A Benchmark for Overgeneration Detection in Biomedical Text Simplification

Berkay Chakar[†], **Liana Ermakova**[‡], **Jaap Kamps**[†]
[†]ILLC, University of Amsterdam, The Netherlands, berkay.chakar2@student.uva.nl, kamps@uva.nl
[‡]HCTI, University of Brest, France, liana.ermakova@univ-brest.fr

Abstract

Large Language Models deployed for biomedical text simplification frequently produce *overgeneration*: extraneous content appended beyond the faithful simplification, including leaked model instructions, ungrounded medical claims, and repetitive or redundant text. Despite its prevalence, this failure mode remains largely unaddressed. We present a benchmark for document-level overgeneration detection, releasing two resources: **SimpleOG-manual**, 500 abstract-level examples with human-validated positive labels, and **SimpleOG-auto**, over 46,000 automatically labeled abstract-level examples derived from submissions to the CLEF 2025 SimpleText Track. Our method exploits the positional regularity of overgeneration in simplification output through sequence alignment, identifying trailing content that lacks a corresponding segment in the source. Human validation of 117 automatically flagged positives confirms ∼95% precision, with leaked model instructions accounting for 75.7% of confirmed cases. Analysis across teams and models reveals that overgeneration is primarily driven by system-level choices, such as prompting and post-processing, rather than by model architecture. We evaluate three detection paradigms and find that sentence similarity (F1 = 0.732, ROC-AUC = 0.921) surprisingly outperforms both NLI-based and LLM-based approaches, suggesting that overgenerated content occupies distinct semantic regions from source material.

Lay Summary: *We investigate a common failure of AI models used for simplifying or summarizing scientific text: overgeneration, where models add extraneous content such as leaked instructions, ungrounded statements, or repetitive text that is not supported by the source. We introduce a benchmark for document-level overgeneration detection and release two resources: SimpleOG-manual, 500 abstract-level examples with human-validated positive labels, and SimpleOG-auto, over 46,000 automatically labeled abstract-level examples derived from CLEF SimpleText 2025 Track submissions.*

Keywords: Natural Language Generation, Text Simplification, Large Language Models, Overgeneration

1. Introduction

Large Language Models (LLMs) have become the dominant approach to text simplification, offering the ability to transform complex technical content into accessible language while aiming to preserve semantic fidelity (Li et al., 2024). This capability is particularly critical in the biomedical domain, where dense clinical literature remains difficult to understand for patients and the general public. The SimpleText shared task at CLEF (Ermakova et al., 2025) has driven this research forward, with the Cochrane Library (Bakker and Kamps, 2024), a collection of high-quality, evidence-based systematic reviews, as its primary testbed (Bakker and Kamps, 2024, 2025). The 2025 iteration challenged participants to simplify these abstracts using systems ranging from fine-tuned encoder-decoder models (BART, T5) to state-of-the-art LLMs (GPT-4, LLaMA-3, Gemini, Mistral).

However, deploying LLMs for text simplification introduces a critical but understudied failure mode: *hallucination*. While hallucination has been extensively investigated in abstractive summarization (Maynez et al., 2020; Kryscinski et al., 2020) and question answering (Li et al., 2023), its manifestation in text simplification remains largely unexplored. Unlike summarization, where the model must compress and select information, simplification requires preserving the full semantic content while transforming linguistic complexity. This fundamental difference creates a distinct failure profile: rather than fabricating facts within the output, simplification models predominantly exhibit *overgeneration*, appending extraneous content such as leaked instructions, unsolicited advice, or ungrounded elaborations beyond the faithful simplification. While overgeneration falls under the broader umbrella of hallucination, the two differ operationally: hallucination detection typically targets factual contradictions or unsupported claims *within* the generated text, whereas overgeneration detection targets extraneous content *appended beyond* the faithful output. We focus on the latter, which we detect via positional analysis of trailing content (Section 3.2). Figure 1 illustrates this at the abstract level: a model faithfully simplifies all three source sentences but leaks its internal strategy between them.

Through automatic analysis of over 707,000 sentence-level simplification pairs from SimpleText 2025 submissions, we sampled 500 abstract-level examples into a curated test set (SimpleOG-manual), of which 117 were automatically flagged as positive by our trailing-content method. Man-

Proceedings of the 2nd Workshop on Evaluating Text Difficulty in a Multilingual Context (DeTermIt! 2026), pages 12–21
11 May 2026.

Abstract-level overgeneration example

Source: *"~~The~~ functionality ~~of the Internet and the World Wide Web is determined in large part by the standards that allow for interoperable implementations, as a result, the privacy of our online interactions depends on the work done within standard-setting~~ organizations. ~~But how~~ do the ~~organizational structure and processes of these multistakeholder~~ groups ~~affect the engineering of values such as~~ privacy~~?~~ This paper ~~reviews the history of considerations for~~ security ~~and privacy in~~ Internet ~~and Web standard-setting, the impact of Snowden surveillance revelations and reactions to them, and some trends in how we review for privacy in Internet and Web~~ standards."*

Prediction: *'The internet's functionality and privacy depend on standards set by certain organizations. (I chose 'rephrase' as the internal simplification strategy) How do the groups that set Internet standards impact privacy. This paper looks at how privacy and security are considered when creating Internet standards."*

~~Strikethrough~~ = deletion Green = insertion Red = trailing content

Figure 1: Abstract-level overgeneration: a 3-sentence source is simplified correctly, but the model leaks its internal strategy between sentences 1 and 2 (red). Annotations show token-level alignment.

ual review of these 117 positives confirmed 111 as true overgeneration (∼95% precision), revealing four primary categories: leaked model instructions (75.7%), ungrounded information injection, repetitive content, and other failures with leaked instructions as the dominant pattern (discussed in Section 4). We term these extraneous additions the *"chatter"* problem: content that, while often linguistically fluent, violates the fundamental faithfulness requirement of text simplification. We also observed instances of *factual distortion* during qualitative analysis, where models alter numbers or invert relationships within the simplified text itself; this falls outside the scope of our trailing-content method and remains an important direction for future annotation.

In this paper, we present two contributions: (1) A benchmark for detecting overgeneration in biomedical text simplification at the abstract level, released as two resources: SimpleOG-manual (500 examples with human-validated positive labels) and SimpleOG-auto (over 46,000 automatically labeled abstract-level examples from 666 source abstracts across all system runs). We exploit the sentence-level structure of shared task submissions to construct reliable automatic labels via trailing-content detection at the sentence level, then aggregate to the document level: an abstract is labeled positive if any of its constituent sentences contains overgeneration (Section 3). (2) A comparative evaluation of three detection paradigms (sentence similarity, natural language inference, and LLM-based classification), finding that, surprisingly, sentence similarity outperforms both more complex approaches.[1]

2. Related Work

Hallucination in NLG. Factual inconsistency in neural text generation has been primarily studied in abstractive summarization. Maynez et al. (2020) provided a comprehensive analysis of faithfulness errors in summaries, distinguishing intrinsic hallucinations (contradicting the source) from extrinsic ones (introducing unsupported content). Automated detection methods include FactCC, a BERT-based entailment model trained on synthetic data (Kryscinski et al., 2020), and SummaC, which reframes consistency checking as an NLI task applied at the sentence level (Laban et al., 2022). The evaluation of FactCC (Kryscinski et al., 2020), QAGS (Wang et al., 2020), FEQA (Durmus et al., 2020), and FactAcc (Goodrich et al., 2019) measures on the TRUE (Honovich et al., 2022) benchmark, which aggregates 11 such datasets covering summarization, dialogue, paraphrasing, and fact verification, showed low F1 and AUPRC (Vendeville et al., 2025). For LLM outputs more broadly, Li et al. (2023) introduced HaluEval, a benchmark covering hallucinations in QA, dialogue, and summarization. However, all these resources target summarization or general-purpose generation. Text simplification, where rephrasing and synonym substitution are expected rather than erroneous, poses distinct detection challenges that these tools do not address. In particular, overgeneration in simplification often follows a recognisable positional pattern, with extraneous content appearing as trailing material after the faithful output, which enables detection through alignment-based methods that complement the entailment and classification approaches used for general hallucination.

Text Simplification Evaluation. Standard evaluation of text simplification relies on metrics such as SARI (Xu et al., 2016), which measures the quality of lexical edits (additions, deletions, and kept words) against reference simplifications. While SARI captures simplification quality, it does not assess faithfulness to the source. Davari et al. (2024) showed that automatic measures such as BERTScore (Zhang et al., 2020), BETS (Zhao et al., 2023), BLEU (Papineni et al., 2002), SARI (Xu et al.,

[1] Code and data are available at https://github.com/chakarberkay/og-benchmark.

2016), FKGL (Kincaid et al., 1975), LENS (Maddela et al., 2023), and others have low correlation with humanly annotated meaning preservation on 9 datasets. A detailed taxonomy of errors in text simplification were proposed in the SALTED benchmark (Vendeville et al., 2025). The evaluation of FactCC (Kryscinski et al., 2020), QAGS (Wang et al., 2020), FEQA (Durmus et al., 2020), and FactAcc (Goodrich et al., 2019) measures on this dataset showed low F1 and AUPRC (Vendeville et al., 2025). The most directly related work is Devaraj et al. (2022), who introduced a taxonomy of factuality errors in text simplification (insertion, deletion, and substitution) and found that such errors are common yet uncaptured by existing metrics. Our work extends this direction in two ways: we focus specifically on overgeneration (trailing content) rather than within-text factual errors, and we construct an automatically labeled benchmark from a large pool of shared task submissions rather than relying on manual annotation alone.

SimpleText Shared Task. The SimpleText track at CLEF (Ermakova et al., 2025) promotes research on making scientific texts accessible. Task 1 addresses sentence-level and document-level simplification of biomedical abstracts from the Cochrane Library, using the Cochrane-auto corpus (Bakker and Kamps, 2024). Our benchmark is built from the 2025 edition of Task 1 submissions, which introduced a Task 2.3: detecting overgeneration in the simplified outputs. This adds a faithfulness evaluation dimension that complements the existing quality-oriented metrics used in the shared task.

3. Dataset Construction

3.1. Source Data

Our dataset is built from submissions to the SimpleText 2025 shared task (Ermakova et al., 2025) (Task 1.1), which requires sentence-level simplification of biomedical abstracts from the Cochrane Library. The source data consists of 666 abstracts comprising 9,160 sentences drawn from systematic reviews. Participating teams submitted simplified versions using a range of models: LLaMA-3, GPT-4, GPT-3.5, BART, T5, Gemini, and Mistral. In total, we processed 707,898 sentence-level pairs, each consisting of a source sentence and the corresponding simplification produced by one system, from 70 system runs across 15 teams. While reference simplifications are available for the source data (enabling evaluation with standard metrics such as SARI (Xu et al., 2016)), we focus on reference-free overgeneration detection, as our goal is to identify extraneous content rather than assess overall simplification quality. Table 1 summarises the source data.

Statistic	Value
Source abstracts	666
Source sentences	9,160
System runs	70
Sentence-level pairs	707,898
SimpleOG-auto (abstract-level)	∼46,000
SimpleOG-manual (abstract-level)	500

Table 1: Source data and dataset overview.

3.2. Trailing-Content Detection

Our key observation is that overgeneration in text simplification frequently manifests as *trailing content*, text appended after the legitimate simplification that has no corresponding segment in the source. We exploit this positional regularity through a sequence alignment algorithm, noting that our method does not capture mid-text overgeneration (e.g., inserted clauses), which we leave to future work.

Algorithm. Given a source sentence and its simplified prediction, we (1) tokenize both using NLTK[2] `word_tokenize`, (2) align them with `difflib.SequenceMatcher`[3] to find the longest contiguous matching blocks, and (3) identify any unmatched prediction tokens that appear after the final aligned block as trailing content. Tokens that do not have matches are classified as:

- **Deletions** (source tokens absent from prediction): expected in simplification, where complex details are removed.
- **Mid-text insertions** (prediction tokens with further matches ahead): expected, as rephrasing introduces new wording.
- **Trailing content** (prediction tokens after the final alignment match): overgeneration candidate.

Not all trailing content constitutes overgeneration: short trailing spans often arise from alignment noise rather than genuine extraneous content. Manual inspection of 30 randomly sampled trailing segments revealed that spans below 25 characters consisted almost entirely of punctuation differences, formatting artifacts, or a few word additions from rephrasing that do not constitute meaningful

[2] https://www.nltk.org/

[3] https://github.com/python/cpython/blob/3.14/Lib/difflib.py

Example 1: Leaked model instructions

Source: *"~~Thus, implementing~~ crowdsourcing ~~applications~~ while ~~preserving the privacy of~~ workers' ~~location~~ is a key issue ~~that needs to be tackled.~~"*

Prediction: *"A key issue is keeping workers' locations private while using crowdsourcing apps.*

(I chose 'rephrase' as the internal simplification strategy)

Trailing content: 59 chars $\geq$ 25 $\rightarrow$ OVERGENERATION

Example 2: Ungrounded information injection

Source: *"SGLT2 inhibitors ~~with or without other background~~ treatments were compared to placebo, ~~standard care,~~ sulfonylurea, ~~DPP-4 inhibitors,~~ or insulin."*

Prediction: *"Medicines called SGLT2 inhibitors were compared to a fake treatment (placebo), usual treatment, sulfonylurea [...], or insulin (a hormone that controls blood sugar).*

These comparisons were made whether or not the patients were also taking other medicines.

Trailing content: 89 chars $\geq$ 25 $\rightarrow$ OVERGENERATION

~~Strikethrough~~ = deletion Green = insertion Red = trailing content

Figure 2: Overgeneration examples from the dataset. Example 1 shows leaked task instructions; Example 2 shows a plausible but unsupported medical statement. Annotations illustrate the three token classifications from our alignment algorithm.

overgeneration. At or above 25 characters, trailing spans consistently contained recognisable overgeneration closer to full-sentence material, such as leaked instruction fragments or injected claims. We therefore apply a character-length threshold of 25 to separate noise from substantive trailing content. Spans below the threshold are retained in the data but not labeled as overgeneration; spans at or above the threshold are labeled as overgeneration candidates. We note that this threshold was calibrated for English biomedical text and may require adjustment for other languages or domains. Figure 2 illustrates this process on two real examples from our dataset.

3.3. Document-Level Aggregation

Although the shared task submissions are sentence-level, our benchmark targets *document-level* overgeneration detection: given a full source abstract and its simplified version, the task is to determine whether the simplification contains overgeneration. We exploit the sentence-level structure solely for automatic label construction, where detecting trailing content is most reliable. We then stitch sentence-level detections into document-level labels: an abstract is labeled POSITIVE if *any* of its constituent sentences contains overgeneration. This OR-rule is deliberately strict: even a single overgenerated sentence renders the entire abstract positive. While this loses severity gradation (a minor trailing phrase is weighted equally to a full injected paragraph), we argue that any unsupported content in a biomedical simplification is potentially misleading and warrants flagging. We note that sentence-level false positives propagate to the document level under this rule, but the high precision of our trailing-content method ($\sim$95%, Section 4) limits this effect in practice. The benchmark also provides sentence-level annotations, enabling future work on graded or severity-aware detection.

Model	Count	Pos.	Rate
LLaMA-3-8b	179	71	39.7%
LLaMA-3-70b	159	29	18.2%
BART	67	1	1.5%
Mistral	32	3	9.4%
Gemini	27	3	11.1%
GPT-4	21	0	0.0%
GPT-3.5	8	0	0.0%
T5	7	4	57.1%
Total	**500**	**111**	**22.2%**

Table 2: Dataset distribution by model family. Overgeneration rates vary from 0% (GPT-3.5, GPT-4) to 57.1% (T5).

3.4. Sampling and Statistics

The full processing yields automatic overgeneration labels for 666 source abstracts across 70 system runs. Each (abstract, system-run) pair constitutes one example, producing over 46,000 abstract-level examples. We release this data as two resources: **SimpleOG-auto**, the complete pool of automatically labeled abstract-level examples, and **SimpleOG-manual**, a stratified sample of 500 examples in which all positive labels have been manually verified (Section 4). Table 2 reports the distribution by model family.

4. Human Validation

Of the 500 examples in SimpleOG-manual, our trailing-content method labeled 117 as positive and 383 as negative. We manually reviewed all 117 positives to assess labeling precision. Of these, 111 were confirmed as true overgeneration ($\sim$95% precision), while 6 were false positives caused by paraphrases that happened to exceed the trailing-content threshold. Negative labels were not independently verified, which we acknowledge as a

Category	Count	%
Leaked model instructions	84	75.7%
Ungrounded info. injection	18	16.2%
Repetitive content	5	4.5%
Other failures	4	3.6%
Total confirmed	**111**	100%

Table 3: Distribution of overgeneration categories among the 111 confirmed positive examples.

limitation: other types of information distortion that do not manifest as trailing content (e.g., factual distortion) may be present among the 383 negatively labeled examples.

Among the 111 confirmed cases, we identified four overgeneration categories:

1. **Leaked Model Instructions:** LLMs leak their internal task framing into the output, producing fragments such as *"Here is the simplified version:"* or *"Rephrase:"*.

2. **Ungrounded Information Injection:** Models inject content from parametric knowledge not present in the source, including unsupported elaborations and misinterpreted abbreviations.

3. **Repetitive Content:** Degenerate outputs containing repeated words, phrases, or entire sentences.

4. **Other Failures:** Unfinished or contextually unrelated sentences and model generation failures.

The relative frequency is shown in Table 3, and the majority of cases are leaked model instructions.

Our main interest is in document-level overgeneration detection. We exploit the fact that overgeneration identification via source attribution is viable for sentence-aligned data, enabling large-scale automatic labeling with minimal human supervision. This yields both SimpleOG-auto, a large pool of weakly labeled training data, and SimpleOG-manual, a smaller test set with human-validated positive labels. Figure 3 shows a full simplified Cochrane abstract in which a single leaked model instruction is embedded among 25 sentences of otherwise faithful medical text, illustrating the difficulty of document-level detection.

5. Baseline Experiments

To establish reference performance on our benchmark, we evaluate three detection paradigms: sentence similarity, natural language inference (NLI), and LLM-based classification. Recall that the data is presented at the document or abstract level, and that predictions contain many sentences (see Figure 3). All three methods share the same sentence-level strategy: the prediction is split into sentences, and the source is chunked if needed. Each prediction sentence is scored against all source chunks, and the most supportive chunk determines the sentence score. The document-level score is the maximum hallucination score across all prediction sentences; if *any* sentence is unsupported, the document is flagged. For methods that produce continuous scores, we tune the decision threshold to maximize F1 on the test set. We emphasize that this yields optimistic estimates and report it as an explicit limitation.

5.1. Methods

Sentence Similarity. We encode sentences using `all-mpnet-base-v2` (Reimers and Gurevych, 2019). For each prediction sentence, we compute its maximum cosine similarity to any source sentence. The hallucination score is $1 - \text{max\_sim}$, and the document score is the maximum across prediction sentences. The intuition is that overgeneration introduces semantically distant content. At the optimal F1 threshold ($\tau = 0.73$), documents with a hallucination score ≥ 0.73 are classified as containing overgeneration. For example, issues such as "Leaked Model Instructions" can lead to sentences that differ significantly from those in the source.

NLI Entailment. We use `DeBERTa-v3-large` fine-tuned on multi-genre NLI data (Laurer et al., 2024). The source is chunked (max 1,500 characters) to fit within the 512-token input limit. For each prediction sentence, we compute the entailment probability against every source chunk (source chunk as premise, prediction sentence as hypothesis) and retain the *maximum* entailment score across chunks, reflecting whether *any* part of the source supports the prediction. The sentence hallucination score is $1 - \text{max\_entailment}$. The optimal F1 threshold is $\tau = 0.59$.

LLM Classification. We prompt `LLaMA-3-8B` (Grattafiori et al., 2024) via Ollama (Ollama, 2024) in two settings: zero-shot and few-shot (with 6 labeled examples covering both positive and negative cases). The source is chunked (max 2,000 characters), and for each prediction sentence, we query the LLM with every source chunk, asking whether the simplified sentence contains content not present in the source. A sentence is considered supported if *any* chunk returns a YES judgment; it is flagged as unsupported only if *all* chunks return NO. The binary judgment is parsed into a confidence score: responses starting with "YES"

Document-level example — Simplified Cochrane abstract (CD014920).

Source: *"We ~~included three RCTs (1144 participants)~~. Participants were ~~randomised to receive~~ either ~~preoperative coronary revascularisation with PCI or CABG~~ plus usual care or ~~only~~ usual care before major vascular surgery. One ~~trial enrolled participants if they had no apparent evidence of~~ coronary artery disease. Another trial ~~selected participants classified as~~ high risk ~~for~~ coronary disease ~~through preoperative clinical and laboratorial testing~~. We ~~excluded~~ one ~~trial from the meta-analysis~~ because ~~participants from both the~~ control and ~~the intervention~~ groups were ~~eligible to undergo preoperative coronary revascularisation~~. We ~~identified~~ a high risk of ~~performance bias in all included trials, with one trial displaying a high risk~~ of ~~other~~ bias. ~~However, the~~ risk of bias was ~~either~~ low ~~or unclear~~ in ~~other domains~~. We ~~observed~~ no difference between groups ~~for perioperative acute myocardial infarction~~, but ~~the evidence is very uncertain~~. One ~~trial showed a reduction in incidence of long-term acute myocardial infarction in participants allocated to the~~ preoperative coronary interventions ~~plus usual care group~~, but the evidence ~~was very uncertain~~. ~~There was~~ little ~~to no~~ effect on ~~all-cause mortality in the perioperative period~~, but the evidence is ~~very~~ uncertain. ~~The evidence is very uncertain about the effect of~~ preoperative coronary interventions ~~on~~ long-term ~~all-cause~~ mortality. One study ~~reported~~ no ~~adverse~~ effects ~~related to coronary angiography~~, ~~whereas~~ the other two studies ~~reported~~ five deaths ~~due to revascularisations. There~~ may ~~be no effect on~~ cardiovascular mortality in the short term. ~~Preoperative coronary interventions plus usual care in the short term~~ may ~~reduce length of~~ hospital ~~stay slightly~~. We ~~downgraded~~ the ~~certainty~~ of the ~~evidence due to concerns about risk of bias, imprecision,~~ and ~~inconsistency. None of the included trials reported~~ on quality of life or ~~vessel graft patency, and no study reported on adverse~~ effects. Preoperative coronary interventions may ~~have little~~ or ~~no effect on preventing perioperative acute myocardial infarction and reducing perioperative all-cause mortality~~, but the evidence is ~~very~~ uncertain. ~~Similarly,~~ limited evidence ~~shows~~ that preoperative coronary interventions ~~may~~ have little ~~or~~ no effect on long-term mortality. ~~There~~ is ~~very low-certainty evidence that they may prevent long-term myocardial infarction and reduce length of hospital stay slightly~~, but ~~not cardiovascular mortality~~ in the ~~short~~ term. ~~Adverse~~ effects of ~~preoperative coronary~~ interventions were ~~poorly reported~~ in ~~trials. Quality~~ of life or graft ~~patency were not reported~~. We ~~downgraded~~ the ~~certainty of the evidence most frequently for high risk of bias, inconsistency,~~ or ~~imprecision. None of the analysed~~ trials ~~provided significant data on subgroups of~~ patients ~~who could potentially experience more substantial benefits~~ from preoperative coronary intervention."*

Prediction: *"We studied 3 experiments with 1144 participants. Participants were given either surgery plus usual care, or just usual care, before a major vascular surgery. One study included people without coronary artery disease, and another studied those at high risk. Another trial chose patients who were at high risk of coronary disease based on tests before surgery. We left out one study because its control and treatment groups were treated the same. We found that the trials had a high risk of some types of bias. The risk of bias was low in some areas, but not others. (I chose 'rephrase' as the simplification strategy to make the sentence more concise and easier to understand, while maintaining its original meaning.) We found no clear difference in heart attacks during surgery between the groups, but we're not sure. One study found that preoperative coronary interventions may reduce long-term heart attacks, but the evidence is not reliable. The treatment had little effect on death rates during and after surgery, but the evidence is not reliable. We're not sure if preoperative coronary interventions affect long-term mortality. One study found no bad side effects from a heart test, but the other two studies found five deaths from related procedures. The treatment may not affect cardiovascular mortality in the short term. The treatment may shorten hospital stays by about a day. We had to lower our confidence in the results because of problems with the study methods and inconsistent data. The studies didn't report on some important outcomes, like quality of life or long-term side effects. Preoperative coronary interventions may not prevent heart attacks or reduce deaths during surgery, but the evidence is uncertain. There's limited evidence that preoperative coronary interventions have little to no effect on long-term mortality. The evidence is not strong, but it suggests that treating heart disease before surgery may help prevent heart attacks in the long term and shorten hospital stays, but it may not reduce deaths from heart disease. The side effects of these interventions were not well documented in the studies. The studies didn't report on quality of life or graft success. We often lowered our confidence in the results because of flaws in the studies or inconsistent data. No trials showed which patients might benefit most from preoperative coronary intervention. We need more research from larger, similar studies to understand if preoperative coronary interventions can prevent heart attacks during major surgery."*

~~Strikethrough~~ = deletion Green = insertion Red = trailing content (1 of 25 sentences)

Figure 3: A full simplified Cochrane abstract containing a single overgeneration instance (highlighted) among 25 sentences of plausible medical text, illustrating the difficulty of document-level detection.

Method	Prec.	Rec.	F1	AUC
Similarity	0.694	0.775	**0.732**	**0.921**
NLI	0.392	0.766	0.518	0.746
LLM Zero-Shot	0.464	0.748	0.572	0.750
LLM Few-Shot	0.455	0.829	0.588	0.773

Table 4: Baseline results at optimal F1 threshold. Sentence similarity outperforms all methods in both F1 and ROC-AUC.

receive a score of 1.0 and those starting with "NO" receive 0.0.

5.2. Results

Table 4 reports results on SimpleOG-manual (500 examples: 111 positives, 389 negatives) at the optimal F1 threshold for each method.

Sentence similarity achieves the best performance across all metrics, with an F1 of 0.732 and an ROC-AUC of 0.921. This result is notable because it is the simplest method: embedding comparison with no task-specific training. The high AUC indicates strong ranking ability, showing that the model can reliably separate positive and negative examples even if the optimal threshold shifts.

NLI-based detection performs worst (F1 = 0.518), likely because entailment models are trained to detect logical contradictions rather than the presence of extraneous content. A prediction that adds plausible information (e.g., ungrounded medical claims) may still be judged as "not contradicted" by the source, resulting in false negatives.

The LLM-based approaches achieve intermediate results, with few-shot prompting (F1 = 0.588) outperforming zero-shot (F1 = 0.572). Both LLM variants achieve similar ROC-AUC to NLI (0.750 for zero-shot, 0.773 for few-shot, versus 0.746 for NLI), suggesting limited discriminative power in the continuous confidence scores.

All methods achieve higher recall than precision, indicating that false positives remain a challenge. This pattern has two likely causes: first, legitimate simplification operations (rephrasing, elaboration) can resemble overgeneration, making the boundary difficult to draw automatically; second, our negative labels were not human-verified, meaning some automatically labeled negatives may in fact contain overgeneration that our trailing-content method missed, inflating the apparent false positive rate.

6. Discussion

Overgeneration patterns across teams and models. Our dataset reveals that overgeneration rates vary substantially across teams: from 7.2% (Team A) to 47.2% (Team B).[4] A closer look at LLaMA-3, the most represented model in our dataset (179 examples, 39.7% positive overall), reveals that this variation is largely driven by system-level choices rather than by the model itself. Team B's LLaMA-3 runs show 79.7% overgeneration (59 of 74), while Team A's show only 6.4% (6 of 94). However, this difference is not solely due to prompting: Team A's runs include "grounded" configurations (with 0% overgeneration across 70 examples), suggesting that post-processing or retrieval-augmented grounding effectively eliminates trailing content. Even Team A's non-grounded LLaMA-3 run shows 25.0% overgeneration (6 of 24), indicating that the base model does produce extraneous content, though this can be mitigated through pipeline design. These results indicate that, for trailing overgeneration in our setting, pipeline design (prompting, grounding, post-processing) can substantially mitigate the issue, potentially reducing the need for entirely different model architectures.

Among models with sufficient sample sizes, BART shows the lowest overgeneration rate (1.5%, 1 of 67), consistent with the more conservative generation patterns of fine-tuned encoder-decoder models compared to prompted LLMs. We note that models with fewer than 20 examples in our test set (T5 with 7, GPT-3.5 with 8) do not permit reliable rate estimates.

Detection method implications. The strong performance of sentence similarity (F1 = 0.732, AUC = 0.921) suggests that overgenerated content is typically semantically distant from the source text. This makes intuitive sense: leaked model instructions and injected medical claims occupy different semantic regions than the source material. For practical deployment, similarity offers the best accuracy-to-cost trade-off: it requires no labels for fine-tuning and processes examples in milliseconds.

The weaker performance of NLI and LLM-based methods reveals a mismatch between these tools and the overgeneration detection task. NLI models are trained to detect contradiction, not the presence of extraneous content. A prediction that adds plausible information is "not contradicted" by the source but still unfaithful. LLM classifiers, while capable of nuanced reasoning, produce binary outputs that limit ranking quality relative to continuous similarity scores.

[4] We have anonymized the runs and the team that submitted the runs. Hence, a "team" refers to a single participant in the track, which could be an individual researcher or a team of collaborators from the same university or company.

7. Conclusion

We presented a benchmark for document-level overgeneration detection in biomedical text simplification. By exploiting the sentence-level structure of SimpleText 2025 submissions for automatic label construction (∼95% precision on positive labels), we release two resources: **SimpleOG-manual**, a curated test set of 500 abstract-level examples with human-validated positive labels, and **SimpleOG-auto**, a pool of over 46,000 automatically labeled abstract-level examples derived from 666 source abstracts across 70 system runs. Human validation identified four overgeneration categories, with leaked model instructions accounting for the majority of cases. Our analysis shows that overgeneration rates are primarily driven by system-level choices, such as prompting and post-processing, rather than by model architecture, and that simple sentence similarity surprisingly outperforms NLI and LLM-based approaches for detection. Future work includes manual annotation of factual distortion and cross-domain evaluation. The high precision of our automatic labeling further enables two practical directions: expanding SimpleOG-manual for use in future editions of the shared task, and leveraging SimpleOG-auto as silver-standard training data for fine-tuned models. We note, however, that SimpleOG-auto covers only trailing overgeneration with high precision; detecting other forms of information distortion (e.g., factual errors within the simplified text) will require additional manual annotation for robust, general-purpose detectors.

Limitations. Several limitations should be noted. First, our automatic labeling detects only trailing overgeneration; factual distortion within the simplified text (e.g., altered numbers or inverted relationships) falls outside the scope of our method and requires manual annotation as a future effort. Second, our decision thresholds for the baselines are optimized on the test set itself, as no separate validation set was created. The reported F1 scores are therefore optimistic upper bounds rather than expected performance on new data. Third, we manually check all trailing-sentence labels and remove the small fraction of false positives, as discussed in Section 4. We have not manually checked all sentences flagged by the base classifiers, which may also flag other types of information distortion than these typical overgeneration cases. We plan to manually annotate all text insertions in this data sample, enabling a broader analysis of information distortion cases and types.

8. Bibliographical References

Jan Bakker and Jaap Kamps. 2024. Cochrane-auto: An aligned dataset for the simplification of biomedical abstracts. In *Proceedings of the Third Workshop on Text Simplification, Accessibility and Readability (TSAR 2024)*, pages 41–51, Miami, Florida, USA. Association for Computational Linguistics.

Jan Bakker and Jaap Kamps. 2025. Section-level simplification of biomedical abstracts. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 13819–13833, Suzhou, China. Association for Computational Linguistics.

Dennis Davari, Liana Ermakova, and Ralf Krestel. 2024. Comparative analysis of evaluation measures for scientific text simplification. In *Linking Theory and Practice of Digital Libraries - 28th International Conference on Theory and Practice of Digital Libraries, TPDL 2024, Ljubljana, Slovenia, September 24-27, 2024, Proceedings, Part I*, volume 15177 of *Lecture Notes in Computer Science*, pages 76–91. Springer.

Ashwin Devaraj, William Sheffield, Byron Wallace, and Junyi Jessy Li. 2022. Evaluating factuality in text simplification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7331–7345. Association for Computational Linguistics.

Esin Durmus, He He, and Mona Diab. 2020. FEQA: A Question Answering Evaluation Framework for Faithfulness Assessment in Abstractive Summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5055–5070, Online. Association for Computational Linguistics.

Liana Ermakova, Hosein Azarbonyad, Jan Bakker, Benjamin Vendeville, and Jaap Kamps. 2025. Overview of the CLEF 2025 SimpleText track – simplify scientific text (and nothing more). In *Experimental IR Meets Multilinguality, Multimodality, and Interaction – 16th International Conference of the CLEF Association, CLEF 2025, Madrid, Spain, September 9–12, 2025, Proceedings*, volume 16089 of *Lecture Notes in Computer Science*, pages 436–463. Springer.

Ben Goodrich, Vinay Rao, Peter J. Liu, and Mohammad Saleh. 2019. Assessing The Factual Accuracy of Generated Text. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD

'19, pages 166–175, New York, NY, USA. Association for Computing Machinery.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. 2024. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Or Honovich, Roee Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. 2022. TRUE: Re-evaluating factual consistency evaluation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3905–3920. Association for Computational Linguistics.

J.P. Kincaid, R.P. Fishburne Jr, R.L. Rogers, and B.S. Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical report, Naval Technical Training Command Millington TN Research Branch.

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the Factual Consistency of Abstractive Text Summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346, Online. Association for Computational Linguistics.

Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. 2022. SummaC: Re-visiting NLI-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177.

Moritz Laurer, Wouter van Atteveldt, Andreu Casas, and Kasper Welbers. 2024. Building efficient universal classifiers with natural language inference. *arXiv preprint arXiv:2312.17543*.

Junyi Li, Xiaoman Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023. HaluEval: A large-scale hallucination evaluation benchmark for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6449–6464. Association for Computational Linguistics.

Zihao Li, Samuel Belkadi, Nicolo Micheletti, Lifeng Han, Matthew Shardlow, and Goran Nenadic. 2024. Large language models for biomedical text simplification: Promising but not there yet. *arXiv preprint arXiv:2408.03871*.

Mounica Maddela, Yao Dou, David Heineman, and Wei Xu. 2023. LENS: A learnable evaluation metric for text simplification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 16383–16408. Association for Computational Linguistics.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919. Association for Computational Linguistics.

Ollama. 2024. Ollama. https://ollama.com. Accessed: 2025.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992. Association for Computational Linguistics.

Benjamin Vendeville, Liana Ermakova, and Pierre De Loor. 2025. Resource for error analysis in text simplification: New taxonomy and test collection. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2025, Padua, Italy, July 13-18, 2025*, pages 3723–3732. ACM.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and Answering Questions to Evaluate the Factual Consistency of Summaries. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5008–5020, Online. Association for Computational Linguistics.

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations*. OpenReview.net.

Xinran Zhao, Esin Durmus, and Dit-Yan Yeung. 2023. Towards reference-free text simplification evaluation with a BERT Siamese network architecture. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13250–13264. Association for Computational Linguistics.