages are retained as the primary answer candidates. The model achieves EM 85.29 and F1 91.84 on SQuAD v1.1.

3.6 Knowledge Graph Retrieval (Stage 4)

A KG $\mathcal{K}$ is pre-built over the full SQuAD corpus as (s, p, o) triples via Named Entity Recognition (NER) and Relation Extraction (RE). For query q, target entity s' and relation p' are extracted. $\mathcal{K}$ is searched for matching triples (s', p', o'). The object o' is returned as a structured answer.

Fig 2 shows a sample Knowledge Graph constructed from SQuAD passages. The KG serves as a logical fallback, which is particularly useful for short, fact-based questions where span extraction is uncertain or multi-hop inference is needed.

3.7 Flan-T5 Fluency Correction (Stage 5)

The google/flan-t5-base is prompted with the instruction `"Answer the question: {q} Context: {o}"` to produce a grammatically fluent, human-readable response. KG-extracted answers are often in the form of triples. Flan-T5-base (250 M parameters) is chosen for its instruction-following ability and low inference cost, making it compatible with single-GPU deployment.

3.8 Final Answer Aggregation

Each query receives four answers: three extractive answers from Stage 3 ranked by RoBERTa confidence and one KG-refined answer from Stage 5. This multi-source output raises coverage, handles query ambiguity, and enables graceful degradation when any single component fails.

4 Experiments and Results

4.1 Evaluation Metrics

Retrieval Metrics. We compute five retrieval metrics for $k \in \{1, 2, 3, 5, 10, 20, 50, 100, \ldots\}$.

MAP averages precision over ranks where relevant documents appear, and since each question has exactly one relevant context, MRR equals MAP.

$$\text{MAP} = \frac{1}{Q} \sum_{q} \text{AP}_q, \qquad \text{AP} = \frac{1}{R} \sum_{k} P(k) \cdot \text{rel}(k). \tag{5}$$

MRR is the mean reciprocal rank of the first relevant result.

$$\text{MRR} = \frac{1}{Q} \sum_{q} \frac{1}{\text{rank}_q}. \tag{6}$$

nDCG penalises relevant documents at lower ranks.

$$\text{DCG@}K = \sum_{i=1}^{K} \frac{2^{\text{rel}_i} - 1}{\log_2(i+1)}, \quad \text{nDCG@}K = \frac{\text{DCG@}K}{\text{IDCG@}K}. \tag{7}$$

Precision@k and *Recall@k* are defined as shown in Eq. (8).

$$Precision@k = \frac{1}{k} \sum_{i=1}^{k} \text{rel}(i), \qquad \text{Recall@}k = \frac{|\text{Rel}_k|}{|\text{Rel}|}. \tag{8}$$

Generation Metrics. ROUGE-n measures n-gram recall overlap between the candidate and reference answer using Eq.(9).

$$\text{ROUGE-}n = \frac{\sum_w \min(\text{count}_{\text{cand}}(w),\, \text{cnt}_{\text{ref}}(w))}{\sum_w \text{count}_{\text{ref}}(w)}. \tag{9}$$

ROUGE-L uses the longest common subsequence. *F1* is the token-level harmonic mean of precision and recall. *Exact Match (EM)* awards full credit only for answers that match the gold string exactly after lower-casing, punctuation removal, and article stripping using Eq.(10).

$$\text{EM} = \frac{\text{exact matches}}{Q}. \tag{10}$$

Table 6. Single-model retrieval comparison at Top-100 (full corpus, 86 769 queries).

Model	Acc. (%)	MRR	nDCG	Recall (%)	Prec.	MAP
TF-IDF	**81.83**	**0.3487**	**0.4444**	**81.83**	**0.3487**	**0.0082**
Contriever [7]	13.83	0.0455	0.0632	13.83	0.0455	0.0014
MiniLM-v6	12.77	0.0600	0.0700	12.77	0.0600	0.0012

Table 7. Retrieval Accuracy@100 vs. dataset size for TF-IDF, Contriever, and MiniLM-v6. Dense models degrade sharply at scale.

Model	2 000	5 000	10 000	20 000	86 769
TF-IDF	85.60	83.06	82.94	81.87	**81.84**
Contriever	*95.40*	*95.40*	82.41	49.64	13.83
MiniLM-v6	*98.15*	*98.12*	81.48	46.97	12.77

4.2 Retriever Comparison at Top-100

Table 6 compares TF-IDF, Contriever, and MiniLM-v6 as standalone retrievers on the full 86 769-question corpus at Top-100. TF-IDF substantially outperforms both dense models. the SQuAD corpus exhibits high lexical self-similarity as questions are crowd-sourced directly from the passage text, making keyword overlap a strong signal. Dense models have a disadvantage here because their embedding spaces are trained on different distributions.

4.3 Retriever Scalability Across Dataset Sizes

A key practical question is whether the choice of retriever should depend on corpus size. Table 7 reports Accuracy@100 for all three models as the evaluation corpus increases from 2 000 to the full 86 769 questions.

On small subsets ($n \leq 5\,000$), both dense models far outperform TF-IDF. MiniLM-v6 achieves 98.15 % vs. TF-IDF's 85.60 % at $n = 2\,000$. However, accuracy declines sharply as the corpus expands. Contriever drops from 95.40 % to 13.83 % and MiniLM-v6 drops from 98.15 % to 12.77 % on the full corpus. Meanwhile, TF-IDF remains stable at 81.84 %. This behaviour stems from the structure of SQuAD where questions are crowd-sourced directly from their answer passages that creates strong lexical overlap that TF-IDF exploits effectively. Dense models trained on different distributions cannot bridge this distribution mismatch at scale without domain-specific fine-tuning. These findings justify TF-IDF as the first-stage retriever and motivate the hybrid cascade.

4.4 Effect of Top-k on TF-IDF Retrieval

Table 8 shows retrieval accuracy as k varies from 100 to 1 000. Accuracy improves monotonically but with diminishing returns beyond $k = 300$, which is an optimal trade-off between recall and computational cost, i.e., 88.03 % $\rightarrow$ 90.52 % at $k =$ 600, a gain of only 2.5% at double the candidate volume.

Table 8. TF-IDF retrieval accuracy versus top-k (full corpus). Gains saturate beyond $k = 300$.

k	Acc. (%)	MRR	nDCG	Recall	Precision	MAP
100	81.84	0.3487	0.4444	0.8184	0.3487	0.0082
200	86.15	0.3491	0.4505	0.8615	0.3491	0.0043
250	87.27	0.3491	0.4519	0.8727	0.3491	0.0035
300	**88.03**	0.3491	0.4528	0.8803	0.3491	0.0029
350	88.68	0.3492	0.4536	0.8868	0.3492	0.0025
400	89.18	0.3492	0.4542	0.8918	0.3492	0.0022
500	89.95	0.3492	0.4551	0.8995	0.3492	0.0018
600	90.52	0.3492	0.4557	0.9052	0.3492	0.0015
700	90.99	0.3492	0.4562	0.9099	0.3492	0.0013
800	91.44	0.3492	0.4567	0.9144	0.3492	0.0011
1000	92.08	0.3492	0.4573	0.9208	0.3492	0.0009

Table 9. Semantic reranking (MiniLM-v6) performance versus top-m (after TF-IDF top-300 candidates).

m	Acc. (%)	MRR	nDCG	Recall	Prec.	MAP
1	53.39	0.5339	0.5339	0.5339	0.5339	0.5339
5	73.91	0.6145	0.6458	0.7391	0.6145	0.1478
10	79.29	0.6218	0.6633	0.7929	0.6218	0.0793
20	83.21	0.6246	0.6733	0.8321	0.6246	0.0416
50	**86.31**	0.6256	0.6796	0.8631	0.6256	0.0173
75	87.12	0.6258	0.6809	0.8712	0.6258	0.0116
100	87.47	0.6258	0.6815	0.8747	0.6258	0.0087

4.5 Semantic Reranking with MiniLM-v6

After TF-IDF retrieval with $k = 300$, MiniLM-v6 reranks and the top-m passages are retained. Table 9 reports retrieval metrics as m varies. At $m = 50$, accuracy is 86.31 %, which is 1.72% below the TF-IDF ceiling while MRR jumps from 0.349 to 0.626. The correct passage now appears much higher in the shortlist, which is critical for the RoBERTa reader that processes only the top-10 reranked passages. Accuracy, MRR, and nDCG all plateau after $m = 50$, which we adopt as the operating point.

4.6 RoBERTa Extractive QA Performance

On SQuAD v1.1 validation, deepset/roberta-base-squad2 achieves EM 85.29 and F1 91.84. On the harder SQuAD v2 split including unanswerable questions achieve EM 79.93 and F1 82.95 as the model's benchmark performance shown in Table 10.

Within the pipeline, only the top-10 of the 50 reranked passages are processed due to inference-budget constraints, causing a 4.98% accuracy reduction from 86.31 % → 81.33 %. Of the 74 893 correctly retrieved contexts (86.31 %

Table 10. Evaluation of deepset/roberta-base-squad2 on SQuAD benchmarks.

Dataset	EM	F1
SQuAD v1.1 (validation, self-reported)	85.29	91.84
SQuAD v2.0 (validation, verified)	79.93	82.95

Table 11. Generation quality of extractive QA vs. KG-augmented output (2 000-sample evaluation set).

Strategy	R-1	R-2	R-L	EM	F1
QA Max	0.555	0.335	0.555	0.465	0.540
KG Only	**0.725**	**0.477**	**0.725**	**0.725**	**0.722**

of 86 769), their distribution across post-reranking ranks is as follows: rank 1: 43 457 (58.0%), rank 2: 8 136 (10.9%), rank 3: 3 650 (4.9%), rank 4: 2 181 (2.9%), rank 5: 1 451 (1.9%), ranks 6–10: 3 575 (4.8%), and ranks 11–50: 6 219 (8.3%). The strong mass at rank 1 confirms that semantic reranking effectively promotes the correct passage to the top position in the majority of queries. The three highest-confidence spans across all processed passages are returned as the primary answer candidates, ranked by $P_{\text{start}}(i^*) \cdot P_{\text{end}}(j^*)$ (Eq. 4).

4.7 Knowledge Graph Integration and Generation Quality

Table 11 and Fig 3 compare two output strategies on a 2 000-sample evaluation set: (a) the best of the three RoBERTa extractive answers (*QA Max*) and (b) the KG-only answer after Flan-T5 correction (*KG Only*).

The KG-only strategy outperforms the extractive reader on every metric. The 26% EM gain from 0.465 → 0.725 confirms that for the short, fact-based answers prevalent in SQuAD, structured triple retrieval followed by Fine-tuned Language Net (Flan)-T5 fluency correction produces cleaner matches than span extraction from reranked passages.

4.8 Pipeline Latency

Table 12 reports average per-query latency over 1 000 queries in two configurations with and without Flan-T5 KG correction. The Flan-T5 correction reduces total latency from 244 ms to 132 ms by replacing a heavier triple-expansion step with a lightweight generation pass. Vector search and QA inference together account for over 82 % of total query time.

Table 13 shows per-query latency for five representative queries spanning the observed range. Query latency is mainly due to QA inference with RoBERTa, which is roughly constant at 68–105 ms. The KG module latency varies from 17 ms for Q9 to 282 ms for Q3. This difference shows how complex the knowledge-graph traversal is for each target entity-relation pair. The total latency for the

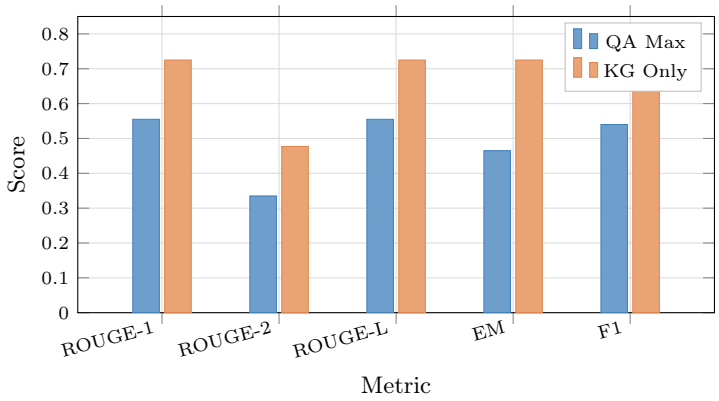

Fig. 3. Generation metrics for extractive QA (QA Max) versus KG-augmented output (KG Only) on the 2 000-sample evaluation set. KG integration improves all five metrics, with the largest absolute gain on Exact Match.

Table 12. Average per-query pipeline latency (1 000 queries).

Stage	without KG Correction	with KG Correction
Vector search (TF-IDF)	0.0332	0.0329
Semantic reranking (MiniLM)	0.0008	0.0059
QA inference (RoBERTa)	0.0775	0.0765
Knowledge graph module	0.1322	0.0167
Scoring logic	0.0001	0.0001
Total	**0.244**	**0.132**

base configuration remains below 150 ms for all queries. This confirms that the pipeline is suitable for real-time use.

4.9 Qualitative Evaluation on Representative Queries

We evaluated the pipeline on ten selected queries that cover different answer types (factual, named entity, numerical, causal). These queries highlight various failure modes such as Q1 (recycling disposal) and Q9 (scale definitions) favour KG-based exact answers, Q3 (nationalism in India) and Q10 (asthma hypothesis) need longer contextual reasoning where extractive spans are more common.

Across the ten queries, the KG-linked configuration provided exact factual answers for queries 1, 4, 5, and 9. These answers were either numerical or named entities. On the other hand, the extractive RoBERTa branch performed better for queries 3, 6, and 10, which required answers at the phrase or clause level. This combination shows that no single answering method works best for all types of questions. This finding supports the answer-per-query aggregation design of ECR-LLM Queries 2, 7, and 8 demonstrated a high level of agreement between

Table 13. Per-query stage-wise latency for five representative queries spanning the observed range i.e., Base (no KG) vs. KG-linked (in seconds).

	Vec. Search		Sem. Rerank		QA Infer.		KG Module		Total	
Q#	Base	KG	Base	KG	Base	KG	Base	KG	Base	KG
1	.034	.033	.001	.006	.078	.077	.018	.067	.130	.183
3	.033	.033	.001	.006	.092	.092	.021	.282	.147	.413
5	.033	.033	.001	.005	.075	.076	.032	.150	.141	.264
8	.034	.033	.001	.006	.105	.100	.001	.070	.140	.209
9	.033	.033	.001	.006	.068	.066	.011	.060	.112	.164

KG and extractive answers. This suggests that both branches consistently deliver reliable results for simple entity lookups.

4.10 Generalisation Protocol for Multi-Hop QA on HotpotQA

SQuAD is a single-hop dataset on which we define the protocol for extending ECR-LLM to HotpotQA [11], a 112 779-question benchmark that requires reasoning across Wikipedia paragraphs per answer. Unlike SQuAD, HotpotQA additionally requires *supporting-fact* identification, scored separately from answer accuracy, with a *joint* EM/F1 that credits a prediction only when both the answer and its full supporting-sentence set are correct.

Single-hop retrieval stages 1–2 of ECR-LLM cannot directly satisfy HotpotQA's two-hop structure, since the second supporting fact is often not lexically or semantically linked to the question. We propose an iterative extension: (i) run Stage 1–2 retrieval on the question to obtain a first-hop passage set; (ii) extract candidate bridge entities from the top-ranked first-hop passages; (iii) re-query TF-IDF/MiniLM-v6 using the question concatenated with each bridge entity to retrieve second-hop passages, following the iterative-retrieval strategy used by Multi-hop Dense Retrieval [12]; (iv) supply the concatenated first- and second-hop passages to the Stage 3 RoBERTa reader with a sentence-level supporting-fact classification head added, and (v) query the KG (Stage 4) over both bridge and target entities before Flan-T5 correction (Stage 5).

4.11 Limitations

Retriever and reader models were used off-the-shelf; the domain fine-tuning would likely yield further gains, particularly for dense retrieval on out of distribution corpora. The KG is invoked unconditionally; a learned confidence-based trigger, for instance, skip KG when RoBERTa confidence exceeds a threshold would eliminate unnecessary latency for simple queries. The hyperparameters k and m were selected by ablation rather than automated tuning. The protocol defined for HotpotQA benchmark in Sections 4.10 is designed to narrow down the single-dataset and module-attribution gaps in future work.

5 Conclusion

We presented ECR-LLM, which is a five-stage hybrid retrieval and answering framework for open-domain QA that combines TF-IDF lexical retrieval, MiniLM-v6 semantic reranking, RoBERTa extractive span prediction, Knowledge Graph retrieval, and Flan-T5 fluency correction. Evaluated on 86 769 SQuAD question-answer pairs, the pipeline achieves 88.03 % initial retrieval accuracy and 86.31 % post-reranking accuracy. KG integration raises Exact Match from 0.465 to 0.725 and F1 from 0.540 to 0.722, and the full pipeline runs in < 250 ms per query on a single GPU. Our scalability analysis reveals a practically important finding that dense retrievers such as MiniLM-v6 and Contriever achieve very high accuracy on small corpora (98.15 % and 95.40 % respectively at $n = 2\,000$) but collapse to below 14 % on the full 86 769-question corpus, while TF-IDF remains stable at 81.84 %. This cross-over effect arising from distribution mismatch between pre-training data and the SQuAD corpus justifies TF-IDF as the first-stage retriever for datasets with high lexical self-similarity and establishes the hybrid cascade as the principled design choice. The qualitative analysis of representative queries confirms that different answer types favour different pipeline branches. KG-based retrieval dominates for short factual numerical, named-entity answers, while extractive span prediction dominates for clause-level answers. The code and dataset are publicly available at https://github.com/veningstonk/Effective-Context-Retrival-For-LLM-FrameWork.

References

1. Mzwri, K., Turcsányi-Szabó, M. Internet wizard for enhancing open-domain question-answering chatbot knowledge base in education. Applied Sciences **13**(14), 8114 (2023). https://doi.org/10.3390/app13148114
2. Chen, D., Fisch, A., Weston, J., Bordes, A. Reading Wikipedia to answer open-domain questions. In: Proc. 55th ACL, vol. 1, pp. 1870–1879. ACL (2017) https://doi.org/10.18653/v1/P17-1171
3. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proc. NAACL-HLT 2019, pages 4171–4186 arXiv preprint arXiv:1810.04805 (2018) https://doi.org/10.48550/arXiv.1810.04805
4. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V. RoBERTa: A robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
5. Guu, K., Lee, K., Tung, Z., Pasupat, P., Chang, M.-W. REALM: Retrieval-augmented language model pre-training. In: Proc. 37th ICML, Article No.: 368, pp. 3929–3938. PMLR (2020) https://doi.org/10.48550/arXiv.2002.08909
6. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., Kiela, D. Retrieval-augmented generation for knowledge-intensive NLP tasks. (2020) In Proc. 34th International Conference on Neural Information Processing Systems (NIPS 2020). Curran Associates Inc., Red Hook, NY, USA, Article 793, 9459–9474.

7. Izacard, G., Caron, M., Hosseini, L., Riedel, S., Bojanowski, P., Joulin, A., Grave, E. Contriever: Unsupervised dense passage retrieval with contrastive learning. arXiv preprint arXiv:2201.12129 (2022), Findings of the Association for Computational Linguistics: ACL 2023, pages 10932–10940. https://doi.org/10.48550/arXiv.2112.09118
8. Wang, L., Yang, N., Huang, X., Jiao, B., Yang, L., Jiang, D., Majumder, R., Wei, F. Text embeddings by weakly-supervised contrastive pre-training. arXiv preprint arXiv:2212.03533 (2024) https://doi.org/10.48550/arXiv.2212.03533
9. Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., Yih, W. Dense passage retrieval for open-domain question answering. In: Proc. EMNLP, pp. 6769–6781 (2020) https://doi.org/10.18653/v1/2020.emnlp-main.550
10. Izacard, G., Grave, E. Leveraging passage retrieval with generative models for open domain question answering. In: Proc. 16th Conference of the European Chapter of the ACL (EACL), pp. 874–880 (2021) https://doi.org/10.18653/v1/2021.eacl-main.74
11. Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W.W., Salakhutdinov, R., Manning, C.D. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In: Proc. EMNLP, pp. 2369–2380 (2018) https://doi.org/10.18653/v1/D18-1259
12. Xiong, W., Li, X.L., Iyer, S., Du, J., Lewis, P., Wang, W.Y., Mehdad, Y., Yih, W., Riedel, S., Kiela, D., Oguz, B. Answering complex open-domain questions with multi-hop dense retrieval. arXiv preprint arXiv:2009.12756 (2020) https://doi.org/10.48550/arXiv.2009.12756
13. Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P. SQuAD: 100 000+ questions for machine comprehension of text. In: Proc. Empirical Methods in Natural Language Processing (EMNLP), pp. 2383–2392 (2016) https://doi.org/10.18653/v1/D16-1264
14. Kim, Y., Wu, J., Abdulle, Y., Wu, H. MedExQA: Medical Question Answering Benchmark with Multiple Explanations. In Proc. 23rd Workshop on Biomedical Natural Language Processing, pp. 167–181 (2024), Bangkok, Thailand. Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.bionlp-1.14
15. Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C.H., Kang, J. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics **36**(4), 1234–1240 (2020) https://doi.org/10.1093/bioinformatics/btz682
16. Ben Abacha, A., Demner-Fushman, D. On the Summarization of Consumer Health Questions. In Proc. 57th Annual Meeting of the Association for Computational Linguistics, pp. 2228—2234 (2019) https://doi.org/10.18653/v1/P19-1215
17. Kpodo, J., Kordjamshidi, P., Nejadhashemi, A. P. AgXQA: A benchmark for advanced Agricultural Extension question answering. Computers and Electronics in Agriculture, 225(1):109349 (2024) https://doi.org/10.1016/j.compag.2024.109349