AIIRLab Systems for CLEF 2025 SimpleText: Cross-Encoders to Avoid Spurious Generation

Nicholas Largey[1,*,†], Deiby Wu[1,†] and Behrooz Mansouri[1]

[1] AIIR Lab, University of Southern Maine, Portland ME 04103, USA

Abstract

This paper investigates the systems proposed by the AIIRLab team in the CLEF 2025 SimpleText track. The team participated in the main tasks: 1) Text Simplification, and 2) Controlled Creativity. For Task 1, the subtasks focus on sentence- and document-level scientific text simplification, where the team's proposed approaches use large language models, including Mistral and LLaMA. During the fine-tuning process, many of the models suffered from hallucinations and extraneous outputs. Better simplification was achieved through the strategic implementation of explicit instructional prompts and output delimiters to guide model behavior and facilitate parsing. Task 2 includes three subtasks: 1) Identify Creative Generation at Document Level, 2) Detect and Classify Information Distortion Errors in Simplified Sentences, and 3) Avoid Creative Generation and Perform Grounded Generation by Design. Four systems are proposed for Subtask 2.1, including a fine-tuned cross-encoder, large language models with majority voting, and two Random Forest classifiers with different textual input feature sets. Similarly, for Subtask 2.2, the proposed techniques consist of majority voting on different large language models' outputs, and fine-tuned bi-encoder and RoBERTa models. Finally, for Subtask 2.3, two approaches were considered for grounded generation, one using the cross-encoder classifier from Subtask 2.1, to check if the generated outputs are spurious, and the other using an instruction for the large language model to provide grounded outputs.

Keywords

Scientific Text Simplification, Large Language Models, Creative Generation Detection.

1. Introduction

The CLEF SimpleText [1, 2, 3] track aims to advance the natural language processing techniques for simplifying scientific text. Despite the advances in the applications of large language models (LLMs), challenges such as balancing simplicity with accuracy demand further exploration. To this end, the SimpleText 2025 lab continues the scientific text simplification task as Task 1 [4] by expanding the corpora and exploring simplification at both sentence and document levels. Task 2 [5], aims to identify and avoid hallucination by focusing on controlled creativity. Inspired by the substantial amount of spurious or over-generated content collected from earlier years, the first subtask is to identify creative generation. The second subtask then focuses on classifying the information distortion. The last subtask aims to avoid overly creative generation.

The Artificial Intelligence and Information Retrieval (AIIR) Lab from the University of Southern Maine participated in two tasks. For Task 1, we proposed 2 approaches for Subtask 1.1 and 4 for Subtask 1.2. For Task 2, we explored 4 approaches for Subtask 2.1, 7 for Subtask 2.2, and 2 for Subtask 2.3.

Our approach for Task 1 relies on two open-source LLMs: LLaMA and Mistral. For Task 2, in addition to LLMs, we relied on bi-encoder and cross-encoder Sentence-BERT [6]. We also studied the Random Forest classifier for Subtask 2.1, using two sets of features: textual similarity (e.g., ROUGE), and Abstract Meaning Representation (AMR)-based features. In the remainder of this paper, each section first introduces the task as a whole and details the subtasks along with their corresponding data. Then, we review our proposed models for each subtask. We conclude each section by reporting on the provided evaluation results for each of our proposed models.

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

[*] Corresponding author.

[†] These authors contributed equally.

Nicholas.Largey@Maine.edu (N. Largey); Deiby.Wu@Maine.edu (D. Wu); Behrooz.Mansouri@Maine.edu (B. Mansouri)
0009-0008-4004-2244 (N. Largey); 0009-0008-0764-1229 (D. Wu); 0000-0002-0400-9761 (B. Mansouri)

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

Table 1
Example Data-point (**pair_id**: ‘CD008131, **para_id**: ‘0‘ **sentence_id**: ‘5‘) from the Cochrane-auto dataset.

complex	No study reported on adverse effects.
label	rephrase
simple	None of the studies measured adverse effects.

2. Task 1: Text Simplification - Simplify Scientific Text

In this section, we will first introduce the two subtasks for text simplification [7], along with the dataset. Then, our proposed approaches and experimental results are provided.

2.1. Subtasks and Dataset

The CLEF 2025 SimpleText track has introduced the Cochrane-auto corpus [8],[1] expanding the Cochrane Database of Systematic Reviews[2] with scientific papers covering various biomedical topics. This corpus builds on methodologies from datasets like Wiki-auto and provides data at the document, paragraph, and sentence levels. It enables true document-level simplification through advanced techniques like sentence merging and reordering. Subtask 1.1 focuses on sentence-level simplification, while Subtask 1.2 focuses on document-level simplification.

Each data point provided in the Cochrane-auto corpus contains a pair of documents, with the ‘complex‘ document containing a technical abstract and the ‘simple‘ document holding its corresponding plain language version. To create Cochrane-auto, a neural alignment model is used to automatically pair corresponding sentences between the ‘complex‘ and ‘simple‘ texts, filtering out unaligned content to ensure the resulting plain language version retains the meaning of it’s technical pair. The dataset includes metadata for each document pair, such as a sentence-level simplification label (e.g., ‘rephrase’, ‘delete’, ‘split’) as shown in Table 1, paragraph and sentence identifiers, and positional information within the document. At the sentence level, the Cochrane-auto dataset offers a more granular view of the same document-level data, breaking down the document pairs into their constituent sentences, with each row containing a single complex sentence and its simplified counterpart. Each sentence pair is linked back to its original document via a pair ID.

2.2. Proposed Models

In this study, we investigate the efficacy of several contemporary LLMs in the simplification of scientific texts. Specifically, we fine-tuned three distinct quantized models: QWEN3-14b [9], LLaMA3.1 [10], and Mistral-7b [11]. LLaMA3.2-3b [12] was also submitted for Subtask 1.2; however, the base model was used to generate the submission, and no fine-tuning was performed. All models utilized were the quantized versions provided by Unsloth [3].

QWEN3-14b failed to produce adequate simplifications according to our evaluation criteria, so none of the results from training were submitted for evaluation. To establish a comparative baseline, the non-quantized versions of LLaMA3.1-8b and Mistral-7b were also evaluated on both subtasks after the official competition ended. The training parameters for each model version are listed in Table 2 (Appendix).

1. **Mistral-7b:** Three iterations of the ‘mistral-7b-instruct-v0.3-bnb-4bit‘ [4] model were fine-tuned. The initial two versions, Mistral-v1 and Mistral-v2, yielded inadequate results, suffering from issues with hallucination and overfitting. Mistral-v1 produced outputs that were excessively long, at times exceeding the length of the source text, and were replete with hallucinations. These

[1] https://github.com/JanB100/cochrane-auto
[2] https://www.cochranelibrary.com/cdsr/reviews
[3] https://huggingface.co/unsloth
[4] https://huggingface.co/unsloth/mistral-7b-instruct-v0.3-bnb-4bit

issues were attributed to a combination of improper instructional format and model overfitting. In an attempt to remedy these issues, a revised approach was undertaken for Mistral-v2. Despite these adjustments, the model continued to exhibit the previously observed problems, though to a lesser extent.

For Mistral-v3, the key modification was the inclusion of the explicit instruction, "Start the Response with 'Simplification:"'. This served as a necessary delimiter in the output, as the model consistently prepended the instructions, input, and response to its output. For submission, the resulting text was programmatically cleaned by splitting the output on this delimiter and stripping any extraneous newlines or whitespace.

2. **LLaMA3.1-8b and LLaMA3.2-3b:** A series of models from the LLaMA family were also trained and evaluated. Specifically, the LLaMA-3.1-8b [5] and LLaMA-3.2-3b [6] architectures were used. For the LLaMA-3.1-8b model, a total of three versions were fine-tuned, with two ultimately producing results that were submitted for evaluation. The initial iteration, LLaMA-3.1-8b-v1, produced output that contained a large amount of noise due to the omission of a simplification-delimiting phrase in the output, as shown to be necessary in the training of Mistral-v3. The extensive cleaning required to process these results was deemed outside the scope of the task; therefore, their outputs were not submitted.

   Subsequent iterations proved more successful. LLaMA-3.1-8b-v2 yielded viable results that were submitted for both Subtask 1.1 and Subtask 1.2. Notably, the model performed better on Subtask 1.2, demonstrating a 33% improvement in score over its performance on Subtask 1.1. A third and final model was trained for Subtask 1.2, LLaMA-3.1-8b-v3, which produced the best results. In addition to the fine-tuned models, the base LLaMA-3.2-3b model was also evaluated, with its results being submitted for Subtask 1.2.

2.3. Evaluation Results

This section presents the performance of our proposed models on the CLEF 2025 SimpleText track's Subtask 1.1 (sentence-level simplification) and Subtask 1.2 (document-level simplification). We evaluate the models using several standard metrics: SARI, BLEU, Flesch-Kincaid Grade Level (FKGL), Levenshtein Similarity (Lev. Sim.), the percentage of Exact Copies, and Lexical Complexity.

Subtask 1.1: Sentence-Level Simplification. For the sentence-level simplification task, both the fine-tuned Mistral-7b and LLaMA-3.1-8b models were evaluated. As shown in Table 2, the Mistral-7b model achieved a higher SARI score of 36.08, indicating a better balance of additions, deletions, and rephrasing compared to LLaMA-3.1-8b-v2, which scored 31.27. However, LLaMA-3.1-8b-v2 produced outputs with a higher BLEU score (19.59 vs. 18.41) and a lower Flesch-Kincaid Grade Level (11.44 vs. 12.78), suggesting its generations were closer to the reference translations and written at a slightly lower grade level.

The Levenshtein Similarity for LLaMA-3.1-8b-v2 was higher (0.83) than for Mistral-7b (0.76), implying that the outputs from the LLaMA model were structurally more similar to the source text. Neither of the models produced exact copies of the input, even though some of the sentences to be simplified were short and reasonably simple. Similar entries in the training data would have been labeled as 'ignore', implying no need for further simplification. Finally, the lexical complexity for both models was comparable, with Mistral-7b at 8.81 and LLaMA-3.1-8b-v2 at 8.83.

Subtask 1.2: Document-Level Simplification. In the document-level simplification task, the fine-tuned Mistral-7b model again demonstrated the strongest performance, achieving the highest SARI score of 42.4, as detailed in Table 2.3. This was followed by the fine-tuned LLaMA-3.1-8b-v3, which obtained a SARI score of 41.07. An interesting finding was that LLaMA-3.1-8b-v2 and the base LLaMA-3.2-3b model both yielded identical SARI scores of 39.14.

Mistral-7b also led in BLEU score (12.97) and produced the simplest text in terms of grade level, with an FKGL of 8.82. However, LLaMA-3.1-8b-v3 had the lowest Levenshtein Similarity (0.43), indicating

[5] https://huggingface.co/unsloth/Meta-Llama-3.1-8B-Instruct-bnb-4bit

[6] https://huggingface.co/unsloth/Llama-3.2-3B-Instruct-bnb-4bit

Table 2
Evaluation results for AIIR Lab models in the SimpleText 2025 Task 1.1: Sentence-Level Simplification

Model	SARI	BLEU	FKGL	Lev. Sim.	Exact Copies	Lexical Comp
Mistral-7b	36.08	18.41	12.78	0.76	0.00	8.81
LLaMA-3.1-8b-v2	31.27	19.59	11.44	0.83	0.00	8.83

Table 3
Evaluation results for AIIR Lab models in the SimpleText 2025 Task 1.2: Document-Level Simplification

Model	SARI	BLEU	FKGL	Lev. Sim.	Exact Copies	Lexical Comp
Mistral-7b	42.40	12.97	8.82	0.52	0.00	8.48
LLaMA-3.1-8b-v3	41.07	8.61	9.22	0.43	0.00	8.44
LLaMA-3.1-8b-v2	39.14	5.62	8.88	0.35	0.00	8.43
LLaMA-3.2-3b	39.14	5.62	8.88	0.35	0.00	8.43

Table 4
Evaluation Results for Post-Competition Base Model Submissions for Task 1 in the SimpleText 2025 Lab.

Model	Subtask 1.1 SARI	Subtask 1.2 SARI
LlaMA-3.1-8b	42.05	42.46
Mistral-7b	42.44	42.57

Table 5
Task 2.1 example data (gen_id: '35623979//T13.1_2139304285//2'). The input system will take in an input text and should decide if it is spurious or not. The systems can use the abstract source for classification.

Abstract Source	Several fairness index measurements have been proposed in the technical literature. Instantaneous fairness property has not been captured.
Spurious Generation	Fairness is not ensured.
Not Spurious Generation	Instantaneous fairness property has not been captured.

more alterations from the source document. As with the sentence-level task, none of the models produced exact copies of the input. The lexical complexity scores were very similar across all evaluated models for this subtask, ranging from 8.43 to 8.48.

Post-Competition Analysis. In the post-competition phase, the non-quantized base versions of LLaMA-3.1-8b and Mistral-7b were evaluated on both subtasks. The results, presented in Table 4, show that these base models achieved higher SARI scores than their fine-tuned counterparts in most cases. For Subtask 1.1, the base LLaMA-3.1-8b and Mistral-7b models obtained SARI scores of 42.05 and 42.44, respectively. In Subtask 1.2, the base models also performed well, with LLaMA-3.1-8b scoring 42.46 and Mistral-7b achieving a score of 42.57. These results suggest that for this particular dataset and task, the base models possessed an existing strong capability for text simplification that was not consistently improved upon through our fine-tuning process.

3. Task 2: Controlled Creativity - Identify and Avoid Hallucination

Task 2 focuses on recognizing and assessing instances where creativity leads to information distortion during text simplification [13]. This section first describes the subtasks and the dataset. It then reviews the proposed approaches for each subtask and provides the results.

3.1. Subtasks and Dataset

For Task 2, three subtasks are considered: 1) identify creative generation at the document level, 2) detect and classify information distortion errors in simplified sentences, and 3) avoid creative generation and

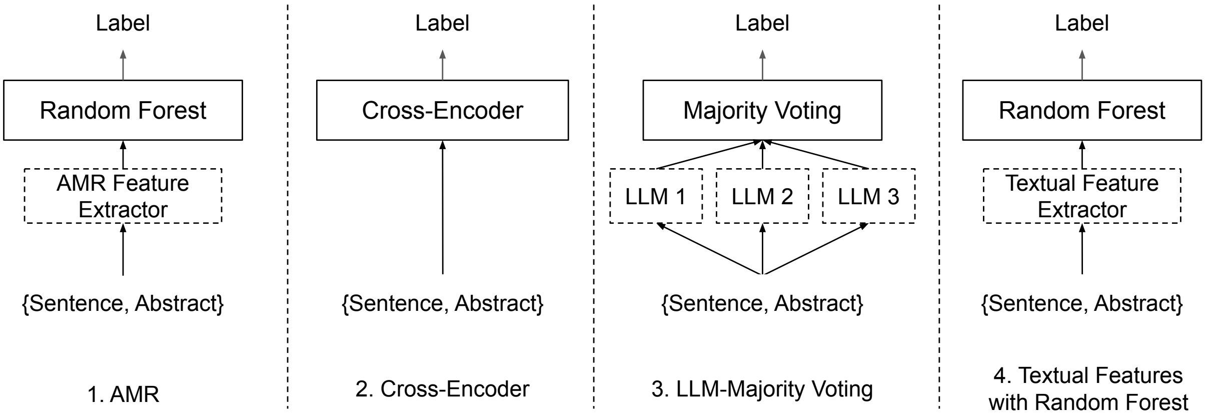

Figure 1: Overview of the four proposed systems by AIIR Lab team for Subtask 2.1.

perform grounded generation by design.

Subtask 2.1 involves developing systems that can detect creative generation at the abstract and document level. These systems should identify sentences fully grounded in the source text, and can either use or ignore the abstract source. Table 5 shows two examples of generated text for the same abstract from the training data, with one annotated as spurious and the other as not spurious.

Subtask 2.2 focuses on detecting information distortion in simplified sentences and classifying the types of errors. The classes are based on four broad categories: Fluency, Alignment, Information, and Simplification [14]. These categories are broken down into 14 subcategories:

- **Fluency**: Random generation, syntax error, contradiction, simple punctuation/grammar errors, and redundancy
- **Alignment**: Format and prompt misalignment
- **Information**: Factuality hallucination, faithfulness hallucination, and topic shift
- **Simplification**: Overgeneralization, overspecification of concepts, loss of informative content, and out-of-scope generation

Systems for this subtask take in both the source sentence and the simplified sentence, and they should predict whether the simplified sentence contains any of these distortions relative to its source sentence. As both Subtask 2.1 and 2.2 are classification tasks, standard metrics such as precision, recall, and F1 are used for evaluation.

Finally, Subtask 2.3 is related to Task 1, where systems should provide a simplified version of input text that is grounded and avoids over-generation. The data and evaluation metrics for this subtask remain the same as Task 1.

3.2. Proposed Models

Our team submitted 4 systems for Subtask 2.1, 7 for Subtask 2.2, and 2 for Subtask 2.3. Here we describe our proposed models for each task.

Subtask 2.1. We consider four different approaches for Subtask 2.1. Figure 1 shows the overview of our proposed approaches. Here we describe each one:

1. **AMR:** Abstract Meaning Representation (AMR) is a directed graph capturing the semantics of the text. AMRs have been widely used in several language processing tasks such as summarization, generative data augmentation, and paraphrase detection [15]. The overview of our proposed approach is shown in Figure 2. Similar to their applications for paraphrase detection, for this approach, we parsed each sentence in the abstract to its corresponding AMR using the model ‘parse_xfm_bart_large’ parser [16], generating $Abs.AMR\ (S_i)$ where S_i corresponds to sentence i in the abstract. Using the same model, the AMR for the simplified sentence, $SMP.AMR$, is

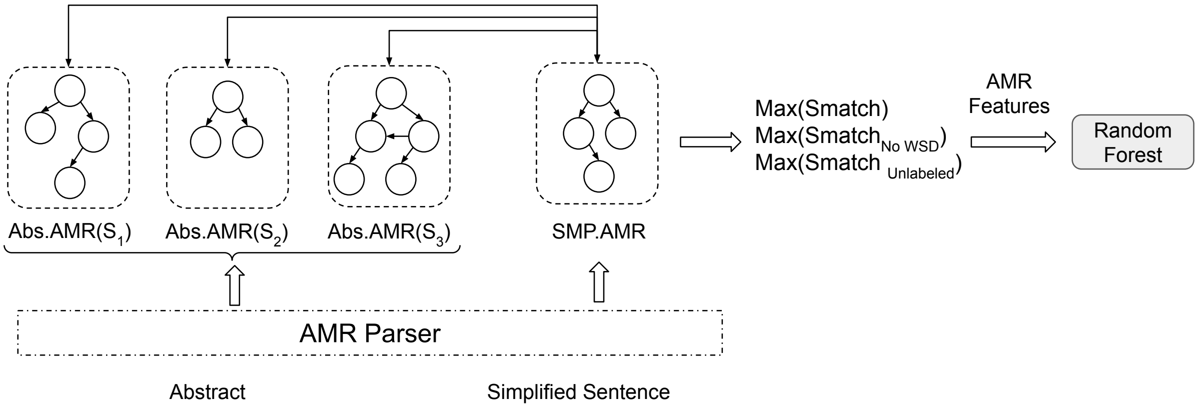

Figure 2: Proposed approach for spurious generation detection with abstract meaning representation (AMR). First, AMRs for each sentence in the abstract and simplified sentence are generated. Then, the simplified sentence AMR (SMP.AMR) is compared against the AMR of each sentence in the abstract, and maximum SMATCH scores are considered as input features for the random forest classifier.

generated and compared against each AMR in the abstract. For similarity features, we have considered the SMATCH score [17], along with two of its variants: Unlabeled and No Word Sense Disambiguation (No WSD). SMATCH is the standard metric for evaluating Abstract Meaning Representation (AMR) parsers by calculating the F1-score based on the overlap of concept-relation triples between a predicted graph and a gold standard. The Unlabeled variant of SMATCH evaluates only the graph's structure by ignoring the specific labels on the edges (e.g., :ARG0), focusing solely on whether the concepts are correctly connected. No WSD is another SMATCH variation that assesses parsing accuracy without penalizing incorrect word senses, meaning a concept like run-01 is considered a match for run-02 since the base word is the same. For classification, we considered these three AMR-based metrics, and used the highest SMATCH score between $SMP.AMR$ and $Abs.AMR\ (S_i)$ for each sentence i in the abstract as shown in Equation 1. These three features were then used to train a Random Forest classifier with the training data.

$$\text{SMATCH\_Feature}(\text{Abstract}, \text{Simplified}) = \max_i \text{SMATCH}(\text{SMP.AMR}, \text{Abs.AMR}(S_i)) \quad (1)$$

2. **Cross-Encoder:** Cross-encoders have been successfully used in our approaches in previous labs, showing high effectiveness for scientific text processing tasks [18, 19]. Therefore, in this approach, we fine-tuned a cross-encoder model, 'ms-marco-MiniLM-L-6-v2', using the available training data by passing the abstract source and the simplified sentence to the model and predicting whether the sentence is spurious. We split the training data into 90-10% train and validation sets, and the model was trained for 20 epochs, with a batch size of 4, and the best model on the validation set was used for the 2025 test set.
3. **LLM-Majority Voting:** In this approach, we prompted three large language models, Llama-3-8B-Instruct, Qwen2.5-7B-Instruct, and Mistral-7B-Instruct-v0.3 to decide whether a simplified sentence is spurious based on its abstract source. All three LLMs were used with few-shot prompting with positive and negative samples from the training data. The prompt used for LLMs is the same as:

> You are a careful evaluator of simplified scientific text. Given a sentence generated from a scientific abstract, your job is to decide whether the sentence is spurious. A sentence is considered spurious if it includes: Information distortion: Factual inaccuracy, misrepresentation, or incorrect attribution of information from the abstract. Creative generation: Introduction of information, ideas, or claims that are not supported or implied by the abstract. Answer only with Yes or No.

Both the sentence and the abstract were passed to LLMs. After the decision is made by each LLM, we use a majority voting technique to decide the final label for each sentence.

4. **Textual Features with Random Forest:** For this technique, we considered textual features between the simplified sentences and their corresponding abstracts. To this end, we measured the following metrics between a simplified sentence and each sentence in its abstract (with the max score as the input feature for the classifier):
   a) *Google BLEU [20]:* This metric adapts the standard BLEU for single sentences by calculating both precision (proportion of n-grams in the generated output that match the reference) and recall (proportion of reference n-grams matched by the output), and then reporting the minimum of the two.
   b) *METEOR [21]:* Metric for Evaluation of Translation with Explicit ORdering aligns candidate and reference sentences using exact, stem, and synonym matches. It computes a harmonic mean of precision and recall, then penalizes fragmentation, resulting in high correlation with human assessments at the sentence level.
   c) *Exact Match:* This is a binary metric indicating whether the simplified sentence exactly matches (word-for-word) any sentence in the abstract.
   d) *MAUVE [22]:* Compares the distributions of generated and human-written text by computing the area under a divergence curve in embedding space.
   e) *Part of Speech:* Measures syntactic correspondence by tagging both sentences (e.g., noun, verb sequences) and comparing the overlap or alignment of these tag sequences.
   f) *ROUGE [23]:* Measures overlap between candidate and reference via: ROUGE-1: unigram (word) recall, and ROUGE-L: longest common subsequence, emphasizing how much of the reference content is captured.
   g) *BERTScore [24]:* Utilizes contextual token embeddings (typically from BERT) to compute cosine similarity between candidate and reference tokens.

Subtask 2.2. We consider three different approaches for Subtask 2.2: a fine-tuned Roberta model, a fine-tuned bi-encoder, and a majority-voting ensemble using three instruction-tuned large language models: LLaMA, Mistral, and OpenChat [25].

1. **RoBERTa:** A RoBERTa [26] base model was fine-tuned as a multi-classifier to predict a single distortion type for each source-simplified sentence pair. The model was trained on labeled source data using cross-entropy loss across 15 classes. Training was performed over five epochs with a batch size of 8, and a learning rate of 1e−5. During inference, this setup was extended to support multi-label predictions. A top-scoring label was first selected, and then it was also included with any other distortion labels that had a probability that exceeded a confidence threshold above 0.9. "No error" was the only label assigned to a sentence pair if it was the top prediction.
2. **LLM-Majority Voting:** Three large language models (LLaMA, Mistral, and Openchat) were utilized to classify simplified sentences according to the distortion types via few-shot prompting. Each model was used independently for each distortion label (excluding "No error"). Prompts included a detailed system message describing the distortion type and emphasizing conservative behavior, followed by three examples, one positive example with a justification for the given answer, and two negative examples with explanations. For a given source-simplified sentence pair, each distortion type was evaluated separately, and the model answered with either "Yes" or "No". A distortion label was given if at least two of the three models agreed with a "Yes". In the case that the sentence pair did not get a label, then it would be defaulted to "No error".

   > System Prompt: You are a binary classifier for [Distortion Type]. [Brief definition of distortion]. Most simplified sentences do not contain this error. Only answer "Yes" if the error is clearly present. Respond only with "Yes" or "No"

   Between the system prompt and the final query, the model was shown three labeled few-shot examples, one positive and two negatives, to demonstrate the expected reasoning behavior.

> User Prompt: Source sentence: [Test Source] Simplified sentence: [Test Simplified]

In addition to majority voting, we also submitted each model's output as separate results.

3. **Bi-encoder classifier:** In this approach, we fine-tune a pretrained MPNet bi-encoder ('paraphrase-mpnet-base-v2'). Each example is presented as a tokenized pair $\langle$simplified, source$\rangle$ (truncated/padded to 512 tokens), and the model's new linear head outputs one sigmoid logit per error category. Training minimizes a summed binary cross-entropy loss, allowing multiple errors to be signaled simultaneously.
   Optimization uses AdamW (learning rate $= 2 \times 10^{-5}$, weight decay $= 0.01$) with batch size $= 32$ over 50 epochs, and the checkpoint with highest micro-F1 on a 10% held-out split is retained. At inference, sigmoid outputs are thresholded at 0.5 to produce binary labels, with a "No error" flag when none are positive. Within the same fine-tuning setting, we also considered the 'all-mpnet-base-v2' (our mpnet system) as another approach.

Subtask 2.3. We considered two approaches for this subtask: one with a prompting technique, and the other using the cross-encoder model developed for Subtask 2.1.

1. **LLaMA Grounded**: In this approach, we used 'LLaMA-3-8B-Instruct' with zero-shot prompting and a system message. For the system message, we specified that the simplified sentences should be grounded, with the following prompt:

   > Your task is to simplify scientific sentences into an easy-to-read sentence while keeping the main content and removing extra data. The simplified sentence should not be spurious and must be grounded in the paragraph from which the sentence is extracted.

   Then the sentence and the paragraph are passed to the model for simplification.
2. **LLaMA Cross-encoder**: This approach relied on the fine-tuned cross-encoder model from Subtask 2.1, where the model decides if the generated simplified text is spurious or not. To simplify the sentences with LLaMA, we use the same prompt as the previous approach for this subtask. However, we then verify the output by passing it to the cross-encoder and checking if it is spurious. With a threshold of 5 times, if LLaMA generates spurious simplification, the output along with the original data is passed again to LLaMA with a similar prompt, but adding 'This simplified result is spurious and not grounded.' at the beginning.

We include our model with no grounding, 'LLaMA3.1-8b-v2', as the baseline for comparison.

3.3. Evaluation Results

Based on the results presented in Table 6, the performance of the AIIR Lab models for detecting spurious generation in Task 2.1 shows variation across different approaches. The CrossEncoder model achieved high performance with an accuracy of 0.98, precision of 0.99, recall of 0.99, and an F1-score of 0.99, demonstrating its effectiveness in identifying spurious generation at the document level. The RandomForest model using textual features also performed well, achieving an accuracy of 0.95 and perfect recall of 1.00, though with slightly lower precision at 0.95, resulting in an F1-score of 0.97. In contrast, the LLMs approach using majority voting performed poorly, with an accuracy of only 0.10 and an F1-score of 0.00, indicating that the three large language models (Llama-3-8B-Instruct, Qwen2.5-7B-Instruct, and Mistral-7B-Instruct-v0.3) struggled with this task despite few-shot prompting. The AMR-based approach could not be evaluated due to technical issues with the online tool, preventing a complete comparison of all proposed methods.

Table 7 shows the performance of the AIIR Lab models for detecting and classifying information distortion errors in Subtask 2.2. The bi-encoder approaches using MPNet models achieved the best overall performance, with the paraphrase-mpnet model leading in most categories, particularly excelling in "No Error" classification with an F1-score of 0.755 and moderate performance across Fluency (A), Alignment (B), Information (C), and Simplification (D) categories with F1-scores ranging from 0.136 to 0.258. The individual large language models and their majority voting ensemble showed considerably

Table 6
Results for AIIR Lab models in the SimpleText 2025 Task 2.1: Detecting over generation with sources.

Method	count	Acc.	Prec	Rec	F1	AUROC	AUPRC
Cross-Encoder	3,379	0.98	0.99	0.99	0.99	0.95	0.99
RandomForest (w. Textual Features)	3,379	0.95	0.95	1.00	0.97	0.77	0.95
LLM-Majority Voting	3,379	0.10	0.00	0.00	0.00	0.50	0.90

Table 7
Results for AIIR Lab models in the SimpleText 2025 Task 2.2. Performance by Error Categories for No error, Fluency (A), Alignment(B), Information (C), and Simplification (D) categories, with F1 and AUC-PR.

Method	No Error		A		B		C		D	
	F_1	AUC	F_1	AUC	F_1	AUC	F_1	AUC	F_1	AUC
paraphrase_mpnet	0.755	0.567	0.255	0.154	0.258	0.113	0.136	0.084	0.147	0.168
mpnet	0.744	0.557	0.255	0.156	0.218	0.099	0.150	0.091	0.147	0.167
OpenChat	0.640	0.421	0.154	0.070	0.141	0.061	0.144	0.080	0.222	0.156
MajorityVoting	0.633	0.415	0.156	0.071	0.110	0.045	0.170	0.088	0.239	0.160
Mistral	0.563	0.357	0.158	0.069	0.104	0.040	0.116	0.070	0.176	0.144
LLaMA	0.544	-	0.135	-	0.070	-	0.171	-	0.300	-
RoBERTa	0.404	-	0.126	-	0.107	-	0.164	-	0.237	-

Table 8
Results for AIIR Lab models in the SimpleText 2025 Task 2.3: Avoiding creative generation by design. (Com. Ratio: Compression Ratio, Sen. Splits: Sentence Splits, Lev. Sim.: Levenshtein Similarity, (Add., Del.) Pro.: (Additions, Deletions) Proportion.

Method	count	SARI	BLEU	FKGL	Com. Ratio	Sen. Splits	Lev. Sim.	(Add., Del.) Pro.
LLaMA Grounded	37	43.63	17.92	11.02	0.63	0.96	0.61	(0.13,0.53)
LLaMA Cross-encoder	37	43.24	17.48	11.16	0.63	0.96	0.61	(0.13,0.53)
LLaMA3.1-8b-v2	37	31.27	19.59	11.44	0.85	1.09	0.83	(0.09,0.25)

lower performance, with OpenChat achieving the highest F1-score of 0.640 for "No Error" detection among the LLM-based approaches, while the majority voting strategy performed slightly worse at 0.633. Notably, all models struggled with the specialized error categories (A, B, C, D), achieving F1-scores below 0.3 across all distortion types, indicating the challenging nature of fine-grained error classification in simplified text. The RoBERTa model showed the poorest performance overall, particularly in "No Error" detection with an F1-score of only 0.404, suggesting that the multi-class to multi-label adaptation approach was less effective than the bi-encoder architectures for this complex classification task.

Finally, our results for Subtask 2.3 are shown in Table 8. Both grounded approaches outperformed the baseline LLaMA3.1-8b model in terms of simplification quality, with the LLaMA Grounded method achieving a SARI score of 43.63 compared to the baseline's 31.27, indicating substantially better simplification performance. The LLaMA Cross-encoder approach performed similarly with a SARI score of 43.24, suggesting that both grounding strategies were equally effective. Notably, the grounded models achieved much higher compression ratios (0.63 vs. 0.85) and deletion proportions (0.53 vs. 0.25), indicating they performed more aggressive simplification while maintaining quality. The lower Levenshtein similarity scores (0.61 vs. 0.83) for the grounded approaches further confirm that they made more substantial modifications to the original text rather than producing conservative, minimally-changed outputs. While the baseline model achieved slightly higher BLEU scores (19.59 vs. $\sim$17.7), this likely reflects its tendency to make fewer changes to the source text, which is less desirable for effective simplification. The comparable Flesch-Kincaid Grade Level (FKGL) scores across all methods ($\sim$11.0-11.4) suggest that readability was maintained regardless of the approach used.

4. Conclusion

This paper presented the Artificial Intelligence and Information Retrieval (AIIR) Lab approaches for the CLEF SimpleText 2025 track. Our team participated in Tasks 1 and 2. For Task 1, we studied Unsloth's quantized versions of Mistral and LLaMA for scientific text simplification, participating in both sentence- and document-level simplification subtasks. Our fine-tuned Mistral-7b achieved the highest SARI scores of 36.08 and 42.4 for sentence-level and document-level tasks, respectively, outperforming LLaMA variants across most metrics. Notably, our post-competition analysis revealed that base models without fine-tuning achieved superior SARI scores (42.44 and 42.57 for Mistral-7b), suggesting that for biomedical text simplification on the Cochrane-auto dataset, the inherent capabilities of these large language models may be sufficient without task-specific fine-tuning.

Task 2 comprised three subtasks focused on controlled creativity and hallucination detection. For Subtask 2.1 (spurious generation detection), we explored four approaches, including AMR-based similarity, a cross-encoder model, LLM majority voting, and textual features with random forest, with the cross-encoder achieving the best performance ($F1 = 0.99$). For Subtask 2.2 (information distortion classification), we implemented RoBERTa, bi-encoder, and LLM-based approaches, where bi-encoder models using MPNet showed the best performance across error categories, though all methods struggled with fine-grained distortion classification. For Subtask 2.3 (grounded simplification), we developed two grounding strategies using LLaMA with explicit grounding prompts and cross-encoder verification, both outperforming the baseline in simplification quality.

For future work, we plan to develop an integrated pipeline that combines our cross-encoder spurious detection model from Subtask 2.1 with our simplification approaches from Task 1 to create a robust grounded simplification system. This would involve implementing an iterative refinement process where initial simplifications are automatically evaluated for spuriousness and information distortion, then regenerated with targeted feedback until satisfactory grounded outputs are achieved. Additionally, we aim to improve fine-grained error classification by exploring more sophisticated multi-label learning approaches and investigating domain-specific training strategies for better performance on the specialized distortion categories identified in Subtask 2.2.

Declaration on Generative AI

During the preparation of this work, the author(s) used Gemini Pro-2 in order to: Grammar and spelling check. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] 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.

[2] L. Ermakova, H. Azarbonyad, J. Bakker, B. Vendeville, J. Kamps, Overview of the CLEF 2025 SimpleText track: Simplify scientific texts (and nothing more), in: [1], 2025.

[3] 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.

[4] J. Bakker, B. Vendeville, L. Ermakova, J. Kamps, Overview of the CLEF 2025 SimpleText Task 1: Simplify Scientific Text, in: [3], 2025.

[5] B. Vendeville, J. Bakker, L. Ermakova, J. Kamps, Overview of the CLEF 2025 SimpleText Task 2: Identify and Avoid Hallucination, in: [3], 2025.

[6] 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. URL: https://aclanthology.org/D19-1410/. doi:10.18653/v1/D19-1410.

[7] J. Bakker, et al., Overview of the CLEF 2025 SimpleText Task 1: Simplify Scientific Text, in: G. Faggioli, et al. (Eds.), Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2025), CEUR Workshop Proceedings, CEUR-WS.org, 2025. http://ceur-ws.org.

[8] 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. URL: https://aclanthology.org/2024.tsar-1.5/. doi:10.18653/v1/2024.tsar-1.5.

[9] A. e. a. Yang, Qwen3 technical report, 2025. URL: https://arxiv.org/abs/2505.09388. arXiv:2505.09388.

[10] A. e. a. Grattafiori, The llama 3 herd of models, 2024. URL: https://arxiv.org/abs/2407.21783. arXiv:2407.21783.

[11] J. A. et al., Mistral 7b, 2023. URL: https://arxiv.org/abs/2310.06825. arXiv:2310.06825.

[12] ai.meta.com., Llama 3.2: Revolutionizing edge ai and vision with open, customizable models., 2024. URL: https://ai.meta.com/blog/llama-3-2-connect-2024-vision-edge-mobile-devices/.

[13] B. Vendeville, et al., Overview of the CLEF 2025 SimpleText Task 2: Identify and Avoid Hallucination, in: G. Faggioli, et al. (Eds.), Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2025), CEUR Workshop Proceedings, CEUR-WS.org, 2025. http://ceur-ws.org.

[14] B. Vendeville, L. Ermakova, P. De Loor, Resource for error analysis in text simplification: New taxonomy and test collection, arXiv preprint arXiv:2505.16392 (2025).

[15] B. Mansouri, Survey of abstract meaning representation: Then, now, future, arXiv preprint arXiv:2505.03229 (2025).

[16] X. Bai, Y. Chen, Y. Zhang, Graph pre-training for AMR parsing and generation, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 6001–6015. URL: https://aclanthology.org/2022.acl-long.415.

[17] S. Cai, K. Knight, Smatch: an evaluation metric for semantic feature structures, in: H. Schuetze, P. Fung, M. Poesio (Eds.), Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Association for Computational Linguistics, Sofia, Bulgaria, 2013, pp. 748–752. URL: https://aclanthology.org/P13-2131/.

[18] N. Largey, R. Maarefdoust, S. Durgin, B. Mansouri, Aiir lab systems for clef 2024 simpletext: large language models for text simplification, in: Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2024), 2024, pp. 3261–3273.

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

[20] Y. W. et al., Google's neural machine translation system: Bridging the gap between human and machine translation, 2016. arXiv:1609.08144.

[21] S. Banerjee, A. Lavie, METEOR: An automatic metric for MT evaluation with improved correlation with human judgments, in: J. Goldstein, A. Lavie, C.-Y. Lin, C. Voss (Eds.), Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, Association for Computational Linguistics, Ann Arbor, Michigan, 2005, pp. 65–72. URL: https://aclanthology.org/W05-0909/.

[22] K. Pillutla, S. Swayamdipta, R. Zellers, J. Thickstun, S. Welleck, Y. Choi, Z. Harchaoui, Mauve: Measuring the gap between neural text and human text using divergence frontiers, in: M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, J. W. Vaughan (Eds.), Advances in Neural Information Processing Systems, volume 34, Curran Associates, Inc., 2021, pp. 4816–4828. URL: https://proceedings.neurips.cc/paper_files/paper/2021/file/260c2432a0eecc28ce03c10dadc078a4-Paper.pdf.

[23] C.-Y. Lin, ROUGE: A package for automatic evaluation of summaries, in: Text Summarization Branches Out, Association for Computational Linguistics, Barcelona, Spain, 2004, pp. 74–81. URL:

https://aclanthology.org/W04-1013/.

[24] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, Y. Artzi, Bertscore: Evaluating text generation with bert, arXiv preprint arXiv:1904.09675 (2019).

[25] G. Wang, S. Cheng, X. Zhan, X. Li, S. Song, Y. Liu, Openchat: Advancing open-source language models with mixed-quality data, 2024. URL: https://arxiv.org/abs/2309.11235. arXiv:2309.11235.

[26] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, 2019. URL: https://arxiv.org/abs/1907.11692. arXiv:1907.11692.

A. Task 1

1. **Instructions Used:**
   a) Write a response that has optimal BLEU, SARI and ROUGE scores and an FKGL score as close to 9.0 as possible. Provide separate simplifications for both the full texts and each sentence in your output. Keep the sentence count to the same or fewer number of sentences as the input text.
   b) Simplify the following input focusing on getting an FKGL score as close to 9.0 as possible while avoiding hallucinations. Maximize the use of simple words and short sentences but include key words from the original text. Start the Response with 'Simplification:' but do not include anything in the Response other than the generated simplification.
   c) Simplify the following input for a student in 9th grade. Maximize the use of simple words and short sentences but include key words from the original text.
   d) Simplify the following input text for a high school student. Maximize the use of simple words and short sentences but include key words from the original text. Start the Response with 'Simplification:' but do not include any other text in the Response other than the generated simplification.
   e) Simplify the following input sentence focusing on optimal BLEU, SARI and ROUGE scores with an FKGL score as close to 9.0 as possible while avoiding hallucinations. Maximize the use of simple words and short sentences but include key words from the original text.
   f) Write a response that has optimal BLEU, SARI and ROUGE scores and an FKGL score as close to 9.0 as possible.

g) **Labels and Corresponding Instruction:**

Label	Instruction
rephrase	Write a response that has optimal BLEU, SARI and ROUGE scores and an FKGL score as close to 9.0 as possible.
delete	Write a response that is an empty ‘list‘
split	Write a response that splits the input into more than one sentence which provides optimal BLEU, SARI and ROUGE scores and an FKGL score as close to 9.0 as possible.
ignore	Write a response that is a copy of the input.
merge	Write a response that combines the current input with the following input which provides optimal BLEU, SARI and ROUGE scores and an FKGL score as close to 9.0 as possible.
none	Write a response that is an empty‘list‘

2. **Training parameters for proposed systems by AIIR Lab team for Task 1.**

Model Version	Instruction	Epochs	Learning Rate
QWEN-v1	f	100	0.0002
QWEN-v2	b	100	0.0002
QWEN-v3	g	60	0.0002
Mistral-v1	a	100	0.00001
Mistral-v2	e	37	0.00001
Mistral-v3	d	20	0.0002
LLaMA-3.1-8b-v1	c	25	0.00002
LLaMA-3.1-8b-v2	d	15	0.00002