UBONLP Report on the SimpleText Track at CLEF 2026

Benjamin Vendeville[1,2,3], Liana Ermakova[1,2] and Pierre De Loor[3]

[1]Université de Bretagne Occidentale, Brest, France
[2]HCTI, Brest, France
[3]Lab-STICC (UMR CNRS 6285), Brest, France

Abstract

This article presents the UBONLP team's participation in Task 1 "Simplify scientific text" of the SimpleText track at CLEF 2026. Our goal is to use recent advances in natural language processing to help the public better understand scientific information. We submitted runs produced by PASTIS, an agentic simplification system built around three cooperating components (a Simplifier, a Quality agent, and a Validity agent) running locally on an open-weight model. We participated in both subtasks. At document level our system reaches median-level SARI while substantially improving readability (FKGL) over the source; at sentence level it is competitive with the strongest open-weight systems. Analysing the refinement loop, we find that iteration improves readability (especially at sentence level) while leaving SARI largely unchanged.

Keywords

LLM, automatic text simplification, agentic generation, science popularization

1. Introduction

The internet has democratized access to scientific research. However, understanding science communication still proves difficult due to the complexity of scientific texts. Text simplification is one way to address this. The CLEF 2026 SimpleText track [1]studies how advances in natural language processing can be applied to this goal. We present here our participation to the lab, and in particular our *Task 1* submission [2]. *Task 1* **Simplify scientific text** has two subtasks:

- **Subtask 1.1**: Simplify sentences.
- **Subtask 1.2**: Simplify abstracts.

We participated in both subtasks. For this task we used PASTIS, an agentic simplification system combining a Simplifier component with a Quality component and a Validity component, run locally on the Gemma 4 (E4B) model [3] on two RTX 4090 GPUs without fine-tuning. We will present the task, our setup, and analyze our results.

2. Task 1: Text Simplification

2.1. Task Description

The goal of this task was to generate simplifications of scientific texts. It was divided into two subtasks: sentence-level simplification (Subtask 1.1) and document-level simplification (Subtask 1.2). This task used the Cochrane-Auto corpus [4], built from the Cochrane systematic reviews and their associated lay summaries. Cochrane-Auto consists of professionally written abstract-summary pairs aligned at sentence, paragraph, and document levels. The dataset was constructed by realigning biomedical abstracts and lay summaries at different levels of granularity-sentence, paragraph, and full document. The alignment is restricted to ensure accurate correspondences, enabling meaningful evaluation at each level. The dataset was split into training and test datasets:

CLEF 2026 Working Notes, 21 – 24 September 2026, Jena, Germany
benjamin.vendeville@univ-brest.fr (B. Vendeville); liana.ermakova@univ-brest.fr (L. Ermakova); deloor@enib.fr (P. D. Loor)
0009-0003-5298-147X (B. Vendeville); 0000-0002-7598-7474 (L. Ermakova); 0000-0002-5415-5505 (P. D. Loor)


- *train*: 4,171 sentences (Task 1.1) and 4,171 paragraphs (Task 1.2)
- *test*: 3,679 sentences (Task 1.1) and 175 abstracts (Task 1.2)

Participants were welcomed to use training data to train models, but we decided to use an untrained, prompt-based approach.

Results are evaluated using a range of standard and simplification-specific metrics provided by EASSE [5]. Flesch-Kincaid Grade Level (FKGL) [6] estimates the reading difficulty of a text based on average sentence length and syllables per word, returning a U.S. school grade level; higher values indicate more complex texts, with a theoretical lower bound of -3.40 and no upper limit.

BLEU [7] assesses n-gram overlap between generated and reference texts. Although originally developed for machine translation, it is commonly applied in simplification by treating standard and simplified English as distinct languages. Scores range from 0 (no overlap) to 1 (perfect match).

SARI [8] is specifically designed for text simplification, comparing the system output not only to references but also to the input. It evaluates the quality of additions, deletions, and words retained, with scores ranging from 0 to 100, where higher indicates better simplification.

To characterize structural transformations, we compute the compression ratio, which compares the token count of the output to that of the reference; higher values reflect more compressed outputs. Sentence splits count the number of input sentences divided into multiple ones in the output, with higher counts indicating more frequent segmentation.

We also use Levenshtein similarity to quantify the edit distance between the input and the output, where higher values denote greater surface similarity. The exact copy rate measures the proportion of output sentences that are identical to sentences in the input.

In addition, we track the proportion of additions and deletions, indicating the extent of lexical changes between input and output. Finally, lexical complexity is computed following Alva-Manchego et al. [5], by aggregating the third quartile of the log-frequency ranks of words, capturing the relative rarity of the vocabulary used.

For sentence-level simplification (Task 1.1) sentences were concatenated into abstract and evaluated as such. Furthermore, two different sets of references were used. One was based on the plain language summary (PLS) from the original Cochrane references and contained references for 175 abstracts, while the 2nd was made from Cochrane-auto and contained references for 32 abstracts.

2.2. Test Data

The provided test data for Task 1.1 was of the form:


```
{
    "pair_id": "CD012520",
    "source": "Cochrane-auto 2026",
    "language": "en",
    "para_id": 0,
    "sent_id": 0,
    "complex": "We included seven cluster-randomised trials with 42,489 patient
        participants from 129 hospitals, conducted in Australia, the UK, China, and the
        Netherlands."
}
```


While test data for Task 1.2 was of the form:


```
{
    "pair_id": "CD012520",
    "source": "Cochrane-auto 2026",
    "language": "en",
    "complex": "We included seven cluster-randomised trials with 42,489 patient
        participants from 129 hospitals, conducted in Australia, the UK, China, and the
        Netherlands. Health professional participants (numbers not specified) included
        nursing, medical and allied health professionals. Interventions in all studies
        included [...]"
```

}

3. Submission Description

We present here our system and setup, then report the official results for each subtask. A full description and evaluation of PASTIS is in preparation.

PASTIS is an agentic pipeline with three roles, described here at a block level:

Simplifier produces the candidate simplification of the input.

Quality agent assesses the candidate along six readability and simplicity dimensions and requests revision. For each dimension the agent is given a set of automatic simplicity/readability measurements, judges whether the candidate is adequate, and returns a list of concrete edits. The dimensions are:

- *Lexical simplicity*: common, everyday words; technical terms glossed on first use.
- *Syntactic simplicity*: short, direct sentences with little clause embedding or passive voice.
- *Cohesion*: explicit connectives and referential continuity between sentences.
- *Concept simplicity*: lower idea density, fewer entities and nominalizations per sentence.
- *Discourse coherence*: a stable logical flow with key concepts maintained across the text.
- *Personalization*: an active, personal voice with direct address ("you", "we").

Validity agent checks the candidate against the source using a validated error taxonomy of four families from [9]. It requests revision, classifying each error and returning a targeted fix:

- **Fluency** (A): random or garbled text, syntax errors, self-contradiction, punctuation/grammar, redundant repetition.
- **Alignment** (B): format artifacts (stray JSON or prompt tags) and prompt leakage (extra questions, sources, or answers).
- **Information** (C): factual hallucination (C1, contrary to general knowledge), faithfulness hallucination (C2, contrary to the source), topic shift (C3, text about the task or a meta-preamble).
- **Simplification** (D): overgeneralization (D1 . 1), overspecification (D1 . 2), loss of informative content (D2 . 1), out-of-scope generation (D2 . 2).

The D-codes are the simplification-specific errors and carry over-step thresholds, so ordinary simplification (reducing precision without misleading the reader) is not flagged.

The three agents operate in an iterative-refinement loop [10] until the Simplifier emits a termination signal or a fixed iteration cap is reached. We ran PASTIS on Gemma 4 (E4B) [3] locally, with constrained JSON output and batched inference. No fine-tuning was used. On the document-level run (175 abstracts), refinement mostly self-terminated rather than reaching the cap: abstracts terminated after a mean of approximately 4.7 iterations (most within six), with a small tail reaching the maximum of 20. We analyse the effect of iteration on output quality in Section 3.4.

3.1. Configuration

We used a prompt-based, agentic setup without fine-tuning. The configuration was as follows:

- **Base model:** Gemma 4 (E4B) [3], served locally on two RTX 4090 GPUs.
- **Inference:** vLLM, batched, with constrained JSON-schema decoding (xgrammar)
- **Decoding:** temperature = 0.
- **Refinement loop:** a maximum of 20 iterations at document level and 8 at sentence level; termination when the Simplifier emits the [[DONE]] signal; 2 agent call(s) per iteration.

run	SARI	BLEU	FKGL
Source (baseline)	8.88	12.40	13.02
Best sentence-level system	47.43	14.21	10.19
Best open-weight system	47.01	11.97	10.55
UBO_Task1.1_PASTIS	44.70	5.21	7.59

Table 1
Sentence-level (Subtask 1.1) results on Cochrane-Auto, scored against `simple_original` (PLS references, 175 abstracts) after concatenating each abstract's sentence predictions. Context rows: the source-as-prediction baseline, the best overall system (Claude Sonnet 4, closed-source) and the best open-weight system (FOSU-YZH-Group). Against `simple_auto` (32 abstracts) PASTIS scores 42.11 / 4.26 / 7.46.

3.2. Reproducibility

All runs use an open-weight base model with no fine-tuning. The initial simplification prompt is given in Appendix A; the full agent prompt specifications are described in full in the complete system description.

3.3. Results

3.3.1. Task 1.1

Table 1 reports sentence-level results with leaderboard context. Following the official evaluation, sentence predictions are concatenated per abstract and scored against the plain-language-summary references (`simple_original`).PASTIS reaches a SARI of 44.70, close to the median of the 127 sentence-level runs and competitive with the strongest open-weight systems; the best overall system (Claude Sonnet 4, closed-source) scores 47.43 and the best open-weight system 47.01. As at document level, PASTIS trades lexical overlap for readability: its FKGL of 7.6 is the lowest among the systems shown and far below the source (13.0), while the low BLEU (5.2) reflects substantial rewriting. Sentences self-terminated after a mean of 2.2 iterations (cap 8).

3.3.2. Task 1.2

Table 2 reports document-level results. PASTIS reaches a SARI of 44.25, around the median of document-level submissions, while producing markedly more readable output than the source: its FKGL of 9.5 is well below the source-as-prediction baseline (13.0). The low BLEU (6.8) is consistent with substantial rewriting rather than near-copying. Scores are computed with EASSE against the official references at the corpus level. Table 2 also places PASTIS among a representative range of Task 1 approaches: participants used direct LLM prompting, retrieval-augmented generation with best-of-N selection, parameter-efficient fine-tuning, multi-agent pipelines, ensembling, and fine-tuned seq2seq models. LLM-based methods cluster in the mid-to-high 40s in SARI while fine-tuned seq2seq models trail, and FKGL varies widely and largely independently of SARI.

3.4. Refinement dynamics

Beyond the final scores, our per-iteration logging lets us examine what the refinement loop actually changes. Figure 1 compares how many steps each subtask takes before the Simplifier self-terminates: sentences converge markedly faster than abstracts (mean 2.2 vs 4.7 iterations; medians 2 vs 3). To test whether the extra iterations pay off, we score every intermediate draft the same official way and plot it against the refinement budget N (Figure 2). Two patterns emerge. First, SARI is essentially flat across the loop for both subtasks: it rises by less than a point and peaks early (documents 44.4 at $N{=}2$; sentences 44.8 at $N{=}3$) before plateauing. Second, readability differs by granularity: at document level the FKGL gain is front-loaded (the first simplification already brings FKGL from 13.0 to 9.6, and further iterations leave it unchanged) whereas at sentence level FKGL keeps decreasing across iterations, from 8.7 to 7.6.

System	Approach	SARI	BLEU	FKGL
deepseek-v3	LLM prompting	48.85	12.22	8.80
RAG + best-of-N	retrieval (Qwen/Llama)	47.57	12.80	12.34
Gemma-2B-IT	QLoRA fine-tune	47.18	10.42	9.91
Claude Sonnet 4†	LLM prompting	46.91	10.20	11.13
Gemma-4	LLM prompting	46.22	9.69	7.77
GPT-5-mini + Gemini†	ensembling	45.84	11.25	10.17
Gemini/Mistral/Llama†	multi-agent	45.34	10.86	9.33
UBO_Task1.2_PASTIS	**agentic (Gemma-4)**	**44.25**	**6.84**	**9.53**
MedGemma	domain LLM	43.84	7.92	8.06
Mistral-7B	DPO + KTO	41.11	5.65	10.74
BARTsent	seq2seq (fine-tuned)	38.97	6.97	12.01
Source (baseline)	—	8.88	12.40	13.02

Table 2
Document-level (Subtask 1.2) results in the context of a representative selection of participant approaches (not a ranking). † closed-source; all others open-weight. Sorted by SARI; our run in bold.

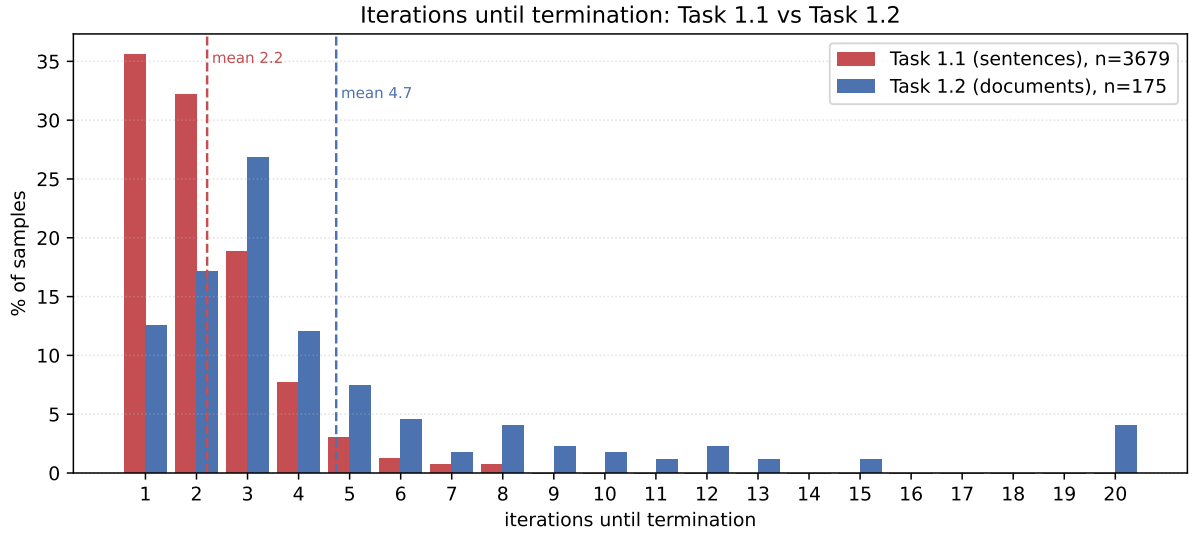

Figure 1: Iterations until termination for the two subtasks, normalised to % of samples. Dashed lines mark the means; sentence-level refinement converges faster.

We read this as evidence that single-reference SARI on biomedical abstracts is largely insensitive to the agents' iterative edits: most of the SARI is set by the first draft, and the loop's measurable benefit surfaces as readability rather than as movement in the official metric.

3.5. Inside the refinement loop

The trajectories in Section 3.4 show *that* iteration helps readability but not SARI; our per-iteration logs let us look at *what* the loop does to produce that outcome. We report three diagnostics.

What the agents flag and resolve. Figure 3 (left) tracks the mean number of issues raised per unit at each iteration. Quality issues stay roughly constant across the loop (around 3.4 per sentence and 3.5–4 per abstract), while the Validity agent's error count declines modestly at document level (1.37 to 1.05 on average). This stability reflects a churn equilibrium rather than a stalled Simplifier: at each step about 18% of below-adequate quality dimensions become adequate (Figure 3, right; from 47% for Concept Simplicity to 8% for Personalization), while a comparable number of previously-adequate

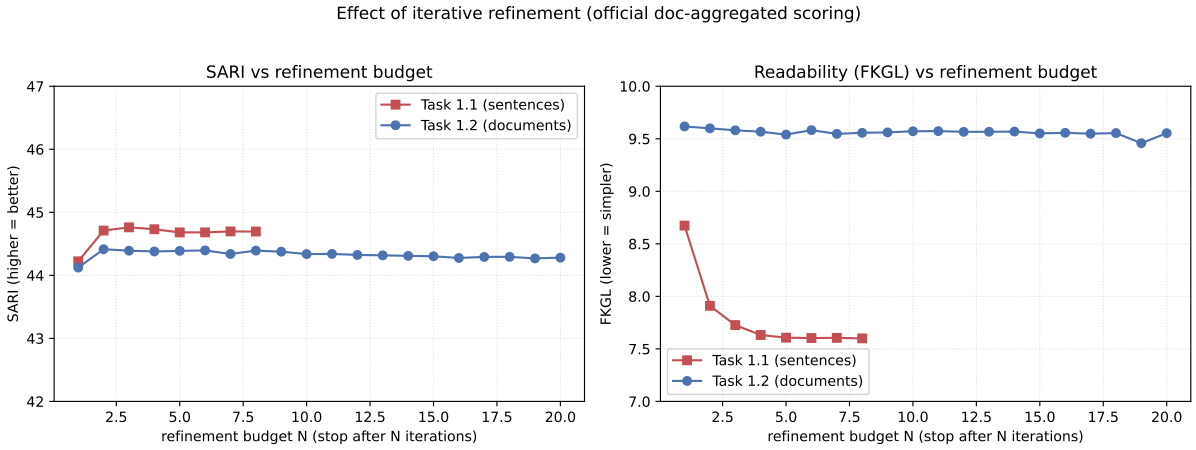

Figure 2: Official doc-aggregated SARI (left) and FKGL (right) vs the refinement budget N. SARI is flat for both subtasks; readability keeps improving with iteration at sentence level but is front-loaded at document level.

dimensions is newly flagged as the rewrite surfaces fresh issues. Each rewrite therefore resolves and creates flagged dimensions in similar numbers, keeping the total roughly constant rather than driving it to zero.

How much the text is rewritten. Figure 4 measures the change between consecutive drafts. We use chrF[11] to measure surface (token) similarity, and Sentence-BERT [12] to measure semantic similarity. Each step rewrites a substantial fraction of the text (around 0.33 chrF dissimilarity per step for sentences and 0.06 for abstracts) and this churn does not decay toward zero: the loop keeps editing rather than settling into micro-edits. The dashed lines track how much of that editing changes meaning (cosine similarity of consecutive drafts, right axis): document drafts stay almost identical in meaning across steps (cosine 0.98–0.99), so the loop reformulates rather than rewrites content, whereas sentence drafts drift more (cosine 0.88–0.91), in line with their larger surface edits.

Where the drafts go. Figure 5 normalises each document by its own source-to-reference distance and shows how far the final draft moves toward the gold reference (0 means no closer than the source, 1 means reaching it), measured two ways. In meaning (cosine) the drafts move slightly toward the reference (median $+0.12$), though a third still end up no closer than the source. In wording (chrF) they do the opposite: the median is -0.03 and 61% move away from the reference's surface form. PASTIS thus approaches the reference's meaning but expresses it in different words, which is precisely why the surface-overlap metrics (SARI, BLEU) register little gain.

Taken together, these observations suggest the loop performs continual surface reformulation that improves readability and modestly reduces validity errors, but reaches a churn equilibrium in quality (each rewrite resolves about as many flagged dimensions as it surfaces) and does not converge on the gold reference. This is consistent with the flat SARI in Section 3.4: single-reference overlap metrics reward reaching the reference, which our loop does not do. A fuller account is left to the complete system description.

4. Conclusion

In this paper we presented the UBONLP team's participation in Task 1 of the SimpleText track at CLEF 2026. We submitted runs from PASTIS, an agentic simplification system combining a Simplifier, a Quality agent, and a Validity agent, run locally on an open-weight model without fine-tuning. At document level our system reaches median-level SARI while substantially improving readability (FKGL) over the source; at sentence level it is competitive with the best open-weight systems. Our analysis of

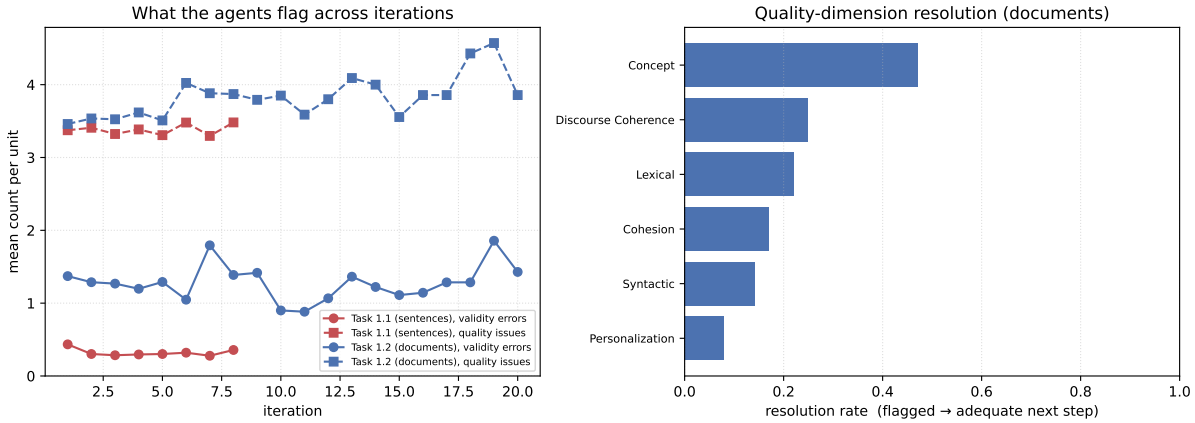

Figure 3: Left: mean issues flagged per unit at each iteration (validity errors solid, quality issues dashed). Right: for documents, the rate at which a quality dimension flagged below-adequate becomes adequate the next step. High-iteration points on the left cover the few documents still active there.

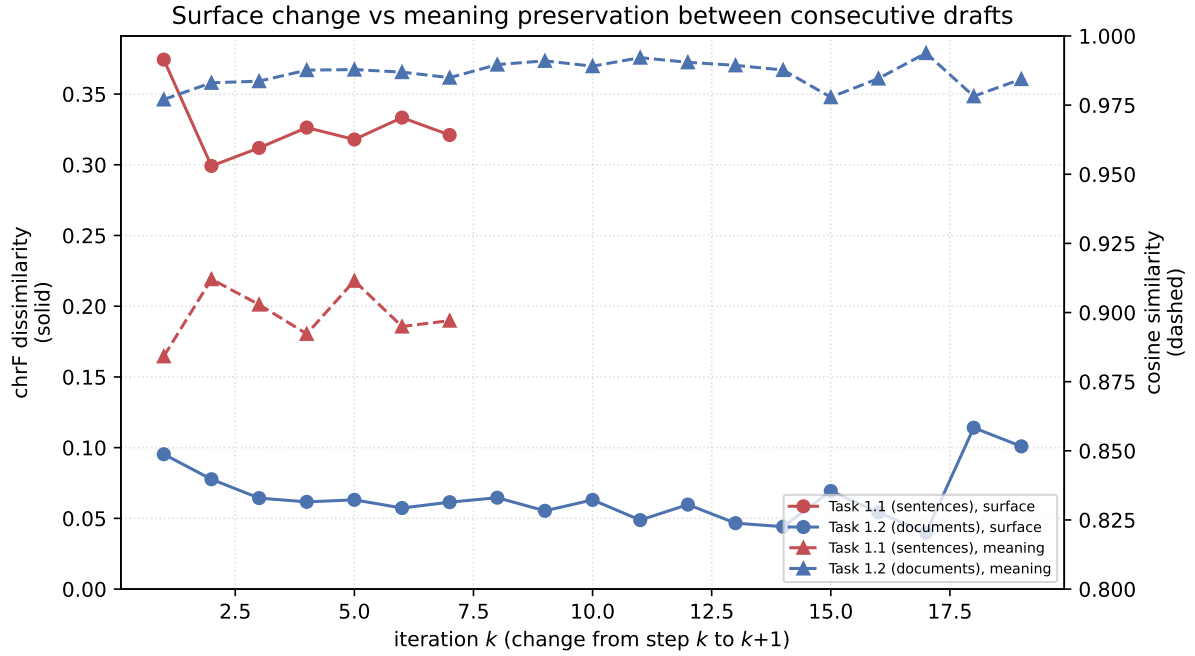

Figure 4: Change between consecutive drafts: surface edits (chrF dissimilarity, left axis, solid) and meaning preserved (cosine similarity, right axis, dashed), for both subtasks. Documents are reworded gently and preserve meaning; sentences are edited more and drift more.

the refinement loop shows that iteration yields readability gains without moving SARI, especially at sentence level. This indicates the loop's benefit is not captured by the official ranking metric.

Acknowledgments

This research was funded by the French National Research Agency (ANR) under the projects ANR-22-CE23-0019-01 and ANR-19-GURE-0001 (program *Investissements d'avenir* integrated into France 2030). We thank the CLEF 2026 SimpleText track organizers for running the task and providing the data.

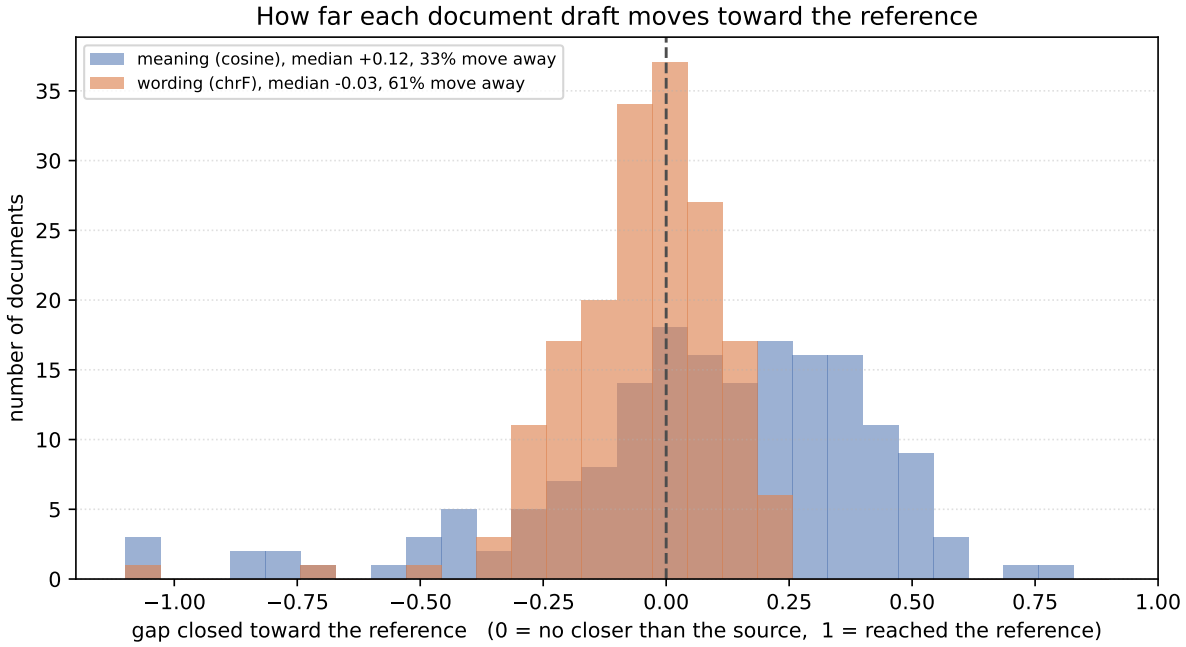

Figure 5: Per-document gap closed toward the gold reference at the final draft, normalised by each document's own source-to-reference distance (0 = no closer than the source, 1 = reached, negative = moved away), for meaning (cosine) and wording (chrF). Drafts approach the reference's meaning but move away from its wording.

Declaration on Generative AI

During the preparation of this work, the author(s) used ChatGPT and Claude in order to: Grammar and spelling check, Paraphrase and reword, and Drafting content. After using these tools, 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, B. Chakar, G. K. Shahi, J. Kamps, Overview of the CLEF 2026 SimpleText track: Simplify scientific texts (and nothing more), in: 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.

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

[3] Gemma Team, Google DeepMind, Gemma 4 model card, https://ai.google.dev/gemma/docs/core/model_card_4, 2026. Open-weights model, Apache 2.0; E4B variant.

[4] J. Bakker, J. Kamps, Cochrane-auto: An Aligned Dataset for the Simplification of Biomedical Abstracts, in: M. Shardlow, H. Saggion, F. Alva-Manchego, M. Zampieri, K. North, S. Štajner, R. Stodden (Eds.), Proceedings of the Third Workshop on Text Simplification, Accessibility and Readability (TSAR 2024), Association for Computational Linguistics, Miami, Florida, USA, 2024, pp. 41–51. doi:10.18653/v1/2024.tsar-1.5.

[5] F. Alva-Manchego, L. Martin, C. Scarton, L. Specia, EASSE: Easier Automatic Sentence Simplification Evaluation, in: Proceedings of the 2019 Conference on Empirical Methods in Natural

Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations, Association for Computational Linguistics, Hong Kong, China, 2019, pp. 49–54. doi:10.18653/v1/D19-3009.

[6] J. P. Kincaid, Jr. Fishburne, R. Robert P., C. Richard L., Brad S., Derivation of New Readability Formulas (Automated Readability Index, Fog Count and Flesch Reading Ease Formula) for Navy Enlisted Personnel:, Technical Report, Defense Technical Information Center, Fort Belvoir, VA, 1975. doi:10.21236/ADA006655.

[7] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, Bleu: A Method for Automatic Evaluation of Machine Translation, in: P. Isabelle, E. Charniak, D. Lin (Eds.), Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, 2002, pp. 311–318. doi:10.3115/1073083.1073135.

[8] 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. doi:10.1162/tacl_a_00107.

[9] B. Vendeville, L. Ermakova, P. D. Loor, Resource for Error Analysis in Text Simplification: New Taxonomy and Test Collection, 2025. doi:10.1145/3726302.3730304. arXiv:2505.16392.

[10] A. Madaan, N. Tandon, P. Gupta, S. Hallinan, L. Gao, S. Wiegreffe, U. Alon, N. Dziri, S. Prabhumoye, Y. Yang, S. Gupta, B. P. Majumder, K. Hermann, S. Welleck, A. Yazdanbakhsh, P. Clark, Self-Refine: Iterative Refinement with Self-Feedback, 2023. doi:10.48550/arXiv.2303.17651. arXiv:2303.17651.

[11] M. Popović, chrF: Character n-gram F-score for automatic MT evaluation, in: O. Bojar, R. Chatterjee, C. Federmann, B. Haddow, C. Hokamp, M. Huck, V. Logacheva, P. Pecina (Eds.), Proceedings of the Tenth Workshop on Statistical Machine Translation, Association for Computational Linguistics, Lisbon, Portugal, 2015, pp. 392–395. doi:10.18653/v1/W15-3049.

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

A. Prompts

PASTIS uses four prompts: an initial simplification prompt and one prompt for each of the three agents (Quality, Validity, and the revision Simplifier). Each has a system and a user part, and the agent outputs are constrained to a JSON schema. We reproduce the initial simplification prompt and an abridged Validity agent prompt below.

A.1. Initial simplification: system prompt

```
You are a scientific text simplification assistant. Your task is to rewrite complex
    texts so they are accessible to readers with no specialized knowledge.

TARGET READER
The simplification is written for a general adult reader with secondary-school education
     who has no specialized training in the source's field (medicine, biology,
    engineering, etc.). They read mainstream news but not textbooks. Aim for CEFR B1-B2
    vocabulary and grade 8-9 readability. The reader wants to understand the gist of the
     text and why it matters to them, not the methodology, the citations, or precise
    statistics unless a number is the main finding.

A good simplification addresses six quality dimensions:
```

```
1. Lexical Simplicity: Use common, everyday words. Avoid jargon, technical terms, and
    advanced vocabulary. When a technical term is necessary, provide a brief explanation
    .

2. Syntactic Simplicity: Use short, straightforward sentences. Avoid deeply nested
    clauses and complex grammatical structures.

3. Cohesion: Connect your sentences clearly. Use linking words (because, therefore,
    however) and maintain referential continuity so the reader can follow the logical
    flow.

4. Concept Simplicity: Manage the density of ideas. Don't introduce too many concepts at
     once. Break down complex ideas into smaller steps.

5. Discourse Coherence: Organize ideas logically. Maintain key concepts throughout the
    text. Ensure each sentence follows naturally from the previous one.

6. Personalization: Engage the reader directly and use an active, personal voice. (a)
    Pronouns and direct address: use "you" and "we" where appropriate. Frame the content
     in terms the reader can relate to personally. (b) Voice choice: prefer the active
    personal voice ("Researchers followed 146 patients", "Aspirin lowers the risk of
    heart attack") over the impersonal passive ("It was found that...", "A reduction in
    risk was observed"). Passive third-person constructions distance the reader from the
     content.

Avoid these common errors:
- Do not add information not present in the original (hallucination)
- Do not contradict the original text
- Do not drop important information during simplification
- Do not overgeneralize specific claims
- Do not produce redundant or repeated content

Output plain text only. Do not use markdown syntax.
```

A.2. Initial simplification: user prompt

```
Simplify the following text for a general audience. Preserve all important information.

Text: "{source}"
```

A.3. Validity agent: system prompt (abridged)

```
You are a text simplification error detector. You identify errors in a simplified text
    by comparing it to the original, using a validated error taxonomy, and suggest
    specific fixes. You check four categories of errors:

A. FLUENCY - is the output fluent, correct language?
   A1 random generation; A2 syntax error; A3 self-contradiction;
   A4 minor punctuation/grammar; A5 redundancy (repeated sentences).

B. ALIGNMENT - did the model follow the prompt and format?
   B1 format misalignment (stray JSON braces/quotes or prompt tags in the output);
   B2 prompt misalignment (leaked questions, extra sources, or extra answers).

C. INFORMATION - does the output preserve the source's facts?
   C1 factuality hallucination (facts contrary to general knowledge);
   C2 faithfulness hallucination (facts contrary to the source);
```

C3 topic shift (text about the task, a leaked in-prompt example, or a meta-preamble such as "Here is the simplified version:").

D. SIMPLIFICATION - is precision reduced without misleading the reader?
Simplification by design reduces precision (generalizing terms, dropping methodology, replacing exact statistics with magnitude-classes). Flag a D-error ONLY when the change crosses a threshold: the reader ends up MISLED about what was studied / found / applies to whom, or LOSES information needed to act on the finding. When in doubt, do not flag.
D1.1 overgeneralization (a scope qualifier dropped, a hedge strengthened, a conditional made absolute, or a subject removed so the claim reads as universal);
D1.2 overspecification (a broad category replaced by a specific example the source did not single out);
D2.1 loss of informative content (the research question, main conclusion, or the population/scope a finding applies to is dropped, or an independent finding lost);
D2.2 out-of-scope generation (opinions, added information, translations, leaked pipeline scaffolding such as "--- Iteration N ---", or meta-commentary).

For each error, report the taxonomy code, the exact problematic text (or null for missing content), an explanation referencing the original, and an exact suggested fix. Report each distinct error separately. If none are found, return errors_found: false.