Team Fosuai at PAN 2026: Source Retrieval via Query-Chunk Provenance Modeling for Generative Plagiarism Detection

Notebook for the PAN Lab at CLEF 2026

Yi Wang[1], Zhongyuan Han[1,*], Chang Liu[1] and Haojie Cao[1]

[1]Foshan University, Guangdong, China

Abstract

The PAN@CLEF 2026 Generative Plagiarism Detection task aims to retrieve potential source documents for a given suspicious document from a large-scale source corpus. Addressing the problem that texts generated by large language models often fuse content from multiple sources, diluting the signal of full-document-level queries, this paper proposes a lightweight source retrieval method based on query chunk provenance modeling. The method splits the suspicious document into paragraph-level query chunks, performs BM25 retrieval independently for each chunk, and aggregates candidate sources via a coverage-weighted voting mechanism, thereby strengthening signals from sources consistently matched across multiple chunks. The system relies only on traditional inverted indexes and standard Python libraries, requiring no GPU or pre-trained language models, and has been submitted as the fosuai team's TIRA entry for the official PAN26 evaluation. Experimental and official results show that the method achieves competitive early-rank performance, while also revealing limitations in output depth and cross-dataset generalization that require further improvement.

Keywords

Generative Plagiarism Detection, Source Retrieval, Query Segmentation, Provenance Modeling, BM25, Coverage-Weighted Voting

1. Introduction

The PAN@CLEF 2026 Generative Plagiarism Detection task [1, 2] addresses a rapidly emerging challenge: Large Language Models (LLMs) can synthesize, recombine, and rewrite content from dozens of source documents into a single suspicious text, where different paragraphs may originate from entirely different sources. The task consists of two stages: Source Retrieval—identify source documents from a large corpus given a suspicious document; and Text Alignment—precisely locate plagiarized passages at character level. This paper focuses on Stage 1.

Existing source retrieval methods predominantly adopt a document-level querying paradigm: the entire suspicious document is encoded as a single query vector or bag-of-words, and a relevance score is computed against each candidate document. While effective for single-source plagiarism, this paradigm suffers from multi-source signal dilution. When a suspicious text composites content from Source A's introduction, Source B's methodology, and Source C's results, the query signal becomes a noisy mixture of multiple provenance fingerprints. Each individual source's matching signal is diluted by noise from other sources, causing the true source to be ranked outside the top retrieval positions.

To address this problem, we propose Query-Chunk Provenance Modeling. Rather than treating the suspicious document as a monolithic query, we decompose it into paragraph-level semantic segments. Each segment independently retrieves candidate sources, and a coverage-weighted voting mechanism aggregates the results. Documents matched by multiple independent segments receive a coverage bonus—a strong signal of genuine multi-paragraph provenance. This approach explicitly models the provenance structure of multi-source plagiarism while maintaining a lightweight implementation.

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

*Corresponding author.

hanzhongyuan@gmail.com (Z. Han)

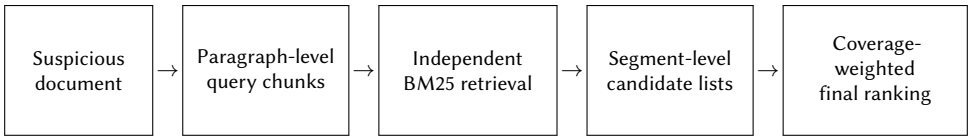

Figure 1: Overview of the Query-Chunk Provenance Modeling pipeline. The suspicious document is split on the query side; each chunk retrieves sources independently, and source candidates are aggregated by coverage-weighted voting.

2. Related Work

Source retrieval has been a core component of PAN plagiarism detection tasks since PAN 2014 [3]. Early approaches [3] relied on lexical matching: word n-gram overlap, fingerprinting, and BM25 [4] have proven effective for verbatim and lightly-paraphrased plagiarism. Dense retrieval methods using sentence encoders such as Sentence-BERT [5] and E5 [6] project documents into a dense vector space for semantic matching, achieving better generalization on heavily paraphrased texts. Chunk-based retrieval [7] splits documents into fixed-size segments for independent encoding and max-pooling aggregation, but applies splitting only on the document side while keeping the query as a single unit. Query decomposition techniques [8] split complex queries into sub-queries for multi-hop QA, but have not been applied to plagiarism-specific segmentation strategies. Our work differs from all of the above by modeling provenance structure on the query side: decomposing the suspicious document itself to isolate individual source signals before retrieval.

3. Method

3.1. Task Formulation

Given a source corpus $C = \{d_1, d_2, \ldots, d_N\}$ ($N = 60{,}592$ academic documents from ClueWeb09) and a suspicious document q, source retrieval aims to produce a ranked list $R(q) = [d_{r_1}, \ldots, d_{r_K}]$ such that the true source document $d^*(q)$ appears as high as possible. In generative plagiarism, q may be synthesized from multiple sources, though evaluation considers one primary source per q. Retrieval quality is measured by Recall@K, nDCG@K, MRR, and Reciprocal Rank (RR).

3.2. Query-Chunk Provenance Modeling

Our method consists of three stages, illustrated in Figure 1.

Stage 1—Paragraph-Level Query Segmentation. The suspicious document q is split by double-newline paragraph boundaries. Adjacent paragraphs are merged until reaching $max\_chars = 3{,}000$, with chunks shorter than $min\_chars = 800$ discarded. Output is capped at 20 chunks. This produces semantically coherent segments roughly corresponding to academic document section structure. For documents lacking paragraph structure, the entire text is treated as a single segment.

Stage 2—Independent Segment Retrieval. Each segment c_i independently performs BM25 retrieval with optimized parameters: $max\_terms = 20$ (vs. 100 for full-document queries, since short segments contain fewer discriminative terms) and $max\_df = 5{,}000$ (filtering terms appearing in $> 5{,}000$ documents, $\sim 8\%$ of the corpus). Without filtering, segment-level BM25 is 85–150$\times$ slower than full-document BM25 because segments contain mostly common academic function words with very long posting lists (30K–50K entries). The max_df threshold eliminates these non-discriminative terms, achieving an $\sim 20\times$ speedup (1.7–3.0s $\rightarrow$ 0.15s per segment). Heap-based top-K extraction (`heapq.nlargest`) replaces full sorting for additional efficiency.

Stage 3—Coverage-Weighted Source Voting. Segment-level rankings are aggregated into a final ranking via: $Score(d) = \sum_i BM25(d \mid c_i) \times (1 + count(d)/n)$, where $count(d)$ is the number of segments that retrieved document d in their top-50 results, and n is the total number of segments. The coverage bonus $(1 + count(d)/n) \in [1, 2]$ rewards documents consistently matched by multiple

independent segments—a strong signal of genuine multi-paragraph provenance rather than incidental single-phrase matching. This mechanism can be viewed as constructing an explicit provenance graph: nodes are query segments and candidate documents, edges represent segment-document matches, and the final ranking reflects each document's global importance in the graph.

3.3. Parameter Selection

The segmentation thresholds were chosen heuristically to match the structure of academic suspicious documents while controlling runtime. The $min\_chars = 800$ threshold removes very short paragraphs that are often dominated by boilerplate or local transition phrases, while $max\_chars = 3{,}000$ keeps chunks close to paragraph or subsection scale. The cap of 20 chunks prevents unusually long documents from dominating runtime; documents without paragraph boundaries fall back to a single query chunk.

Retrieval parameters were selected empirically on development samples and then kept fixed for TIRA submission. For segment retrieval, $max\_terms = 20$ and $max\_df = 5{,}000$ were chosen after the 200-query ablation in Table 4, where they preserved R@10 while reducing per-chunk search time from 1.7–3.0s to approximately 0.15s. Segment-level depth was set to top-50 to retain enough candidates for voting without accumulating many weak one-segment matches. Full-document BM25 baselines use $max\_terms = 100$ and top-100 output as stronger lexical baselines rather than as tuned values for our method.

4. Experiments

4.1. Experimental Setup

4.1.1. Dataset

Development data comes from the PAN 2025 generative plagiarism detection dataset converted to PAN26 format: 60,592 source documents from ClueWeb09 (avg. 11,765 tokens), 42,444 LLM-generated suspicious documents (training), 5,522 held-out queries with 7,949 source documents, and 43,140 query-document relevance annotations (qrels). The PAN26 official spot-check dataset (4 queries) was used only for format verification and TIRA submission validation. Official main test set results are from the TIRA evaluation platform [9] (accessed May 25, 2026).

4.1.2. Preprocessing

All texts are lowercased and tokenized via regex `[a-z0-9]+`. An inverted index is built with 1,273,004 unique terms. BM25 parameters: $k_1 = 1.2$, $b = 0.75$. Paragraph segmentation: $min\_chars = 800$, $max\_chars = 3{,}000$. Segment retrieval: $max\_terms = 20$, $max\_df = 5{,}000$, $top\_k = 50$. Standard retrieval: $max\_terms = 100$, $top\_k = 100$. All metrics are evaluated using `scripts/evaluate.py` (supporting multi-relevant qrels).

4.1.3. Evaluation Metrics

Four standard IR metrics are used: (1) Recall@K ($K = 10, 100$)—proportion of queries where the correct source appears in the top-K results; (2) nDCG@10—Normalized Discounted Cumulative Gain with position weighting; (3) MRR—Mean Reciprocal Rank; (4) RR—Reciprocal Rank, used by the official TIRA evaluation. All metrics range $[0, 1]$, higher is better.

4.1.4. Baselines

Three comparison categories: (1) Standard BM25—full document as single query, $max\_terms = 100$. (2) E5 Chunking—256-token document chunks encoded with E5-base-v2, coverage-aware scoring for BM25 blind-spot queries. (3) Ablation variants—fixed-chunk segmentation (3,000-char rolling windows)

Table 1
Development data method overview with the exact evaluation subset shown for each row.

Method	R@10	R@100	nDCG@10	MRR	Data Scale
Standard BM25	0.9196	0.9769	0.8184	0.7856	42,444q (full train)
+ Query Decomposition	0.9317	0.9890	0.8274	0.7937	982q (Cat1 blind-spot)
+ E5 Chunking	0.9830	0.9917	0.8716	0.8355	42,444q (full train)
Segmented BM25 (ours)*	0.9700	0.9912	0.9242	0.9092	800q sample (TIRA code)

Table 2
Full held-out validation on 7,949 source documents and 5,522 queries.

Method	R@10	R@100	nDCG@10	MRR
Standard BM25	0.9804	0.9874	0.9323	0.9174
Segmented BM25 (ours)	0.9947	0.9969	0.9770	0.9724

Table 3
Segmentation strategy comparison (held-out).

Strategy	R@10	MRR	Segmented%	Time (s)
No segmentation (BM25)	0.9804	0.9174	–	33
Paragraph split (ours)	0.9947	0.9724	99.9%	358
Fixed chunk	0.9933	0.9705	99.9%	300
Semantic split (section headers)	0.9746	0.9195	51.6%	49

and semantic segmentation (section header pattern matching). All methods share identical inverted indexes and BM25 parameters for fair comparison.

4.2. Results and Analysis

To make the evaluation setting reproducible, we keep development-sample, full held-out, and official TIRA results separate. Table 1 is a development-data overview whose last column identifies the exact subset used for each row. Table 2 reports the full held-out validation set, and Table 5 reports only the official PAN26 main test results.

*Segmented BM25 result from 800-query random sample (seed=20260520) using `submit_pan26.py` (exact TIRA submission code). Query Decomposition recovered 514/982 Cat1 blind-spot queries (52.3% recovery rate). E5 Chunking attains higher R@10 but requires a 440MB E5 model; our method is numpy-only.

Table 3 reveals several findings: (1) Paragraph segmentation achieves the best overall performance. (2) Fixed-chunk segmentation performs close to paragraph split but is slightly inferior, confirming that paragraph boundaries provide more meaningful semantic units than arbitrary character windows. (3) Semantic segmentation based on section header pattern matching (Introduction, Method, Results, etc.) actually degrades performance below standard BM25 (R@10 0.9746 vs. 0.9804), because only 51.6% of queries contain explicit section headers. This negative result is retained as a credibility-enhancing finding: simple rule-based semantic segmentation is not viable for this dataset.

The ablation study demonstrates that the three optimization strategies (max_df filtering, max_terms reduction, heapq top-K) together achieve an ~ 15–$20\times$ speedup with negligible accuracy loss (ΔR@10 = 0.002, within statistical noise).

On the official main test, our system achieves RR=0.9350 (ranked 3rd), demonstrating strong early-rank performance consistent with the design goal of coverage-weighted voting. Recall@10 and nDCG@10 both improve over the BM25 baseline (+9.2% and +13.2% respectively), indicating that when the method identifies the correct source, it tends to rank it very early.

Table 4
Ablation study—speed optimization (200 queries).

Configuration	R@10	nDCG@10	MRR	Time/chunk
(a) Standard BM25	0.900	0.805	0.775	~ 0.02s
(b) + segmentation ($max\_terms = 50$)	0.975	0.942	0.931	1.7–3.0s
(c) + $max\_df = 5{,}000$	0.973	0.940	0.929	~ 0.30s
(d) + $max\_terms = 20$ + heapq	0.975	0.942	0.931	~ 0.15s

Table 5
Official PAN26 main test results (TIRA, accessed May 25, 2026).

System	Recall@10	Recall@100	nDCG@10	nDCG@100	RR
BM25 baseline (official)	0.5767	0.7825	0.5532	0.6068	0.8318
fosuai / oscillating-list (ours)	0.6300	0.6300	0.6263	0.6263	0.9350
JLP / jlp (best system)	0.8542	0.9629	0.7994	0.8326	0.9760

The equality of Recall@100 and Recall@10 (both 0.6300) is an output-depth limitation of our submitted run, not a property of the underlying ranking formula. The run file submitted to TIRA contained only top-10 documents per query, while the task format allowed substantially deeper ranked lists. As a result, Recall@100 and nDCG@100 could not benefit from any additional candidates beyond rank 10. Future submissions should emit top-100 or top-1,000 candidates per query.

The separate gap between development-set R@10 (~ 0.97–0.99) and official test-set R@10 (0.63) reflects cross-dataset generalization rather than the output-depth issue. The converted development data and official main test differ in corpus scale, domain mixture, query generation, and plagiarism type distribution. This suggests that robust retrieval under unseen generative plagiarism distributions remains the central challenge of the task.

5. Conclusions

We presented Query-Chunk Provenance Modeling, a lightweight (numpy-only) source retrieval method for PAN26 generative plagiarism detection. The method decomposes suspicious documents into paragraph-level segments, independently retrieves sources per segment via BM25, and aggregates results through coverage-weighted voting to construct an explicit provenance graph. On development data, our method consistently outperforms standard BM25. On the official PAN26 main test, our system (fosuai / oscillating-list) achieved RR=0.9350 (3rd by RR). The submitted run's top-10 output limit constrained Recall@100; future work should output top-1000 and explore topic-based or embedding-based segmentation strategies for improved semantic coherence.

Code Availability

The implementation, including the TIRA Docker entry point `scripts/submit_pan26.py`, is available at https://github.com/Abraham-wy/Generative-Plagiarism-Detection. The official submitted system used team identifier `fosuai` and software name `oscillating-list`.

Acknowledgments

This work is supported by the National Social Science Foundation of China (24BYY080). We thank the PAN@CLEF 2026 organizers for providing the evaluation dataset and the TIRA platform.

Declaration on Generative AI

During the preparation of this work, the author(s) used Claude in order to: English abstract translation, and English language proofreading. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] 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, Overview of PAN 2026: Voight-Kampff Generative AI Detection, Text Watermarking, Multi-Author Writing Style Analysis, Generative Plagiarism Detection, and Reasoning Trajectory Detection, 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.

[2] 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.

[3] M. Potthast, M. Hagen, T. Gollub, M. Tippmann, J. Kiesel, P. Rosso, E. Stamatatos, B. Stein, Overview of the 5th International Competition on Plagiarism Detection, in: Working Notes of the CLEF 2013 Evaluation Labs, volume 1179 of *CEUR Workshop Proceedings*, CEUR-WS.org, Valencia, Spain, 2013. URL: https://ceur-ws.org/Vol-1179/CLEF2013wn-PAN-PotthastEt2013.pdf.

[4] S. E. Robertson, H. Zaragoza, The Probabilistic Relevance Framework: BM25 and Beyond, Foundations and Trends in Information Retrieval 3 (2009) 333–389. doi:10.1561/1500000019.

[5] N. Reimers, I. Gurevych, Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 3982–3992. doi:10.18653/v1/D19-1410.

[6] 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).

[7] V. Karpukhin, B. Oğuz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, W. tau Yih, Dense Passage Retrieval for Open-Domain Question Answering, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online, 2020, pp. 6769–6781. doi:10.18653/v1/2020.emnlp-main.550.

[8] S. Min, V. Zhong, L. Zettlemoyer, H. Hajishirzi, Multi-Hop Reading Comprehension through Question Decomposition and Rescoring, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL), Association for Computational Linguistics, Florence, Italy, 2019, pp. 6097–6109. doi:10.18653/v1/P19-1613.

[9] 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: 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.