AIIRLab Systems at SimpleText CLEF 2026: Ensemble and Multi-Agent Pipelines for Scientific Text Simplification

Ella Hawkins, Kristina Zbinden, Ellis Fitzgerald and Behrooz Mansouri

Artificial Intelligence and Information Retrieval Lab, University of Southern Maine, Portland, Maine, USA

Abstract

This paper presents the systems developed by the Artificial Intelligence and Information Retrieval Lab (AIIRLab) for the CLEF 2026 SimpleText Track, addressing three tasks: scientific text simplification (Task 1), overgeneration detection (Task 2), and research area classification (Task 3). For sentence-level simplification (Task 1.1), an ensemble reranker using Llama-3.1-70B-Instruct to combine outputs from three base systems, Mistral-8B with gist extraction and supervised fine-tuning, the same model with additional direct preference optimization, and a context-free Gemma-4 simplifier, achieved the highest SARI score of 46.24. For document-level simplification (Task 1.2), an LLaMA baseline with a coverage-first prompting strategy attained the highest SARI of 42.67, while an APIO-based Gemma-4 few-shot system yielded the best BLEU (13.06) and Levenshtein similarity (0.659). For overgeneration detection (Task 2), a majority-voting ensemble of two DeBERTa-v3 cross-encoders and a fine-tuned LLaMA-3.1-8B achieved a document-level F1 score of 0.8197 and a multiclass accuracy of 0.8269. For research area classification (Task 3), a fine-tuned SPECTER2 classifier outperformed a generative LLaMA classifier, achieving an exact match of 0.6539 for discipline area and 0.7353 for field area, with high embedding distance scores indicating semantic proximity. The experimental results show that ensemble methods, multi-agent pipelines, and fine-tuned scientific language models are effective for scientific text simplification, hallucination detection, and automated subject classification.

Keywords

Scientific Text Simplification, Creative Generation Detection, Large Language Models

1. Introduction

The CLEF SimpleText track aims to advance natural language processing techniques for making scientific text more accessible [1]. Despite the remarkable progress of large language models (LLMs), key challenges remain, including balancing simplification quality with factual accuracy, detecting and avoiding overgeneration (hallucination), and enabling reliable automatic classification of scientific publications. The SimpleText 2026 [2, 3] lab addresses these challenges through three complementary tasks. Task 1 continues the scientific text simplification task, expanding the Cochrane-auto corpus and evaluating both sentence-level (Task 1.1) and document-level (Task 1.2) simplification. Task 2 focuses on identifying and evaluating creative generation and information distortion, with two subtasks: binary overgeneration detection (Task 2.1) and fine-grained multi-class classification of error types (Task 2.2). Task 3 is a new pilot task on classifying scientific publications into research areas according to the DFG taxonomy, supporting automatic subject indexing in digital repositories.

The Artificial Intelligence and Information Retrieval Lab (AIIRLab) at the University of Southern Maine participated in all three tasks. For Task 1, two distinct systems are developed for sentence-level simplification: a four-stage pipeline built around Mistral-8B with gist extraction, operation-conditioned generation, and Direct Preference Optimization (DPO) fine-tuning (GSI and DPO runs), and a prompt-based simplification system using Gemma-4. These base systems are further combined via an LLM-based ensemble reranker (Llama-3.1-70B-Instruct), which produced the highest SARI scores. For document-level simplification (Task 1.2), four systems are implemented: a five-agent collaborative pipeline, a single-agent APIO-fewshot prompt, a three-stage sequential simplification with LLaMA, and a direct

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

ella.hawkins@maine.edu (E. Hawkins); kristina.zbinden@maine.edu (K. Zbinden); ellis.fitzgerald@maine.edu (E. Fitzgerald); behrooz.mansouri@maine.edu (B. Mansouri)

0009-0008-2630-5626 (E. Hawkins); 0009-0001-2646-7587 (K. Zbinden); 0009-0004-0601-0688 (E. Fitzgerald); 0000-0002-0400-9761 (B. Mansouri)

prompting baseline (LLaMA Baseline). For Task 2, three independent overgeneration detectors are built: two DeBERTa-v3 cross-encoders (baseline and enhanced with threshold calibration and two-stage hierarchical classification) and a fine-tuned LLaMA-3.1-8B-Instruct, and combined via majority voting. For Task 3, two classifiers are developed: a fine-tuned SPECTER2 multi-class classifier over discipline labels (deriving field labels deterministically from the taxonomy) and a fine-tuned generative LLaMA classifier.

The remainder of this paper is organized as follows. Section 2 describes Task 1, detailing the data, the proposed approaches for both subtasks, and the evaluation results. Section 3 covers Task 2, including task description, the three overgeneration detection systems and majority voting ensemble, and quantitative results. Section 4 presents Task 3, the two classifiers, and the evaluation metrics and outcomes. Section 5 concludes the paper with a summary of findings and future directions.

2. Task 1: Simplify Scientific Text

This section first introduces Task 1, Simplify Scientific Text, and then explains our proposed approaches for both Tasks 1.1 and 1.2, and then provides the evaluation results.

2.1. Task and Data

The CLEF 2026 SimpleText Track Task 1 [4] focuses on scientific text simplification, specifically targeting biomedical abstracts from Cochrane systematic reviews. Participants are asked to generate simplified versions of source abstracts that are coherent as standalone texts and consistent with the conclusions of the full review. The task is divided into two subtasks: sentence-level simplification (Task 1.1), where individual sentences from a Cochrane abstract are simplified in isolation while retaining paragraph and sentence structure, and document-level simplification (Task 1.2), where an entire abstract is simplified as a single document. Both subtasks draw on the Cochrane-auto corpus, which provides authentic parallel data in the form of author-written plain language summaries (PLS) aligned to their corresponding abstracts, enabling evaluation against genuine human simplifications.

Evaluation is conducted against the newly crawled *Cochrane-auto 2026* test set, comprising 175 abstracts and 3,679 sentences. For a subset of 32 abstracts with high-quality sentence-level alignments between the abstract and PLS, sentence-level evaluation is possible; for all 175 abstracts, both sentence-level and document-level submissions can be evaluated at the document level using the full PLS as a reference. Alongside this primary evaluation data, the test input also includes supplementary data from prior CLEF SimpleText tracks and new pilot task data, supporting broader experimental analysis. The total dataset used across training, validation, and evaluation spans over 8,000 documents and nearly 49,000 sentences, drawing from sources including Cochrane reviews, Medline abstracts, and earlier SimpleText benchmark data.

2.2. Our proposed Systems

We submitted four runs for **Task 1.1**, based on two distinct system architectures. Based on our previous participation in the SimpleText track, we designed our systems to leverage large language models that provided promising results [5, 6].

Gist-Preserving Simplification with Mistral (GSI and DPO runs). The first architecture is a four-stage pipeline built around Mistral-8B.[1]

Stage 1: Gist Extraction. Before generation, each source sentence undergoes Semantic Role Labelling (SRL) using an AllenNLP predictor [7], which extracts a structured gist frame comprising five semantic slots: agent, predicate, outcome, quantity, and uncertainty markers. When AllenNLP is unavailable, a regex fallback is used instead. These slots are serialized as special prefix tokens (e.g., [AGENT:

[1] https://huggingface.co/mistralai/Ministral-8B-Instruct-2410

. . .], [QTY: . . .]) that are prepended to the generation prompt, ensuring critical clinical content is preserved throughout simplification.

Stage 2: Operation-Conditioned Generation. Each sentence is associated with a simplification operation label (rephrase, split, merge, delete, or copy) derived from the training data. The operation is encoded as a control token (e.g., [REPHRASE]) and combined with a positional token [POS:k] that encodes the sentence's fractional position within its source document, discretized into quantiles. Together with the gist prefix, these conditioning signals form the generation prompt. Batched greedy decoding is used for efficiency, with an Out-of-Memory (OOM) fallback to single-sentence inference.

Stage 3: Multi-Component Reward Computation. On the training split, a composite reward is computed for each generated hypothesis. The reward combines five components with fixed weights: semantic preservation (cosine similarity between source and hypothesis embeddings, w=0.30); readability (proximity to a target Flesch-Kincaid Grade Level of 8.0, w=0.20); gist fidelity (slot recall of the extracted frame, w=0.25); operation correctness (structural match to the intended operation, w=0.15); and a clinical risk penalty (NLI-based contradiction score plus a hedging-loss penalty, w=0.10).

Stage 4: DPO Preference Pair Construction. Training examples are converted into preference pairs for Direct Preference Optimization [8]. The gold simplification from the Cochrane-auto training set serves as the *chosen* response. Synthetic *rejected* responses are constructed by applying targeted semantic negation (inverting clinical directionality, e.g. "reduces" $\rightarrow$ "increases") or by stripping uncertainty hedges. Pairs where the chosen response fails a minimum gist-fidelity threshold are discarded.

The **GSI** run uses the model fine-tuned via supervised fine-tuning (SFT) only, while the **DPO** run further fine-tunes the SFT checkpoint with DPO (β=0.1, LoRA with r=16). Both runs use early stopping on validation loss.

Simplification with Gemma (No-Context run). The second system employs Gemma-4[2] loaded in bfloat16 precision. No contextual information is provided to the model; document-level abstract context, paragraph sibling sentences, and positional hints are all ignored. Each sentence is simplified in isolation using only the source sentence itself.

The system prompt is explicitly designed to target SARI score components [9]. The model is instructed to: (1) *add* simplified content by replacing medical jargon with everyday English; (2) *delete* non-essential content such as subordinate clauses and parenthetical statistical detail; and (3) *keep* the main finding, uncertainty markers (e.g. *may*, *low-certainty*), and all numerical values exactly as they appear in the source. Two in-context examples illustrating the desired condensation style are included in the prompt.

Generation uses greedy decoding with a mild repetition penalty (penalty=1.15) and a 4-gram no-repeat constraint. Post-processing retains only the first sentence of the output and applies several quality guards: degenerate outputs, those containing leaked instruction text, and those exceeding 1.3$\times$ the source word count are all replaced with the original source sentence as a fallback.

LLM-Based Ensemble Reranker. This run combines the outputs of the three previous systems via an LLM-based reranker. First, the intersection of all three prediction sets is computed; only sentences for which all three systems produced a prediction are retained. For each retained sentence, a prompt is constructed containing the original complex sentence, a truncated document context (up to 600 characters), and the three candidate simplifications. A Llama-3.1-70B-Instruct[3] model, loaded in 4-bit NF4 quantization across two GPUs, is then prompted to produce a single best simplification. The reranker is instructed explicitly to maximize the SARI score [9]: retaining words that should be kept, deleting jargon, and adding clarifying words only when necessary, while preserving all medical findings. Greedy decoding is used for determinism.

We submitted four runs for **Task 1.2**, targeting document-level simplification of full Cochrane abstracts.

[2] https://huggingface.co/google/gemma-4-31B-it

[3] https://huggingface.co/meta-llama/Llama-3.1-70B-Instruct

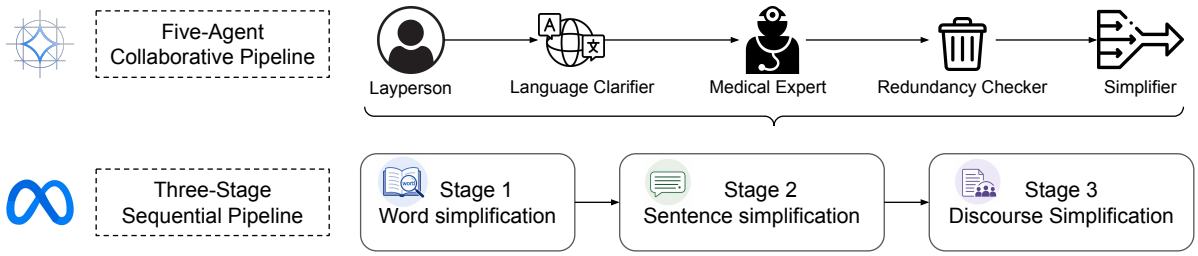

Figure 1: Architectural overview of our multi-step simplification pipelines: the Five-Agent Collaborative Pipeline (Gemma-4-31B-Instruct) and the Three-Stage Sequential Simplification system (LLaMA-3.1-8B-Instruct).

Five-Agent Collaborative Pipeline (multi-agent run). The first system employs a five-agent architecture (depicted in Figure 1) in which specialized agents collaborate sequentially on each document, all sharing a single loaded Gemma-4-31B-Instruct model to avoid redundant memory allocation. The agents are executed in order:

1. *Layperson Agent*: identifies clinical terms, acronyms, statistical jargon, and Latin expressions that a general reader would not understand, producing a line-by-line list;
2. *Language Clarifier Agent*: proposes plain-English replacements for each flagged term and flags syntactic issues (e.g. passive voice, overly long sentences);
3. *Medical Expert Agent*: validates each proposed replacement for medical accuracy, approving or correcting terms where dosages, anatomical references, GRADE certainty labels, or statistics would be distorted;
4. *Redundancy Checker Agent*: identifies verbose filler phrases, repeated information, and parenthetical statistical details that can be safely condensed or removed;
5. *Simplifier Agent*: synthesises all four agents' feedback, writes a brief internal translation plan, and produces the final simplified text.

Each intermediate agent operates within a tight token budget (256–384 new tokens), while the final Simplifier agent receives up to 768 new tokens. Prompts exceeding 2,048 tokens are left-truncated before generation. At each agent call, the KV-cache is explicitly released to manage VRAM across two GPUs. An in-context exemplar is selected from a small seed pool via token-level Jaccard similarity and injected into the Simplifier's prompt. Greedy decoding is used throughout with a mild repetition penalty (penalty=1.10).

Single-Agent APIO-FewShot Prompting (single-agent run). The second system uses the same Gemma-4-31B-Instruct model but simplifies each document in a single generation call. The prompt follows an Atomic Plain-language Instruction Optimization (APIO) structure: a discrete, ordered instruction list targeting each SARI sub-component: replacing jargon (ADD), preserving GRADE certainty labels and numerical findings exactly (KEEP), and removing verbose structure (DELETE). A single in-context exemplar is selected from the same seed pool via Jaccard similarity and prepended to the prompt. Generation uses greedy decoding with up to 1,024 new tokens and a repetition penalty of 1.10.

Three-Stage Sequential Simplification with LLaMA (multi-stage run). The third system processes each document through three sequential simplification agents (also shown in Figure 1), all sharing a single LLaMA-3.1-8B-Instruct model. The stages are:

1. *Stage 1: Word Simplification Agent.* Each sentence is processed independently. A curated jargon dictionary maps common biomedical terms to plain-English equivalents (e.g., *mortality* → *death rate*; *prophylaxis* → *preventive treatment*). Sentences containing any flagged vocabulary or long words (defined as ≥7 characters, excluding common stopwords) are rewritten by the model using explicit replacement hints; sentences with no complex vocabulary are passed through unchanged.

Table 1
AIIRLab submission results for Task 1.1 on the *Cochrane-auto 2026* test set. Results are reported against two reference types: *orig* (simple_original) and *auto* (simple_auto).

System	Ref	SARI	BLEU	FKGL	Compr.	Sent. Split	Levenshtein	Additions	Deletions	Lex. Compl.
Mistral GSI	orig	42.44	5.58	10.45	0.865	1.180	0.518	0.420	0.554	8.39
	auto	40.21	4.49	10.54	0.778	1.116	0.517	0.361	0.586	8.37
Mistral DPO	orig	42.68	5.80	9.84	1.015	1.377	0.533	0.486	0.446	8.37
	auto	40.47	4.68	9.79	0.935	1.357	0.537	0.455	0.488	8.33
Gemma4 No-Context	orig	41.97	11.04	8.66	0.624	0.979	0.599	0.153	0.531	8.52
	auto	43.17	13.89	8.69	0.592	0.977	0.595	0.138	0.548	8.48
Ensemble (LLM Reranker)	orig	**46.24**	10.80	11.36	0.774	0.979	0.448	0.327	0.566	8.40
	auto	**45.08**	10.04	11.35	0.733	0.977	0.450	0.302	0.582	8.35

2. *Stage 2: Sentence Simplification Agent.* Sentences exceeding 30 words are further rewritten using a few-shot prompt. Two exemplar complex–simple pairs are retrieved from the training set via word-overlap scoring (requiring no embedding model) and prepended as in-context demonstrations. Shorter sentences are passed through unchanged.
3. *Stage 3: Discourse Simplification Agent.* The reassembled document from Stage 2 is rewritten as a whole in a single generation call with a larger token budget (up to 1,024 new tokens). The agent is instructed to produce flowing, connected prose without markdown or headers, to spell out all abbreviations on first use, and to preserve every number and study detail. Post-processing strips any residual markdown artifacts. Low-temperature sampling ($T = 0.2$) is applied across all three stages.

Direct Prompting with Llama (baseline run). The fourth system uses Llama-3.1-8B-Instruct with a coverage-first prompting strategy designed to balance SARI and FKGL. The model is instructed to cover every comparison arm and outcome from the abstract, preserve all numerical values while explaining them in plain English, and produce output at approximately 65% of the input word count using sentences of at most 20 words. To prevent over-generation, the maximum number of new tokens is set dynamically to $\min(1.8 \times w_{\text{in}}, 800)$, where w_{in} is the input word count. Post-processing applies a proportional word-count ceiling of 72% of the input, a 25-sentence cap, and a rambling detector that truncates the output at the first sentence exceeding 55 words or exhibiting excessive word repetition. Near-greedy sampling is used (T=0.1) with a 4-gram no-repeat constraint and repetition penalty of 1.15.

2.3. Evaluation Results

Table 1 reports the performance of our four Task 1.1 submissions on the test set. The *Ensemble (LLM Reranker)* system achieves the highest SARI scores across both reference types, reaching 46.24 against *orig* and 45.08 against *auto*, indicating that combining the three base systems via an LLM reranker yields consistent gains over any individual system.

Among the base systems, the two Mistral variants perform comparably on SARI, with *Mistral DPO* (42.68 / 40.47) edging out *Mistral GSI* (42.44 / 40.21), suggesting that the DPO fine-tuning step provides a small improvement. The DPO run also achieves a lower FKGL (9.84 vs. 10.45), indicating that preference-pair training encourages more readable output. However, its higher compression ratio (1.015 vs. 0.865) and sentence-split rate (1.377 vs. 1.180) suggest a tendency to expand and restructure source sentences more aggressively than the SFT-only model.

While the *Gemma4 No-Context* system's SARI against *orig* (41.97) is slightly lower than those of both Mistral runs, it achieves the highest SARI against *auto* among the base systems (43.17), the best BLEU scores (11.04 / 13.89), and the lowest FKGL (8.66 / 8.69), reflecting its explicit targeting of readability and lexical simplification. Its low addition ratio (0.153 / 0.138) and compression below 1.0 show that the system primarily simplifies by deletion and paraphrase rather than by elaboration. The *Ensemble system* inherits the readability trade-off; its FKGL (11.36) is the highest among all runs, likely because

Table 2
AIIRLab submission results for Task 1.2 on the *Cochrane-auto 2026* test set. Results are reported against two reference types: *orig* (simple_original) and *auto* (simple_auto).

System	Ref	SARI	BLEU	FKGL	Compr.	Sent. Split	Levenshtein	Additions	Deletions	Lex. Compl.
LLaMA Multi-Stage	orig	41.74	8.00	14.31	1.109	1.066	0.569	0.447	0.377	8.55
	auto	40.30	7.60	14.17	1.088	1.100	0.575	0.428	0.387	8.53
Multi-agent Gemma	orig	42.00	4.04	13.03	1.282	1.406	0.482	0.601	0.451	8.53
	auto	39.97	3.71	13.26	1.209	1.373	0.480	0.587	0.491	8.50
LLaMA Baseline	orig	**42.67**	5.48	11.06	0.596	0.838	0.461	0.269	0.713	8.63
	auto	42.16	7.90	10.94	0.615	0.905	0.473	0.277	0.706	8.59
APIO Gemma4 Few-Shot	orig	41.41	11.83	13.83	0.889	0.999	0.659	0.264	0.441	8.49
	auto	**42.21**	13.06	13.49	0.862	1.050	0.659	0.244	0.446	8.48

the Llama-3.1-70B reranker tends to select outputs that maximize SARI component coverage even at the cost of reading ease.

Table 2 shows results for our four Task 1.2 submissions on the same test set. Performance differences across systems are smaller here than in Task 1.1, with SARI scores clustered between 41 and 43 regardless of reference type.

The *LLaMA Baseline* achieves the highest SARI against *orig* (42.67), while *APIO Gemma4 Few-Shot* leads against *auto* (42.21). The Baseline system's strong deletion behavior (deletion ratio of 0.713 / 0.706) and compression well below 1.0 (0.596 / 0.615) reflect its explicit word-count targeting strategy, which appears well-aligned with how *orig* references are constructed. Conversely, APIO Gemma4 achieves the best BLEU scores (11.83 / 13.06) and the lowest Levenshtein distance (0.659), suggesting its outputs overlap more closely with the automatic reference, consistent with its instruction-optimized prompting approach.

The *Multi-agent Gemma* system shows the most aggressive addition and sentence-splitting behavior (additions: 0.601 / 0.587; sentence-split: 1.406 / 1.373), with a higher compression ratio (1.282 / 1.209) indicating output expansion. Despite this, its SARI (42.00 / 39.97) remains competitive against *orig* but drops more sharply against *auto*, suggesting that the elaborative outputs align less well with the automatic references.

The *LLaMA Multi-Stage* system produces the highest FKGL values across both tasks (14.31 / 14.17), indicating that the sequential pipeline does not sufficiently reduce syntactic complexity at the document level despite the dedicated word- and sentence-simplification stages. This points to a limitation of the bottom-up approach: local simplifications in Stages 1 and 2 may not propagate into globally coherent plain-language prose without stronger discourse-level constraints in Stage 3.

Overall, SARI scores on Task 1.2 are lower and more tightly clustered than on Task 1.1, which is consistent with the greater difficulty of document-level simplification and the broader set of phenomena (discourse coherence, coreference, structural reorganization) that a document-level system must handle.

3. Task 2: Identify and Avoid Hallucination

This section discusses Task 2, Identify and Avoid Hallucination, explaining the task and dataset, our proposed methods for both subtasks, and evaluation results.

3.1. Task and Data

Task 2 [10] focuses on identifying and evaluating creative generation and information distortion in text simplification. Using annotations of CLEF 2025 track submissions, the task targets *overgeneration*: content in simplified outputs that cannot be attributed to tokens in the source data. Overgeneration takes many forms, ranging from a model continuing to generate text after correctly simplifying a sentence, to inline commentary on the simplification process, leftover extraction noise such as leaked prompts, and conversational asides from the underlying model.

The task is divided into two subtasks. Task 2.1 asks systems to identify overgeneration at the document level, detecting which sentences are fully grounded in the source input, either (a) without or (b) with access to the source documents, and thereby flagging those that introduce significant new content. Sentence-level labels are Boolean: `"None"` or `"OVERGENERATION"`, and a document is considered positive if any of its sentences is flagged. Task 2.2 extends this to fine-grained classification, requiring systems to assign one of six labels to each sentence, as described in Table 3. Both subtasks are evaluated using the same Boolean aggregation logic as Task 2.1.

Table 3
Overgeneration labels for Task 2.2.

Label	Description
`"None"`	No overgeneration detected
`"LEAKED_INSTRUCTIONS"`	Leaked task framing, assistant chatter, or simplification rationale
`"UNGROUNDED_INJECTION"`	Unsupported factual or interpretive additions
`"GENERATION_FAILURE"`	Catastrophic or unusable output (degenerate, truncated, or corrupted)
`"REPETITIVE_CONTENT"`	Repetition loops
`"GROUNDED_OVERGENERATION"`	Source-supported content beyond simplification scope

The primary evaluation metric is document-level F1, used as the official leaderboard ranking metric. All labels are first binarized: `"None"` and equivalent negative values map to False, and any other label maps to True, then aggregated so that a document is positive if any sentence is flagged. The evaluation script additionally reports document-level precision and recall, sentence-level F1, precision, and recall, and for multi-class submissions, category accuracy over sentences with positive ground-truth labels.

Two important caveats apply to the test set. First, some source abstracts appear in both the training and test sets by design, to support system-level generalization testing where the source is seen but the model output is from an unseen system. Second, sentence-level ground-truth labels were produced using an automatic trailing-content detection algorithm and subsequently refined with an LLM-based taxonomy classifier (Gemini 2.5 Flash); sentences initially flagged as trailing but reclassified as non-overgeneration were corrected to `"None"`. The current leaderboard is therefore preliminary, and final rankings may be revised following human vetting of a subset of the data.

3.2. Our proposed Systems

We developed three independent systems for overgeneration detection, subsequently combined via majority voting. Following our previous years' systems, we relied on both fine-tuned LLMs and cross-encoder architecture [11, 12].

Cross-Encoder Classifier (Baseline). The first system fine-tunes DeBERTa-v3-base [13] as a six-class sequence-pair classifier. Each input is formed by concatenating a retrieved source context with the target sentence. Because the source document can be long, we apply a lightweight lexical retrieval step before tokenization: for each sentence under evaluation, the top-k source sentences ranked by Jaccard overlap are selected and joined as the context passage, keeping the combined input within 384 tokens. The model is trained with Focal Loss [14] weighted by inverse class frequency to counteract the heavy label imbalance (most sentences carry the None label). Optimization uses AdamW with a linear warm-up schedule over six epochs; the best checkpoint is selected by macro-F1 on the development set. We also conducted the same experiment using DeBERTa-v3-large.

Cross-Encoder Classifier (Enhanced). The second system extends the baseline with three targeted improvements. *(i) Binary-F1 checkpoint selection:* because the primary evaluation metric is binary F1 (error vs. no error), the checkpoint criterion is changed from macro-F1 to binary F1, computed by collapsing all non-None predictions into a single positive class. *(ii) Threshold calibration:* after training, a sweep over $P(\text{None})$ thresholds is run on the development set; the threshold maximizing binary F1 is

saved and applied at inference time, replacing the default argmax decision. *(iii) Two-stage hierarchical classification:* an optional two-stage mode first trains a binary detector (error vs. no error) and then trains a separate five-class fine-grained model on error-only examples; at inference, sentences predicted as errors by Stage 1 are forwarded to Stage 2 for fine-grained typing.

Fine-tuned LLaMA-3.1-8B-Instruct. The third system adapts a large generative model to the classification task via supervised fine-tuning. LLaMA-3.1-8B-Instruct [15] is loaded in 4-bit quantization and fine-tuned with LoRA (r=16, α=32) targeting the query, key, value, output, and feed-forward projection layers using the Unsloth framework.[4] Training examples are formatted as instruction-following chat turns: a system prompt describing the six labels and their meanings is prepended, followed by the full source document and target sentence as the user turn, with the gold label as the assistant turn. To reduce the dominance of `None` examples, oversampling is applied so that `None` instances constitute at most twice the count of all error instances. The model is trained with a cosine learning rate schedule, 8-bit AdamW, and early stopping monitored on development-set loss. At inference, the model generates a single token (the label string); a multi-tier parsing strategy first attempts an exact match, then case-insensitive matching, and finally substring search, with up to three retries before defaulting to `None`.

Majority Voting Ensemble. The predictions of the three systems (DeBERTa-v3-base baseline and enhanced, with fine-tuned LLaMA) are combined by majority voting over the per-sentence label assignments. In cases of a three-way tie, a predefined priority ordering among the individual systems is used to break the tie.

3.3. Evaluation Results

Table 4 reports results for our five AIIRLab submissions to Task 2. The majority voting ensemble, which combines the fine-tuned DeBERTa and LLaMA 3.1 systems, achieves the best overall performance; it attains the highest document-level F1 of 0.8197, the highest document-level precision of 0.8149, and the highest sentence-level F1 of 0.8633, indicating that combining complementary classifiers via voting yields more balanced predictions than any individual system.

The enhanced cross-encoder (DeBERTa-v3-base with threshold calibration) reaches a document-level F1 of 0.8030, showing that binary-F1-oriented checkpoint selection and threshold calibration provide meaningful gains over the baseline cross-encoder (0.7243), which scores lowest on both document-level F1 and precision (0.6355).

The fine-tuned LLaMA 3.1-8B achieves the highest sentence-level precision of 0.8374, indicating that the generative model is comparatively conservative in flagging overgeneration, at the cost of lower document-level recall (0.7337) relative to the other systems.

The DeBERTa-v3-large baseline, despite having the lowest document-level precision (0.6601), achieves the highest document-level recall (0.8632), the highest sentence-level recall (0.9114), and the highest Task 2.2 multiclass accuracy (0.8326). This pattern suggests that the larger model is highly sensitive to overgeneration signals and captures a broader range of error categories, but at the cost of more false positives. Across all submissions, sentence-level recall consistently exceeds precision, reflecting the difficulty of avoiding false positives when the label distribution is heavily skewed toward `None`.

4. Task 3: Research Area Classification

This section provides an overview of the pilot task, Research Area Classification, and then discusses our two proposed systems, followed by the evaluation results.

[4] https://github.com/unslothai/unsloth

Table 4
AIIRLab submission results for Task 2.

	Task 2.1 (doc)			Task 2.1 (snt)			
System	**F1**	**Prec**	**Recall**	**F1**	**Prec**	**Recall**	**2.2 Acc**
Majority Voting (DeBERTa + LLaMA 3.1)	**0.8197**	**0.8149**	0.8246	**0.8633**	0.8297	0.8998	0.8269
Cross-Encoder (Enhanced)	0.8030	0.7674	0.8421	0.8486	0.8019	0.9012	0.8161
Fine-tuned LLaMA	0.7724	0.8153	0.7337	0.8483	**0.8374**	0.8593	0.7693
Cross-Encoder (Baseline - base)	0.7243	0.6355	0.8421	0.7783	0.6930	0.8874	0.8024
Cross-Encoder (Baseline - large)	0.7481	0.6601	**0.8632**	0.8083	0.7263	**0.9114**	**0.8326**

4.1. Task and Data

Task 3 [16] is a pilot task focusing on the classification of scientific publications into research areas according to a predefined taxonomy derived from the DFG (German Research Foundation) classification system. The goal is to support automatic subject classification in scientific repositories and digital libraries, improving metadata consistency across collections such as arXiv and OpenAIRE. The task may be approached either as a hierarchical classification problem or through the use of large language models for label prediction [17, 18]. Participants are asked to predict two hierarchical labels for each publication: a broad `field_area` and a fine-grained `discipline_area`, where each discipline is a subset of its corresponding field.

The dataset consists of scientific publication records in CSV format, each containing a unique identifier, title, and abstract alongside the two label columns. The training set contains 12,301 records with ground-truth labels, and the test set contains 3,106 records for which labels are withheld for evaluation.

Submissions are evaluated using three complementary measures. Exact Match (EM) is a binary metric that scores a prediction as correct only if it matches the ground truth exactly. String Distance (SD) computes normalized Levenshtein similarity between the predicted and actual labels, capturing partially correct answers where wording differs slightly from the expected result. Embedding Distance (ED) uses BERT sentence embeddings and cosine similarity to assess semantic equivalence, rewarding predictions that convey the correct meaning even when expressed with different wording.

4.2. Our proposed Systems

We developed two systems for research area classification under the DFG taxonomy, targeting both *field* and *discipline* labels.

Fine-tuned Scientific BERT Classifier. The first system fine-tunes SPECTER2 [19] as a single-head multi-class classifier over discipline labels; field labels are then derived deterministically from the taxonomy mapping, requiring no additional prediction head. Each input is formed by concatenating the paper title and abstract with a `[SEP]` separator, truncated to 256 tokens, sufficient to cover the vast majority of inputs. The `[CLS]` token representation is passed through a dropout layer and a linear classification head. To handle the severe class imbalance across 19 discipline classes, the cross-entropy loss is weighted by inverse class frequency computed on the training split. The encoder and classification head are optimized jointly with AdamW using a $10\times$ higher learning rate for the head than for the encoder, with a linear warm-up schedule over 10% of total training steps. The data is split into 90% training and 10% validation using stratified sampling; the best checkpoint is selected by validation discipline accuracy over six epochs.

Fine-tuned LLaMA Classifier. The second system fine-tunes Meta-Llama-3.1-8B-Instruct [15] as a generative classifier over field-discipline label pairs. Rather than treating this as a classification problem over a fixed set of labels, we frame it as a generative task: the model is prompted to predict the field

Table 5
AIIRLab submission results for Task 3.

	Discipline Area			Field Area		
Run	**EM**	**SD**	**ED**	**EM**	**SD**	**ED**
Fine-tuned LLaMA Classifier	0.5838	0.6746	0.7696	0.7288	0.8244	0.8698
Fine-tuned Scientific BERT Classifier	0.6539	0.7356	0.8068	0.7353	0.8270	0.8695

and discipline of each publication with a structured chat template that clearly separates and marks the title and abstract. The base model is loaded in 4-bit quantization with bfloat16 precision and fine-tuned using LoRA with rank 16, alpha 32, and dropout 0.05. The model is trained for up to five epochs with a cosine learning rate schedule (LR $= 2 \times 10^{-4}$) and early stopping with patience 20, evaluated every 50 steps. We randomly split the data into training (90%) and validation (10%), with a seed of 3407. For inference, the model generates predictions using greedy decoding.

4.3. Evaluation Results

Table 5 reports the results for our two AIIRLab submissions to Task 3. Fine-tuned Scientific BERT outperforms Fine-tuned LLaMA across all discipline-area metrics, achieving an exact match score of 0.6539 compared to 0.5838, with corresponding gains in string distance (0.7356 vs. 0.6746) and embedding distance (0.8068 vs. 0.7696). At the field-area level the gap narrows, with Fine-tuned Scientific BERT again leading on exact match (0.7353 vs. 0.7288) and string distance (0.8270 vs. 0.8244), while both runs achieve nearly identical embedding distance scores (0.8695 vs. 0.8698). Across both runs, field-area classification is consistently easier than discipline-area classification, as expected given the coarser granularity of the field label. The higher embedding distance scores relative to exact match and string distance suggest that even when predictions do not match the ground truth exactly, they tend to be semantically close to the correct label.

5. Conclusion

This paper reported the AIIRLab systems for the CLEF 2026 SimpleText Track, addressing scientific text simplification (Task 1), overgeneration detection (Task 2), and research area classification (Task 3). For Task 1, the LLM reranker ensemble combining Mistral GSI, Mistral DPO, and Gemma4 no-context outputs achieved the highest SARI scores among the submitted runs (46.24 and 45.08 against the two reference sets) in sentence-level simplification, indicating that aggregating diverse generation strategies outperforms any single model. The DPO-fine-tuned Mistral produced more readable outputs (FKGL 9.84), while Gemma4 no-context delivered the lowest FKGL (8.66) and highest BLEU, reflecting a conservative simplification style. For document-level simplification, the LLaMA Baseline attained the top SARI (42.67), whereas the APIO Gemma4 Few-Shot system achieved the best BLEU (13.06) and Levenshtein similarity (0.659), indicating that structured single-call prompting preserves closer alignment with reference summaries. In Task 2, the majority voting ensemble over two DeBERTa cross-encoders and a fine-tuned LLaMA-3.1-8B achieved the strongest overgeneration detection performance (document-level F1 0.8197, multiclass accuracy 0.8269), with the fine-tuned DeBERTa-v3-large achieving the highest Task 2.2 multiclass accuracy (0.8326). For Task 3, the best submission attained an exact match of 0.6539 for discipline classification and 0.7353 for field classification, with high embedding distance scores confirming semantic proximity even when exact matches were not achieved.

Future work will focus on three directions. First, improving factual consistency in document-level simplification through stronger medical knowledge constraints. Second, extending overgeneration detection to finer-grained causal reasoning, distinguishing grounded overgeneration from harmful hallucination. Third, exploring hierarchical classification architectures that jointly optimize field and discipline predictions for the DFG taxonomy.

Acknowledgments

We thank the SimpleText organizers for their continued efforts in organizing this shared task and fostering a collaborative community dedicated to advancing automatic text simplification.

Declaration on Generative AI

During the preparation of this work, the authors used *Claude* in order to: **Grammar and spelling check** and **Paraphrase and reword**. After using these tools/services, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] L. Ermakova, H. Azarbonyad, J. Bakker, G. K. Shahi, B. Vendeville, J. Kamps, Clef 2026 simpletext track: Simplify scientific text (and nothing more), in: European Conference on Information Retrieval, Springer, 2026, pp. 215–224.

[2] L. Ermakova, H. Azarbonyad, J. Bakker, B. Chakar, G. K. Shahi, J. Kamps, Overview of the CLEF 2026 SimpleText track: Simplify scientific texts (and nothing more), in: [3], 2026.

[3] M. Hagen, M. Potthast, B. Stein, P. Schaer, E. Zangerle, S. MacAvaney, J. M. Struß, E. Sánchez Salido, A. Barrón Cedeño, A. García Seco de Herrera (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Seventeenth International Conference of the CLEF Association (CLEF 2026), Lecture Notes in Computer Science, Springer, 2026.

[4] J. Bakker, L. Ermakova, J. Kamps, Overview of the CLEF 2026 SimpleText Task 1: Simplify Scientific Text, in: [20], 2026.

[5] N. Largey, R. Maarefdoust, S. Durgin, B. Mansouri, Simpletext best of labs in clef-2024: Application of large language models for scientific text simplification, in: J. Carrillo-de Albornoz, A. García Seco de Herrera, J. Gonzalo, L. Plaza, J. Mothe, F. Piroi, P. Rosso, D. Spina, G. Faggioli, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction, Springer Nature Switzerland, Cham, 2026, pp. 128–141.

[6] N. Largey, R. Maarefdoust, S. Durgin, B. Mansouri, Aiir lab systems for clef 2024 simpletext: Large language models for text simplification., in: CLEF (Working Notes), 2024, pp. 3261–3273.

[7] P. Shi, J. Lin, Simple bert models for relation extraction and semantic role labeling, arXiv preprint arXiv:1904.05255 (2019).

[8] R. Rafailov, A. Sharma, E. Mitchell, C. D. Manning, S. Ermon, C. Finn, Direct preference optimization: Your language model is secretly a reward model, in: A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, S. Levine (Eds.), Advances in Neural Information Processing Systems, volume 36, Curran Associates, Inc., 2023, pp. 53728–53741. URL: https://proceedings.neurips.cc/paper_files/paper/2023/file/a85b405ed65c6477a4fe8302b5e06ce7-Paper-Conference.pdf.

[9] W. Xu, C. Napoles, E. Pavlick, Q. Chen, C. Callison-Burch, Optimizing statistical machine translation for text simplification, Transactions of the Association for Computational Linguistics 4 (2016) 401–415. URL: https://aclanthology.org/Q16-1029/. doi:10.1162/tacl_a_00107.

[10] B. Chakar, J. Bakker, L. Ermakova, J. Kamps, Overview of the CLEF 2026 SimpleText Task 2: Identify and Avoid Hallucination, in: [20], 2026.

[11] B. Mansouri, S. Durgin, S. Franklin, S. Fletcher, R. Campos, Aiir and liaad labs systems for clef 2023 simpletext., in: CLEF (Working Notes), 2023, pp. 3017–3026.

[12] N. Largey, D. Wu, B. Mansouri, Aiirlab systems for clef 2025 simpletext: Cross-encoders to avoid spurious generation (2025).

[13] P. He, X. Liu, J. Gao, W. Chen, Deberta: Decoding-enhanced BERT with disentangled attention, CoRR abs/2006.03654 (2020). URL: https://arxiv.org/abs/2006.03654. arXiv:2006.03654.

[14] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollar, Focal loss for dense object detection, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017.

[15] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan, et al., The llama 3 herd of models, arXiv preprint arXiv:2407.21783 (2024).
[16] G. K. Shahi, L. Ermakova, J. Kamps, Overview of the CLEF 2026 SimpleText Task 3: Research Area Classification, in: [20], 2026.
[17] G. Shahi, O. Hummel, On the effectiveness of large language models in automating categorization of scientific texts, in: Proceedings of the 27th International Conference on Enterprise Information Systems - Volume 1: ICEIS, INSTICC, SciTePress, 2025, pp. 544–554. doi:10.5220/0013299100003929.
[18] G. K. Shahi, O. Hummel, Automating categorization of scientific texts with in-context learning and prompt-chaining in large language models, arXiv preprint arXiv:2604.23430 (2026).
[19] A. Singh, M. D'Arcy, A. Cohan, D. Downey, S. Feldman, Scirepeval: A multi-format benchmark for scientific document representations, in: Conference on Empirical Methods in Natural Language Processing, 2022. URL: https://api.semanticscholar.org/CorpusID:254018137.
[20] E. Sánchez Salido, A. Barrón Cedeño, A. García Seco de Herrera, 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.