# Overview of the Plagiarism Detection Task at PAN 2026

André Greiner-Petter[1,2,*], Yun-Ang Wu[3], Yuka Kitamura[2], Kon Woo Kim[2,4], Maik Fröbe[5], Bela Gipp[1], Akiko Aizawa[2] and Martin Potthast[6,7,8]

[1]Georg-August-Universität, Göttingen, Germany
[2]National Institute of Informatics (NII), Tokyo, Japan
[3]Research and Development Center for Large Language Models, NII, Tokyo, Japan
[4]The Graduate University for Advanced Studies (SOKENDAI), Kanagawa, Japan
[5]Friedrich-Schiller-Universität Jena, Jena, Germany
[6]University of Kassel, Kassel, Germany
[7]hessian.ai, Darmstadt, Germany
[8]ScaDS.AI, Leipzig, Germany

**Abstract**

The generative plagiarism detection task at PAN 2026 aims at retrieving the original scientific sources for given automatically generated textual plagiarism. We created two novel datasets of synthetically generated plagiarism following the premise of a malicious actor trying to generate papers with minimal effort while evading detection of undisclosed re-use. We utilized four widely available LLMs: GPT-OSS 120B, Llama 3.1 8B, Qwen 3.5 27B FP8, and DeepSeek-R1 Distill Llama 8B. In this task overview paper, we outline the creation of this dataset and summarize and compare the results of all participants with three baselines. For this year's task, we received 26 submissions from six teams that significantly outperformed our baselines.

**Keywords**

PAN, Plagiarism Detection, Semantic Document Retrieval, Semantic Similarity

## 1. Generative Plagiarism Retrieval

With the rapid advancements of large language models (LLMs), modern plagiarism detection systems must undergo a paradigm shift [1], moving away from traditional lexical-overlap approaches toward more sophisticated semantic analysis. Modern LLMs are already capable of generating novel, highly sophisticated scientific publications that can occasionally even pass rigorous academic peer review [2, 3, 4]. Consequently, academics increasingly and legitimately use AI to proofread, polish, or expand their scientific texts. This practice has also gained widespread acceptance within the community over the past few years.[1,2]

However, alongside these legitimate uses, synthetically generated articles can conceal severe cases of plagiarism and hallucinations [5]. Whether intentional or unintentional, these texts pose a significant risk to the integrity of the academic community. In this landscape, the mere presence of AI-generated text does not necessarily imply academic misconduct. Hence, modern plagiarism detection systems cannot simply rely on AI classifiers. Plagiarism detection must remain focused on identifying the uncredited reuse of ideas and text, necessitating new datasets to train and evaluate state-of-the-art approaches in a post-LLM academic landscape.

*CLEF 2026 Working Notes, 21 – 24 September 2026, Jena, Germany*
*Corresponding author.
greinerpetter@gipplab.org (A. Greiner-Petter); yunangwu@nii.ac.jp (Y. Wu); ykitamura@nii.ac.jp (Y. Kitamura); ken1204@nii.ac.jp (K. W. Kim); pan@webis.de (M. Fröbe); gipp@gipplab.org (B. Gipp); aizawa@nii.ac.jp (A. Aizawa); pan@webis.de (M. Potthast)
https://gipplab.org/team/dr-andre-greiner-petter/ (A. Greiner-Petter); pan.webis.de (M. Potthast)
0000-0002-5828-5497 (A. Greiner-Petter); 0009-0002-3292-2571 (Y. Wu); 0009-0004-3587-9841 (Y. Kitamura); 0000-0002-6970-5370 (K. W. Kim); 0000-0002-1003-981X (M. Fröbe); 0000-0001-6522-3019 (B. Gipp); 0000-0001-6544-5076 (A. Aizawa); 0000-0003-2451-0665 (M. Potthast)


[1]https://2023.aclweb.org/blog/ACL-2023-policy/
[2]https://neurips.cc/Conferences/2025/LLM

To foster research in this direction, we successfully revived the plagiarism detection task at PAN last year [6], providing a novel dataset featuring realistic, future-proof examples of sophisticated, generative plagiarism. While last year's edition focused on the alignment task, i.e., identifying the exact source paragraphs for a given pair of source and plagiarized documents, this year, we shift our focus to the source retrieval subtask.

For this year's challenge, we developed two novel automated pipelines that mimic the behavior of a malicious actor. This hypothetical actor's motivation is to generate realistic scientific publications with minimal effort while evading detection. Our pipeline feeds well-known, widely available LLMs with multiple genuine scientific articles, instructing the models to synthesize a new, combined, plagiarized article based on those provided sources. Participants are tasked with retrieving all sources for a given disingenuous, auto-generated article from a corpus of 100,000 scientific articles.

For this year's edition, we received a total of 26 submissions from six teams. Four of these teams provided notebook submissions explaining their approaches [7, 8, 9, 10], all from different countries. Furthermore, four teams provided their source code, enabling us to evaluate their methods on future datasets and potentially varying task configurations. In the following, we discuss the dataset creation in detail and perform a more comprehensive analysis of the submissions. This paper is an extended version of the brief summary in the PAN 2026 overview publication [11].

## 2. Dataset

As in the previous year, we utilized ar5iv[3] (an HTML5 mirror of arXiv) as the underlying source corpus for our dataset. From this data, we constructed 4-tuples $S_k = (S_{k1}, S_{k2}, S_{k3}, S_{k4})$ as the sources for each plagiarized document $P_k$.

### 2.1. Source Tuples Selection

To construct realistic, non-trivial source sets for generative plagiarism, we developed a two-stage, category-aware selection pipeline over a retrieval pool of 320,878 ar5iv document embeddings based on Specter [12] (768 dimensions). In the first stage, the pipeline selects 1,000 *seed documents* to serve as anchors ($S_{k1}$) for each source set. These seeds are sampled proportionally across arXiv categories with a per-category cap. This constraint maximizes topical diversity across the final dataset and prevents large subject areas from dominating the sample. In the second stage, the pipeline constructs the rest of the 4-tuple by selecting three additional companion documents ($S_{k2}, S_{k3}, S_{k4}$) for each seed. To ensure these companion documents are highly related but do not degenerate into trivial near-duplicates, candidates are retrieved from the seed's semantic neighbors and filtered using the following criteria:

- **Semantic Similarity Band:** Candidates are strictly restricted to the 60th to 95th percentile of per-source semantic similarity. This effectively prunes weakly related papers on the lower end and excludes near-duplicate matches on the upper end.
- **Category Overlap:** Candidates are evaluated based on their arXiv category overlap with the seed document to ensure contextual cohesion.
- **Overlap Penalties:** The pipeline applies penalties for excessive lexical overlap (measured via character 5-gram Jaccard similarity on titles and abstracts) and structural overlap (estimated from section-title signatures).

### 2.2. Sequential Plagiarism Generation

Our primary dataset features a sequential generation pipeline. At a high level, the sequential method merges papers via a chain: given $N$ source papers $s_1, \ldots, s_N$, it produces intermediate outputs as $(s_1, s_2) \rightarrow g_1$, $(g_1, s_3) \rightarrow g_2$, …, $(g_{N-2}, s_N) \rightarrow g_{N-1}$. For this dataset, we set $N \in 2, 3, 4$. Each

[3] https://ar5iv.labs.arxiv.org/

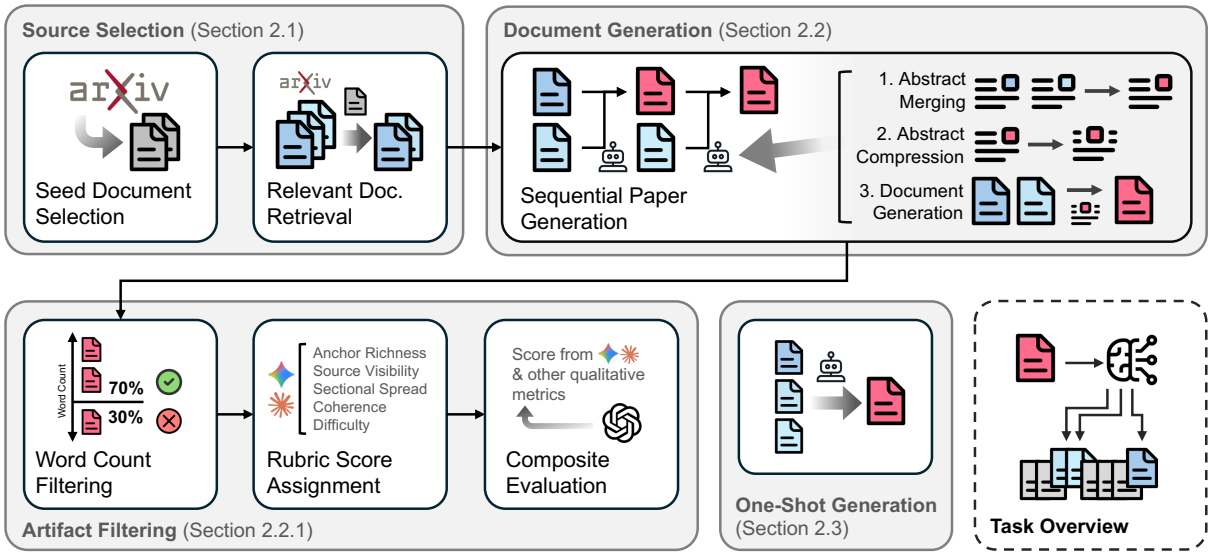


**Figure 1:** Overview of the dataset generation pipeline. After artifact filtering, the set of sequentially generated documents serves as the primary evaluation task data.

merge consists of three LLM stages: abstract merging, abstract compression, and full paper generation. The final output is a JSON object with six sections: abstract, introduction, methodology, experiment, discussion, and conclusion.

The design is motivated by two technical challenges when generating a paper from multiple sources. First, models tend to merge papers in an obvious way. Since each source has a distinct narrative, the model often produces survey-like writing rather than a coherent, unified paper. To address this, we first ask the model to merge only the abstracts, as synthesizing at a higher level of abstraction makes it easier to construct a coherent argument before generating the full text. Second, we are concerned about context length: if the prompt is too long, the model may fail to produce meaningful output. By applying merges sequentially rather than all at once, we keep each prompt shorter and each individual merge higher in quality.

We split the 1,000 source paper sets randomly into four groups of 250, each assigned to a different model: gpt-oss-120b [13] (GPT-OSS), Llama-3.1-8B-Instruct [14] (Llama 3.1), Qwen3.5-27B-FP8 [15] (Qwen 3.5), and DeepSeek-R1-Distill-LLaMA-8B [16] (DeepSeek-R1). Within each group, we allocate 50, 150, and 50 instances for $N = 2$, 3, and 4, respectively. Due to the left-fold design, earlier source papers have less direct influence on the final generated paper than later ones. We reflect this in our relevance scoring: for $N = 2$ the scores are $[1, 1]$, for $N = 3$ they are $[1, 1, 2]$, and for $N = 4$ they are $[1, 1, 2, 3]$. Not every generation succeeds; some fail due to excessive prompt length or malformed model outputs.

This yielded 898 usable generated query artifacts across the four generator LLMs. The remaining 102 attempts did not yield usable artifacts for subsequent filtering, most commonly because the model attempted to generate LaTeX-style text containing backslashes, which caused JSON decoding issues during the parsing stage. Specifically, 8 attempts failed for GPT-OSS, 52 for Llama 3.1, 8 for Qwen 3.5, and 34 for DeepSeek-R1.

### 2.2.1. Artifact Filtering

While the generated documents had a coherent structure, they still may contain obvious cases of copy-paste textual reuse or LLM artifacts. Motivated by the fact that a human malicious actor would run some quick filtering in order to conceal the reuse, we also applied quality filtering. First, we applied an LLM-specific word-count filter to the usable outputs. Rather than imposing a single global cutoff, we selected LLM-specific thresholds to retain approximately 70% of the usable outputs within each generator group. The remaining queries were independently scored by Gemini 3 Flash [17] and Claude Sonnet 4.6 [18] using a heuristic scoring rubric based on retrieval-anchor richness, multi-source

**Table 1**
Source distributions for each LLM generator on the sequential dataset. The upper half shows the distribution for the number of sources for the 200 plagiarized documents. The lower half shows all (a source might have multiple categories) major arXiv categories of the sources respectively.

| | – | GPT-OSS | Llama 3.1 | Qwen 3.5 | DeepSeek-R1 | Total | Share |
|---|---|---|---|---|---|---|---|
| Sources | 2 | 5 | 10 | 7 | 10 | 32 | 16.0% |
| | 3 | 30 | 27 | 35 | 30 | 122 | 61.0% |
| | 4 | 15 | 13 | 8 | 10 | 46 | 23.0% |
| | *Sources total* | 50 | 50 | 50 | 50 | 200 | 100.0% |
| arXiv category | Physics | 33 | 30 | 32 | 37 | 132 | 66.0% |
| | Computer Science | 16 | 17 | 17 | 13 | 63 | 31.5% |
| | EESS | 7 | 4 | 6 | 0 | 17 | 8.5% |
| | Quantitative Biology | 0 | 1 | 0 | 3 | 4 | 2.0% |
| | Statistics | 0 | 4 | 0 | 1 | 5 | 2.5% |
| | Mathematics | 1 | 0 | 0 | 2 | 3 | 1.5% |
| | *Category total* | 57 | 56 | 55 | 56 | 224 | 100.0% |

visibility, sectional spread of retrieval clues, coherence as a suspicious generated paper, and difficulty balance. The total score was summed across all rubric categories. GPT-5.4 [19] served as an automated judge for the final selection of 50 queries per LLM based on the rubric scores. GPT-5.4 was prompted to consider the score outputs together with qualitative evidence of retrieval anchors, multi-source traces, coherence, template-like phrasing, overly explicit source cues, and obvious merge artifacts.

The final selection therefore favored queries that retained recoverable but non-trivial source traces while remaining coherent as suspicious generated papers. Candidates could be down-ranked despite favorable scorer totals when they were overly generic, too explicit, template-like, or visibly stitched together. The surviving queries have a mean rubric score of 8.03 and 6.05 from Gemini and Claude, respectively. In contrast, the filtered queries have mean rubric scores of 5.89 and 4.82 from Gemini and Claude, respectively. Table 1 provides a distribution overview of the sources across the final 200 plagiarized documents.

## 2.3. One-Shot Generation

The sequential generation pipeline, in combination with quality filtering, provides a controlled and expandable mechanism for generating plagiarism. However, it also provides significant guardrails for the LLM to generate reused content. Although this provides us with a relatively controlled environment to evaluate retrieval approaches, it is also significantly less realistic, as a malicious actor would not actively ensure traceable amounts of source material in the generated output. This motivated us to create a second, considerably less restrictive dataset internally as another testing ground.

For this set, we feed $N$ source papers to the model within a single prompt, and the model is asked to generate the synthesized paper (a JSON object) in one shot. The source papers are passed to the model directly, together with a prompt explaining the task. The only guidelines within this prompt are aimed at length control. Within the prompt, we include a guide specifying how many words we want for each section in order to steer the model toward generating an appropriate amount of text.

For one-shot generation, we apply the method to the same 1,000 source paper sets used for sequential generation. Because the model must read the full text of every source and synthesize it directly, the generation step is more challenging; we therefore use only GPT-OSS, the most capable of our four selected models. Additionally, we fix the number of input papers to $N = 3$. Since all papers contribute to the input in equal proportion, we set the relevance scores to $[1, 1, 1]$. After removing erroneous outputs, the one-shot generation dataset contained 933 plagiarized documents.

## 3. Evaluation

In the following, we evaluate all submissions and baselines using normalized discounted cumulative gain and recall, both evaluated at a specific cutoff (nDCG@$N$ and R@$N$), and mean reciprocal rank (MRR) without a cutoff. All metrics are averaged across the evaluation corpus. Let $P = \{P_1, \ldots, P_{|P|}\}$ denote the complete set of plagiarized documents (queries). For a given plagiarized document $P_k$, let $D_k$ represent the ranked list of retrieved source documents returned by a system, and let $S_k$ be the ground-truth set of true source documents. The evaluation metrics are defined as follows:

**Recall at Cutoff N (R@N)** Recall at a specific cutoff measures the fraction of total ground-truth relevant documents successfully retrieved within the top-$N$ ranks:

$$R_k@N = \frac{|D_k[1 \ldots N] \cap S_k|}{|S_k|}$$

where $D_k[1 \ldots N]$ is the subset of documents found in the top-$N$ positions of the system's ranking. R@$N$ is the average of $R_k@N$ over all queries.

**Mean Reciprocal Rank (MRR)** Reciprocal rank targets the position of the first relevant document retrieved. If the highest-ranked true source document for a query $P_k$ occurs at rank $s_k$, its reciprocal rank is $1/s_k$ (and 0 if no relevant documents are retrieved). MRR is the average of these reciprocal ranks across all evaluation queries:

$$\text{MRR} = \frac{1}{|P|} \sum_{k=1}^{|P|} \frac{1}{s_k}$$

**Normalized Discounted Cumulative Gain at Cutoff N (nDCG@N)** Discounted Cumulative Gain (DCG) accounts for the positions of all relevant documents within the top $N$ positions, penalizing relevant items if they appear lower in the ranking. For a single query, DCG@$N$ is defined as:

$$\text{DCG}_k@N = \sum_{i=1}^{N} \frac{2^{\text{rel}_{ik}} - 1}{\log_2(i+1)}$$

where $\text{rel}_{ik}$ is the relevance score for the retrieved document at position $i$ in $D_k$. As the number of relevant documents varies across $P_k$ (between two and four sources), nDCG normalizes DCG by the ideal (iDCG) ordering of retrievable sources. iDCG therefore represents the maximum DCG for a given query.

$$\text{nDCG}_k@N = \frac{\text{DCG}_k@N}{\text{iDCG}_k@N}$$

As defined in Section 2.1, our ideal relevance lists for the sequentially generated dataset depend on the number of sources. For two sources, the relevance scores are $[1, 1]$; for three sources, they are $[2, 1, 1]$; and for four sources, they are $[3, 2, 1, 1]$. For the one-shot generation dataset, the relevance scores are binary $\{0, 1\}$. nDCG@$N$ is again defined as the average over all queries.

### 3.1. Baselines

We provided three baselines: BM25, LinQ embeddings [20], and Numerical Matching. The BM25 baseline ranks all documents based on their BM25 similarity to the query document. Similarly, the LinQ baseline ranks documents based on cosine similarity, but utilizes LinQ embeddings instead of BM25 vectors. The selection of LinQ was motivated by its status as the best-performing model in the previous year's alignment task [6]. Furthermore, guided by our manual analysis of several generated documents, we employed a simple numerical matching baseline. This approach is based on the observation that generated documents likely exhibit an unusually high number of matching numerical values with their source texts, as LLMs typically treat numbers as hard facts and rarely alter them during paraphrasing.

### 3.2. Team Submissions

We received 26 submissions from six teams. Two teams (Fsu-AIGC-Detector and NLP-DE) did not provide notebooks explaining their methodologies and are therefore listed separately in the results. Four teams (ENGsTF, FosuAI, Bedratiuk-Lab, and Fsu-AIGC-Detector) provided code submissions, allowing us to automatically evaluate their approaches on the internal one-shot generation dataset as well. Most teams submitted multiple runs, of which we focus only on the major increments explained by the teams.

**Team JLP [7]** achieved the highest scores on the sequential dataset across all metrics. Team JLP's system decomposes each suspicious document into sentence-level spans, using each sentence as an independent query against two complementary indices: a full-document BM25 index for lexical overlap and a dense GTE-Base v1.5 bi-encoder index over structure-aware paragraph chunks. A key differentiator from other teams with similar pipelines is that Team JLP constructed a 200-query synthetic development set from the source corpus, enabling systematic hyperparameter tuning via Optuna. This allowed them to explore variables that other teams left untuned. Most notably, span length, where single-sentence spans surprisingly outperformed all larger windows (up to 10 sentences or full-document queries). This suggests that tighter topical focus at the sentence level is critical for isolating source-specific evidence.The team submitted two runs differing only in the final aggregation step: BM25 + GTE RRF (jlp-1) sums raw RRF span scores via CombSUM, while BM25 + GTE RRF minmax (jlp-2) applies min-max normalization per span before summing.

**Team ENGsTF [8]** employed a two-stage retrieval pipeline that is representative of several submissions. In the first stage, source documents are split into overlapping chunks and retrieved using a hybrid dense–sparse approach. Dense retrieval is performed either with E5 or the long-context BGE-M3 encoder, while sparse retrieval uses BM25. Chunk-level dense scores are aggregated to the document level using max pooling and linearly combined with the BM25 scores. In the second stage, the highest-ranked candidates are reranked with the BGE-reranker-v2-m3 cross-encoder. Since the reranker cannot process full-length queries, the queries are chunked, and the maximum score across all query chunks is used. The submitted runs differ primarily in the dense retriever and the use of reranking, namely E5 + FAISS, E5 + BM25, BGE-M3 + BM25, BGE-M3 + BM25 + reranker, and BGE-M3 + BM25 + reranker with score blending (bl.), where the final run linearly combines the reranker and retrieval scores instead of replacing the latter.

**Team FosuAI [9]** proposed a lightweight lexical retrieval approach based on Query-Chunk Provenance Modeling. Instead of querying with the entire suspicious document, the document is split into paragraph-level chunks that independently retrieve candidate sources using BM25. The resulting rankings are then combined through a coverage-weighted voting scheme, which favors documents that are retrieved consistently across multiple query chunks rather than through isolated lexical matches. Unlike most other submissions, the approach relies solely on traditional lexical retrieval without dense encoders or rerankers, making it computationally lightweight while explicitly modeling multi-source provenance. The submitted run, however, returned only the top 10 candidates, limiting the reported Recall@100 and nDCG@100 to the same values as their @10 counterparts.

**Team Bedratiuk-Lab [10]** proposed a two-stage retrieval pipeline combining dense retrieval with several local reranking signals. Candidate documents are first retrieved using a MiniLM-based dense encoder. The reranking stage combines three complementary signals. Fragment matching aggregates sentence-level alignments with a coverage signal, while semantic attention softly aggregates similarities across multiple candidate sentences instead of relying only on the best match. The third component compares normalized local embedding configurations to capture structural semantic correspondence beyond pairwise similarity. The final ranking is obtained by combining the dense retrieval score with the three reranking signals, allowing the system to exploit both global document similarity and local semantic evidence.

**Table 2**
Overall results for the plagiarism detection task, grouped by team and ranked by nDCG@10 performances. Best results are marked in bold. FosuAI, Bedratiuk-Lab, and Fsu-AIGCC-Detector (unk-1) only retrieved 10 results and are excluded from nDCG@100 and R@100.

| Team | Approach | nDCG@10 | nDCG@100 | R@10 | R@100 | MRR |
|---|---|---|---|---|---|---|
| JLP [7] | BM25 + GTE RRF | **0.799** | **0.833** | **0.854** | 0.963 | **0.976** |
| | BM25 + GTE RRF minmax | 0.779 | 0.817 | 0.838 | **0.965** | 0.961 |
| FosuAI [9] | BM25 Segmented | 0.626 | 0.626 | 0.630 | 0.630 | 0.935 |
| ENGsTF [8] | BGE-M3 + BM25 + RR bl. | 0.642 | 0.687 | 0.671 | 0.840 | 0.908 |
| | BGE-M3 + BM25 | 0.613 | 0.658 | 0.648 | 0.832 | 0.912 |
| | E5 + BM25 | 0.585 | 0.637 | 0.612 | 0.820 | 0.876 |
| | BGE-M3 + BM25 + RR | 0.491 | 0.561 | 0.565 | 0.823 | 0.695 |
| | E5 + FAISS | 0.385 | 0.448 | 0.433 | 0.676 | 0.647 |
| Bedratiuk-Lab [10] | MiniLM + Reranker | 0.385 | 0.385 | 0.386 | 0.386 | 0.619 |
| Fsu-AIGC-Detector | unk-1 | 0.643 | 0.643 | 0.657 | 0.657 | 0.892 |
| | unk-2 | 0.301 | 0.374 | 0.360 | 0.643 | 0.498 |
| NLP-DE | unk-1 | 0.076 | 0.111 | 0.098 | 0.225 | 0.137 |
| | unk-2 | 0.040 | 0.067 | 0.057 | 0.157 | 0.076 |
| Baselines | BM25 | 0.553 | 0.607 | 0.577 | 0.782 | 0.832 |
| | LinQ Embeddings | 0.391 | 0.467 | 0.445 | 0.726 | 0.699 |
| | Numeral Matching | 0.184 | 0.199 | 0.170 | 0.220 | 0.321 |

## 3.3. Discussion and Results

Table 2 summarizes the results for the major approaches, ranked by the best nDCG@10 performance of each team. Notably, BM25 proved to be a surprisingly strong baseline. The primary reason for this robust performance likely lies in the prevalence of core scientific and mathematical disciplines within the arXiv repository. Scientific articles in domains such as physics and computer science contain rich, highly specific terminology that often remains even after heavy paraphrasing. These untouched, concise text fragments are ideal for TF-IDF-based retrieval methods. Provided the texts are sufficiently long to contain multiple such concise fragments (a condition further supported by our generation and filtering pipeline), BM25 establishes a very solid foundation. Recognizing this, several participating teams exhibited a predominant bias toward BM25 variants in their systems and often combined BM25 with a dense retrieval approach based on embeddings. The main difference between approaches came down to decomposing queries and sources and combining the resulting scores as efficiently as possible for the deployed embedding models.

**Table 3**
Overall results on the one-shot generation dataset. FosuAI, Bedratiuk-Lab, and Fsu-AIGCC-Detector (unk-1) only retrieved 10 results and are excluded from nDCG@100 and R@100. Fsu-AIGC-Detector did not submit a notebook and is listed separately.

| Team | Approach | nDCG@10 | nDCG@100 | R@10 | R@100 | MRR |
|---|---|---|---|---|---|---|
| FosuAI [9] | BM25 Segmented | **0.299** | 0.299 | **0.286** | 0.286 | **0.451** |
| ENGsTF [8] | E5 + BM25 | 0.292 | **0.311** | 0.283 | **0.344** | 0.446 |
| | BGE-M3 + BM25 | 0.256 | 0.277 | 0.245 | 0.320 | 0.407 |
| | BGE-M3 + BM25 + RR | 0.189 | 0.223 | 0.210 | 0.321 | 0.293 |
| | E5 + FAISS | 0.185 | 0.214 | 0.191 | 0.292 | 0.299 |
| Bedratiuk-Lab [10] | MiniLM + Reranker | 0.147 | 0.147 | 0.141 | 0.141 | 0.254 |
| Fsu-AIGC-Detector | unk-1 | **0.313** | 0.313 | **0.295** | 0.295 | **0.461** |
| | unk-2 | 0.148 | 0.180 | 0.153 | 0.263 | 0.258 |
| Baselines | BM25 | 0.209 | 0.231 | 0.209 | 0.281 | 0.332 |
| | LinQ Embeddings | 0.190 | 0.218 | 0.189 | 0.289 | 0.325 |
| | Numeral Matching | 0.031 | 0.040 | 0.036 | 0.070 | 0.051 |

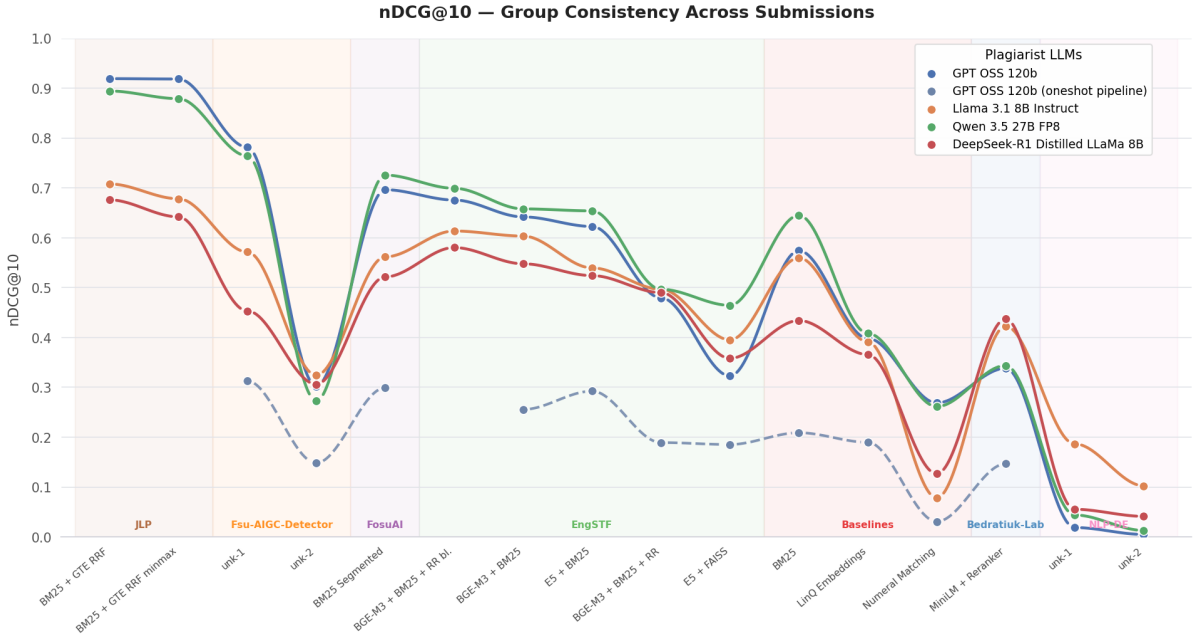


**Figure 2:** nDCG@10 for each generation model across all submissions and both datasets.

Table 3 provides the results of all code submissions on the one-shot generation dataset. Team ENGsTF's best-performing approach exceeded the computing cap of 48 hours on our Tira [21] instance,[4] and is therefore not included in the results. On this dataset, all approaches, including all baselines, perform significantly worse. The main reason for this severe performance drop of over 30% comes down to the increased freedom in the generation of plagiarism. While the sequential pipeline was designed to ensure a relatively consistent level of source material within the generated document (via sequential feeding and quality filtering), the one-shot prompting approach allowed the LLM to hallucinate significantly more and choose which sources to use or not use. As such, MRR is the better metric in Table 3, because we cannot guarantee that all sources are still traceable within the generated document. On MRR, the performance drops are less significant, considering that an MRR of 0.5 means the first correctly retrieved document was, on average, still in the second position. The high MRR values above 0.9, even for low nDCG or Recall scores on the sequential dataset, show that it was significantly easier to retrieve at least one source.

Figure 2 shows nDCG@10 scores across all generation models for all submissions. GPT-OSS and Qwen 3.5 sources consistently score higher in comparison to Llama 3.1 and DeepSeek-R1, with the exception of some weaker submissions. While the proximity of Llama 3.1 and DeepSeek-R1 to each other can be explained by the selection of the distilled variant of Llama 3.1 for DeepSeek-R1, it is unclear why GPT-OSS and Qwen 3.5 exhibit similarly close performance across all submissions. Figure 3 provides the same overview but with respect to MRR scores. Here, no clear trend across different generation models emerges. This indicates that Llama 3.1 and DeepSeek-R1 lose more traceable source material from previous iterations during sequential feeding in comparison to GPT-OSS and Qwen 3.5.

Figure 4 further highlights the bias toward retrieving at least one source. This figure provides the distribution of $nDCG_k@10$ scores for all queries across all submissions. With a finite number of retrievable documents, nDCG is discrete and exhibits clear score clusters at certain retrieval combinations. Figure 4 highlights the most common clusters on the right. For example, all teams except JLP and NLP-DE have a clear cluster of scores around the '$[2, 0, 0]$ in $[2, 1, 1]$' mark, i.e., they retrieved the top-relevant source (the last in the source chain) of three sources, while none of the other initial sources (the initial two documents fed to the model at the same time) were retrieved. Team JLP's missing cluster around the same mark highlights that the initial two sources were still traceable within the plagiarized document using their approach.

[4]The instance has 5 CPU cores, 50GB RAM, and an A100 GPU with 40GB VRAM.

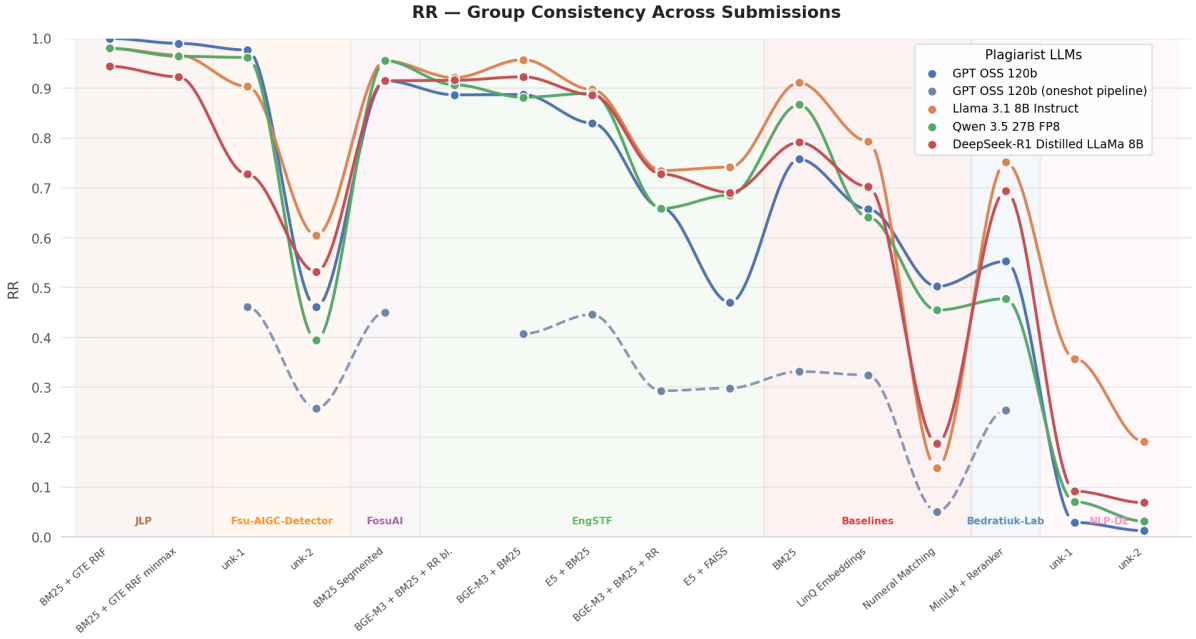


**Figure 3:** MRR for each generation model across all submissions and both datasets.

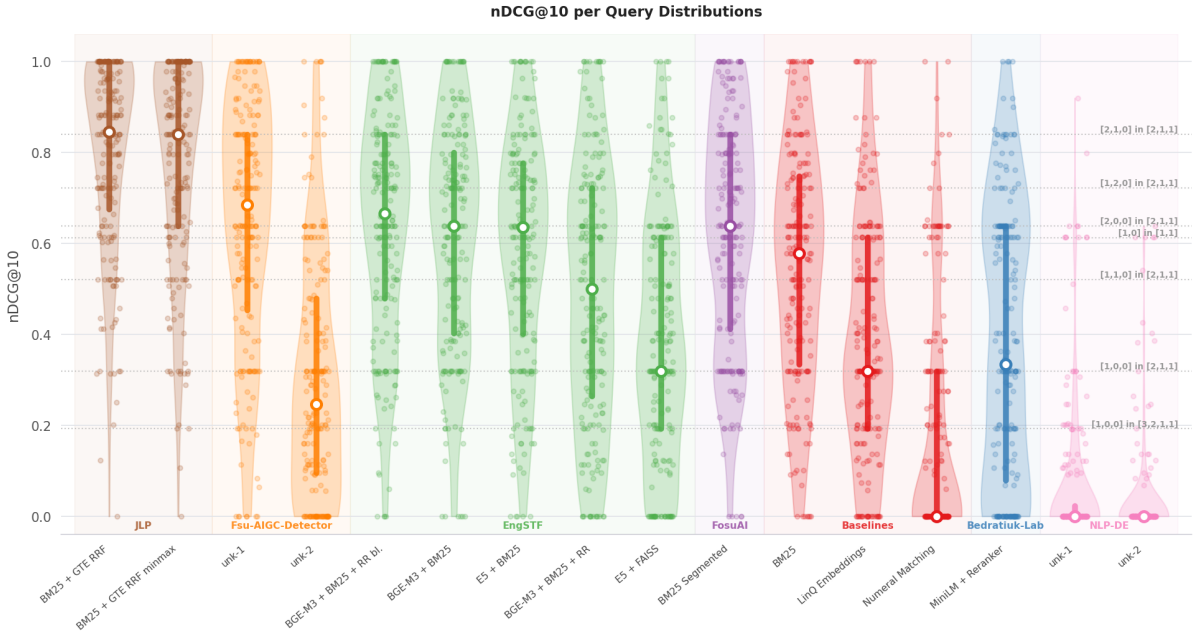


**Figure 4:** Distributions of all queries' nDCG@10 scores across all submissions.

Figure 5 provides a full overview of all 200 queries grouped by their underlying generative model and sorted from easier to harder (on average) across all submissions. We can see that only one query, the left-most query generated by DeepSeek-R1, resulted in a score of 0 across all submissions and baselines. This indicates that the quality measures and guardrails during generation have been largely successful in ensuring that all plagiarized documents still contain sufficient amounts of source material. Manual investigation of this single query shows that it does indeed appear to have no obvious connections to the provided sources and instead seems to be fully hallucinated. The query abstract focuses on digital health tools and mental well-being, while the first source discusses AI-supported story-to-video production and the second source discusses contextual vision transformers.

Figure 6 provides the same overview but for the one-shot dataset. Here, 396 (42.4%) of the queries

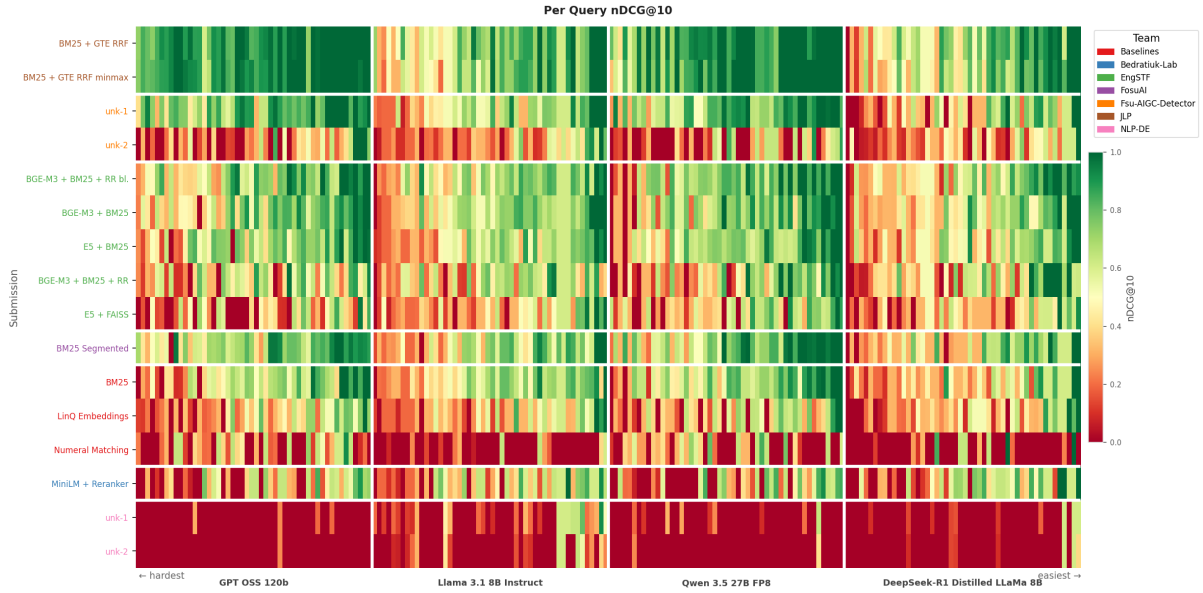


**Figure 5:** Full sequential dataset nDCG@10 scores for each query and submission. The queries are grouped by the generative model and than sorted within each group by the averaged nDCG@10 over all submissions from left to right increasing. In other words, more left queries have been, the harder they were for the submitted approaches, while the more right a query appears, the more easier it has been.

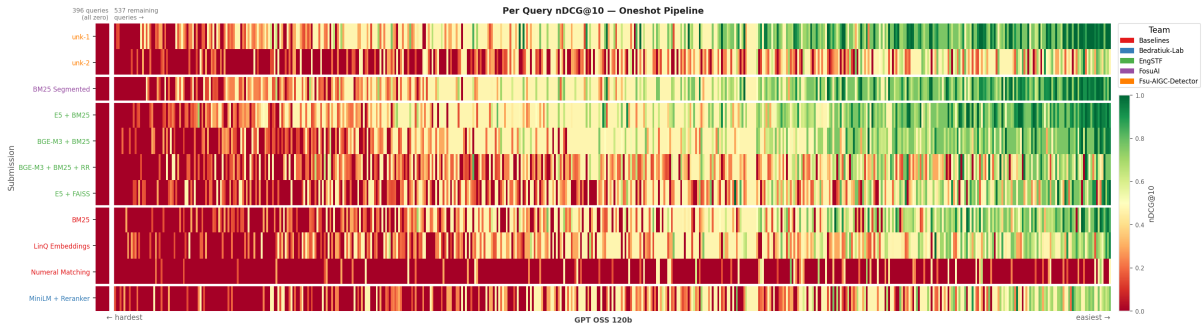


**Figure 6:** Full one-shot dataset nDCG@10 scores for each query and submission. The queries are sorted by the averaged nDCG@10 over all submissions from left to right increasing. The queries achieving score 0 (left-most) are squashed.

scored 0 across all submissions and baselines, i.e., no source was retrieved within the first 10 retrieved documents. Note that most submissions were designed to return only the top 10 results since the task's primary metrics focused mainly on the top 10 results.[5] This further highlights the increased difficulty of the one-shot dataset but likewise also raises concerns about whether those queries could still be categorized as plagiarism. On the other hand, this dataset is also significantly more realistic. This trade-off between robust task design and realism primarily prompted us to treat the one-shot generation dataset as an internal test set for now and leave more sophisticated quality assurance for future iterations.

## 4. Conclusion & Future Work

For this year's edition, we curated two novel datasets to evaluate source retrieval methodologies for plagiarism detection systems and received 26 submissions from six teams. Four of the six teams were

[5]For nDGC@100, 392 queries achieved a score of 0 across all submissions.

able to surpass the BM25 baseline by combining BM25 with dense retrieval approaches based on model embeddings. Team JLP submitted the winning approaches, utilizing the provided corpus to curate a synthetic development set that allowed them to fine-tune critical components of their pipeline. Our second dataset, based on a one-shot strategy without guardrails or quality control, was significantly more difficult for the submitted approaches. While this dataset serves as a more realistic, real-world scenario, it also raises the question of whether some generated documents still contain sufficient traceable source material to constitute plagiarism. Without a clear answer to this question, we did not use this dataset as the primary evaluation set for this year.

## Acknowledgments

This work has been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 554559555.

## Declaration on Generative AI

The authors utilized Gemini, Claude, and ChatGPT solely for language editing and code assistance. The authors reviewed all AI-assisted content, take ultimate responsibility for the final manuscript, and confirm that any additional uses of AI are explicitly noted in the text.

## References

[1] J. P. Wahle, T. Ruas, T. Foltýnek, N. Meuschke, B. Gipp, Identifying machine-paraphrased plagiarism, in: M. Smits (Ed.), Information for a Better World: Shaping the Global Future - 17th International Conference, iConference 2022, Virtual Event, February 28 - March 4, 2022, Proceedings, Part I, Lecture Notes in Computer Science, Springer, 2022, pp. 393–413. URL: https://doi.org/10.1007/978-3-030-96957-8_34. doi:10.1007/978-3-030-96957-8_34.

[2] C. Si, D. Yang, T. Hashimoto, Can llms generate novel research ideas? A large-scale human study with 100+ NLP researchers, in: The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025, OpenReview.net, 2025. URL: https://openreview.net/forum?id=M23dTGWCZy.

[3] C. Lu, C. Lu, R. T. Lange, J. Foerster, J. Clune, D. Ha, The ai scientist: Towards fully automated open-ended scientific discovery, arXiv (2024). doi:10.48550/arxiv.2408.06292.

[4] Y. Yamada, R. T. Lange, C. Lu, S. Hu, C. Lu, J. Foerster, J. Clune, D. Ha, The ai scientist-v2: Workshop-level automated scientific discovery via agentic tree search, arXiv (2025). doi:10.48550/arXiv.2504.08066.

[5] T. Gupta, D. Pruthi, All that glitters is not novel: Plagiarism in AI generated research, in: W. Che, J. Nabende, E. Shutova, M. T. Pilehvar (Eds.), Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025, Association for Computational Linguistics, 2025, pp. 25721–25738. URL: https://aclanthology.org/2025.acl-long.1249/.

[6] A. Greiner-Petter, M. Fröbe, J. P. Wahle, T. Ruas, B. Gipp, A. Aizawa, M. Potthast, Overview of the plagiarism detection task at PAN 2025, in: Working Notes of the Conference and Labs of the Evaluation Forum, CLEF 2025, Madrid, Spain, 9-12 September 2025, CEUR Workshop Proceedings, CEUR-WS.org, 2025, pp. 3575–3585. URL: https://ceur-ws.org/Vol-4038/paper_280.pdf.

[7] L. K. Cody, P. F. Panzitt, J. E. Cox, Team JLP at PAN 2026: Span-Based Hybrid Source Retrieval for Generative Plagiarism Detection, 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-WS.org, 2026.

[8] F. Alkayyem, T. Alkhateeb, R. Sonbol, R. Alhossny, Team ENGsTF at PAN 2026: Hybrid Sparse-Dense Retrieval with Two-Stage Score Fusion for Generative Plagiarism Detection, 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-WS.org, 2026.

[9] Y. Wang, Z. Han, C. Liu, H. Cao, Team Fosuai at PAN 2026: Source Retrieval via Query-Chunk Provenance Modeling for Generative Plagiarism Detection, 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-WS.org, 2026.

[10] L. Bedratyuk, Team bedratiuk-lab at PAN 2026: From Embedding Similarity to Semantic Canonization for Generated Plagiarism Detection, 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-WS.org, 2026.

[11] 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.

[12] A. Cohan, S. Feldman, I. Beltagy, D. Downey, D. S. Weld, SPECTER: document-level representation learning using citation-informed transformers, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020, Association for Computational Linguistics, 2020, pp. 2270–2282. doi:10.18653/V1/2020.ACL-MAIN.207.

[13] OpenAI, S. Agarwal, L. Ahmad, J. Ai, S. Altman, A. Applebaum, et al., gpt-oss-120b & gpt-oss-20b model card, 2025. URL: https://arxiv.org/abs/2508.10925. arXiv:2508.10925.

[14] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, et al., The llama 3 herd of models, 2024. URL: https://arxiv.org/abs/2407.21783. arXiv:2407.21783.

[15] Qwen Team, Qwen3.5: Towards native multimodal agents, 2026. URL: https://qwen.ai/blog?id=qwen3.5.

[16] DeepSeek-AI, Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL: https://arxiv.org/abs/2501.12948. arXiv:2501.12948.

[17] Google, Gemini 3 flash is now available in gemini cli, Google Developers Blog, 2025. URL: https://developers.googleblog.com/gemini-3-flash-is-now-available-in-gemini-cli/.

[18] Anthropic, Introducing claude sonnet 4.6, 2026. URL: https://www.anthropic.com/news/claude-sonnet-4-6.

[19] OpenAI, Introducing gpt-5.4, 2026. URL: https://openai.com/index/introducing-gpt-5-4/.

[20] J. Kim, S. Lee, J. Kwon, S. Gu, Y. Kim, M. Cho, J. yong Sohn, C. Choi, Linq-embed-mistral: Elevating text retrieval with improved gpt data through task-specific control and quality refinement, Linq AI Research Blog, 2024. URL: https://getlinq.com/blog/linq-embed-mistral/.

[21] 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.

# A. Sequential Pipeline Prompts

### Step 1: Abstract Merging

**System:** You are an expert academic writer.

**Task:** Read Abstract A and Abstract B. Write ONE merged abstract for a new survey/merged paper that integrates both sources. Keep it faithful: do not invent claims that are not supported by either abstract. If the two sources disagree, mention the disagreement neutrally. Prefer concise, information-dense writing.

**Output format:** Output ONLY the merged abstract text (no title, no bullets, no JSON).

**Abstract A:** `{{ abstract_a }}`
**Abstract B:** `{{ abstract_b }}`

### Step 2: Abstract Compression

Rewrite the following abstract to conform to the length and tone of a formal academic abstract. Prioritize clarity, coherence, and conciseness; nonessential details may be omitted. Reduce the length to approximately one-third of the original. Do not make it sound like a combination of several works; instead, make it a coherent piece with a logical flow. Please do not write in bullet points (or a similar style).

**Abstract:** `{{ abstract }}`

### Step 3: Full Paper Generation

**System:** You are an expert academic writer.
You will be given a compressed abstract for the target merged paper, Source Materials A (may be an original paper excerpt OR an already-generated paper), and Source Materials B (same).

**Task:** Write a coherent merged paper that integrates A and B. Do NOT copy large chunks verbatim; synthesize and paraphrase. Do NOT invent unsupported experiments, datasets, metrics, or numeric results. If you must be uncertain, explicitly hedge. Keep style consistent across sections.

Return STRICT JSON with EXACTLY the following keys: `{"abstract", "introduction", "methodology", "experiment", "discussion", "conclusion"}`

Each value must be a non-empty string. Output ONLY valid JSON (no markdown fences, no commentary).

**Compressed Abstract:** `{{ abstract }}`
**Source Materials A:** `{{ materials_a }}`
**Source Materials B:** `{{ materials_b }}`

# B. One-Shot Pipeline Prompt

### One-Shot Generation Prompt

**System:** You are an academic paper generator. You will be given {n} source papers. Your task: synthesize them into a single new, original research paper that draws on ideas, methods, and findings from all sources.

Each section must be SUBSTANTIAL and DETAILED:

- "abstract": 300–400 words
- "introduction": 600–800 words
- "methodology": 800–1000 words
- "experiment": 600–800 words
- "discussion": 600–800 words
- "conclusion": 300–400 words

The total paper should be at least 3500 words. Do NOT write short, superficial sections. Expand on details, provide thorough explanations, and develop arguments fully, as a real published academic paper would.

Respond with ONLY a JSON object (no markdown fences, no commentary) with exactly these keys, each a non-empty string of well-written academic prose: "abstract", "introduction", "methodology", "experiment", "discussion", "conclusion"

**User:** Here are the {n} source papers to synthesize:
`[Source Paper 1] [Source Paper 2] ... [Source Paper n]`

Now generate a single synthesized paper as a JSON object.

# C. LLM Scoring Rubric

## Scorer Prompt Used for Gemini and Claude

**User:** We are selecting 50 high-quality queries from each group for the PAN 2026 generated plagiarism detection setting.

**Context:**

- Each query is a generated suspicious paper intended for source detection.
- A good query is not just long or fluent.
- It should preserve meaningful retrieval signals that help recover the original source papers.
- At the same time, it should not be trivially easy.
- We have already applied a first-pass length filter.
- Your job is **not** to make the final selection.
- Your job is **only** to score each query and provide a short justification.
- Final selection will be done manually by a human based on your scores.

**Your task:** For each query, assign scores for the criteria below and provide a short retrieval-oriented justification.

**1. Retrieval Anchor Richness** *(0–2)*

- 0 = very few concrete retrieval signals
- 1 = some useful searchable signals
- 2 = many clear and discriminative retrieval signals

**2. Multi-Source Visibility** *(0–2)*

- 0 = mostly looks like one dominant source
- 1 = some evidence of multiple source strands
- 2 = clear traces of at least two distinguishable source contributions

**3. Sectional Spread of Retrieval Signals** *(0–2)*

- 0 = clues are sparse or concentrated in one small part
- 1 = some spread across the document
- 2 = useful retrieval clues are distributed across multiple sections

**4. Coherence as a Suspicious Generated Paper** *(0–2)*

- 0 = fragmented or poorly integrated
- 1 = moderately coherent
- 2 = coherent and plausible while still preserving source traces

**5. Difficulty Balance** *(0–2)*

- 0 = too easy or too hard
- 1 = somewhat balanced but imperfect
- 2 = good moderate difficulty for source detection

**Difficulty balance guidance:**

- Down-score queries that are trivially easy because of overly explicit or overly concentrated clues.
- Down-score queries that are too vague, too generic, or nearly unrecoverable.
- Prefer queries that are recoverable but still realistically challenging.

**Output format:** Query ID; Anchor Richness: X/2; Multi-Source Visibility: X/2; Sectional Spread: X/2; Coherence: X/2; Difficulty Balance: X/2; Total: X/10; Short justification: 1–3 sentences.

**Important:** Do not output final keep/drop decisions. Do not rank the final top 50. Only score and justify. Keep the reasoning retrieval-oriented. Do not overvalue writing polish alone.