

DOI: https://doi.org/10.48009/3_iis_2026_130

Retrieval-augmented question answering over a Romanian news corpus: An end-to-end system with an open-weight LLM

Adrian Vintilă, *Bucharest University of Economic Studies*, adi.vint@gmail.com
Constanța-Nicoleta Bodea, *Bucharest University of Economic Studies*, bodea@ase.ro

Abstract

Retrieval-augmented generation (RAG) is a promising approach for grounded question answering, but its practical use in lower-resource languages and low-infrastructure settings remains less documented. This paper presents the design and evaluation of a RAG system that grounds an open-weights large language model (LLM) in a Romanian news corpus. The system answers Romanian-language questions over 14,263 fixed news articles. A single Romanian-optimized LLM performs both query understanding and grounded answer generation, supported by hybrid sparse and dense retrieval, recency-aware reranking, article selection, and a fixed refusal protocol for questions outside the corpus. The full pipeline runs on a single consumer workstation, making the design relevant for small organizations that cannot rely on large infrastructure. Evaluation uses a hand-authored 70-item gold set covering factual lookup, temporal lookup, entity disambiguation, multi-document synthesis, and not-in-corpus refusal. Retrieval, refusal, and generation are measured separately, and three retrieval-only ablations test the contribution of hybrid retrieval, recency weighting, and query understanding. Under one locked configuration, the system achieves overall Recall@5 of 0.838 and MRR of 0.825 on answerable items, refusal F1 of 0.952 with no false negatives, keypoint coverage of 0.819, and a mean judge-faithfulness of 3.28 out of 5. The results show that open-weights Romanian news RAG is feasible on consumer hardware and performs best on focused factual and entity-disambiguation questions. A complementary error analysis locates the residual failure modes in wide-scope multi-document queries, a small number of temporal misses, and generation-side errors involving numeric facts or unsupported detail from adjacent articles. The study introduces a fixed Romanian news corpus, a gold set of authored evaluation questions, and an evaluated end-to-end pipeline for grounding an open-weights LLM in Romanian news.

Keywords: retrieval-augmented generation (RAG), Romanian news corpus, open-weight large language models, news question answering, LLM-as-judge evaluation, hybrid retrieval

Introduction

Retrieval-augmented generation (RAG) is a practical pattern for grounded question answering over a defined text collection (Lewis et al., 2020). It separates evidence selection from answer generation: the retriever identifies relevant documents or passages, and the language model produces a fluent response constrained by that evidence. This separation is especially important for news applications, where answers depend on recent, domain-specific material that may not be represented reliably in a model's parameters. Romanian, like many other languages, is comparatively under-resourced in openly available pre-trained models and public benchmarks (Dumitrescu et al., 2020; Martins et al., 2025). At the same time, the system

is aimed at settings such as small newsrooms, academic groups, and civic-technology projects, where low-infrastructure deployment is often desirable. This points to a narrower but practically important and less documented setting: a non-English news question answering RAG system using open-weights components on consumer hardware.

This paper reports the design and end-to-end evaluation of a retrieval-augmented question-answering system for that setting. Its fixed evidence base is a corpus of 14,263 Romanian news articles, constructed and frozen for this study. All retrieval and generation components are open-weights, including a Romanian-optimized large language model (LLM) that handles both the query-understanding stage and the final answer generation. The full pipeline runs on a mid-range desktop with a consumer GPU. The design centres on five selected query classes: factual lookup, temporal lookup, entity disambiguation, multi-document synthesis, and not-in-corpus refusal.

The study is organised around three research questions. First, it asks what retrieval and generation quality such a system can achieve on grounded Romanian-language news across the supported query classes. Second, it asks how three design choices, hybrid sparse-plus-dense retrieval, recency-aware reranking, and an LLM-driven query-understanding stage, perform. Third, it analyses the main failure modes observed in the resulting system and what they suggest about the limits of query-conditioned retrieval for news question answering.

The system is exercised against a seventy-item author-created gold set spanning the five query classes, under a single binding configuration and without any hyperparameter tuning on the evaluation set. Retrieval, generation, and refusal are measured separately, and the three design choices above are evaluated through retrieval-only ablations. The paper therefore makes three coordinated contributions: the frozen Romanian news corpus, the author-created gold set, and the evaluated end-to-end RAG pipeline that uses them, all developed under an open-weights, consumer-hardware constraint.

Literature Review

Retrieval-augmented generation couples a neural retriever over a non-parametric evidence base with a parametric language model that conditions its output on the retrieved passages. Lewis et al. (2020) introduce RAG models in which a dense retriever feeds a pre-trained sequence-to-sequence generator, achieving strong open-domain question answering (QA) performance while preserving interpretable, updateable external memory. Following this line, our pipeline is staged, and we report retrieval, generation, and refusal as separate components rather than collapsing them into a single end-to-end score.

Classical probabilistic retrieval, in the form of BM25 within the Probabilistic Relevance Framework (Robertson & Zaragoza, 2009), remains a strong baseline that Dense Passage Retrieval (Karpukhin et al., 2020) was shown to surpass on top-k recall through dual-encoder training over question-passage pairs. More recent multilingual work consolidates sparse and dense scoring: M3-Embedding (Chen et al., 2024) unifies dense, multi-vector, and sparse retrieval in a single multilingual encoder, while the BGE family of resources (Xiao et al., 2024) packages compatible multilingual rerankers. Cross-encoder rescoring of first-stage candidates, popularised by Nogueira and Cho (2019), is now the standard reranking pattern, and Reciprocal Rank Fusion (Cormack et al., 2009) supplies a robust parameter-free combiner for sparse and dense lists. Our system combines BM25, dense M3 embeddings, and a cross-encoder reranker, and a dense-only ablation isolates the added value of the hybrid and reranking stages.

Large language models can also act as front-end controllers that rewrite or expand user prompts to better match the retriever. Hypothetical-document methods such as HyDE (Gao et al., 2023) generate pseudo-documents from the input query to drive zero-shot dense retrieval, while the Rewrite-Retrieve-Read pipeline of Ma et al. (2023) adapts queries to the needs of a frozen retriever and reader, with consistent gains on open-domain QA. Time-based language models (Li & Croft, 2003) formalise how document timestamps and a reference time should enter retrieval scoring for temporally scoped questions. Our

pipeline includes an LLM-based query understanding stage performing query rewriting, intent classification, entity extraction, and temporal resolution, and we ablate this stage.

Recency-aware ranking is especially consequential in news, where relevance decays rapidly. Independently, abstention has become a first-class evaluation dimension: SQuAD 2.0 (Rajpurkar et al., 2018) introduces unanswerable questions explicitly, and retrieval-conditioned self-reflection (Asai et al., 2024) shows that systems can be trained to abstain when evidence is weak, trading recall for reliability. Our design combines a time-aware exponential recency decay applied to reranker scores when the query carries no explicit date filter with a fixed-string refusal protocol aligned with this prior work on unanswerable QA and calibrated RAG.

Evaluating RAG pipelines requires metrics that separately probe retrieval, grounding, and generation. Standard measures such as Recall@k and Mean Reciprocal Rank remain the workhorses of retrieval evaluation in open-domain QA. RAG-specific frameworks such as RAGAS (Es et al., 2024) and ARES (Saad-Falcon et al., 2024) introduce automated metrics covering context relevance, answer faithfulness, and answer relevance, often using LLMs as verifiers over retrieved passages. LLM-as-judge setups carry their own pathologies, and Zheng et al. (2023) document position and verbosity biases that motivate careful protocol design. We adopt a two-track evaluation comprising automated retrieval and answer metrics together with a reference-based, keypoint-coverage LLM-as-judge protocol on a 70-item gold set tailored to Romanian news question answering.

Large transformer models for Romanian have only recently become available. The Romanian BERT of Dumitrescu et al. (2020) shows that monolingual pre-training improves a range of downstream tasks over multilingual baselines, and EuroLLM (Martins et al., 2025) provides open-weight models covering all twenty-four EU official languages with explicit attention to lower-resource ones such as Romanian. In parallel, work on efficient inference has made on-device deployment of multi-billion-parameter models feasible: GPTQ (Frantar et al., 2022) shows that three- to four-bit weight-only quantization can compress large models with minimal accuracy loss, and open inference runtimes for quantized weights bring on-device deployment within reach on commodity hardware. To our knowledge, however, there is still little published work on fully open-weights Romanian news RAG pipelines evaluated end-to-end under an 8 GB consumer-GPU constraint. The present paper addresses this gap, situating Romanian news QA at the intersection of low-resource language modeling, hybrid retrieval, LLM-driven query understanding, time-aware ranking, and on-device RAG deployment.

Contribution

The main contribution is an end-to-end measurement package for Romanian news question answering. It combines a frozen corpus of 14,263 Romanian-language news articles fetched once from a Romanian news provider’s content API, a 70-item gold set authored for this study, and a working RAG pipeline evaluated under a single locked configuration.

The gold set contains factual, temporal, entity-disambiguation, multi-document, and not-in-corpus items. Each answerable item includes expected source articles and answer keypoints; temporal items also carry expected date filters where applicable. The pipeline composes LLM-based query understanding, hybrid retrieval, recency-aware reranking, article selection, and LLM-based grounded answer generation with a fixed-string refusal protocol. Together, these artifacts answer the three research questions: the headline run measures retrieval, generation, and refusal quality; the ablations isolate the impact of the main design choices; and the per-item records support the qualitative failure analysis. The corpus, gold set, and pipeline are structured for follow-on reuse.

Methodology

This section describes the system from corpus construction to evaluation. The work focuses on grounded question answering over a frozen Romanian-language news corpus and is evaluated across five query categories: factual lookup, temporal lookup, entity disambiguation, multi-document synthesis, and not-in-corpus refusal. The design is also constrained to consumer hardware, using a quantized 12-billion-parameter open-weights Romanian LLM, a mid-range CPU, and a consumer GPU with 8 GB of VRAM.

Corpus construction

The article corpus was built using a standalone upstream fetcher that queries the DatoCMS API of a Romanian online news publisher and serialises the result as `articles-25-7apr26.json`. The fetcher is not part of the per-query path; its output is the source consumed by ingestion. The frozen snapshot contains 14,263 articles. Each article has five fields: `slug`, `title`, `content`, `url`, and `published_at`. The `slug` is used throughout the pipeline as the primary article identifier, while `published_at` is stored as an ISO 8601 datetime with timezone offset.

The corpus is immutable for the duration of the study. The filename embeds the export date, while the semantic corpus identifier “`articles_v1`” is written into every Qdrant payload and every run log. A future re-export must increment the semantic identifier and trigger a full re-index. This separation prevents a file rename from being mistaken for the same provenance state.

System architecture and hardware envelope

As shown in Figure 1, the system is organized into five independently testable stages around a single Qdrant index. One offline stage performs chunking and embedding of the frozen JSON corpus and writes the result into the index. The remaining four stages run per query: query understanding, hybrid retrieval over the same index with optional recency weighting, cross-encoder reranking with article selection, and grounded answer generation with refusal when evidence is insufficient. A FastAPI backend exposes the online stages, and a React frontend consumes streamed output through Server-Sent Events.

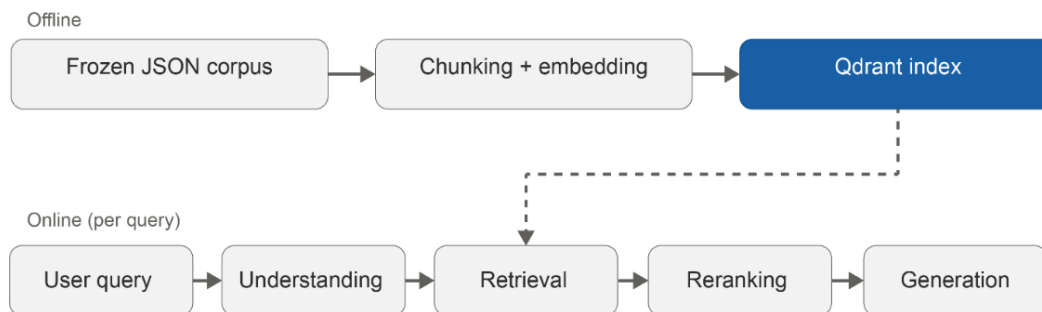


Figure 1. System architecture

Qdrant is used in embedded mode with local disk persistence. It stores dense and sparse vectors in the same collection and supports payload filtering on publication dates for temporal queries. The embedder is BAAI/bge-m3, which supports multilingual retrieval and produces both a 1024-dimensional dense vector and a sparse lexical vector from a single model. The reranker is BAAI/bge-reranker-v2-m3, a multilingual cross-encoder. The answer-generation LLM is jobautomation/OpenEuroLLM-Romanian, a Gemma 3-based 12.2 billion parameter model served through Ollama in Q4_K_M quantization. The same model is used for query understanding and answer generation, avoiding a second large model in memory.

Hardware placement follows the 8 GB VRAM constraint of the NVIDIA RTX 5060. The LLM runs on the CPU, an AMD Ryzen 7 9700X, and remains resident in system memory across query understanding and

generation. The embedder and reranker run on the GPU. Offline ingestion reuses the same workstation and may use the GPU for embedding, because the generation model is not loaded during that stage.

Ingestion and indexing

Ingestion runs once against the frozen JSON file and writes all chunks into one Qdrant collection. The process is deterministic given the same corpus, tokenizer, and embedding model revision. Articles are split on double-newline paragraph boundaries and greedily packed under a target size of 450 tokens with a hard maximum of 600 tokens, measured with the bge-m3 tokenizer. The article title is prepended to every chunk and counted against the same budget, so the final title-plus-body text never exceeds the hard cap. A sentence-boundary overlap of about 75 tokens is carried from one chunk to the next.

Each chunk is embedded with bge-m3 in a single forward pass that produces both dense and sparse vectors. These are stored in the same Qdrant point through named-vector slots, with payload metadata for slug, title, url, published_at, chunk_index, article_chunk_count, corpus_version, and the full chunk text. Storing the text in the payload allows the debug interface and error analysis to show excerpts without reloading the source JSON.

Query understanding

Each user query first passes through a structured query-understanding call to OpenEuroLLM-Romanian via Ollama JSON mode. The output has four fields: rewritten_query, entities, date_filter, and intent. The rewritten query is used for embedding and retrieval; entities are recorded for diagnostics; date_filter is either an inclusive ISO date window or null; and intent is one of factual_lookup, summary, timeline, comparison, or not_answerable.

The rewrite is intentionally narrow. It performs text normalization, including Romanian abbreviation expansion such as SUA to “Statele Unite”, UE to “Uniunea Europeană”, and CM to “Cupa Mondială”, accepts common spelling variants, and resolves temporal expressions against an injected reference date. It does not expand roles or titles from the model’s parametric knowledge, because the corpus includes events after the model’s pretraining horizon. Role and title disambiguation is left to retrieval, where the relevant phrases must be present in the corpus itself.

Temporal expressions are resolved against the corpus snapshot end date, 7 April 2026, which is injected into the prompt at every call. The handled expressions include “azi” (today), “ieri” (yesterday), “săptămâna trecută” (last week), “luna trecută” (last month), “în ultimele X zile” (last X days), “anul acesta” (this year), and named months. Tying temporal interpretation to the corpus end date rather than the host clock makes evaluation runs reproducible.

The not_answerable intent is a fast path for queries that are clearly outside a news corpus, such as recipes, real-time weather, or technical tutorials. It skips retrieval and routes directly to the refusal protocol. All other intents use the same retrieval and reranking path. The structured output is schema-validated; malformed JSON or an invalid intent triggers up to two retries at temperature zero. If all attempts fail, the system falls back to the raw query, an empty entity list, a null date filter, and factual_lookup intent. This conservative fallback mirrors the no-query-understanding baseline and is recorded in the per-item provenance. The query-understanding call is computationally expensive because it uses the same CPU-resident 12 billion parameter model as generation. Its value is therefore not assumed: the evaluation includes a query-understanding-off ablation to test whether the extra latency is repaid by retrieval quality.

Hybrid retrieval with recency weighting

The rewritten query is embedded once with bge-m3, producing dense and sparse query representations. Qdrant returns the top 40 dense neighbors and the top 40 sparse neighbors from the same hybrid collection. The two ranked lists are fused with Reciprocal Rank Fusion using a fusion constant of 60: each chunk receives a score equal to the sum, across the lists in which it appears, of the reciprocal of the sum of the

fusion constant and its rank. The top 20 fused chunks are passed to reranking. Fusion is computed in Python so that dense and sparse ranks and scores remain available for diagnostics and for the dense-only ablation. When query understanding emits a `date_filter`, that filter is applied to `published_at` before fusion. Bounds are inclusive: the from date is lifted to the start of its UTC day and the to date to the end of its UTC day. The corpus stores publication timestamps with timezone offsets, and Qdrant parses both sides to UTC for comparison.

Recency weighting is applied after reranking. The final chunk score is the reranker score multiplied by an exponential decay in article age, with age measured in days against the fixed corpus end date. The decay rate is fixed a priori at 0.0074 per day, so a 30-day-old article retains about 80% of its score, a 60-day-old article about 64%, and an article roughly 94 days old about 50%. The value is not tuned on the evaluation set, and the recency-on-off ablation tests whether the weighting is useful. Recency is disabled whenever a `date_filter` is present, because an explicit user time window already expresses the temporal constraint and should not be further biased toward the newest article inside that window.

Reranking and article selection

The 20 fused chunks are scored by BAAI/bge-reranker-v2-m3 using the rewritten query and the chunk text already stored in Qdrant. Because the title was prepended at indexing time, it is not reattached at reranking. After recency weighting where applicable, chunks are grouped by article slug. Each article receives the maximum final score among its chunks, which promotes an article based on its strongest evidence rather than averaging in weaker passages.

The top five articles are retained, and each carries forward up to two highest-scoring chunks into the generation context. These caps bound the prompt at ten chunks of no more than 600 tokens each, plus the system prompt and few-shot examples. Two further settings are fixed a priori at their most conservative values and are not tuned on the evaluation set: a reranker score floor of 0.0, which drops only chunks scored below zero, and a minimum-articles-required value of one, so the system proceeds to generation when the retained set is non-empty and otherwise emits the refusal string.

Generation and refusal protocol

The answer-generation LLM receives the system prompt, four hand-written Romanian few-shot examples, the user question, and the selected article context in a single Ollama chat request. Three examples show grounded answers, while the fourth demonstrates refusal. The generated answer is constrained to Romanian prose, one paragraph per source article, in reranked article order. The prompt forbids cross-article fact mixing within a paragraph, markdown formatting, introductions, conclusions, and inline source references. The source footer shown to the user is not generated by the LLM. The backend sends a structured source list containing slug, title, URL, publication date, and rank; the frontend renders the footer deterministically. This separation removes one possible hallucination channel, because the model cannot fabricate source titles or URLs.

Generation uses temperature 0.0, top_p 0.9, a fixed seed where supported, and a maximum output of 1024 tokens. The context window is set to `num_ctx = 8192`.

The refusal protocol is single-tier and exact. The canonical string is “Acest subiect nu se află în baza de articole disponibilă.” (This topic is not in the available article database.). No source footer is rendered on a refusal. The same string is emitted when query understanding classifies a query as `not_answerable`, when retrieval returns no article to generate from, or when the model itself refuses because the retrieved evidence does not answer the question.

Evaluation methodology

The evaluation is organized around four questions: whether retrieval finds the expected articles, whether generation grounds its answers in those articles, whether the refusal protocol behaves correctly, and whether the main design decisions are defensible against simple alternatives. The gold set, metrics,

ablations, and LLM-as-judge protocol are aligned to these questions. End-to-end query latency is additionally recorded for each item as operational context for the consumer-hardware setting. It is not one of the four evaluation questions, and the ablations and judge are not evaluated against it.

Gold dataset

The evaluation uses a hand-authored 70-item Romanian-language gold set derived from manual inspection of the frozen corpus. Each item contains an id, query, expected_article_slugs, query_type, expected_answer_keypoints, notes, and, where relevant, an expected_date_filter. The expected slugs are flat article identifiers matched directly against retrieved article slugs. The keypoints are short Romanian phrases used for local coverage scoring. For answerable items, the expected_date_filter field provides an inclusive ISO date window when the query carries temporal scope. For not-in-corpus items, expected slugs and keypoints are empty; the only required behavior is refusal. The distribution across the five query categories is given in Table 1.

Table 1. Distribution of the 70 gold-set items across the five query categories

Category	Count	Item codes
factual	20	F01 – F20
temporal	15	T01 – T15
entity_disambiguation	10	E01 – E10
multi_doc	15	M01 – M15
not_in_corpus	10	N01 – N10
Total	70	

Individual gold items are identified by a code combining the query-class initial with a two-digit index, for example F03 or T10. The categories probe distinct system properties. Factual items ask for a specific fact. Temporal items combine a topic with an explicit time window and therefore test both date resolution and filtered retrieval. Entity-disambiguation items use entities whose roles or affiliations changed within the corpus window, requiring retrieval to surface the corpus-relevant context instead of relying on pretraining associations. Multi-document items require evidence from more than one article. Not-in-corpus items are out-of-scope questions, but they are authored to include some vocabulary that could overlap with the corpus, so refusal is not measured only on trivially unrelated queries.

Gold-set authoring is therefore part of the contribution rather than only a measurement step. The queries are written in Romanian, tied to concrete article slugs, and accompanied by short keypoints and notes that document the authoring rationale. This structure separates three evaluation targets that are often combined in RAG evaluation: whether the right articles were retrieved, whether the answer covered the expected facts, and whether the system refused when the answer was absent. The construction preserves traceability from each metric back to the underlying corpus articles and makes the gold set reusable for inspection of individual failures.

Metrics

Headline retrieval quality is measured by Recall@5 and Mean Reciprocal Rank. Recall@5 is the fraction of expected article slugs appearing in the top five retrieved articles. This makes multi-document items fractional rather than any-hit: retrieving one of four expected articles gives partial credit rather than full credit. MRR is the reciprocal rank of the first expected article, averaged across eligible items. Not-in-corpus items are excluded from both metrics because they have no expected article slug.

Generation quality is measured locally by keypoint coverage and externally by judge faithfulness. Keypoint coverage normalises both the answer and each expected keypoint by lowercasing, stripping punctuation, tokenising on whitespace, and applying the Romanian Snowball stemmer; a keypoint is considered covered

if its stemmed token sequence appears as an order-preserving subsequence in the stemmed answer. Diacritics are preserved. Stemming is required because Romanian is morphologically rich and, without it, a keypoint such as “guvernul” would fail to match the inflected form “guvernului” in the answer. Items without expected keypoints are excluded. Judge faithfulness is a 1 to 5 score described below and is the headline judge metric; judge relevance and completeness are produced by the same protocol but reported as supplementary metrics.

Refusal precision, recall, and F1 are computed across the full 70-item gold set. A gold-positive item is a not-in-corpus query. A predicted positive item is any answer that exactly equals the Romanian refusal string after whitespace stripping. The not-in-corpus category contributes to refusal metrics only, while the four other categories contribute to retrieval, keypoint coverage, and judge metrics. Retrieval and generation values are reported by query category and overall; refusal values are reported overall, computed over the full 70-item gold set from the confusion matrix. Overall retrieval and generation values are item-weighted means over the 60 answerable items rather than means of slice means. Per-stage and end-to-end wall-clock latency is captured for query understanding, retrieval and reranking, and generation, and aggregated across the gold set as operational context rather than as a quality metric.

Ablations

Three retrieval-only ablations are reported against the headline configuration. Each changes one configuration value while holding the gold set, models, hardware, seeds, and temporal reference constant. The dense-only ablation disables the sparse retrieval and tests whether hybrid retrieval improves over dense retrieval when reranking is kept on. The recency-off ablation sets the decay rate to zero, making every decay factor equal to one and testing whether time-weighting helps in a recent-news corpus. The query-understanding-off ablation bypasses the structured-output call, feeds the raw query to retrieval, leaves the date filter null, and treats the intent as `factual_lookup`. All three ablations are evaluated on Recall@5 and MRR only, without rerunning generation or the judge.

LLM-as-judge

Keypoint coverage cannot detect unsupported extra claims, so the evaluation also uses a reference-based LLM-as-judge protocol. For each eligible gold item, the judge receives the Romanian query, the gold reference articles looked up by slug from the frozen corpus, the expected keypoints, and the system answer. It returns structured 1 to 5 scores for faithfulness, relevance, and completeness, with a short rationale for each dimension. The judge is not asked to produce its own answer and is not told which model generated the system answer. The judge model is Claude Opus 4.7, accessed through the Anthropic API at temperature zero.

The protocol is reference-based rather than head-to-head. The judge sees the gold reference articles rather than the retrieved chunks so that generation quality is assessed against the correct evidence, while retrieval quality is measured separately by Recall@5 and MRR. This prevents a wrong, but plausible, retrieved article from making an unsupported answer appear faithful. Each item is judged three times; the per-item value is the median of the three runs, and slice and overall values reported are means over items. The judge averages cover the 60 items in the four answerable categories; the ten not-in-corpus items are excluded because their correct output is a refusal rather than a generated answer. Faithfulness is the headline judge score because it is the direct measure of unsupported content in a grounded RAG system.

Qualitative error analysis

The Discussion section complements the aggregate metrics with a small qualitative error analysis drawn from the headline run artifacts. A few examples are selected from the per-item records to illustrate recurring failure modes rather than to form a formal taxonomy. Each example reports the original Romanian query, a short excerpt of the wrong or incomplete answer, and a concise diagnosis of the stage at which the failure occurred.

Reproducibility

Every evaluation run writes a run directory containing the resolved runtime configuration, per-item raw outputs, aggregated metrics, and pre-formatted tables. The resolved configuration records model identifiers, hyperparameters, device placement, seed, host name, git commit, run id, timestamp, and command-line overrides. Per-item outputs include the query-understanding result, retrieved slugs and scores, articles passed to generation, the generated answer, keypoint decisions, refusal decisions, and judge scores when enabled. Retrieval diagnostics retain dense rank, sparse rank, RRF score, reranker score, recency factor, and final score where available, which lets the qualitative analysis distinguish failures caused by retrieval, reranking, date filtering, or generation. Aggregated metrics are written in CSV and JSON files.

Three reproducibility decisions are especially important. First, greedy decoding at temperature zero and a global seed reduce run-to-run variance. Second, the corpus end date, 7 April 2026, is the reference time used both by temporal query understanding and by recency weighting, preventing host-clock drift from changing evaluation behavior. Third, provenance for the corpus and gold set is established by filename and internal semantic version strings, with model identifiers recorded in the resolved configuration.

Results

This section reports the empirical findings of the system described above, evaluated against the 70-item gold set under the single binding configuration recorded in the run directory of the headline run. All 70 items were processed without execution errors. Fifty-nine items produced grounded answers and eleven produced the canonical Romanian refusal string. The findings are presented as a single headline block covering retrieval, refusal, and generation, followed by the three retrieval-only ablations and per-stage latency. The supplementary judge dimensions are reproduced at the end of the section.

Headline end-to-end metrics

Table 2 reports the headline metrics by query class and overall, combining retrieval, refusal, and generation columns into a single view that follows the evaluation harness output. The overall row uses two denominators: Recall@5, MRR, and keypoint coverage are item-weighted means over the 60 answerable items, so the overall MRR of 0.825 equals $(20 \times 0.933 + 15 \times 0.800 + 10 \times 0.900 + 15 \times 0.656) / 60$, while the overall refusal precision, recall, and F1 are computed over all 70 items from the confusion matrix in Table 3. Three subsections below unpack the same table by metric family, with each subsection referring back to the relevant columns of Table 2 rather than reproducing them.

Table 2. Headline end-to-end metrics

Slice	n	R@5	MRR	Refusal Precision	Refusal Recall	Refusal F1	Keypoint coverage	Judge faithfulness
factual	20	1.000	0.933	n/a	n/a	n/a	0.901	3.300
temporal	15	0.767	0.800	n/a	n/a	n/a	0.814	3.733
entity_disambiguation	10	1.000	0.900	n/a	n/a	n/a	0.853	3.200
multi_doc	15	0.583	0.656	n/a	n/a	n/a	0.693	2.867
not_in_corpus	10	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Overall	60/70	0.838	0.825	0.909	1.000	0.952	0.819	3.283

Retrieval

Across the 60 answerable items, retrieval reaches an overall Recall@5 of 0.838 and an MRR of 0.825. Performance is highest on factual and entity-disambiguation queries, where Recall@5 reaches 1.000 and

the first expected article appears at rank one on most items, yielding MRR values of 0.933 and 0.900 respectively. Temporal queries are noticeably harder, with a Recall@5 of 0.767 and an MRR of 0.800, and multi-document queries are the weakest slice at 0.583 and 0.656. Because multi-document items can list two to four expected articles and Recall@5 is computed as a fractional hit rate over those expected slugs rather than as an any-hit indicator, a value of 0.583 means that on average a little over half of the expected articles per item enter the top-five retained set. Several multi-document items still recover all expected articles, so the slice average is dragged down by a smaller number of items whose evidence spans a wider topic.

Refusal

Refusal is evaluated as a binary classifier over the full 70-item gold set, with the fixed Romanian refusal string defining the positive class through exact match. The full confusion matrix across all 70 items is shown in Table 3.

Table 3. Confusion matrix for refusal classification

	Predicted: refusal	Predicted: answer	Total
gold: not_in_corpus	10 (TP)	0 (FN)	10
gold: answerable	1 (FP)	59 (TN)	60
Total	11	59	70

From these counts, precision is $10 / (10 + 1) = 0.909$, recall is $10 / (10 + 0) = 1.000$, and F1 is 0.952. The 10 true positives are the not-in-corpus items, all of which were correctly refused. The 59 true negatives are answerable items that received a grounded answer rather than a refusal. The single false positive is one answerable item, and there are no false negatives, so no not-in-corpus item received a grounded answer instead of the refusal string.

The false positive is item T10: the gold chunks survive both the February 2026 date filter and hybrid retrieval, but the cross-encoder scores them below zero, so the 0.0 score floor empties the retained set and the result is routed to the refusal branch. T10 is also a retrieval miss in the temporal row of Table 2, so it contributes once to the Recall@5 shortfall and once to the overall refusal precision shortfall.

Generation

Generation quality is summarised by two complementary metrics, both reported in the rightmost columns of Table 2. Local keypoint coverage across the 60 answerable items is 0.819 overall, with per-class values of 0.901 for factual, 0.853 for entity-disambiguation, 0.814 for temporal, and 0.693 for multi-document items. Mean judge faithfulness on the same 60 items is 3.283 out of 5, with multi-document the weakest slice at 2.867 and temporal the strongest at 3.733. The pattern across classes is broadly aligned: the slice with the lowest keypoint coverage is also the one with the lowest judge faithfulness, and the slices where retrieval is most reliable obtain the highest keypoint coverage. The two metrics measure different properties of the generated text. Keypoint coverage rewards each Romanian keypoint that appears as an order-preserving stem-subsequence anywhere in the answer, and so cannot detect unsupported claims that are inserted alongside covered keypoints. The reference-based judge can detect such claims and penalise them through the faithfulness rubric, which is why the absolute scale of the two metrics differs even when the slice ordering is similar.

Retrieval-only ablations

Three retrieval-only ablations were run against the same gold set, models, hardware, and seeds, each altering one configuration value while holding the rest of the pipeline fixed. Table 4 reports Recall@5 and MRR per class and overall for each configuration, with the headline row repeated for reference. No re-generation or re-judging was performed for the ablations.

Table 4. Retrieval-only ablations against the headline configuration

Slice	Metric	Headline	Dense only	Recency off	QU off
factual	Recall@5	1.000	1.000	0.925	0.975
	MRR	0.933	0.933	0.950	0.929
temporal	Recall@5	0.767	0.733	0.767	0.733
	MRR	0.800	0.767	0.800	0.606
entity_disambiguation	Recall@5	1.000	0.900	1.000	1.000
	MRR	0.900	0.800	0.883	0.900
multi_doc	Recall@5	0.583	0.561	0.572	0.572
	MRR	0.656	0.633	0.656	0.656
Overall	Recall@5	0.838	0.807	0.810	0.818
	MRR	0.825	0.794	0.828	0.775

Disabling sparse retrieval and using only the dense top-40 lowers overall Recall@5 from 0.838 to 0.807 and overall MRR from 0.825 to 0.794. The losses are largest on entity-disambiguation, where Recall@5 falls from 1.000 to 0.900 and MRR from 0.900 to 0.800, and on temporal items, where Recall@5 falls from 0.767 to 0.733 and MRR from 0.800 to 0.767. Factual items are unaffected, with both metrics identical to the headline.

Setting the decay rate to zero leaves overall MRR essentially unchanged, at 0.828 against the headline 0.825, and slightly lowers overall Recall@5 to 0.810. The effect is concentrated on the factual slice, where Recall@5 falls from 1.000 to 0.925 while MRR rises slightly to 0.950, indicating that recency weighting promotes the relevant article into the top five on a small number of factual items where multiple plausible articles compete at similar reranker scores. Temporal and entity-disambiguation results are essentially identical to the headline, as expected: temporal items disable recency through their explicit date filter, and entity-disambiguation items mostly resolve to a single article whose ranking is not sensitive to decay.

Bypassing the query-understanding stage and feeding the raw query to retrieval, with a null date filter and a factual-lookup intent, produces an overall Recall@5 of 0.818, slightly below the headline 0.838, and an overall MRR of 0.775, the lowest of the four configurations. The degradation is concentrated on temporal items, whose MRR falls from 0.800 to 0.606, because the missing date filter allows older lexically similar articles to outrank the in-window expected articles. Factual, entity-disambiguation, and multi-document slices are essentially unchanged in Recall@5, with the multi-document MRR identical to the headline.

Per-stage latency

Table 5 reports the per-stage latency distribution measured during the headline run on the binding hardware profile described in the Methodology. The end-to-end p50 is 77.0 seconds per query, with a p95 of 112.1 seconds. Generation dominates the budget at roughly three quarters of the end-to-end p50, query understanding is the second-largest component because the same CPU-resident model performs both stages, and retrieval and reranking together account for less than 2 percent of the budget.

Table 5. Per-stage latency from the headline run, p50 and p95 in milliseconds

Stage	p50 (ms)	p95 (ms)
query understanding	17,300	23,374
retrieval (hybrid + fusion + date filter)	899	1,860
reranking	347	483
generation	57,485	93,931
end-to-end total	77,017	112,056

Per-stage percentiles are computed independently and do not sum to the end-to-end total: the query at the median for one stage is generally not the median query for another.

Supplementary judge dimensions

Table 6 reports the two supplementary judge dimensions, relevance and completeness, produced by the same reference-based judge protocol and the same per-item judge records as the headline faithfulness score, so no additional inference was performed to obtain them. Overall means on the 60 answerable items are 4.133 for relevance and 4.150 for completeness. Entity-disambiguation is the strongest slice on completeness, at 4.600, while multi-document is the weakest, at 3.533, mirroring the headline pattern in Table 2.

Table 6. Judge relevance and completeness results

Slice	n	Judge relevance (mean)	Judge completeness (mean)
factual	20	4.100	4.450
temporal	15	4.200	4.067
entity_disambiguation	10	4.300	4.600
multi_doc	15	4.000	3.533
overall	60	4.133	4.150

Discussion and Qualitative Error Analysis

The headline results describe a system whose per-class quality is largely set at the retrieval stage and then propagates, with some compression, through generation, although several generation-side failures persist even when retrieval succeeds. Per-class generation quality in Table 2 tracks per-class retrieval quality across all four answerable slices: factual and entity-disambiguation items reach a Recall@5 of 1.000 and the highest keypoint coverage values, at 0.901 and 0.853; temporal items sit one step below at 0.767 and 0.814; and multi-document items are the weakest on both retrieval and generation, at 0.583 and 0.693. The closeness of the two metric families suggests that retrieval is the main driver of class-level generation quality, as predicted by the staged RAG formulation of Lewis et al. (2020), though individual items still show generation-side errors, particularly on numeric facts and details imported from adjacent retrieved articles. The temporal slice does post the highest judge faithfulness mean of any class, at 3.733, which is most plausibly explained by the date filter acting as a hard constraint on the evidence supplied to the generator and leaving little room for drift.

The three retrieval-only ablations in Table 4 isolate the contribution of each design choice. Disabling sparse retrieval costs the most on entity disambiguation, whose Recall@5 falls from 1.000 to 0.900, consistent with the sparse signal helping most when proper names disambiguate the relevant article from a topically crowded neighbourhood, and with the observation of Karpukhin et al. (2020) that dense retrievers underperform on rare entities that exact lexical matching captures directly. Disabling recency weighting leaves overall MRR essentially unchanged, with the small Recall@5 effect concentrated on factual items where multiple plausible articles compete at similar reranker scores. This qualifies the temporal priors formalised for time-sensitive retrieval (Li & Croft, 2003): in this recent-news corpus, recency acts mainly as a tie-breaker between similarly scored candidates rather than as a primary ranking signal. Bypassing query understanding is the most consequential ablation, but its effect is overwhelmingly concentrated on the temporal slice, whose MRR falls from 0.800 to 0.606 once the date filter is removed. This narrows the broad gains reported for LLM-based query rewriting in open-domain QA (Gao et al., 2023; Ma et al., 2023): here, the rewrite stage is repaid almost entirely through temporal resolution rather than through general reformulation. Table 5 shows where the time lands: generation accounts for roughly three quarters of the

median budget, retrieval and reranking together for less than two percent, and query understanding for the remaining 17.3 seconds at p50, which is therefore repaid on temporal items and not meaningfully elsewhere. The per-item records show a more concrete reading of the residual errors. The single largest contributor to the retrieval shortfall is a recall ceiling on broad multi-document queries: on items such as M03, on Romania's World Cup 2026 playoff participation, and M12, on drone incidents in the Black Sea area, none of the expected articles entered the top-five retained set, because hybrid retrieval correctly identified the topic but could not promote the small named subset over its larger topical neighbourhood. This recall ceiling is a limit of single-pass, query-conditioned retrieval rather than of any single component tested, and it is the case for which adaptive, self-reflective retrieval such as Self-RAG has been proposed (Asai et al., 2024). M11, which tracks the judicial trajectory of a single public figure, illustrates the partial-hit variant: retrieval returned five articles on that same figure, of which only one matched the expected slugs, and the generator produced a confident multi-paragraph answer covering only part of the trajectory. These items account for most of the gap between Recall@5 and keypoint coverage on the multi-document slice.

The most informative single-item failure is F03, which asks for the score of the Turkey-Romania playoff. Retrieval returned the live-text article at rank one, the article the gold set treats as the canonical source for the 1-0 result, but the LLM reported a score of 3-2. The same article was retrieved by the dense-only and recency-off ablations, which indicates that the failure is a property of the answer-generation LLM on this prompt rather than of the retriever. F03 is the cleanest demonstration in the gold set that retrieval success does not guarantee generation faithfulness on numeric facts. F02 shows a related failure with no retrieval shortfall: both expected articles were returned, but the answer states that the subject studied at "UNATC", a detail absent from both sources and most plausibly traceable to the generator's pretraining corpus, precisely the parametric knowledge that grounding in an external evidence base is intended to override (Lewis et al., 2020). A third pattern appears on F05, where retrieval returns the expected article alongside a topically related but irrelevant one and the generator pulls details from the irrelevant article into the answer, which most plausibly accounts for the persistent gap between keypoint coverage and judge faithfulness. The system performs best on focused factual and entity-disambiguation queries, remains usable but weaker on temporal queries, and struggles most on multi-document items.

Conclusions

This paper has reported the design and end-to-end evaluation of an open-weights Romanian-news RAG pipeline running on a single workstation with an 8 GB consumer GPU. Under one locked configuration, the system retrieves the expected articles with an overall Recall@5 near 0.84 and an MRR near 0.83, refuses out-of-corpus questions with an F1 of 0.95 and no false negatives, and produces grounded answers with a keypoint coverage near 0.82 and a mean judge-faithfulness of 3.3 out of 5, at a generation-dominated median latency near 77 seconds per query.

These findings answer the three research questions set out in the Introduction. Retrieval and refusal are usable in this configuration, and the ablations confirm that hybrid retrieval, recency-aware reranking, and LLM-driven query understanding each contribute to the headline numbers. The residual quality ceiling has two distinct sources: a retrieval recall ceiling on wide-scope multi-document items, whose gold expectation is a named subset of a larger topical cluster, and on a smaller number of temporal items whose in-window expected articles never enter the top five; and generation-side errors on a small number of items where the model contradicted a numeric fact or imported unsupported detail from an adjacent retrieved article. Further quality gains will therefore require work on both retrieval and the answer-generation LLM, since each is the limiting factor on a different part of the gold set.

For research, the implication is that evaluations of news RAG pipelines should report retrieval, refusal, and generation as separate quantities rather than collapsing them into a single end-to-end score. The headline run here shows why: an overall Recall@5 near 0.84 still leaves room for faithfulness errors that the retrieval

metric cannot detect, and only a generation-level metric exposes them. The implication for practice is that the deployment is feasible: an open-weights Romanian RAG pipeline can be run end-to-end on consumer hardware. Practitioners should however expect LLM-dominated latency on the order of one minute per query under this hardware envelope, and they should treat numeric facts and wide-scope multi-document synthesis as higher-risk cases that may warrant additional verification before a user-facing answer is rendered. When targeting another language, expect to replace the morphological stemmer, the text-normalisation dictionary, the generation LLM together with its few-shot examples and refusal string, and the gold set; the multilingual retrieval and reranking backbone applies unchanged.

A first limitation of this study concerns sample size: the headline metrics come from a single run against a 70-item gold set, with per-slice counts between 10 and 20, and are reported as point estimates without confidence intervals. Per-slice values are therefore sensitive to the addition or rewording of a single item, and small differences between query classes should not be over-interpreted. The retrieval-only ablations share this protocol and inherit the same per-slice variance. Beyond sample size, several scope limitations apply. The pipeline uses a single answer-generation model and a corpus drawn from a single Romanian news provider, so neither the model-level nor the publisher-level generality of the results can be claimed from this study alone. Generation quality is assessed in part by an LLM-as-judge protocol, which carries the position, verbosity, and stylistic biases documented in prior work and was not cross-checked against human raters in this study. No multi-model comparison was performed, so the headline numbers describe one open-weights configuration rather than the relative strength of competing models on the same Romanian news evaluation.

These limitations define the future work: the generation assessment will be repeated with human judges and their ratings compared against those of the LLM judge; the gold set will be enlarged to support interval estimates; and additional answer-generation LLMs will be evaluated on the same protocol to determine whether other models produce better results.

References

- Asai, A., Wu, Z., Wang, Y., Sil, A., & Hajishirzi, H. (2024). Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *Proceedings of the 12th International Conference on Learning Representations (ICLR)*.
- Chen, J., Xiao, S., Zhang, P., Luo, K., Lian, D., & Liu, Z. (2024). Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*.
- Cormack, G. V., Clarke, C. L. A., & Büttcher, S. (2009). Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '09)* (pp. 758–759). ACM.
- Dumitrescu, S. D., Avram, A.-M., & Pyysalo, S. (2020). The birth of Romanian BERT. In *Findings of the Association for Computational Linguistics: EMNLP 2020* (pp. 4324–4328). Association for Computational Linguistics.
- Es, S., James, J., Anke, L. E., & Schockaert, S. (2024, March). Ragas: Automated evaluation of retrieval augmented generation. In *Proceedings of the 18th conference of the european chapter of the association for computational linguistics: system demonstrations* (pp. 150–158).

- Frantar, E., Ashkboos, S., Hoefler, T., & Alistarh, D. (2022). Gptq: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*.
- Gao, L., Ma, X., Lin, J., & Callan, J. (2023). Precise zero-shot dense retrieval without relevance labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics* (Vol. 1, pp. 1762–1777). Association for Computational Linguistics.
- Karpukhin, V., Oğuz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., & Yih, W. (2020). Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 6769–6781). Association for Computational Linguistics.
- Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems* (Vol. 33, pp. 9459–9474).
- Li, X., & Croft, W. B. (2003). Time-based language models. In *Proceedings of the 12th International Conference on Information and Knowledge Management (CIKM '03)* (pp. 469–475). ACM.
- Ma, X., Gong, Y., He, P., Zhao, H., & Duan, N. (2023). Query rewriting in retrieval-augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 5303–5315). Association for Computational Linguistics.
- Martins, P. H., Fernandes, P., Alves, J., Guerreiro, N. M., Rei, R., Alves, D. M., ... & Martins, A. F. (2025). Eurollm: Multilingual language models for europe. *Procedia Computer Science*, 255, 53–62.
- Nogueira, R., & Cho, K. (2019). Passage re-ranking with BERT. *arXiv preprint arXiv:1901.04085*.
- Rajpurkar, P., Jia, R., & Liang, P. (2018). Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics* (Vol. 2, pp. 784–789). Association for Computational Linguistics.
- Robertson, S., & Zaragoza, H. (2009). *The probabilistic relevance framework: BM25 and beyond* (Vol. 4). Now Publishers Inc.
- Saad-Falcon, J., Khattab, O., Potts, C., & Zaharia, M. (2024). ARES: An automated evaluation framework for retrieval-augmented generation systems. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 338–354). Association for Computational Linguistics.
- Xiao, S., Liu, Z., Zhang, P., & Muennighoff, N. (2024). C-Pack: Packed resources for general Chinese embeddings. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24)* (pp. 641–649). ACM.
- Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E. P., Zhang, H., Gonzalez, J. E., & Stoica, I. (2023). Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems* (Vol. 36).