DUTH at CLEF 2025 SimpleText Track: Tackling Scientific Text Simplification and Hallucination Detection

Georgios Arampatzis*, Avi Arampatzis

Democritus University of Thrace, Department of Electrical and Computer Engineering, Xanthi, Greece

Abstract

This paper presents the participation of the DUTH team in the CLEF 2025 SimpleText Track, focusing on the automatic simplification of scientific texts and the detection of hallucinations. For the simplification tasks, we employed large instruction-tuned language models (LLMs), such as FLAN-T5-Large and BART-SAMSum. Experiments at both the sentence level (Task 1.1) and the document level (Task 1.2) showed that scaling up the model and curating the content significantly improve simplification quality. The models demonstrated the ability to preserve semantic accuracy, even in complex contexts.

In the field of hallucination detection (Task 2), we applied both binary and multi-class classification methods, based on lexical and semantic representations. Tree-based ensemble learning models, such as Extra Trees and Random Forest, achieved top performance in identifying erroneous content, under both posthoc and sourced conditions. However, the fine-grained classification of error types (Task 2.2) revealed substantial challenges—particularly in detecting semantic deviations, such as hallucinations of reality.

Future work will focus on incorporating contextual embeddings, applying few-shot learning, and enhancing the robustness of the models.

Keywords

Simplification, Hallucination, Scientific Texts, Instruction Tuning, Document-level Simplification, Large Language Models, Posthoc Annotation, Ensemble Methods

1. Introduction

Scientific texts are often characterized by dense terminology and complex syntactic structures, making them difficult to understand for lay audiences. As a result, non-experts frequently avoid engaging with primary scientific literature, instead relying on simplified or secondary sources—such as blogs or social media—which may contain distorted or unreliable interpretations [1]. Automatic text simplification, particularly within scientific domains, aims to bridge this accessibility gap by transforming complex content into more comprehensible forms while preserving factual accuracy and intended meaning.

The CLEF 2025 SimpleText Track [1] was introduced to support the systematic evaluation of scientific text simplification systems and to address the growing concern of hallucinations—spurious content not grounded in the source—often produced by generative language models. In this context, we participated in two core tasks of the track:

- **Task 1: Simplify Scientific Text**, which includes:
  - **Task 1.1:** Sentence-level simplification

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

*Corresponding author.

†These authors contributed equally.

geoaramp@ee.duth.gr (G. Arampatzis*); avi@ee.duth.gr (A. Arampatzis)

0009-0003-3840-4537 (G. Arampatzis*); 0000-0003-2415-4592 (A. Arampatzis)

  - **Task 1.2:** Document-level simplification [2]
- **Task 2: Identify and Avoid Hallucination**, which focuses on detecting erroneous or fabricated information in simplified outputs [3]

Prior work in scientific text simplification includes both supervised approaches using aligned corpora [4, 5] and prompt-based techniques leveraging large language models (LLMs) such as GPT-3 or T5 [6, 7]. Although LLMs generate fluent outputs, they are prone to hallucinations—posing a significant risk in scientific applications where factual consistency is paramount. Consequently, recent research has increasingly focused on evaluating and mitigating hallucinated content in generated text [8, 9].

This paper presents our submission to Tasks 1 and 2 of the CLEF 2025 SimpleText track. For full details on task definitions, datasets, and evaluation protocols, we refer the reader to the official track and task overview papers [10, 11, 12].

The remainder of this paper is organized as follows: Section 2 describes our methodology, including data preprocessing, system architecture, and prompt engineering strategies. Section 3 presents the experimental results and analysis. Section 4 concludes with key findings and discusses directions for future work.

2. Experimental Setup

2.1. Task 1:Text Simplification: Simplify scientific text

2.1.1. Task 1.1: Sentence-level Scientific Text Simplification

2.1.1.1. Dataset Table 1 provides a detailed overview of the dataset configuration used in Task 1.1, which focuses on the simplification of scientific texts at the sentence level. Each split is characterized by the number of unique complex sentences, the total number of entries, and the presence or absence of reference simplifications.

The training set contains 11,452 unique complex sentences and 11,510 total entries, each paired with a corresponding simplification. The validation and internal test sets consist of 1,695 and 1,510 unique sentences, respectively, and are used for model tuning and intermediate evaluation.

In contrast, the final test set comprises 9,086 unique sentences and 9,160 entries, but does not include reference simplifications. This test set is used for the official system evaluation and leaderboard submission via the Codabench platform. The absence of gold outputs ensures an unbiased, blind evaluation of system performance.

The difference between the number of unique sentences and total entries stems from cases in which multiple simplifications are provided for a single complex sentence, as defined by the dataset schema in the CSV files.

Table 1
Summary of the Task 1.1 Sentence-level Simplification Dataset

Dataset	Unique Complex Sentences	Total Entries	Simplifications Included
Train Set	11,452	11,510	Yes
Validation Set	1,695	1,697	Yes
Test Set	1,510	1,512	Yes
Final Test Set	9,086	9,160	No

2.1.1.2. Methodology The simplification approach adopted for Task 1.1 leverages large pretrained language models tailored for sequence-to-sequence tasks. These models operate in a zero-shot setting and are prompted to rewrite complex scientific sentences into simpler, more accessible forms while preserving core meaning and factual accuracy.

To guide generation, task-specific prompts were applied where relevant, encouraging the models to produce outputs that align with simplification goals. The decoding strategy emphasized determinism and brevity, ensuring that the generated sentences were concise and syntactically well-formed.

This methodology exploits the generalization capabilities of large-scale foundation models to perform scientific text simplification without additional supervision, demonstrating their feasibility for specialized communication tasks in scientific domains.

2.1.2. Task 1.2 – Document-level Scientific Text Simplification

2.1.2.1. Dataset Table 2 presents an overview of the document-level dataset distribution used in Task 1.2, which focuses on the simplification of full scientific abstracts. Each split is characterized by the number of unique complex documents, total entries (rows), and the availability of reference simplifications.

The training set comprises 3,967 documents, each paired with corresponding simplifications. The validation and internal test sets contain 500 and 502 documents, respectively, and include gold-standard simplifications. These subsets are intended for model development, hyperparameter tuning, and preliminary evaluation.

The final test set includes 666 complex documents without reference simplifications. It serves as the blind evaluation input for official submissions on Codabench. The absence of target outputs ensures fair and unbiased scoring of participants' systems.

Notably, the number of total entries equals the number of unique documents across all subsets, indicating that each row corresponds to a single abstract. Unlike Task 1.1, document-level simplification requires handling discourse structure, paragraph segmentation, and sentence-level transformations in a holistic and coherent manner.

Table 2
Overview of the Task 1.2 dataset for document-level scientific text simplification.

Dataset	Unique Documents	Total Entries	Reference Simplifications
Train Set	3967	3967	Yes
Validation Set	500	500	Yes
Test Set	502	502	Yes
Final Test Set	666	666	No

2.1.2.2. Methodology The approach followed for Task 1.2 leverages a large pretrained language model to simplify full scientific abstracts, addressing the broader context and discourse structure inherent to document-level content. The model is guided through natural language prompts that explicitly define the simplification objective.

Each document is treated as a single input unit, and the model is prompted to generate a simplified version that preserves terminological precision, semantic fidelity, and coherence across multiple sentences. This is achieved by prepending structured instructions (e.g., "Simplify the following scientific document:") to the complex abstract.

The methodology highlights the ability of large-scale instruction-tuned models to handle extended scientific discourse and produce simplified outputs that remain faithful to the original content—without the need for fine-tuning on domain-specific simplification data.

2.1.3. Implementation and Environment

All experiments were implemented in **Python 3.10**, using the `Transformers` library from Hugging Face and `PyTorch` (v2.1.0). Execution was performed on a compute node equipped with an NVIDIA RTX A6000 GPU.

2.2. Controlled Creativity: Identify and Avoid Hallucination

2.2.1. Task 2.1 – Identify Creative Generation at Document Level

2.2.1.1. Dataset Table 3 presents a quantitative summary of the posthoc subset used in Task 2 of the CLEF 2025 SimpleText track, which evaluates hallucination detection in simplified scientific texts. This subset consists of system-generated simplifications that were annotated after generation to determine whether they contain hallucinated content.

The table reports both the number of unique simplified sentences and the total number of entries, which may include duplicates due to multiple annotations or metadata variants. Specifically:

The posthoc training set contains 13,137 unique simplified sentences and 13,519 total entries, indicating that some examples appear more than once due to secondary annotations, such as annotator disagreement or metadata variation.

The posthoc test set includes 3,249 unique sentences and 3,293 entries, and is used to evaluate hallucination detection models under controlled conditions.

This subset is particularly valuable for training models that must generalize to noisy, real-world outputs from text simplification systems. Its posthoc nature provides a realistic evaluation setting, where hallucinations are assessed independently of the system that produced the simplification.

The distinction between unique instances and total entries offers insight into the annotation process and potential variance introduced by human labeling or system generation artifacts.

Table 3
Overview of the posthoc subset used in Task 2 for hallucination detection, including the number of unique simplified sentences and total annotated entries.

Dataset	Unique Sentences	Total Entries
Posthoc Train Set	13,137	13,519
Posthoc Test Set	3,249	3,293

Table 4 presents a summary of the sourced subset used in Task 2 of the CLEF 2025 SimpleText Track, which focuses on detecting hallucinations in simplified scientific texts. Unlike the posthoc subset, the sourced data are constructed such that each simplification is directly aligned with a known and verifiable source text. This design enables explicit grounding assessment by checking whether all information in the output is traceable to the input.

Each split is characterized by two key metrics: the number of unique simplified sentences and the total number of entries, which may include duplicates due to repeated annotations or system variants.

The sourced training set contains 13,120 unique simplified sentences and 13,514 total entries. Minor redundancy may arise from sentence variants, additional annotations (e.g., multiple annotators), or metadata replication.

The sourced test set includes 3,318 unique sentences and 3,379 entries, and serves as the benchmark for evaluating hallucination detection models on source-aligned simplifications.

The explicit grounding offered by this dataset makes it particularly suitable for supervised learning and fine-grained hallucination evaluation. In combination with the posthoc subset, it supports robust model development across both real-world and controlled hallucination scenarios.

Table 4
Overview of the sourced subset used in Task 2, including the number of unique simplified sentences and total annotated entries for hallucination detection.

Dataset	Unique Sentences	Total Entries
Sourced Train Set	13,120	13,514
Sourced Test Set	3,318	3,379

2.2.1.2. Methodology To address the detection of spurious or hallucinated content in simplified scientific sentences, we adopted a supervised binary classification framework based on lexical features. Two parallel models were developed—one for the sourced and one for the posthoc subsets—using the same processing pipeline.

Each classifier was trained to distinguish between factually accurate and spurious simplifications using an ensemble-based learning approach. Specifically, we employed the `ExtraTreesClassifier`, a non-parametric ensemble method that aggregates multiple randomized decision trees to improve robustness and generalization.

Input sentences were vectorized using a TF-IDF representation over a vocabulary of the 3,000 most informative terms. To address class imbalance in the training data, the minority class was upsampled via random oversampling, resulting in a balanced training set. Predictions were then generated for the test instances and exported in structured format for evaluation.

This approach demonstrates the effectiveness of combining simple lexical representations with ensemble learning methods for hallucination detection in scientific text simplification.

2.2.2. Task 2.2 – Detect and Classify Information Distortion Errors in Simplified Sentences

2.2.2.1. Dataset Table 5 summarizes the dataset used in Task 2.2 of the CLEF 2025 SimpleText Track, which targets fine-grained error annotation in sentence-level simplifications. The dataset supports supervised training and evaluation of systems capable of identifying specific simplification errors, such as hallucinations, faithfulness violations, and discourse-level inconsistencies.

The training set comprises 42,392 annotated entries corresponding to 35,621 unique simplified sentences. Each entry includes one or more categorical labels indicating the presence of error types (e.g., factuality hallucination, topic shift, overgeneralization). Due to the multi-label structure and multiple annotations per sentence, individual sentences may appear more than once in the dataset.

The test set contains 2,659 entries derived from 1,537 unique complex source sentences. Unlike the training data, test instances do not include simplified outputs, enabling blind evaluation: systems must infer likely errors solely based on the input sentence.

This dataset plays a key role in advancing error-aware simplification systems, providing a structured foundation for training multi-class classifiers and enabling performance breakdown by error type across diverse semantic and pragmatic dimensions.

Table 5
Overview of the dataset used in Task 2.2 for hallucination and error-type annotation.

Dataset	Unique Sentences	Total Entries
Task 2.2 Train Set	35,621	42,392
Task 2.2 Test Set	1,537	2,659

2.2.2.2. Methodology For the fine-grained detection of hallucination errors in simplified scientific text, we adopt a multi-label classification framework grounded in semantic similarity. Sentence pairs—comprising the original and the simplified version—are embedded into dense semantic vectors using a pretrained sentence encoder (`all-mpnet-base-v2`), enabling the model to capture meaning-preserving or distorting transformations.

A multi-output classifier is trained on these embeddings to predict the presence of specific hallucination categories, as defined by a structured error taxonomy. To address class imbalance and data sparsity, the training set is augmented with synthetic examples and oversampling techniques. Label-wise thresholds are tuned via validation-based F1 maximization to ensure calibrated predictions across error types.

Implementation Details. We used the `all-mpnet-base-v2` model from the SentenceTransformers library to generate fixed-size semantic embeddings. The encoder operated in inference-only mode, with **no task-specific fine-tuning**; that is, its parameters remained frozen during training. Only the downstream classifier—a `MultiOutputClassifier` using either Logistic Regression or Random Forest as the base estimator—was trained on the extracted embeddings. This lightweight architecture enables efficient yet effective classification of hallucination error types in scientific simplifications.

2.3. Implementation and Environment

All experiments were implemented in **Python 3.10**, using the `PyTorch` framework (v2.1.0) in conjunction with the Hugging Face `transformers` and `sentence-transformers` libraries. For the classification tasks, models were built using `scikit-learn`, including both tree-based ensemble methods (e.g., Extra Trees, Random Forest) and linear classifiers (e.g., Logistic Regression).

3. Results

3.1. Evaluation Metrics

We evaluate sentence simplification and hallucination detection using a combination of reference-based, semantic, and classification-based metrics.

3.1.1. Sentence Simplification (Tasks 1.1 and 1.2)

The main evaluation metrics include:

- **SARI** [13]: Evaluates the quality of added, deleted, and retained n-grams with respect to reference simplifications.
- **BLEU** [14]: Measures n-gram overlap with reference texts, though it is less sensitive to simplification quality.
- **BERTScore** [15]: Computes semantic similarity between system outputs and references using contextual embeddings.
- **LENS** [16]: A learned metric trained on human-annotated simplification quality ratings.
- **SLE** [5]: A classifier-based, reference-less metric that distinguishes simplified from non-simplified outputs.

3.1.2. Hallucination Detection (Task 2.1).

For binary classification of hallucinated content, we report standard classification metrics:

- **Accuracy** [17]: Proportion of correct predictions over all predictions.
- **Precision** [18]: Proportion of predicted positives that are correct.
- **Recall** [18]: Proportion of actual positives that are correctly predicted.
- **F1-score** [18]: Harmonic mean of precision and recall.
- **ROC AUC** [17]: Area under the ROC curve, indicating overall class separability.

3.2. Task 1.1 – Sentence-level Scientific Text Simplification

3.2.1. Experimental Results

Table 6 presents detailed results for sentence-level scientific text simplification (Task 1.1), evaluated using three metrics: *SARI (original)*, *SARI (auto)*, and the final *Score*. These metrics assess simplification quality in terms of information added, deleted, and retained, incorporating both reference-based and automatic evaluations.

The **FLAN-T5-Large** model outperforms all others, achieving a **SARI (original) of 35.35** and a high **SARI (auto) of 38.73**. This indicates a strong ability to generate simplified outputs that preserve core semantic content while enhancing accessibility. Its consistent performance across human and automatic references demonstrates the robustness of large-scale, instruction-tuned models in zero-shot settings.

The `BART-SAMSum` model, despite being pretrained on dialogue summarization data, performs competitively with a **SARI (original) of 29.68**, surpassing the generic `BART` model (*23.84*). This suggests that pretraining on abstractive, paraphrastic tasks can effectively transfer to scientific simplification, even in the presence of domain mismatch.

In contrast, smaller variants such as `FLAN-T5-Base` and `FLAN-T5-XL` yield significantly lower scores (*19.51* and *18.78*, respectively), underscoring the impact of model scale on simplification quality. These results support the hypothesis that both size and instruction tuning are key factors in enabling generalization without task-specific supervision.

Finally, the gap between *SARI (original)* and *SARI (auto)* offers additional insights into evaluation alignment. The top-performing `FLAN-T5-Large` exhibits strong agreement across both metrics, suggesting its outputs align well with both human references and automated paraphrases—further validating its generalization capacity.

Table 6
Simplification performance for Task 1.1 (Sentence-level), reported per model using SARI (original), SARI (auto), and the final evaluation score.

Model	SARI (original)	SARI (auto)	Score
flan-t5-large	35.348	38.730	35.348
bart-samsum	29.677	32.184	29.677
bart	23.835	27.587	23.835
flan-t5-base	19.512	23.280	19.512
flan-t5-xl	18.784	22.749	18.784

3.3. Task 1.2 – Document-level Scientific Text Simplification

3.3.1. Experimental Results

The evaluation results for Task 1.2 indicate that models incorporating domain adaptation or content cleaning strategies yield improved performance in document-level scientific text simplification. The **top-performing system**, `bart-samsum_clean`, achieved a score of **36.998**, demonstrating the benefit of leveraging dialogue-style summarization pretraining combined with targeted refinement.

Closely following, `flan-t5-xl_clean` and `flan-t5-xxl_clean` achieved scores of **36.620** and **35.813**, respectively, confirming the positive effect of scaling and data curation. The `flan-t5-large_co` variant, presumably optimized with contrastive objectives, also performed competitively with a score of **34.612**.

In contrast, `flan-t5-base`—the smallest model—achieved a lower score of **33.130**, suggesting a performance ceiling for models lacking sufficient capacity or instruction tuning. This reinforces the sensitivity of document-level simplification to both model scale and pretraining configuration.

Overall, the results highlight the importance of instruction tuning, scaling, and input refinement in achieving high-quality simplifications that preserve coherence and semantic fidelity—crucial for expert-to-lay communication in scientific domains.

Table 7
Simplification performance for Task 1.2 (Document-level), reported per model using SARI (original), SARI (auto), and the official final score.

Model	SARI (original)	SARI (auto)	Score
bart-samsum_clean	36.998	36.251	36.998
flan-t5-xl_clean	36.620	36.653	36.620
flan-t5-xxl_clean	35.813	34.733	35.813
flan-t5-large_co	34.612	32.553	34.612
flan-t5-base	33.130	–	33.130

3.4. Task 2.1 – Identify Creative Generation at Document Level

3.4.1. Experimental Results

Table 8 presents detailed classification results for Task 2.1, focusing on posthoc hallucination detection in scientific text simplification. The evaluation covers Accuracy, Precision, Recall, F1-score, ROC AUC, and a unified Score to comprehensively assess model performance.

The best-performing model is the **Extra Trees** classifier, which achieves an F1-score and Score of **0.948**, alongside a high Recall (0.974) and Accuracy (0.904). These results underscore the model's robustness in identifying hallucinated content, even in imbalanced or sparse feature settings, affirming the strength of ensemble tree-based methods.

Random Forest follows closely with a Score of **0.945** and an F1-score of 0.945, further confirming the efficacy of ensemble approaches. Both models effectively balance precision and recall, which is essential for minimizing both false positives and false negatives in hallucination detection pipelines.

Support Vector Classifier and XGBoost achieve moderate performance, with Scores of 0.879 and 0.874, respectively. Despite being more complex learners, they lag behind the ensemble methods, possibly due to their sensitivity to data representation or hyperparameter tuning.

Linear models like **Logistic Regression** and **Ridge Regression** also perform competitively, reaching F1-scores of 0.863 and 0.862. Their success suggests that even without complex architectures, high-dimensional TF-IDF representations can be effectively leveraged to detect semantic inconsistencies.

At the lower end, *Gradient Boosting* and *KNN* scored lowest (**0.784**, **0.210**), reflecting limited generalization in sparse, high-dimensional spaces. The weak KNN performance aligns with prior findings [19] on its inefficacy in complex multi-label settings.

Overall, the findings confirm that tree-based ensemble models, particularly **Extra Trees** and **Random Forest**, are highly effective in posthoc hallucination detection. Their ability to handle feature sparsity, combined with robust discriminative performance, makes them suitable for integration into real-world simplification quality assurance systems.

Table 8
Detailed classification metrics for Task 2.1 (Posthoc Hallucination Detection)

Model	Accuracy	Precision	Recall	F1-score	ROC AUC	Score
Extra Trees	0.904	0.924	0.974	0.948	0.621	0.948
Random Forest	0.898	0.925	0.965	0.945	0.625	0.945
Support Vector Classifier	0.795	0.937	0.827	0.879	0.662	0.879
XGBoost	0.789	0.945	0.814	0.874	0.690	0.874
Logistic Regression	0.773	0.947	0.792	0.863	0.696	0.863
Ridge Regression	0.769	0.938	0.797	0.862	0.659	0.862
SGD Classifier	0.762	0.951	0.776	0.855	0.706	0.855
Naive Bayes	0.754	0.949	0.768	0.849	0.695	0.849
Gradient Boosting	0.669	0.950	0.668	0.784	0.673	0.784
K-Nearest Neighbors	–	–	–	–	–	0.210

The results presented in Table 9 demonstrate that ensemble-based classifiers achieve superior performance in the *sourced hallucination detection* setting. Specifically, *Extra Trees* and *Random Forest* attain the highest overall scores (F1-score: **0.950** and **0.945**, respectively), indicating their robustness in capturing subtle lexical or semantic cues related to spurious content. Both models exhibit excellent recall (**0.974** and **0.964**) while maintaining high precision, suggesting a balanced ability to identify hallucinated instances without overfitting.

Among the linear models, *Ridge Regression* and *Logistic Regression* perform consistently well (F1-scores: **0.861** and **0.860**), showing that even without non-linear transformations, TF-IDF-based representations provide strong discriminative power. The *Support Vector Classifier* also demonstrates notable performance (F1-score: **0.881**), with an accuracy of **0.799** and ROC AUC of **0.688**, confirming its capacity to construct expressive hyperplanes for this binary classification task.

SGD Classifier and *Naive Bayes* yield slightly lower performance (F1-scores: **0.842** and **0.838**, re-

spectively), yet still maintain reasonable balance between precision and recall, affirming their utility as lightweight and interpretable alternatives.

The lowest performing model is *Gradient Boosting*, with an F1-score of **0.768** and recall of just **0.642**, despite a high precision of **0.955**. This suggests that while the model is highly conservative in predicting hallucinations (yielding few false positives), it fails to recall a significant portion of true hallucinated cases — potentially due to overfitting or an inability to generalize across sparse lexical input.

Overall, the sourced hallucination detection results corroborate the effectiveness of tree-based ensembles and strong linear classifiers, which consistently achieve a desirable trade-off between precision and recall, making them well-suited for reliable identification of hallucinated content in scientific text.

Table 9
Detailed classification metrics for Task 2.1 (Sourced Hallucination Detection)

Model	Accuracy	Precision	Recall	F1-score	ROC AUC	Score
Extra Trees	0.909	0.928	0.974	0.950	0.656	0.950
Random Forest	0.900	0.927	0.964	0.945	0.650	0.945
Support Vector Classifier	0.799	0.941	0.827	0.881	0.688	0.881
Ridge Regression	0.770	0.941	0.794	0.861	0.679	0.861
Logistic Regression	0.769	0.945	0.789	0.860	0.692	0.860
SGD Classifier	0.745	0.947	0.758	0.842	0.694	0.842
Naive Bayes	0.739	0.945	0.753	0.838	0.683	0.838
Gradient Boosting	0.651	0.955	0.642	0.768	0.688	0.768

The lowest performing model is *Gradient Boosting*, with an F1-score of **0.768** and recall of just **0.642**, despite a high precision of **0.955**. This suggests that while the model is highly conservative in predicting hallucinations (yielding few false positives), it fails to recall a significant portion of true hallucinated cases — potentially due to overfitting or an inability to generalize across sparse lexical input.

Overall, the sourced hallucination detection results corroborate the effectiveness of tree-based ensembles and strong linear classifiers, which consistently achieve a desirable trade-off between precision and recall, making them well-suited for reliable identification of hallucinated content in scientific text.

3.5. Task 2.2 – Detect and Classify Information Distortion Errors in Simplified Sentences

3.5.1. Evaluation Metrics

The evaluation of Task 2.2 (Detect and Classify Information Distortion Errors) is framed as a multi-label classification problem, where each simplified sentence may exhibit multiple error types drawn from a predefined taxonomy.

System performance is assessed using:

- Precision, Recall, and F1-score per error class;
- Macro-averaged F1-score across all labels, to account for class imbalance and to provide an overall measure of system effectiveness.

This evaluation setup enables a fine-grained assessment of a model's ability to detect both surface-level issues (e.g., grammar errors) and deeper semantic inconsistencies (e.g., factual hallucinations), in line with prior work on multi-label learning [20] and factual consistency evaluation in text generation [8]. The task design and metric definitions follow the CLEF 2025 SimpleText guidelines [21].

3.5.2. Experimental Results

The classification results for Task 2.2 reveal substantial variability in performance across error categories, highlighting the inherent difficulty of multi-label hallucination detection in scientific simplification.

The system demonstrates strong performance in detecting sentences labeled as having **no errors**, with an F1-score of 0.496, driven by high recall (0.937) but limited precision. This suggests that the model tends to over-predict error-free cases, successfully retrieving many valid simplifications, albeit with a high false-positive rate.

In contrast, performance on hallucination-related categories remains low. For instance, **Factuality Hallucination (C1)** and **Prompt Misalignment (B2)** achieve F1-scores of only 0.025 and 0.014, respectively—reflecting the subtle, context-dependent nature of these phenomena and the challenge of reliably capturing them from limited input representations.

Some categories, such as **Loss of Informative Content (D2.1)** and **Faithfulness Hallucination (C2)**, yielded relatively higher F1-scores (0.290 and 0.185), suggesting that models are more capable of detecting content reduction or minor semantic inconsistencies compared to abstract hallucination types.

Overall, these findings indicate that while surface-level or structural errors (e.g., syntactic mistakes or overgeneralization) are more tractable, deeper semantic distortions and hallucinations remain difficult to detect using current feature-based classifiers—emphasizing the need for richer contextual modeling or task-specific representation learning.

Table 10
Detailed classification results for Task 2.2 grouped by error category

Category	Label	Precision	Recall	F1-score
No Error				
	No error	0.338	0.937	0.496
Category A: Linguistic Errors				
	A1. Random generation	0.020	0.128	0.035
	A2. Syntax error	0.066	0.508	0.117
	A3. Contradiction	0.000	0.000	0.000
	A4. Punctuation / Grammar	0.082	0.340	0.132
	A5. Redundancy	0.032	0.012	0.017
Category B: Alignment Issues				
	B1. Format misalignment	0.022	0.071	0.034
	B2. Prompt misalignment	0.026	0.010	0.014
Category C: Semantic Hallucinations				
	C1. Factuality hallucination	0.013	0.483	0.025
	C2. Faithfulness hallucination	0.170	0.204	0.185
	C3. Topic shift	0.054	0.185	0.084
Category D: Content Loss and Scope Shift				
	D1.1. Overgeneralization	0.123	0.382	0.186
	D1.2. Overspecification of Concepts	0.068	0.128	0.089
	D2.1. Loss of Informative Content	0.212	0.460	0.290
	D2.2. Out-of-Scope Generation	0.111	0.008	0.015

4. Discussion and Conclusions

This work presented our participation in the CLEF 2025 SimpleText Track, addressing both scientific text simplification and hallucination detection. For simplification (Tasks 1.1 and 1.2), instruction-tuned large language models—such as *FLAN-T5-Large* and *BART-SAMSum*—demonstrated strong zero-shot capabilities, particularly when scaled or enhanced through content cleaning. Notably, sentence-level simplification benefited from increased model capacity, while document-level tasks required coherence-aware prompting strategies.

Looking ahead, we plan to investigate few-shot prompting for Tasks 1.1 and 1.2, incorporating in-context examples to further improve simplification quality—especially in domains requiring terminological precision and semantic fidelity.

For hallucination detection (Task 2.1), tree-based ensemble classifiers (*Extra Trees*, *Random Forest*) proved highly effective in both posthoc and sourced conditions. While these methods perform well using lexical features, future work will explore transformer-based classifiers (e.g., fine-tuned *BERT*) to assess whether contextualized embeddings can better capture subtle inconsistencies beyond shallow representations.

Task 2.2 further revealed limitations in capturing fine-grained semantic distortions, with substantial variation in performance across error categories. To address this, we aim to incorporate contextual embeddings from transformer encoders (e.g., *BERT*, *RoBERTa*) into classification pipelines, and apply hierarchical modeling and curriculum learning to better capture inter-error dependencies. Enhancing data diversity via augmentation and annotation bootstrapping will also be critical for improving generalization in underrepresented categories.

Overall, our goal is to develop models that are both simplification-aware and hallucination-resilient, supporting faithful and accessible communication of scientific content to non-expert audiences.

5. Acknowledgments

We gratefully acknowledge the organizers of the CLEF 2025 SimpleText Track for their dedicated efforts in designing and coordinating the track. The datasets, tools, and evaluation infrastructure they provided formed a crucial basis for the development and assessment of our systems.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] L. Ermakova, et al., Overview of clef 2025 simpletext track: Simplify scientific texts (and nothing more), in: Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Sixteenth International Conference of the CLEF Association (CLEF 2025), LNCS, Springer, 2025.

[2] J. Bakker, et al., Overview of the clef 2025 simpletext task 1: Simplify scientific text, in: Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2025), CEUR Workshop Proceedings, 2025.

[3] B. Vendeville, et al., Overview of the clef 2025 simpletext task 2: Identify and avoid hallucination, in: Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2025), CEUR Workshop Proceedings, 2025.
[4] X. Zhang, M. Lapata, Sentence simplification with deep reinforcement learning, Transactions of the Association for Computational Linguistics (TACL) 5 (2017) 365–378.
[5] L. Martin, B. Muller, P. J. O. Suarez, D. Seddah, B. Sagot, Controllable sentence simplification with a constrained seq2seq model, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, p. 3537–3550.
[6] Y. Jiang, Y. Shen, et al., Prompting gpt for text simplification: A case study, in: Findings of the Association for Computational Linguistics: EMNLP, 2022.
[7] D. Madaan, et al., Text simplification with large language models, in: arXiv preprint arXiv:2302.13971, 2023.
[8] J. Maynez, S. Narayan, B. Bohnet, R. McDonald, On faithfulness and factuality in abstractive summarization, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, p. 1906–1919.
[9] N. Dziri, O. R. Zaiane, et al., Evaluating the factual consistency of abstractive text summarization, in: Findings of the Association for Computational Linguistics: NAACL, 2022.
[10] L. Ermakova, H. Azarbonyad, J. Bakker, B. Vendeville, J. Kamps, Overview of the CLEF 2025 SimpleText track: Simplify scientific texts (and nothing more), in: J. Carrillo de Albornoz, J. Gonzalo, L. Plaza, A. García Seco de Herrera, J. Mothe, F. Piroi, P. Rosso, D. Spina, G. Faggioli, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Sixteenth International Conference of the CLEF Association (CLEF 2025), Lecture Notes in Computer Science, Springer, 2025.
[11] J. Bakker, B. Vendeville, L. Ermakova, J. Kamps, Overview of the clef 2025 simpletext task 1: Simplify scientific text, in: [22], 2025.
[12] B. Vendeville, J. Bakker, L. Ermakova, J. Kamps, Overview of the clef 2025 simpletext task 2: Identify and avoid hallucination, in: [22], 2025.
[13] W. Xu, C. Callison-Burch, C. Napoles, Optimizing statistical machine translation for text simplification, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL), 2016, p. 560–570.
[14] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, Bleu: a method for automatic evaluation of machine translation, in: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, 2002, p. 311–318.
[15] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, Y. Artzi, Bertscore: Evaluating text generation with bert, International Conference on Learning Representations (ICLR) (2020).
[16] T. Goyal, L. Martin, D. Kumar, G. Durrett, Lens: A learned evaluation metric for sentence simplification, in: Proceedings of ACL, 2022.
[17] T. Fawcett, An introduction to roc analysis, Pattern recognition letters 27 (2006) 861–874.
[18] D. M. W. Powers, Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation, Journal of Machine Learning Technologies 2 (2011) 37–63.
[19] G. Arampatzis, V. Perifanis, S. Symeonidis, A. Arampatzis, DUTH at SemEval-2023 Task 9: An Ensemble Approach for Twitter Intimacy Analysis, in: Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023), Association for Computational Linguistics, 2023, pp. 1225–1230. URL: https://aclanthology.org/2023.semeval-1.170. doi:10.18653/v1/2023.semeval-1.170.
[20] G. Tsoumakas, I. Katakis, Multi-label classification: An overview, International Journal of Data

Warehousing and Mining (IJDWM) 3 (2007) 1–13.
[21] B. Vendeville, L. Ermakova, J. Bakker, S. Wrigley, Overview of the clef 2025 simpletext task 2: Identify and avoid hallucination, in: Working Notes of CLEF 2025 – Conference and Labs of the Evaluation Forum, CEUR Workshop Proceedings, CEUR-WS.org, 2025. To appear.
[22] G. Faggioli, N. Ferro, P. Rosso, D. Spina (Eds.), Working Notes of CLEF 2025: Conference and Labs of the Evaluation Forum, CEUR Workshop Proceedings, CEUR-WS.org, 2025.

Appendix A: Submission Information

Team name: DUTH
Track: CLEF 2025 SimpleText
Submitted Tasks: Task 1.1, Task 1.2, Task 2.1, Task 2.2
Run IDs:

- 5 runs for Task 1.1
- 5 runs for Task 1.2
- 10 runs for Task 2.1 (Posthoc)
- 8 runs for Task 2.1 (Sourced)
- 1 run for Task 2.2

Appendix B: Prompt Examples for Task 1

Task 1.1 – Sentence-level Simplification (FLAN-T5-XL).

The following prompt was used in a zero-shot setting:

> `Prompt:` Simplify: The medicine caused drowsiness and fatigue.
> `Output:` The medicine made the person tired and sleepy.

Task 1.1 – Sentence-level Simplification (BART-SAMSUM).

For BART, no explicit instruction was used. The model received the raw sentence directly as input:

> `Input:` The medicine caused drowsiness and fatigue.
> `Output:` The drug made people feel tired and sleepy.

Task 1.2 – Document-level Simplification.

The prompt used for document-level simplification was:

> `Prompt:` Simplify the following scientific document:
> In this study, we investigate the structural behavior of graphene-based materials under varying thermal and mechanical conditions. Our findings demonstrate significant improvements in tensile strength and flexibility when integrated into polymer composites.
>
> `Output:` This study looks at how graphene materials behave under heat and stress. The results show they become stronger and more flexible in plastics.