Overview of the Plagiarism Detection Task at PAN 2025

André Greiner-Petter[1,2,*], Maik Fröbe[3,†], Jan Philip Wahle[1,†], Terry Ruas[1,†], Bela Gipp[1], Akiko Aizawa[2] and Martin Potthast[4,5,6]

[1] *Georg-August-Universität, Göttingen, Germany*
[2] *National Institute of Informatics, Tokyo, Japan*
[3] *Friedrich-Schiller-Universität Jena, Jena, Germany*
[4] *University of Kassel, Kassel, Germany*
[5] *hessian.ai, Darmstadt, Germany*
[6] *ScaDS.AI, Leipzig, Germany*

Abstract

The generative plagiarism detection task at PAN 2025 aims at identifying automatically generated textual plagiarism in scientific articles and aligning them with their respective sources. We created a novel large-scale dataset of automatically generated plagiarism using three large language models: Llama, DeepSeek-R1, and Mistral. In this task overview paper, we outline the creation of this dataset, summarize and compare the results of all participants and four baselines, and evaluate the results on the last plagiarism detection task from PAN 2015 in order to interpret the robustness of the proposed approaches. We found that the current iteration does not invite a large variety of approaches as naive semantic similarity approaches based on embedding vectors provide promising results of up to 0.8 recall and 0.5 precision. In contrast, most of these approaches underperform significantly on the 2015 dataset, indicating a lack in generalizability.

Keywords

PAN, Plagiarism Detection, Generative AI Detection, Semantic Similarity

1. Generative Plagiarism Detection

Plagiarism detection has a long-standing tradition at PAN, with the main tasks running from 2009 [1] to 2015 [2]. Over time, the focus gradually shifted toward specialized intrinsic tasks, such as the still active authorship analysis challenges. However, the recent breakthrough of generative artificial intelligence (AI) has dramatically transformed the landscape of plagiarism detection. For the first time in history, large language models (LLMs) can serve as so-called automatic plagiarists [3]. At the same time, major scientific venues adjust their submission policies to allow (at least partially) AI-generated content [4, 5, 6]. The annual conference on AI (AAAI) recently announced to deploy an AI-assistend peer review assessment system for 2026[1]. This shift inspired us to revive a classic plagiarism detection task for 2025, this time centered on automatically generated plagiarism using LLMs.

For the 2025 edition, we adhered to the well-established foundations of the 2015 plagiarism detection task, particularly in evaluation methodology and dataset formatting [3]. Following the same formats will later allow us to evaluate new submissions on the older datasets to investigate the robustness of new approaches. Therefore, this format allows us to re-run the old baselines on this new dataset to judge the overall challenge of the new data versus the previous dataset. The participants receive an annotated synthetic dataset of pairs of documents (S, P), where S is a source document and P is the plagiarism

CLEF 2025 Working Notes, 9 – 12 September 2025, Madrid, Spain

[*] Corresponding author.

[†] These authors contributed equally.

greinerpetter@gipplab.org (A. Greiner-Petter); pan@webis.de (M. Fröbe); wahle@uni-goettingen.de (J. P. Wahle); ruas@gipplab.org (T. Ruas); 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); 0000-0002-1003-981X (M. Fröbe); 0000-0002-2116-9767 (J. P. Wahle); 0000-0002-9440-780X (T. Ruas); 0000-0001-6522-3019 (B. Gipp); 0000-0001-6544-5076 (A. Aizawa); 0000-0003-2451-0665 (M. Potthast)

[1] https://aaai.org/aaai-launches-ai-powered-peer-review-assessment-system/

document in which some paragraphs p are replaced with paraphrased versions s' of paragraphs s in S using an LLM without citation. This setup closely mirrors the 2015 PAN text alignment task[2].

The 2025 PAN task has received four submissions in total, outperforming all our baselines. Since all of these submissions (and our baselines) follow a similar approach of aligning text fragments based on their semantic similarity in terms of vector representations, we set up a fourth baseline using the Linq-Embed-Mistral model [7][3]. Linq outperforms all submissions, indicating that specialized models for the text retrieval task might suit the task for plagiarism detection particularly well. Note that this summary is an extended and in-depth version of the *Overview of PAN 2025* paper [8].

2. Dataset

To the best of our knowledge, no large-scale dataset with automatically generated cases of textual reuse exists. Some studies suggest that LLMs can disguise plagiarism via paraphrasing the original source [9, 10]. Additionally, LLMs have already been successfully used to replace human paraphrasing on scale [11]. For this task revival, we aim to create a novel dataset with realistic cases of textual reuse disguised via automated paraphrasing. To make this dataset large enough to enable possible fine-tuning approaches, we automated the full dataset creation pipeline.

For this year's iteration, we focus on the text alignment task setup, i.e., we provide participants with pairs of source and plagiarized documents (S, P) and the participants are asked to identify and align the LLM-generated, plagiarized paragraphs s' in P with their respective source paragraphs s in S.

2.1. Data Creation

We use arXiv as the source corpus for our novel dataset. Specifically, the ar5iv[4] release from 2025 of arXiv. This dataset contains all arXiv documents in a structured HTML5 format, which allows us to avoid most parsing problems of identifying paragraph splits, author identifications, citations, and more. We sample a subset of 100,000 documents with an even distribution across all arXiv categories (also known as archives), to ensure a wide variety of topics. These 100,000 documents serve as candidates for S. Afterwards, we use the SPECTER model [12] to create document embeddings and identify the semantically most similar documents (in terms of cosine similarity) to each S. This gives us 100,000 pairs of (S, P).

For each document pair (S, P), we first select a random number of paragraphs in P that should be replaced with paragraphs from S. Additionally, we add paragraphs p that cite S to the pool, as otherwise the document could contain genuine, referenced materials from S. For each selected p, we than find the most semantically similar paragraphs s based on three criteria. The alignment score is computed as a weighted aggregate: 50% semantic similarity via SPECTER sentence embeddings, 40% lexical similarity using TF-IDF vector similarity, and 10% section title similarity using again SPECTER embeddings. The inclusion of similarity in the title of the section helps discourage the alignment of paragraphs from unrelated sections of the documents and preserve a more coherent document structure within P. For each pair (S, P), we select one of three LLMs: LLaMA-3 [13] (3.3 70B Instruct), DeepSeek-R1 [14] (Distill-Qwen-32B) or Mistral [15] (7B Instruct v0.3), and replace all selected p in each aligned paragraph (s, p) with LLM-paraphrased versions s' derived from paragraphs s in S.

2.2. Categorization

To support a more detailed analysis of system performances, we establish several categories of document pairs, which later allows us to slice the dataset and investigate performances (e.g., least recall) on specific subsets of the data. First, 5% of the 100,000 pairs remain unchanged, i.e., both S and P are original arXiv documents without textual reuse. An additional 20% of pairs do not contain any plagiarism, but some

[2] http://www.uni-weimar.de/medien/webis/events/pan-15/pan15-web/plagiarism-detection.html
[3] in the following referred to as Linq
[4] https://ar5iv.labs.arxiv.org/

paragraphs in P have been paraphrased by an LLM independently of S. These examples are useful for evaluating systems that aim to detect LLM-generated content rather than plagiarism specifically. We want to discourage such approaches, as the use of LLMs in modern research does not necessarily indicate academic misconduct or even plagiarism [16]. Those document pairs are called **altered**. The remaining 75% of document pairs are constructed as plagiarism pairs as described above. In about half of these plagiarized documents, we also add 10% of altered paragraphs so that plagiarized documents may also contain LLM-generated but otherwise genuine paragraphs.

2.2.1. Severity.

We classify the severity of plagiarism in P into three levels: low, medium, and high. These refer to the proportion of paragraphs in P that are replaced with paraphrased versions from S. In 30% of the document pairs, the severity is *low*, with 20% to 40% of paragraphs replaced. In 40% of the pairs, severity is *medium*, with 40% to 60% replaced. The remaining 30% has *high* severity, where 70% to 100% of paragraphs in P are substituted.

2.2.2. Paraphrasing Prompts.

For paraphrasing, we use three prompt types: simple, default, and complex. While severity is defined on a document pair level, each pair of paragraphs within one document pair can use different types of prompts. For each pair, we follow a distribution of 60% simple prompts, 30% default prompts, and 10% complex prompts. The *simple prompt* instructs the LLM to paraphrase a given paragraph without additional constraints.

Simple Paraphrasing Prompt

```
Paraphrase the given paragraph for a professional audience.
```

We found that, especially technical texts, like the ones we often find in scientific articles from arXiv, do not produce sufficient paraphrasing. This is especially prominent to see if the texts contain mathematical formulae. To encourage the LLMs to generate more sophisticated paraphrasing, we use different *default prompt* that elevates the use of a complete reformulation rather than slight adjustments.

Default Paraphrasing Prompt

```
Reformulate the given paragraph in a sophisticated manner while preserving its
meaning. Modify sentence structure, reword phrases, and incorporate elements of
general knowledge to ensure coherence. The less token overlap, the better.
```

As the synthetic data faces the issue of replacing paragraphs from an existing, genuine document, one could potentially identify incoherent logical steps from one paragraph to the other in order to identify replaced paragraphs. In order to make this a more realistic setup, we define a third type of prompt that tries to take the previous paragraph into account as a context for the LLM to generate slightly more appropriate paraphrasing.

Complex Paraphrasing Prompt Structure with Context

```
Completely rephrase the given paragraph in your own words.  Feel free to
incorporate elements from general knowledge to ensure coherence, flow, and
better understanding.

{context_before}
```

All prompts include additional instructions to output only the paraphrased content, avoiding any explanatory text. Special tokens are used to suppress verbose output, tailored to each LLM. For

Table 1
Plagiarism alignment dataset and LLM splits. The bottom percentages refer to the total amount of samples in the final dataset.

	Llama-3		DeepSeek-R1		Mistral		Alt.	Orig.	Total
Train	18,423	79.80%	18,452	79.46%	6,265	79.65%	15,101	3,918	62,159
Validation	2,353	10.19%	2,383	10.26%	802	10.20%	1,919	518	7,975
Test	2,310	10.01%	2,386	10.28%	799	10.16%	1,919	490	7,904
Total	23,086	42.62%	23,221	42.86%	7,866	14.52%	18,939	4,926	78,038

DeepSeek-R1, a custom `<thinking>. . . </thinking>` block was used to suppress the model's internal reasoning steps, which would otherwise significantly slow down the generation. It is worth noting that Mistral performed poorly in following prompt instructions. It often produces explanatory content, hallucinated facts, or gets stuck in output loops, an issue reminiscent of neural network architectures before the attention mechanism era [17]. We presume the 7B parameter model variant is simply too small to perform paraphrasing of highly technical texts. In total, the final dataset consists of 78,038 document pairs, divided into training, validation, and test subsets. The training and validation sets are provided to participants, while the test set is kept private for the evaluation phase. The data splits and sizes are given in Table 1.

3. Evaluation

All systems are submitted and evaluated on the TIRA platform [18]. The participants are tasked with identifying all the paragraphs s' in P and aligning each with the corresponding paragraph s in S. The training and validation sets contain all alignments (s, s') for each pair of documents (S, P), together with the full text of both documents. The evaluation is carried out using the original scripts from the 2015 PAN plagiarism detection task. We used granularity as well as the micro-averaged and macro-averaged variants of `plagdet`, recall, and precision for comparability purposes with past plagiarism detection tasks [19]. All of these metrics take into account the exact character spans of the source and plagiarism and calculate the overlap regions in comparison to the truth values. While the micro-averaged variants take the length of plagiarism spans into account, the macro-averaged variants are length independent. The micro-averaged variants made especially sense for the old task setups at PAN, as earlier iterations infused plagiarism on sentence and sometimes even subsentence levels. As our dataset is constructed based on paragraph borders, the micro-variants are less indicative for our evaluations. For the sake of completeness, we evaluated all algorithms on both variants.

The granularity metric counts how often a truth case is detected on average. This metric is useful as we want to avoid a single case of plagiarism being detected multiple times. The domain of the granularity metric is $[1, |D|]$ where $|D|$ is the number of detections for a single document pair. A perfect score of 1 means that every truth case of plagiarism is detected at most once by the given algorithm. As a reminder, `plagdet` is defined via the F_1 score and with respect to the granularity:

$$\texttt{plagdet}(P, D) = \frac{F_1(P, D)}{\log_2(1 + \text{gran}(P, D))}, \tag{1}$$

where P indicates the actual case of plagiarism in the truth data and D the detected cases in (S, P).

3.1. Baselines

We implement three new baselines that use semantic similarity with large language models and the baseline from the 2012 edition of PAN [20] that uses lexical similarity. For the three large language model baselines, we split S and P into their paragraphs. For each paragraph in p we take the semantically closest paragraph in S in terms of cosine similarity based on *Linq* [7], *Qwen2 7B instruct*[5] [21], and *Llama-3.3 70B Instruct*[6] [13]. For each model, we define a cut-off threshold that classifies the closest pairs as plagiarism. Pairs below that threshold are then discarded. The threshold is determined by calculating the ideal cut-offs on the training split of the data. To compare this class of semantic plagiarism detectors to previous lexical approaches, we also include the baseline from the 2012 edition of the plagiarism detection task at PAN. The 2012 baseline tokenizes the text while normalizing white spaces and punctuation and then detects sequences of overlapping n-grams between S and P as plagiarism cases.

3.2. Team Submissions

Four teams participated in the task by submitting software.

3.2.1. Team chi-zi-zhi-xin-dui.

Su et al. [22] split the document of each pair into sentences and aligned the sentences of S and P according to the SBERT, MPNet, TF-IDF, or BERT score, whichever passed a pre-defined threshold, which was also determined based on the training data. After the alignment, they performed a merging logic to combine subsequences of detected sentences into single blocks.

3.2.2. Team foshan-university.

Tang et al. [23] also pre-processed documents by splitting them into sentence chunks and aligned all sentences from P with sentences from S based on E5 embeddings (`intfloat/e5-base-v2`). Again, the threshold was determined with the training data. They also performed a span aggregation if two spans have been categorized as plagiarism within a distance of 30 characters.

3.2.3. Team jrluo.

Jieren et al. [24] also split the documents into sentences and first aligned pairs by using TF-IDF vector similarities. For each pair, he calculated the word-based Jaccard similarity and discarded all pairs below a given threshold. All remaining sentence pairs were classified as plagiarism or genuine by a BERT classifier fine-tuned on the training data.

3.2.4. Team yukino.

Mo et al.[25] also splits the data into chunks of sentences. Each sentence gets a vector representation as the averaged vector representation of each token based on Glove (6B model with 300 dimensions). Afterwards, all sentences are aligned according to their cosine similarities. Like all other teams, Mo et al. also employed a merging strategy for positive detections based on position proximity, semantic coherence (based on cosine similarity), and a minimum length constraint.

3.3. Discussion and Results

Table 2 shows the evaluation results for all submissions and baselines on our new dataset. The final score is the average of all sub-scores and is reported as the final score in the lab overview paper [8]. While Linq seems to outperform most other approaches, the best performers vary in terms of precision and granularity. This is especially surprising as the baselines Linq, Qwen2, and Llama have been deployed for paragraph splitting rather than sentence splitting with subsequent merging techniques. We would assume these baselines have a slight advantage, especially on the granularity score. It should also be noted that Linq was deployed afterwards to investigate the performance of a special model that aimed

[5] In the following referred to as Qwen2.
[6] In the following referred to as Llama

Table 2
Overall results of baselines and submissions. r and p refer to recall and precision, respectively. Sore is the mean average of all individual scores.

	Micro			Macro				
submission	*plagdet*	r	p	*plagdet*	r	p	gran.	score
qwen2	0.39	0.57	0.32	0.26	0.54	0.18	1.06	0.38
linq	**0.61**	**0.82**	0.58	**0.53**	**0.83**	0.45	1.15	**0.64**
llama	0.28	0.36	0.23	0.21	0.40	0.14	1.01	0.27
pan12	0.08	0.08	0.59	0.06	0.05	**0.54**	2.20	0.23
foshan-university	0.42	0.57	0.33	0.31	0.56	0.21	**1.00**	0.40
jrluo	0.16	0.12	0.53	0.14	0.10	0.53	1.39	0.26
chi-zi-zhi-xin-dui	0.42	0.38	**0.67**	0.34	0.32	0.51	1.21	0.44
yukino	0.49	0.50	0.48	0.45	0.46	0.45	**1.00**	0.47

Table 3
Overall results of baselines and submissions on the old PAN12 dataset. r and p refer to recall and precision, respectively. Sore is the mean average of all individual scores.

	Micro			Macro				
submission	*plagdet*	r	p	*plagdet*	r	p	gran.	score
qwen2	0.03	**0.49**	0.04	0.01	0.38	0.02	7.49	0.16
linq	0.17	0.45	0.71	0.14	**0.42**	0.44	7.99	0.39
llama	0.02	0.34	0.05	0.01	0.19	0.02	10.94	0.10
pan12	**0.29**	0.35	**0.99**	**0.22**	0.24	0.93	2.38	**0.50**
foshan-university	0.03	0.03	0.03	0.03	0.04	0.02	**1.04**	0.03
jrluo	0.08	0.10	0.98	0.08	0.10	**0.97**	3.69	0.38
chi-zi-zhi-xin-dui	0.01	0.01	0.22	0.02	0.02	0.08	2.38	0.06

towards text retrieval tasks. Otherwise, most submissions outperform the baselines with the exception of team jrluo. Team jrluo has a relatively low recall compared to high precision scores. We suspect this is related to an agressive filtering of the initial TF-IDF similarity calculations.

Table 3 shows the same results on the old PAN12 dataset. Unfortunately, team yukino could not be evaluated as we ran into issues when applying the old datasets. All submissions (except the original PAN12 baseline) face a significant drop in performance. This is not as surprising for the baselines, as the paragraph splitting simply should not have been applied to the old dataset. This is also evident when looking at the high granularity scores. The team submissions perform significantly better in terms of granularity. An outlier is again team jrluo with very high precision values. It seems the two-stage filtering approach is particularly useful on the older dataset.

Figure 1 shows the results as a heatmap layout. We can see that team yukino performs overall similarly to Linq but loses significant on recall. It is also noteworthy that the new dataset is significantly easier in terms of granularity, as entire paragraphs have been plagiarized. It is therefore relatively rare that multiple detections detect the same plagiarized paragraph.

3.3.1. Data Subsets.

To investigate the performance on specific subsets of the data, we calculate the recall values on slices of the data. We only calculate the recall metrics of all approaches on the new data, as precision, plagdet, and granularity would require us to rerun all submissions on a pre-filtered dataset. However, the recall

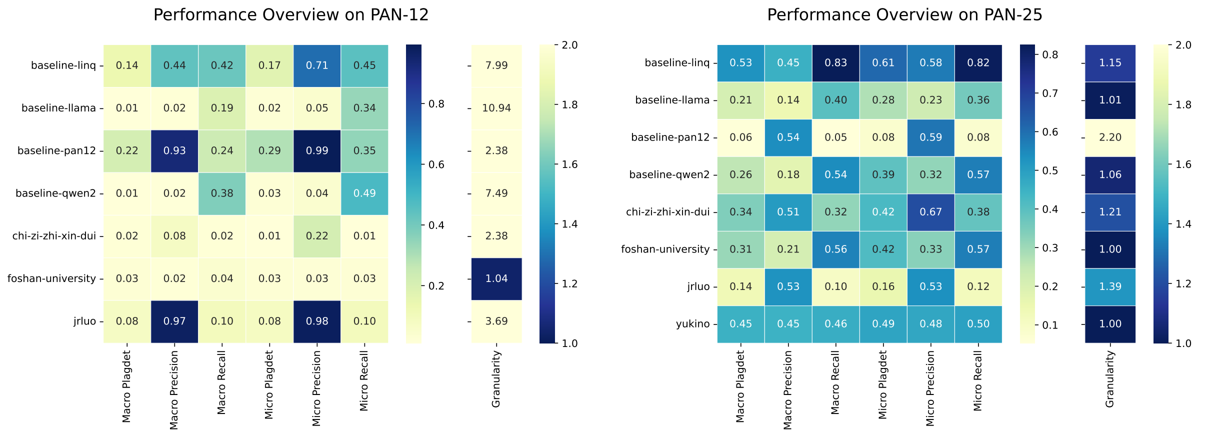

Figure 1: Comparison heatmap of performances on the PAN-12 and PAN-25 datasets.

performances are sufficient to identify trends across all submissions. Figure 2 shows the recall values of all algorithms on different models (i.e., which model is used to generate the paraphrasing) and the obfuscation level (i.e., which prompt has been used to generate the paraphrasing).

We can see that overall, Mistral is easier to detect by almost all approaches. A possible explanation is the fact that Mistral, with 7B parameters, is also the smallest of the used models and as such did not provide as high-quality paraphrasing as the other models did. Counterintuitively, the Llama baseline performs significantly worse on detecting paraphrasing generated with Llama. This might be surprising as classical LLM-detection methods typically perform best when the same model was used for the detection as for the generation of the texts [26, 27, 28]. However, the results should not be confused with classical LLM-detection approaches that are often based on logit-value comparisons. The approaches here are marely based on cosine similarities of content embeddings rather than logit values between tokens.

Another trend is also visible in the obfuscation level overview. The recall values per obfuscation level confirm a clear difference between prompt types. Almost all approaches find more plagiarism generated with simple prompts. Likewise, all approaches have the lowest success detection rate with complex prompts. While some approaches, such as by *chi-zi-zhi-xin-dui*, are more susceptible to model changes, some approaches are relatively stable regardless of prompt or model type, such as *foshan-university*.

Lastly, Figure 3 shows the recall performances on the actual plagiarism cases compared to all altered cases. Detecting an altered case is considered a false-positive. We want approaches that minimize these false classifications, as they could be interpreted as potentially harmful false accusations when handling plagiarism detections. Surprisingly, We can identify a clear difference between participant's submissions and two of our baselines even though the underlying approaches are not particularly diverse. We can see that all submissions by participants show a significantly lower recall on altered cases, sometimes up to 20% lower. The baselines of Llama and Qwen2 are particularly noteworthy as opposing approaches. as the recall on altered cases is significantly higher (in the case of Llama, even twice as high) than on actual plagiarized cases. That means, an identified case of plagiarism with these approaches is significantly more likely to be a wrong accusation than an actual case of plagiarism. We assume this discrepancy comes from the construction of the dataset, as all pairs (S, P) have been constructed to be semantically close. We can therefore assume a relatively high, general similarity across all paragraphs between S and P even without infused plagiarism. It seems Llama and Qwen2 have particular issues with differentiating these nuances in semantic similarities based on these embeddings.

In summary, the results mostly underperform our expactations. All submitted approaches and baselines follow a simple detection approach based on cosine similarities of content embeddings and achieve mostly values below 0.6 in plagdet. In comparison, on the 2014 edition of the text alignment

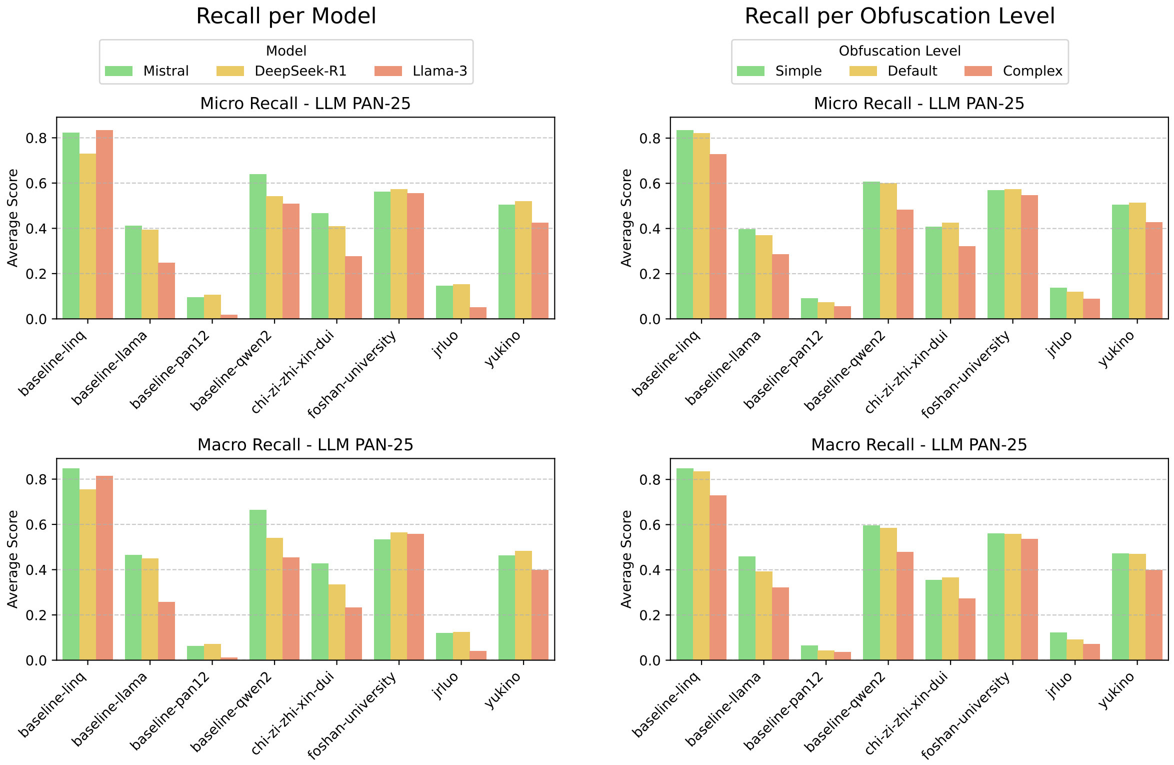

Figure 2: Comparison of recall values per used models and used prompts.

task[7] the majority of submissions achieved plagdet scores above 0.8. Unfortunately, it is unclear if this can be attributed to a more difficult task setup or the simplicity of detection approaches. The comparison to the the PAN12 dataset indicates that all approaches are not robust against changes in the data. However, this also includes the previous PAN12 baseline as it outperforms other methodologies on the PAN12 task but significantly underperforms on the new dataset.

4. Future Work

The revival of the plagiarism detection task can be summarized as successful. However, there are a few crucial improvements that can be made to make this task more realistic. The main point of criticism is the actual generation of plagiarism in the new dataset. The current pipeline starts with two genuine documents and infuses synthetic plagiarism by replacing a subset of paragraphs with a paraphrased version of another article. Typically, the textual content of scientific articles is not that interchangeable. Likewise, real-world plagiarism typically does not start with an existing publication and adds paragraphs from other works to it. In order to overcome this issue, in future iterations, we will start from multiple genuine documents (or a single document) and generate a new article by paraphrasing the content of each source rather than replacing paragraphs within an existing document excerpt. This should also promote a larger variety of detection approaches, as all submissions have been following very similar approaches. The new pipeline will also allow us to revive the important retrieval aspect of plagiarism detection tasks, in which participants start from a suspicious document without knowing if it is genuine or what the sources are. Another shortcoming is the relatively narrow domain of arXiv. As we have seen with the evaluations on the PAN12 dataset, all approaches, including the PAN12 baseline, are not very robust and perform vastly different on different datasets. This means newer iterations of this task must incorporate a larger variety of types and possibly domain of plagiarism. In the future, we will

[7] https://pan.webis.de/clef14/pan14-web/text-alignment.html

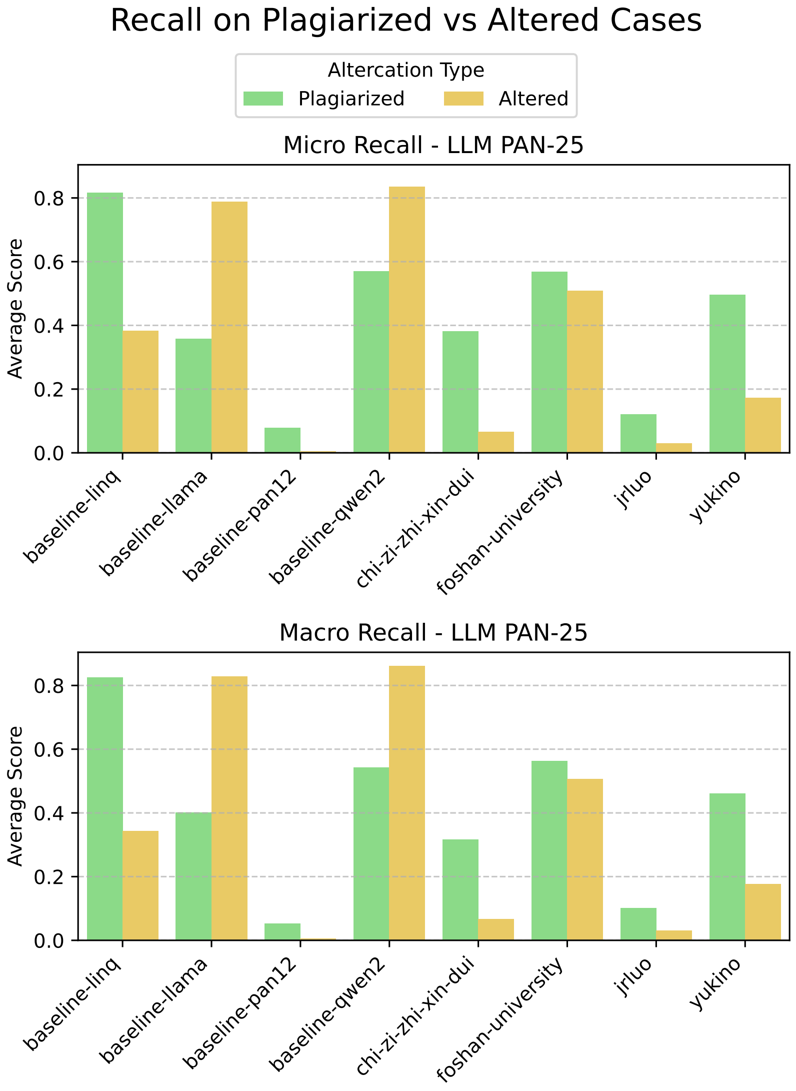

Figure 3: Recall on plagiarism cases versus altered genuine cases.

incorporate especially the medical domain to bring more variety to the dataset.

Another challenge is the rapid development of LLMs and plagiarism in itself. Recently, Zochi, a scientific LLM has generated a publication that passed the scrutiny of peer reviews at a reputable international conference[8]. This shows that LLMs are capable of generating genuine, new scientific texts without plagiarizing existing work. Nonetheless, plagiarizing existing work is now easier than ever for perpetrators. Future iterations of this task must therefore focus more on proper citations and the actual case of idealogical reuse or copying of reasoning-chains to stay relevant. Proper citation of S in P was only touched on the surface in the creation of this iteration's dataset and not separately evaluated. Lastly, this development also deemphasizes the alignment task because, moving forward, there will be fewer straightforward cases of matching sources to plagiarism. Instead, indicators such as structural, ideological, or reasoning chain similarities will have to be utilized to detect plagiarism. We will therefore reframe future iterations of this task to ensure that the dataset and plagiarism detection approaches stay relevant regardless of the development of LLMs.

[8] https://www.intology.ai/blog/zochi-acl

Acknowledgments

This work has been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 554559555, 564661959, 437179652; the Lower Saxony Ministry of Science and Culture, and the VW Foundation.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] M. Potthast, B. Stein, A. Eiselt, A. Barrón-Cedeño, P. Rosso, PAN plagiarism corpus 2009 (PAN-PC-09) (version 1), 2009. doi:10.5281/zenodo.3250083.

[2] E. Stamatatos, M. Potthast, F. M. R. Pardo, P. Rosso, B. Stein, Overview of the PAN/CLEF 2015 evaluation lab, in: 6th International Conference of the CLEF Association, CLEF 2015, Toulouse, France, September 8-11, 2015, Proceedings, volume 9283 of *Lecture Notes in Computer Science*, Springer, 2015, pp. 518–538. doi:10.1007/978-3-319-24027-5_49.

[3] A. Barrón-Cedeño, M. Potthast, P. Rosso, B. Stein, Corpus and evaluation measures for automatic plagiarism detection, in: Proceedings of the International Conference on Language Resources and Evaluation, LREC 2010, 17-23 May 2010, Valletta, Malta, European Language Resources Association, 2010. URL: http://www.lrec-conf.org/proceedings/lrec2010/summaries/35.html.

[4] J. Boyd-Graber, N. Okazaki, A. Rogers, ACL 2023 policy on ai writing assistance, 2023. URL: https://2023.aclweb.org/blog/ACL-2023-policy/.

[5] A. for the Advancement of Artificial Intelligence, AAAI publication policies & guidelines, 2025. URL: https://aaai.org/aaai-publications/aaai-publication-policies-guidelines/.

[6] E. Brunskill, K. Cho, B. Engelhardt, Clarification on large language model policy llm, 2023. URL: https://icml.cc/Conferences/2023/llm-policy.

[7] C. Choi, J. Kim, S. Lee, J. Kwon, S. Gu, Y. Kim, M. Cho, J. Sohn, Linq-embed-mistral technical report, CoRR abs/2412.03223 (2024). doi:10.48550/ARXIV.2412.03223.

[8] J. Bevendorff, D. Dementieva, M. Fröbe, B. Gipp, A. Greiner-Petter, J. Karlgren, M. Mayerl, P. Nakov, A. Panchenko, M. Potthast, A. Shelmanov, E. Stamatatos, B. Stein, Y. Wang, M. Wiegmann, E. Zangerle, Overview of the Generative Plagiarism Detection Task at PAN 2025, in: CLEF 2025 Proceedings, CEUR-WS.org, 2025.

[9] J. P. Wahle, T. Ruas, N. Meuschke, B. Gipp, Are neural language models good plagiarists? A benchmark for neural paraphrase detection, in: ACM/IEEE Joint Conference on Digital Libraries, JCDL 2021, Champaign, IL, USA, September 27-30, 2021, IEEE, 2021, pp. 226–229. doi:10.1109/JCDL52503.2021.00065.

[10] J. P. Wahle, T. Ruas, F. Kirstein, B. Gipp, How large language models are transforming machine-paraphrase plagiarism, in: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022, ACL, 2022, pp. 952–963. doi:10.18653/V1/2022.EMNLP-MAIN.62.

[11] J. Cegin, J. Simko, P. Brusilovsky, Chatgpt to replace crowdsourcing of paraphrases for intent classification: Higher diversity and comparable model robustness, in: H. Bouamor, J. Pino, K. Bali (Eds.), Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023, Association for Computational Linguistics, 2023, pp. 1889–1905. doi:10.18653/V1/2023.EMNLP-MAIN.117.

[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] AI@Meta, Llama 3 Model Card, https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md, 2024. Accessed: 2024-12-14.

[14] DeepSeek-AI, Deepseek-v3 technical report, 2024. URL: https://arxiv.org/abs/2412.19437. arXiv:2412.19437.

[15] MistralAI, Mistral 7b instruct v0.3 Model Card, https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3, 2024. Accessed: 2025-02-14.

[16] A. M. Jarrah, Y. Wardat, P. Fidalgo, Using chatgpt in academic writing is (not) a form of plagiarism: What does the literature say?, Online Journal of Communication and Media Technologies 13 (2023). doi:10.30935/ojcmt/13572.

[17] Z. Fu, W. Lam, A. M. So, B. Shi, A theoretical analysis of the repetition problem in text generation, in: Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI Press, 2021, pp. 12848–12856. doi:10.1609/AAAI.V35I14.17520.

[18] 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.

[19] M. Potthast, B. Stein, A. Barrón-Cedeño, P. Rosso, An evaluation framework for plagiarism detection, in: COLING 2010, 23rd International Conference on Computational Linguistics, Posters Volume, 23-27 August 2010, Beijing, China, Chinese Information Processing Society of China, 2010, pp. 997–1005. URL: https://aclanthology.org/C10-2115/.

[20] M. Potthast, T. Gollub, M. Hagen, J. Kiesel, M. Michel, A. Oberländer, M. Tippmann, A. Barrón-Cedeño, P. Gupta, P. Rosso, B. Stein, Overview of the 4th international competition on plagiarism detection, in: CLEF 2012 Evaluation Labs and Workshop, Online Working Notes, Rome, Italy, September 17-20, 2012, volume 1178 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2012. URL: https://ceur-ws.org/Vol-1178/CLEF2012wn-PAN-PotthastEt2012.pdf.

[21] J. Bai, S. Bai, Y. Chu, Z. Cui, K. Dang, X. Deng, Y. Fan, W. Ge, Y. Han, F. Huang, B. Hui, L. Ji, M. Li, J. Lin, R. Lin, D. Liu, G. Liu, C. Lu, K. Lu, J. Ma, R. Men, X. Ren, X. Ren, C. Tan, S. Tan, J. Tu, P. Wang, S. Wang, W. Wang, S. Wu, B. Xu, J. Xu, A. Yang, H. Yang, J. Yang, S. Yang, Y. Yao, B. Yu, H. Yuan, Z. Yuan, J. Zhang, X. Zhang, Y. Zhang, Z. Zhang, C. Zhou, J. Zhou, X. Zhou, T. Zhu, Qwen technical report, arXiv preprint arXiv:2309.16609 (2023).

[22] Z. Su, Y. Han, Y. Jia, L. Kong, Hierarchical Generative Plagiarism Detection Method, in: Working Notes of CLEF 2025 - Conference and Labs of the Evaluation Forum, CEUR-WS.org, 2025.

[23] J. Tang, Q. Hu, Z. Han, Efficient Plagiarism Detection via Sentence Embeddings and FAISS-based Retrieval, in: Working Notes of CLEF 2025 - Conference and Labs of the Evaluation Forum, CEUR-WS.org, 2025.

[24] L. Jieren, H. Mancheng, L. Biao, H. Zhongyuan, Two-Stage Generative Plagiarism Detection: From TF-IDF/Jaccard Filtering to Transformer Classification, in: Working Notes of CLEF 2025 - Conference and Labs of the Evaluation Forum, CEUR-WS.org, 2025.

[25] D. Mo, H. Zhang, X. Zhang, L. Kong, Using GloVe for Fragment Feature Matching and Overlap Ratio Optimized Generated Plagiarism Detection Method, in: Working Notes of CLEF 2025 - Conference and Labs of the Evaluation Forum, CEUR-WS.org, 2025.

[26] E. Mitchell, Y. Lee, A. Khazatsky, C. D. Manning, C. Finn, Detectgpt: Zero-shot machine-generated text detection using probability curvature, in: A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, J. Scarlett (Eds.), International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA, volume 202 of *Proceedings of Machine Learning Research*, PMLR, 2023, pp. 24950–24962. URL: https://proceedings.mlr.press/v202/mitchell23a.html.

[27] J. Ji, J. Guo, W. Qiu, Z. Huang, Y. Xu, X. Lu, X. Jiang, R. Li, S. Li, "i know myself better, but not really greatly": Using llms to detect and explain llm-generated texts, CoRR abs/2502.12743 (2025). doi:10.48550/ARXIV.2502.12743.

[28] R. Tang, Y. Chuang, X. Hu, The science of detecting llm-generated text, Commun. ACM 67 (2024) 50–59. URL: https://doi.org/10.1145/3624725. doi:10.1145/3624725.