Team ENGsTF at PAN 2026: Hybrid Sparse-Dense Retrieval with Two-Stage Score Fusion for Generative Plagiarism Detection*

Notebook for the PAN Lab at CLEF 2026

Farah Alkayyem[1,*], Tala Alkhateeb[1], Raghad Alhossny[1] and Riad Sonbol[1,2]

[1]Faculty of Artificial Intelligence Engineering, Syrian Private University, Damascus, Syria
[2]Department of Informatics, Higher Institute for Applied Sciences and Technology, Damascus, Syria

Abstract

Generative plagiarism generated by large language models preserves semantic content while exhibiting low lexical overlap, which limits sparse retrieval methods and complicates dense retrieval over long documents. In this work, we present a hybrid source retrieval system for the PAN 2026 Generative Plagiarism Detection task. Our approach combines BM25-based sparse retrieval with dense semantic retrieval using BGE-M3, integrating both signals via weighted score fusion. To better handle long scientific documents, source papers are segmented into overlapping token-based chunks, while queries are encoded as full-length inputs using the long-context capability of BGE-M3. Retrieved candidates are then reranked using BGE-reranker-v2-m3, with scores fused back into the retrieval scores via a second weighted combination. Experiments show that the hybrid approach outperforms sparse-only and dense-only baselines. Long-context embeddings improve retrieval quality, while score fusion is crucial for maintaining performance, as replacing retrieval scores with reranker outputs degrades effectiveness. On the PAN 2026 test set, the system achieves an nDCG@10 of 0.6422 and Recall@10 of 0.6708, the best performance among evaluated configurations.

Keywords

Generative Plagiarism Detection, Source Retrieval, Hybrid Retrieval, Dense Retrieval, BM25, BGE-M3, Long-Context Embedding, Reranking

1. Introduction

The widespread adoption of large language models has made fluent text paraphrasing readily accessible, creating new challenges for academic integrity and plagiarism detection [1]. Unlike traditional plagiarism, which typically involves verbatim or lightly modified copying, LLM-generated plagiarism can rewrite source material so fluently that surface-level overlap is minimal while the underlying content is substantially reproduced [2], making lexical matching methods insufficient on their own. Detecting this kind of reuse requires systems that can bridge the semantic gap between a generated passage and its origin [3].

The PAN 2026 Generative Plagiarism Detection task frames this as a source retrieval problem: given a set of generated scientific documents and a large corpus of human-written arXiv papers, participants must retrieve the source documents that each generated document draws from, ranked by relevance [4, 5].

Sparse retrieval methods such as BM25 [6] rely on lexical overlap between query and document, which is inherently weak when source material has been paraphrased by an LLM. Dense retrieval methods address this by operating in semantic space, but standard dense models such as E5 [7] are limited to short input windows – typically 512 tokens – forcing long scientific documents to be fragmented into many small chunks, which dilutes document-level context and introduces aggregation noise. Prior approaches to this task [3] operated at the sentence or paragraph level, which is well-suited for text

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

*Source code available at: https://github.com/farah040/pan2026-engstf

*Corresponding author.
farah_alkayyem@spu.edu.sy (F. Alkayyem)

alignment but does not scale naturally to corpus-level source retrieval over full documents. While encoding query documents that draw from multiple sources into a single dense vector may produce somewhat diluted representations, combining long-context dense retrieval with sparse signals helps compensate for this limitation, making a hybrid approach more robust than dense retrieval alone [8].

This paper describes our submission to this task, built around a hybrid retrieval pipeline combining dense semantic search with BM25-based sparse retrieval, followed by a cross-encoder reranking stage. The central design choice is the use of BGE-M3 [9], a long-context embedding model capable of encoding sequences up to 8,192 tokens, which allows query documents to be encoded as single vectors without truncation. Score fusion and reranker blending are each controlled by a separate weighting parameter, giving the pipeline two stages of score combination. We show that this design yields meaningful improvements over both sparse and dense baselines.

The remainder of this paper is organized as follows. Section 2 presents the system architecture and pipeline design. Section 3 describes the experimental development process and reports results on both the proxy and official test datasets. Section 4 concludes the paper and discusses directions for future work.

2. Method

Our system is a hybrid retrieval pipeline combining dense vector search with BM25 sparse retrieval, followed by a cross-encoder reranking stage, built on top of the BGE-M3 embedding model and the PyTerrier [10] framework. This design combines two established information retrieval practices. First, fusing sparse and dense retrieval scores is a well-known strategy for exploiting the complementary strengths of exact lexical matching and semantic similarity, rather than relying on either signal in isolation [8]. Second, the overall pipeline follows a retrieve-then-rerank paradigm, a standard approach for balancing effectiveness and efficiency: a cheap first-stage retriever narrows a large candidate pool before a more accurate but computationally expensive model refines the ranking [11]. Since a cross-encoder jointly processes each query-document pair to improve ranking quality at the top [12], scoring every candidate document with one is infeasible at corpus scale; the retrieval stage is therefore optimized to maximize recall, while reranking is reserved for a narrowed candidate pool. The pipeline proceeds in three stages: document chunking and indexing, dual retrieval with score fusion, and cross-encoder reranking (Figure 1).

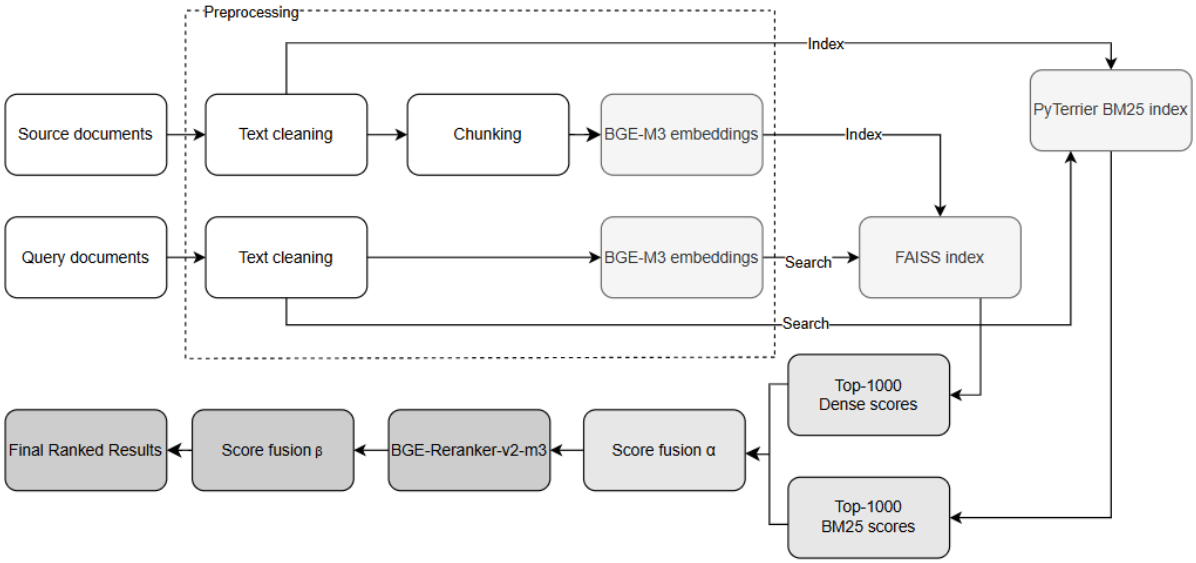

Figure 1: Overview of the hybrid retrieval and reranking pipeline.

2.1. Chunking and Indexing

Before chunking, we apply a lightweight text cleaning step that removes display math expressions and replaces inline math with an equation tag, and removes LaTeX footnotes – both are token-heavy constructs that contribute noise rather than retrievable semantic content for this task.

Source documents are then split into overlapping token-based chunks using a sliding window of 4,096 tokens with a stride of 3,687 tokens, giving approximately 10% overlap between consecutive chunks. The chunk size is set to half of BGE-M3's maximum context length – while the model supports up to 8,192 tokens, operating near the upper limit risks embedding quality degradation on sequences longer than those well-represented in training data; the 10% overlap is chosen to preserve continuity across chunk boundaries while keeping the number of vectors per document computationally manageable. Each chunk is encoded using BGE-M3 and indexed in a FAISS [13] flat inner product index. Separately, the full cleaned source documents are indexed for BM25 retrieval using PyTerrier's term-based indexer.

Query documents are encoded as single vectors without chunking. BGE-M3's large context window is sufficient to represent full query documents in a single embedding, preserving their complete semantic content rather than fragmenting it across multiple vectors.

2.2. Hybrid Retrieval and Fusion

At retrieval time, each query vector is searched against the FAISS index to retrieve the top-1,000 most similar source chunks. Chunk-level scores are aggregated to the document level by keeping the maximum score across all chunks belonging to the same source document. In parallel, the pre-built BM25 index is queried using PyTerrier, also retrieving the top-1,000 candidates per query.

The two result sets are combined using weighted linear fusion. Both score lists are first normalized to $[0, 1]$ using min-max normalization, then combined as:

$$\text{score}_{\text{fused}} = \alpha \cdot \text{score}_{\text{dense}} + (1 - \alpha) \cdot \text{score}_{\text{sparse}} \tag{1}$$

where $\alpha = 0.57$, tuned on our development experiments, as detailed in Section 3.6. Documents retrieved by only one method contribute a score of zero for the other component. The fused list is sorted and truncated to the top 1,000 documents per query.

2.3. Reranking

The top-150 candidates from the fused list are reranked using BGE-reranker-v2-m3 [14], consistent with the effectiveness-efficiency trade-off described in Section 2.

Since query documents are too long to feed directly into the cross-encoder, we split each query into 512-token chunks using the reranker's tokenizer. Source documents are truncated to their first 1,000 tokens, which for arXiv papers typically covers the abstract and the beginning of the introduction. For each query-source pair, we score all combinations of query chunks against the truncated source and take the maximum cross-encoder score across query chunks as the final reranker score – consistent with the retrieval aggregation strategy of identifying the strongest match between any region of the query and the source.

The reranker score is then blended with the original fusion score using a second weighting parameter:

$$\text{score}_{\text{final}} = \beta \cdot \text{score}_{\text{reranker}} + (1 - \beta) \cdot \text{score}_{\text{fused}} \tag{2}$$

where both scores are normalized within the rerank window before blending and $\beta = 0.7$, as detailed in Section 3.6. Documents outside the top-150 rerank window are kept in their original fusion order.

3. Experimental Results and Analysis

Table 2 summarizes all evaluated configurations on the PAN 2026 test set. The experiments are presented in development order, but they also form a natural component-wise ablation: each row adds or modifies

exactly one element relative to the previous, making it possible to isolate the contribution of each design decision to the final system (Experiment 6).

3.1. Proxy Evaluation Dataset

Since no validation set was provided for the PAN 2026 task, we constructed a lightweight proxy dataset from the PAN 2025 text alignment validation set to support early-stage development. We sampled 100 suspicious documents with their aligned source pairs, chunked sources at paragraph level, and derived query chunks from the ground-truth annotations. This setup was intentionally minimal – the goal was fast pipeline verification and rough comparisons between early configurations, not rigorous evaluation.

It is worth noting that the PAN 2025 task involves text alignment between pre-paired documents, whereas PAN 2026 requires retrieval over a large, unpaired source corpus. This structural difference means the proxy dataset does not reflect the actual task well: sources are paragraph-level chunks rather than full documents, and there is no retrieval over a large candidate pool. We therefore treat results on this dataset as sanity checks only. Once the core pipeline was stable, we switched to evaluating directly against the official PAN 2026 test set via TIRA [15].

3.2. Early Retrieval Experiments

On this dataset, we first evaluated the PAN-provided BM25 baseline using PyTerrier, then built a dense retrieval system using E5-base-v2 [7] with FAISS indexing. Dense retrieval outperformed BM25 alone, and combining both via weighted score fusion improved results further across all metrics. Experiments 1–3 show that each retrieval signal is individually insufficient: BM25 alone (Experiment 1) and E5 alone (Experiment 2) both underperform their combination (Experiment 3), confirming that sparse and dense signals capture complementary aspects of relevance. These results are summarized in Table 1. Since each query in this dataset is paired with a single relevant source, Recall@1 and Recall@10 are equivalent here; we report Recall@1 as it more directly reflects whether the single correct source was retrieved.

Table 1
Retrieval Performance on the Proxy Evaluation Dataset

Experiment	Approach	nDCG@10	Recall@1
1	BM25 + PyTerrier	0.850	0.807
2	E5-base-v2 + FAISS	0.876	0.844
3	Hybrid Pipeline (E5 + BM25)	0.896	0.865

3.3. Long-Context Retrieval

Experiment 4 isolates the contribution of long-context encoding by replacing E5 with BGE-M3 while keeping the rest of the pipeline fixed. Before adopting this change, we conducted several exploratory experiments within the E5-based hybrid that are not included in Table 2, as they informed this design decision rather than forming part of the final ablation. Specifically, we explored sentence-based sliding window chunking against token-based chunking, and max pooling against mean pooling for aggregating chunk-level scores to the document level. Token-based chunking performed marginally better. Max pooling consistently outperformed mean pooling – which is expected given the nature of the task: since a query document may draw from multiple sources, the strongest signal is the highest similarity between any region of the query and any region of a candidate source. We also briefly explored adding a cross-encoder reranking step to the E5 hybrid, which produced a significant drop in performance, attributed to the reranker receiving only a single chunk pair as input – a fragment that lacked sufficient context for reliable relevance judgments.

Rather than continuing to explore chunking configurations within E5's limited context window, we shifted to BGE-M3 – a model specifically designed for long-document retrieval. Unlike E5, whose

512-token limit forces long arXiv papers into many small chunks, BGE-M3 supports sequences up to 8,192 tokens, though encoding at the model's maximum limit can introduce noise into the resulting vector; we therefore chunk source documents at 4,096 tokens. This produced a clear improvement in both nDCG@10 and Reciprocal Rank (RR) over the best E5-based configuration, as shown in Table 2.

3.4. Cross-Encoder Reranking

Experiment 5 isolates the effect of adding cross-encoder reranking to the BGE-M3 hybrid (Experiment 4), with everything else held fixed. We rerank the top-150 candidates using BGE-reranker-v2-m3. Somewhat surprisingly, this produced a significant drop across all metrics – nDCG@10 fell from 0.6128 to 0.4908 and RR from 0.9123 to 0.6955. We attribute this to the reranker scores fully replacing the fusion scores for the reranked candidates, effectively discarding retrieval signal that was already strong. The reranker, operating only on truncated source text, appears to introduce noise rather than refinement when its scores are used in isolation.

3.5. Score Blending with β Parameter

Experiment 6, the final system, addresses this by introducing a second weighting parameter β that blends the reranker score with the original fusion score rather than replacing it. This recovered and improved most metrics over the unranked hybrid (Experiment 4): nDCG@10 improved from 0.6128 to 0.6422 and Recall@10 from 0.6483 to 0.6708. RR showed a slight decrease compared to the unranked hybrid, from 0.9123 to 0.9076, suggesting that while the reranker improves overall ranking quality, it occasionally displaces a correct top-1 result. These results are summarized in Table 2.

Table 2
Retrieval Performance on the PAN 2026 Test Set

Experiment	Approach	Recall@10	Recall@100	nDCG@10	nDCG@100	RR
1	BM25 (PAN baseline)	0.5767	0.7825	0.5532	0.6068	0.8318
2	E5-base-v2 + FAISS	0.4329	0.6758	0.3852	0.4479	0.6470
3	Hybrid Pipeline (E5 + BM25)	0.6121	0.8200	0.5851	0.6368	0.8760
4	Hybrid Pipeline (BGE-M3 + BM25)	0.6483	0.8321	0.6128	0.6584	0.9123
5	Hybrid Pipeline (BGE-M3 + BM25) + Reranker	0.5654	0.8229	0.4908	0.5611	0.6955
6	Hybrid Pipeline (BGE-M3 + BM25) + Reranker (β blending)	0.6708	0.8404	0.6422	0.6871	0.9076

3.6. Experimental Setup

Table 3 summarizes the key hyperparameters used across all experiments. Unless otherwise stated, these values were held constant throughout development.

The fusion weight α was determined through a parameter sweep on the proxy dataset, testing values across the range $[0, 1]$ and selecting the value that maximized nDCG@10. Performance is relatively stable across a broad region around the optimum, roughly between $\alpha = 0.5$ and $\alpha = 0.65$, rather than peaking sharply at a single value, and Recall@1 follows a similar trend across the sweep, indicating that the fused system is not highly sensitive to the exact setting of α in this range.

Unlike α, the reranker blend weight β was not tuned via a systematic search. Performing such a search directly on the official test set would lead to tuning on the evaluation data itself, which we avoided as invalid practice, and the proxy dataset's simplicity, with short source chunks rather than full documents, does not reflect how the reranker behaves on full-length test documents, making it unsuitable for selecting this parameter either. We therefore set $\beta = 0.7$ based on the intuition that

the reranker score should dominate, since cross-encoders are generally more accurate at relevance judgment than retrieval-stage scores [12], while still retaining a meaningful contribution from the fusion score, motivated by the degradation observed when fully replacing fusion scores with reranker outputs in Experiment 5. We acknowledge that this choice was not validated through systematic tuning, and we treat this as a limitation of the current system rather than a properly determined value.

Table 3
System Hyperparameters

Parameter	Value
Chunk size (retrieval)	4,096 tokens
Stride	3,687 tokens
Retrieval top-k	1,000
Fusion weight α	0.57
Rerank window	150
Query chunk size (reranker)	512 tokens
Source truncation (reranker)	1,000 tokens
Reranker blend weight β	0.7

4. Conclusion

We presented a hybrid source retrieval system for the PAN 2026 Generative Plagiarism Detection task, combining BGE-M3 dense retrieval with BM25, followed by a cross-encoder reranking stage with score blending. The central finding is that neither signal alone is sufficient – combining both consistently outperforms either in isolation. Switching to BGE-M3 over E5 improved retrieval quality by enabling longer document chunks with less fragmentation, and the β-blending parameter further improved ranking quality by retaining fusion scores rather than replacing them with reranker outputs.

Several aspects of the system warrant further investigation. Both the fusion weight α and the reranker blend weight β were set without a fully representative validation set; both would benefit from optimization on proper held-out data. The reranking stage also raises open questions around input design – the effect of query chunk size, source truncation length, and the number of candidates passed to the reranker all remain underexplored and may offer additional gains.

Acknowledgments

We thank the PAN 2026 task organizers for preparing the dataset and defining the evaluation framework, and the TIRA team for providing the evaluation infrastructure.

Declaration on Generative AI

During the preparation of this work, the authors used Claude (Anthropic) in order to: Grammar and spelling check, Paraphrase and reword, Improve writing style. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] S. Pudasaini, L. Miralles-Pechuan, D. Lillis, M. Llorens Salvador, Survey on AI-generated plagiarism detection: The impact of large language models on academic integrity, Journal of Academic Ethics (2024). ArXiv:2407.13105.

[2] K. Krishna, Y. Song, M. Karpinska, J. Wieting, M. Iyyer, Paraphrasing evades detectors of AI-generated text, but retrieval is an effective defense, in: Advances in Neural Information Processing Systems, 2023. ArXiv:2303.13408.
[3] A. Greiner-Petter, M. Fröbe, J. P. Wahle, T. Ruas, B. Gipp, A. Aizawa, M. Potthast, Overview of the generative plagiarism detection task at PAN 2025, in: Working Notes of CLEF 2025, volume 4038 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2025.
[4] A. Greiner-Petter, Y.-A. Wu, Y. Kitamura, K. W. Kim, M. Fröbe, B. Gipp, A. Aizawa, M. Potthast, Overview of the Plagiarism Detection Task at PAN 2026, in: A. Barrón-Cedeño, A. G. S. de Herrera, E. S. Salido, S. MacAvaney, J. M. Struß (Eds.), Working Notes of CLEF 2026 – Conference and Labs of the Evaluation Forum, CEUR Workshop Proceedings, CEUR-WS.org, 2026.
[5] J. Bevendorff, M. Fröbe, A. Greiner-Petter, A. Jakoby, M. Mayerl, P. Nakov, H. Plutz, M. Potthast, B. Stein, M. N. Ta, Y. Wang, E. Zangerle, in: M. Hagen, M. Potthast, B. Stein, P. Schaer, E. Zangerle, A. Barrón-Cedeño, A. G. S. de Herrera, E. S. Salido, S. MacAvaney, J. M. Struß (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, Springer, Berlin Heidelberg New York, 2026.
[6] S. Robertson, H. Zaragoza, The probabilistic relevance framework: BM25 and beyond, Foundations and Trends in Information Retrieval 3 (2009) 333–389.
[7] L. Wang, N. Yang, X. Huang, B. Jiao, L. Yang, D. Jiang, R. Majumder, F. Wei, Text embeddings by weakly-supervised contrastive pre-training, arXiv preprint arXiv:2212.03533 (2022).
[8] Y. Luan, J. Eisenstein, K. Toutanova, M. Collins, Sparse, dense, and attentional representations for text retrieval, Transactions of the Association for Computational Linguistics 9 (2021) 329–345.
[9] 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, arXiv preprint arXiv:2402.03216 (2024).
[10] C. Macdonald, N. Tonellotto, S. MacAvaney, I. Ounis, PyTerrier: Declarative experimentation in Python from BM25 to dense retrieval, in: Proceedings of CIKM 2021, ACM, 2021, pp. 4526–4533.
[11] L. Wang, J. Lin, D. Metzler, A cascade ranking model for efficient ranked retrieval, in: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2011, pp. 105–114.
[12] R. Nogueira, K. Cho, Passage re-ranking with BERT, arXiv preprint arXiv:1901.04085 (2019).
[13] J. Johnson, M. Douze, H. Jégou, Billion-scale similarity search with GPUs, IEEE Transactions on Big Data 7 (2019) 535–547.
[14] BAAI, bge-reranker-v2-m3, https://huggingface.co/BAAI/bge-reranker-v2-m3, 2024.
[15] M. Fröbe, M. Wiegmann, N. Kolyada, B. Grahm, T. Elstner, F. Loebe, M. Hagen, B. Stein, M. Potthast, Continuous Integration for Reproducible Shared Tasks with TIRA.io, in: J. Kamps, L. Goeuriot, F. Crestani, M. Maistro, H. Joho, B. Davis, C. Gurrin, U. Kruschwitz, A. Caputo (Eds.), Advances in Information Retrieval. 45th European Conference on IR Research (ECIR 2023), Lecture Notes in Computer Science, Springer, Berlin Heidelberg New York, 2023, pp. 236–241. URL: https://link.springer.com/chapter/10.1007/978-3-031-28241-6_20. doi:10.1007/978-3-031-28241-6_20.