VerbaNexAI at LongEval 2026 Task 4: Fully Local Temporal and Pairwise Evidence Selection for Scientific RAG

LongEval at CLEF 2026

Isaac David Sánchez Sánchez[1,*], Jairo Enrique Serrano Castañeda[1], Edwin Puertas[1] and Juan Carlos Martínez-Santos[1]

[1]Team VerbaNexAI, Universidad Tecnologica de Bolivar (UTB), Cartagena, Colombia

Abstract

This paper describes the VerbaNexAI submissions to LongEval-RAG, Task 4 of the LongEval Lab at CLEF 2026. The task evaluates citation-grounded retrieval-augmented generation over scientific documents. We ran all retrieval, reranking, judging, and generation locally on a Mac Mini M4 with 16 GB of unified memory, without external LLM APIs. We submitted 2 automatic systems. The first, `qwen3-minmax-temp`, combines candidate-document ingestion, cross-encoder reranking with per-query min–max normalization, exponential temporal decay, and anchored generation with Qwen3-14B-4bit through MLX. The second, `pairjudge-qwen3-compact`, treats the candidate set as distractor-heavy evidence, ranks document pairs using lexical, neural, temporal, coverage, and complementarity signals, and uses a local LLM judge to select a compact citation set before answer generation. Official evaluation results show complementary behavior: the temporal min–max run achieved high citation precision and nugget coverage, while the PairJudge run produced more compact answers with better ROUGE-L, BERTScore, and TFC1 scores. These results suggest that, under tight local-compute constraints, researchers should optimize evidence selection and answer style jointly rather than independently.

Keywords

Retrieval-augmented generation, scientific information retrieval, temporal reranking, evidence selection, local language models, LongEval

1. Introduction

LongEval-RAG, part of the LongEval Lab at CLEF 2026 [1, 2], evaluates systems that must answer scientific queries using a small set of candidate documents and return citations into that reference set [3]. The task is not only a generation problem: answer quality depends on finding the right evidence, avoiding distractor documents, preserving temporal relations between scientific results, and producing a concise response that evaluators can compare against reference answers and citation-grounded evaluation criteria.

Our participation focused on a deliberately constrained research question: how far can a fully local RAG system go without proprietary APIs or cloud-hosted LLM judges? We produced all submitted runs locally on a Mac Mini M4 with 16 GB of RAM. This constraint shaped the model and verification choices: we prioritized cached intermediate artifacts, quantized local generation, and lightweight verification over large ensembles or API-based judging.

We submitted 2 automatic runs. The first run emphasizes temporally aware chunk reranking and grounded answer generation. We developed the second run after inspecting the task shape more closely: each query has only 10 candidate documents, many of which are distractors, and the official examples are short, usually citing only 2 documents. The second run separates evidence selection from answer writing and uses pairwise evidence scoring before local LLM-based judging.

CLEF 2026 Working Notes, 21 – 24 September 2026, Jena, Germany

*Corresponding author.

sanchezi@utb.edu.co (I. D. S. Sánchez); jserrano@utb.edu.co (J. E. S. Castañeda); epuerta@utb.edu.co (E. Puertas); jcmartinezs@utb.edu.co (J. C. Martínez-Santos)

0009-0003-2014-4749 (I. D. S. Sánchez); 0000-0001-8165-7343 (J. E. S. Castañeda); 0000-0002-0758-1851 (E. Puertas); 0000-0003-2755-0718 (J. C. Martínez-Santos)

2. Task Setting

Each LongEval-RAG instance contains a narrative query and a fixed list of candidate scientific documents. A system must output an answer and a list of citation indices into the provided references. In our submitted files, both runs contain 47 test queries, each with 10 candidate references. The official evaluation report compares generated answers to gold answers using ROUGE-L and BERTScore [4, 5], measures precision of cited documents, and includes TREC AutoJudge-based measures such as nugget coverage and IR axiom scores [6, 7, 8].

The task poses a citation-grounded scientific summarization problem with 2 main risks. First, a model can write a plausible answer from weak evidence. Second, it can retrieve or cite documents that appear relevant but do not support the exact query. The 2 systems address these risks differently: the temporal run tries to preserve strong semantic and temporal evidence in the prompt, while the PairJudge run tries to reduce the context to the smallest sufficient evidence set before generation.

3. Resources and Common Infrastructure

Both submitted runs reuse the same ingestion layer. The task provides a fixed candidate-reference list for each query, and only these documents can support the final answer and citations. We first parse the query file to build a hash set of candidate document identifiers, then scan the corpus shards line by line using a fast identifier check before full JSON parsing. This design treats non-candidate records as outside the task evidence space rather than as retrieval targets. For matched documents, we select text through a fallback cascade: `fullText`, then `abstract`, then `title`. We cache the resulting candidate-document tables as Parquet files. In the submitted experiments, this process produced 47 query records, 470 reference slots, and 468 unique extracted documents; the difference comes from 2 document identifiers that appear in more than one query.

We used `mlx-community/Qwen3-14B-4bit` as the local generation model for both runs and served it through MLX on Apple Silicon [9, 10]. We used Qwen3's chat template with thinking disabled to keep outputs direct and avoid emitting hidden traces of reasoning. For neural reranking, we used `BAAI/bge-reranker-v2-m3`; we cite the BGE-M3 family as the closest published model family reference [11].

4. Run 1: Temporal Min–Max RAG

The `qwen3-minmax-temp` run follows a 3-phase RAG architecture. It first extracts candidate evidence, then ranks text chunks with semantic and temporal signals, and finally generates grounded answers from the selected extracts.

4.1. Candidate-Focused Ingestion

The first phase extracts only the documents required by the query-reference map. We made this task-driven choice because LongEval-RAG answers must use the provided candidate references as evidence. Documents outside that set are not valid evidence for the submitted response. The line-by-line implementation also keeps the process efficient, with complexity linear in the number of required identifiers and matched records rather than in the full corpus size.

4.2. Temporal Cross-Encoder Reranking

The system splits documents into 250-word windows with 50-word overlap. It scores each query–chunk pair with the BGE reranker. We normalize raw cross-encoder logits per query using min–max normalization:

$$\mathrm{MinMax}_q(x_i) = \frac{x_i - \min_q}{\max_q - \min_q}. \tag{1}$$

We then fuse the normalized semantic score with an exponential temporal freshness signal. The following scoring function combines topical fit with publication-time evidence.

$$\mathrm{Score}(q, c) = \alpha\, \mathrm{MinMax}_q(\mathrm{CE}(q, c)) + (1 - \alpha)e^{-\lambda \Delta t(c)}. \tag{2}$$

We established the hyperparameters α and λ heuristically from the task structure and local development checks rather than from labeled validation feedback. The calibration target was ranking stability: semantic relevance had to remain dominant, while temporal freshness served only as a weak prior. We set $\alpha = 0.7$, so semantic relevance would strictly dictate the primary ranking, preventing recent but topically tangential documents from overshadowing highly relevant foundational work. We set the temporal decay rate $\lambda = 0.5$ to model the natural half-life relevant to science. Under this exponential formulation, a one-year-old document retains approximately 60% of its temporal weight, while a two-year-old document retains roughly 36%. This calibration creates a soft inductive bias towards recent advancements without aggressively discarding valid prior knowledge and addresses the temporal evolution of scientific consensus evaluated in Task 4. When a document lacks a publication date, the system assigns a neutral temporal value of 0.5. Following this temporal-semantic reranking phase, the top 5 highest-scoring chunks are retrieved and passed to the generator.

4.3. Anchored Generation

The generator receives numbered extracts and a strict system prompt that requires direct answers grounded only in those extracts. We ask it to append extract citations, such as `[1]` or `[1, 3]`, to each sentence. The export script then maps these extract-level citations back to the original reference list.

This run generated relatively long answers: across the 47 submitted responses, the average answer length was 112.1 words, ranging from 25 to 184. It used 1 citation for 29 queries, 2 citations for 17 queries, and 3 citations for 1 query.

5. Run 2: PairJudge Compact RAG

We designed the `pairjudge-qwen3-compact` run to match the evidence pattern observed in the examples: short answers and usually 2 citations. Instead of first writing from a broad top-k context, this run separates evidence selection from answer writing.

5.1. Document Cards and Snippets

For each candidate document, the system builds a document card containing metadata and high-scoring snippets. Initial snippet scoring uses lexical overlap with the query, title, abstract bonuses, and phrase-level evidence. When the cross-encoder is enabled, the selector rescores and normalizes the selected snippets. The final snippet score is:

$$s_{\mathrm{snippet}} = 0.35 s_{\mathrm{lexical}} + 0.65 s_{\mathrm{CE}}. \tag{3}$$

The document score combines the best snippet, the average snippet score, and a title bonus:

$$d_i = \min(1,\ 0.72 \max_{s \in S_i} s + 0.23\, \mathrm{avg}_{s \in S_i}\, s + 0.05 J(T(q), T(\mathrm{title}_i))). \tag{4}$$

Here, S_i is the selected snippet set for document i, $T(\cdot)$ denotes tokenization, and J is Jaccard overlap. The selector then keeps only the top documents for pair construction, which reduces the number of distractor combinations shown to the judge.

5.2. Pairwise Evidence Selection

The run ranks candidate document pairs because many task queries ask about relations, validation, changes, trade-offs, or combined findings. We define the pair score as follows.

$$s_{\text{pair}}(a,b) = 0.48 s_{\text{relevance}} + 0.19 s_{\text{temporal}} + 0.23 s_{\text{coverage}} + 0.10 s_{\text{complementarity}}. \quad (5)$$

For a pair (a, b), $s_{\text{relevance}}$ combines the stronger document score with the pair average: $0.58 \max(d_a, d_b) + 0.42(d_a + d_b)/2$. The coverage term measures the fraction of query terms covered by the selected snippets and titles in the pair. The temporal term rewards date gaps and ordered evidence when the query suggests validation, comparison, change, or another relation across studies. The complementarity term uses Jaccard overlap between the selected snippets: it rewards moderate overlap, assigns a lower value to nearly unrelated pairs, and penalizes near-duplicate evidence.

We established the linear combination weights heuristically, guided by the task's structural demands, and refined them through local checks for plausible evidence coverage, valid citations, and compact answers. We assigned the highest weight to semantic relevance (0.48), so the scoring function would prioritize strict topical alignment and aggressively filter out dense distractor documents. Coverage (0.23) acts as the primary secondary constraint, rewarding pairs that collectively address all facets of complex, multi-hop queries. The temporal bridge feature (0.19) rewards chronologically ordered evidence (e.g., an earlier hypothesis followed by a later empirical validation), which is critical for capturing the evolving scientific knowledge that Task 4 evaluates. Finally, complementarity (0.10) serves as a redundancy penalty, acting as a tie-breaker that favors diverse document pairs over those that overlap. The selector also scores single-document fallbacks for queries that a single source can fully answer.

5.3. Local Judge and Answer Writer

The selector computes the top 14 pair candidates and shows the top 8 to a local LLM judge. Each prompt item includes the heuristic score, reference indices, document identifiers, publication dates, titles, and up to 2 snippets per document. The judge prompt asks for the smallest sufficient citation set, prefers 2 documents for relation or comparison queries, uses 1 document when it fully supports the answer, and returns JSON only with selected references, confidence, relation type, support facts, and an answer plan. We use 3 judging passes with a temperature of 0.15. A separate answer writer then produces 1 concise sentence of 18–45 words from the selected evidence, with citations returned as a separate JSON field rather than in-text markers. The verifier checks candidate outputs for citation validity, answer length, absence of preambles, lexical query coverage, and pair score.

This run generated substantially shorter answers than the temporal run: the average answer length was 28.3 words, ranging from 18 to 42. It used 1 citation for 12 queries and 2 citations for 35 queries. The validator found all submitted citation indices valid with respect to their reference lists.

6. Results

Table 1 reports the organizer-provided evaluation results for the 2 submitted runs and the official naive baseline. Higher values are better for all listed measures.

Table 1
Organizer-provided evaluation results for the submitted VerbaNexAI runs.

Run	ROUGE-L	BERTScore	Precision	Nugget cov.	TFC1
official-naive-baseline	0.082	-0.118	0.200	0.124	-0.758
`pairjudge-qwen3-compact`	**0.206**	**0.263**	0.681	0.251	**0.809**
`qwen3-minmax-temp`	0.139	0.126	**0.975**	**0.493**	-0.197

Both submitted systems improve over the naive baseline on the selected measures. However, the 2 runs optimize different aspects of the task. The temporal min–max run obtains the strongest citation precision and nugget coverage. This result suggests that the cross-encoder plus temporal fusion strategy selected useful evidence and preserved information needed by the nugget-based evaluator. Its weaker ROUGE-L, BERTScore, and TFC1 results likely reflect answer style: the run often produced long, cautious paragraphs that remained faithful to evidence but less aligned with the compact gold-answer style. The TFC1 gap is especially consistent with this interpretation because the axiom rewards query-term occurrences: PairJudge explicitly checks lexical query coverage in compact answers, whereas the temporal run often expands or paraphrases the evidence.

The PairJudge run shows the opposite tendency. It obtains the best ROUGE-L, BERTScore, and TFC1 scores among our submissions and the baseline. This behavior is consistent with its short answer constraint and explicit validation against example-like answer length. Yet its citation precision and nugget coverage are lower than the temporal run. These scores indicate that compact answers alone are not sufficient: the local judge sometimes selected evidence that supported a fluent answer but did not match the gold citation set, the full nugget bank, or the temporal run.

7. Discussion

The results are encouraging under the local-compute constraint. A fully local system, without external LLM APIs or cloud reranking services, can outperform the naive baseline across all reported measures. More importantly, the 2 runs expose a useful trade-off. Strong evidence retrieval does not automatically produce answers that match reference-answer metrics, and compact answer generation does not automatically maximize citation precision or nugget coverage.

For scientific RAG, this trade-off matters. Researchers should not optimize citation-based systems solely for answer fluency, because citations are part of the scientific claim. At the same time, evaluators may under-reward high-quality evidence when the generated answer is too verbose, hedged, or mismatched to the expected answer granularity. These results suggest that teams should train or calibrate evidence selection and answer compression together.

The hardware constraint also affected design choices. The Mac Mini M4 with 16 GB of memory made large-scale model ensembles, multi-model judging, and extensive hyperparameter sweeps impractical. We used cached artifacts, 4-bit local generation, and simple deterministic verifiers. While limiting, this setup also makes the system reproducible for teams without access to expensive cloud GPUs or paid LLM APIs.

8. Limitations and Future Work

The main limitation is that the 2 strongest properties appear in separate runs. A natural next system would combine the temporal min–max run's high-precision evidence selection with the PairJudge run's compact answer writer. The PairJudge local verifier also uses heuristic answer scoring rather than learned calibration against task-specific labels. Better local calibration could reduce distractor citations without requiring external judges.

The reported weights and answer-length bounds are manually calibrated. We refined them through task-shape inspection, qualitative review of selected evidence, and local validity checks for JSON parsing, citation validity, and answer length rather than through a labeled validation split. The study should be read as a careful engineering comparison of 2 submitted systems rather than a controlled ablation study.

This working note does not include component ablations, statistical significance testing, or additional standard RAG baselines beyond the organizer-provided naive baseline. The official report available to us provides aggregate scores for 47 queries, so the reported differences should be read as descriptive rather than as claims of statistical significance. Future work should explore sentence-level evidence extraction after document-pair selection, citation-aware answer compression, explicit optimization for

nugget coverage, and query-adaptive temporal signals. We also plan to release cleaned notebooks and scripts after removing local paths and respecting task-data redistribution constraints.

9. Conclusion

We presented 2 fully local VerbaNexAI systems for LongEval-RAG at CLEF 2026. The temporal min–max run prioritizes evidence quality through cross-encoder reranking and temporal fusion, achieving high citation precision and nugget coverage. The PairJudge run prioritizes compact, example-aligned answer writing through pairwise evidence selection and local LLM judging, achieving stronger ROUGE-L, BERTScore, and TFC1 results. Together, the submissions show that local scientific RAG is feasible under modest hardware constraints, but that future systems must jointly optimize citation fidelity and answer style.

Acknowledgments

We thank the CLEF 2026 LongEval organizers for designing the LongEval-RAG task and providing the evaluation resources for this work. We also acknowledge Team VerbaNexAI and Universidad Tecnologica de Bolivar (UTB) for their support of this participation.

Declaration on Generative AI

During the preparation of this work, the authors used OpenAI Codex/ChatGPT for grammar and spelling checks, paraphrase and reword suggestions, LaTeX formatting, consistency checks, and assistance preparing source files. The authors wrote the scientific content, verified the technical claims, reviewed and edited the assisted text as needed, and take full responsibility for the publication's content.

References

[1] S. Bouaraba, T. Breuer, M. Cancellieri, A. El-Ebshihy, M. Fröbe, P. Galuscáková, L. Goeuriot, G. Iturra-Bocaz, J. Keller, P. Knoth, A. K. Kruff, P. Mulhem, F. Piroi, D. Pride, P. Schaer, D. Schwab, Overview of the clef 2026 longeval lab on longitudinal evaluation of model performance, in: M. Hagen, M. Potthast, B. Stein, P. Schaer, E. Zangerle, S. MacAvaney, J. M. Struß, E. Sánchez Salido, A. Barrón-Cedeño, A. García Seco de Herrera (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Seventeenth International Conference of the CLEF Association (CLEF 2026), Lecture Notes in Computer Science (LNCS), Springer, 2026.

[2] S. Bouaraba, T. Breuer, M. Cancellieri, A. El-Ebshihy, M. Fröbe, P. Galuscáková, L. Goeuriot, G. Iturra-Bocaz, J. Keller, P. Knoth, A. K. Kruff, P. Mulhem, F. Piroi, D. Pride, P. Schaer, D. Schwab, Extended overview of the clef 2026 longeval lab on longitudinal evaluation of model performance, in: E. Sánchez Salido, A. Barrón-Cedeño, A. García Seco de Herrera, S. MacAvaney, J. M. Struß (Eds.), CLEF 2026 Working Notes, CEUR Workshop Proceedings, CEUR-WS.org, 2026.

[3] LongEval Organizers, LongEval-RAG: Retrieval augmented generation at CLEF 2026, 2026. URL: https://clef-longeval.github.io/tasks/, cLEF 2026 LongEval Lab, Task 4.

[4] C.-Y. Lin, ROUGE: A package for automatic evaluation of summaries, in: Text Summarization Branches Out, 2004, pp. 74–81.

[5] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, Y. Artzi, BERTScore: Evaluating text generation with BERT, in: International Conference on Learning Representations, 2020. URL: https://openreview.net/forum?id=SkeHuCVFDr.

[6] L. Dietz, N. Farzi, E. Yang, D. Lawrie, Too many questions: Deriving concise and effective nugget banks, 2026. Manuscript cited in the organizer-provided evaluation report.

[7] N. Farzi, T. Hagen, E. Yang, M. Fröbe, R. Pradeep, H. A. Rahmani, X. Wang, O. Zendel, L. Dietz, Autojudge: A cross-task benchmark for comparing LLM judges for citation-grounded RAG systems, 2026. Manuscript cited in the organizer-provided evaluation report.

[8] J. H. Merker, M. Fröbe, B. Stein, M. Potthast, M. Hagen, Axioms for retrieval-augmented generation, in: Proceedings of the 2025 International ACM SIGIR Conference on Innovative Concepts and Theories in Information Retrieval, 2025, pp. 67–77. doi:10.1145/3731120.3744601.

[9] Qwen Team, Qwen3, 2025. URL: https://huggingface.co/mlx-community/Qwen3-14B-4bit, large language model family used through the 4-bit MLX community checkpoint.

[10] Apple Machine Learning Research, MLX: An array framework for apple silicon, 2023. URL: https://github.com/ml-explore/mlx.

[11] J. Chen, S. Xiao, P. Zhang, K. Luo, D. Lian, Z. Liu, BGE M3-Embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation, 2024. URL: https://arxiv.org/abs/2402.03216. arXiv:2402.03216.