UBONLP Report on the SimpleText Track at CLEF 2024

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

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

Abstract

This article presents the UBONLP team's participation at the SimpleText lab of CLEF 2024 in tasks 1 "Selecting passages to include in a simplified summary", 2 "Difficult concept identification and explanation", and 3 "Given a query, simplify passages from scientific abstracts". Our goal is to use recent advances in natural language processing to help the public better understand scientific information. In Task 1 we show a method using TF_IDF and a neural reranker to retrieve scientific texts. In Task 2 we use a non fine-tuned Phi3 mini to extract complicated terms. Task 3 we use a LLM pipeline with separate syntactic and lexical simplifications.

Keywords

LLM, Ranking, information retrieval, Neural reranking, Term difficulty, Automatic text simplification, Science popularization, Lexical simplification, Syntactic simplification

1. Introduction

The internet has democratized access to scientific research. However, understanding science communication still proves to be a problem due to the complexity of scientific texts. Text simplification is a way to solve this issue. The CLEF 2024 SimpleText lab [1] aims to study how advances in natural language processing can be applied to this goal. The lab is divided into four tasks:

- **Task 1:** What is in (or out)? Selecting passages to include in a simplified summary.
- **Task 2:** What is unclear? Difficult concept identification and explanation (definitions, abbreviation deciphering, context, applications, …) with three subtasks:
  - **Subtask 2.1:** To predict what are the terms in a passage of a document and their difficulty as e, m or d (Easy/Medium/Difficult)
  - **Subtask 2.2:** To generate a definition and an explanation only for the difficult terms
  - **Subtask 2.3:** To retrieve the provided definitions of the difficult terms and rank them in the "correct" order: manual (2, ground truth), generated positive 1 (1, correct definitions), generated positive 2 (1, correct definitions), generated negative 1 (0, incorrect definitions), generated negative 2 (0, incorrect definitions).
- **Task 3:** Rewrite this! Given a query, simplify passages from scientific abstracts. Two subtasks are considered:
  - **Subtask 3.1:** Sentence-level simplification

CLEF 2024: Conference and Labs of the Evaluation Forum, September 09–12, 2024, Grenoble, France
benjamin.vendeville@univ-brest.fr (B. Vendeville); liana.ermakova@univ-brest.fr (L. Ermakova); deloor@enib.fr (P. D. Loor)
0009-0003-5298-147X (B. Vendeville); 0000-0002-7598-7474 (L. Ermakova); 0000-0002-5415-5505 (P. D. Loor)

- **Subtask 3.2:** Abstract-level simplification
- **Task 4:** SOTA: Tracking the state-of-the-art in scholarly publications.

We participated in Tasks 1, 2 (subtask 1), and 3 (subtasks 1 and 2). For Task 1 we use Pyterrier[1] [2] to index documents, TF_IDF to rank them, and MonoT5 [3] to rerank the top results. For Task 2 we used Phi3 mini [4], a LLM, to extract and score complex terms in a one-shot prompt context [5], using no fine-tuning. For Task 3 we used Phi3 mini in a pipeline that separated syntactic and lexical simplifications. Again, the model was not fine-tuned and used a one-shot prompt. We further tested this method on data.

We will first present our method and results for the Task 1. Then we will present the method, prompts, and results for Task 2. In chapter 4 we will present the method for Task 3 and study the results in details. We will see that our method for Task 3 can produce some results when separating lexical and syntactic simplification.

2. Task 1: Passage Selection for a Simplified Summary

In this task, participants were provided with a dataset of abstracts with their metadata (author names, title, year of publication…). Participants are also provided with a set of references for training, and a test dataset of queries. Task 1 consists of, for each query, retrieving the 100 most relevant documents.

For Task 1, we first used PyTerrier [1] [2], a framework for creating information retrieval pipelines, to index all documents. We wanted to use an LLM to rank abstracts, but the number of initial documents was too great to practically run any model. Instead, we used TF_IDF to first rank all documents based on their abstracts and titles and kept the 4000 most relevant documents. Then we could use the MonoT5 reranker [3, 6] provided by PyTerrier to rerank all extracted documents and kept the 100 best.

2.1. Metrics

To measure the quality of simplifications, we will use the following metrics as provided by the EASSE library [7]:

- **MRR:** The Mean Reciprocal Rank is a metric used to evaluate the performance of search engines, recommendation systems, and other information retrieval systems. It measures the average rank at which the first relevant item is found in the search results. The results vary from 0 to 1, with 1 being a perfect score, where relevant items appear at the top position for all queries.
- **Prec10:** Precision 10 is a metric used to evaluate the performance of information retrieval systems. It measures the proportion of relevant items among the top 10 results returned by the system. The value ranges from 0 to 1, with 1 being a perfect score where all of the top 10 results are relevant and 0 meaning no relevant results among the top 10.
- **Prec20:** Precision 20 is a metric used to evaluate the performance of information retrieval systems. Like Precision10, it measures the proportion of relevant items, but focusing instead on the top 20 results returned by the system. The value ranges from 0 to 1, with 1 being a perfect score where all of the top 20 results are relevant and 0 meaning no relevant results among the top 20.

[1] https://pyterrier.readthedocs.io/en/latest/

- **NDCG10:** The Normalized Discounted Cumulative Gain 10 metric is based on a normalization of the Discounted Cumulative Gain, which gives a score based on the relevance of every result in the top 10, weighted by their position. The values range from 0 to 1 with 1 being a perfect score where the most relevant results appear at the top of the top 10 results, and 0 meaning no relevant results among the top 10.
- **NDCG20:** The metric is the same as NDCG10 but focusing on the top 20. The values range from 0 to 1 with 1 being a perfect score where the most relevant results appear at the top of the top 20 results, and 0 meaning no relevant results among the top 20.
- **Bpref:** The Binary Preference is a metric used to evaluate the performance of information retrieval systems. It is designed to handle situations where not all documents have been judged for relevance. It measures the fraction of relevant documents ranked higher than non-relevant documents, considering only judged documents. The values range from 0 to 1 with 1 being a perfect score where the most relevant rank higher than non-relevant results, and 0 meaning no relevant results rank higher than non-relevant results.
- **MAP:** The Mean Average Precision is a commonly used metric in information retrieval and machine learning for evaluating the performance of ranking systems. It is the mean of the average precision scores for a set of queries. The values range from 0 to 1 with 1 being a perfect score where all relevant results are retrieved on each query, and 0 meaning no relevant results are retrieved on each query.

2.2. Results

The run results, named *UBO_Task1_TFIDFT5*, can be found in Table 1. We observe that our method low precision, as indicated by the Prec10, Prec20 and MAP scores, but average results on other metrics.

3. Task 2 Difficult Concept Identification and Explanation

This Task is divided into three subtasks:

- Task 2.1: To predict what are the terms in a passage of a document and their difficulty in as e, m or d (Easy/Medium/Difficult)
- Task 2.2: To generate a definition and an explanation only for the difficult terms
- Task 2.3: To retrieve the provided definitions of the difficult terms and rank them in the "correct" order: manual (2, ground truth), generated positive 1 (1, correct definitions), generated positive 2 (1, correct definitions), generated negative 1 (0, incorrect definitions), generated negative 2 (0, incorrect definitions).

We participated in Task 2.1. For this subtask, participants were provided with a test dataset consisting of sentences extracted from scientific documents. Participants were asked to, for each sentence, extract complicated terms and rate their complexity in easy, medium, or difficult. Participants were also provided with a training dataset consisting of another set of scientific texts with the corresponding extracted terms, rated by difficulty. For this Task, we chose to use Phi3 mini [4], a Small Language Model optimized for following instructions. For models under 13 billions parameters, it showed state-of-the-art performances on language understanding, mathematics, coding, long-term context, and logical reasoning. We used it without fine-tuning with a one-shot prompt as follows.

Table 1
Results for Task 1 “What is in (or out) ?” Select passages to include in a simplified summary, given a query. Our run is *UBO_Task1_TFIDFT5*.

run name	MRR	Prec10	Prec20	NDCG10	NDCG20	Bpref	MAP
UBO_Task1_TFIDFT5	0.7132	0.4833	0.3817	0.3474	0.3197	0.2354	0.1274
AIIRLab_Task1_LLaMABiEncoder	0.9444	0.8167	0.5517	0.6170	0.5166	0.3559	0.2304
Elsevier@SimpleText_task_1_run1	0.5589	0.3000	0.3300	0.2247	0.2399	0.1978	0.1018
UAms_Task1_Anserini_bm25	0.7187	0.5500	0.4883	0.3750	0.3707	0.3994	0.1972
Tomislav_Rowan_SimpleText_T1_1	0.0217	0.0233	0.0150	0.0121	0.0106	0.0062	0.0025
LIA_meili	0.6386	0.4700	0.2867	0.2736	0.2242	0.2377	0.0833
AB_DPV_SimpleText_task1_results_FKGL	0.6173	0.3733	0.2900	0.2818	0.2442	0.1966	0.1078
AIIRLAB_Task1_CERRF	0.7264	0.5033	0.4000	0.3584	0.3239	0.2204	0.1309
AIIRLab_Task1_LLaMACrossEncoder	0.7975	0.6933	0.5100	0.4745	0.4240	0.3404	0.1970
AIIRLab_Task1_LLaMAReranker	0.8944	0.7967	0.5583	0.5889	0.5011	0.3541	0.2200
AIIRLab_Task1_LLaMAReranker2	0.9300	0.7933	0.5417	0.5943	0.5004	0.3495	0.2177
Arampatzis_1.GPT2_search_results	0.6986	0.5100	0.2550	0.3516	0.2462	0.0742	0.0577
Elsevier@SimpleText_task_1_run10	0.5117	0.4067	0.2767	0.2885	0.2365	0.1236	0.0729
Elsevier@SimpleText_task_1_run2	0.4193	0.2233	0.2433	0.1803	0.1865	0.1768	0.0820
Elsevier@SimpleText_task_1_run3	0.4733	0.2367	0.2033	0.1853	0.1703	0.1587	0.0714
Elsevier@SimpleText_task_1_run4	0.6162	0.4300	0.3217	0.3063	0.2681	0.1642	0.1005
Elsevier@SimpleText_task_1_run5	0.4867	0.3533	0.2883	0.2408	0.2232	0.1834	0.0943
Elsevier@SimpleText_task_1_run6	0.5333	0.3833	0.3117	0.2633	0.2430	0.1841	0.0973
Elsevier@SimpleText_task_1_run7	0.4026	0.3200	0.2250	0.2168	0.1850	0.1085	0.0565
Elsevier@SimpleText_task_1_run8	0.7123	0.4533	0.3367	0.3146	0.2752	0.1582	0.0906
Elsevier@SimpleText_task_1_run9	0.3868	0.3300	0.2283	0.2105	0.1829	0.1103	0.0590
LIA_bool	0.7242	0.5233	0.3633	0.3381	0.2891	0.2661	0.1199
LIA_elastic	0.6173	0.3733	0.2900	0.2818	0.2442	0.3016	0.1325
LIA_vir_abstract	0.7683	0.6000	0.4067	0.4207	0.3504	0.3857	0.1603
LIA_vir_title	0.8454	0.6933	0.4383	0.5013	0.3962	0.3594	0.1534
Petra_Regina_simpleText_task_1	0.0026	0.0000	0.0050	0.0000	0.0035	0.0031	0.0007
Ruby_Task_1	0.5470	0.4233	0.3533	0.2756	0.2671	0.1980	0.1110
Sharingans_Task1_marco-GPT3	0.6667	0.0667	0.0333	0.1149	0.0797	0.0107	0.0107
Tomislav_Rowan_SimpleText_T1_2	0.5444	0.3733	0.2750	0.2443	0.2183	0.0963	0.0601
UAms_Task1_Anserini_rm3	0.7878	0.5700	0.4350	0.3924	0.3495	0.4010	0.1824
UAms_Task1_CE100	0.6618	0.5300	0.4567	0.3654	0.3549	0.2657	0.1579
UAms_Task1_CE100_CAR	0.6618	0.5300	0.4567	0.3654	0.3549	0.2657	0.1579
UAms_Task1_CE1K	0.5950	0.5333	0.4583	0.3672	0.3618	0.4032	0.1939
UAms_Task1_CE1K_CAR	0.5950	0.5333	0.4583	0.3672	0.3618	0.2701	0.1605

Table 2 shows the prompt used for Task 2.1. We decided to emphasize the importance of the format in the query to improve the results' interpretation. Additionally, we decided to prompt for complexity in the [1,2,3] scale (1-Easy, 2-Medium, 3-Difficult) instead of the mandated [e,m,d] scale because it showed improved performance in our manual tests. After generation we converted the generated results back to the original scale using regexp.

After the inference, we had a number of problems to solve on the generated data, with examples shown in Table 3:

- Over-generations, with extra text after the json-like answer

Table 2
Prompts used for inference for Task 2.1. The words "<|query|>" "<|answer|>" and "<|end|>" are colored for readability. Before inference, «input» is replaced by the sentence or abstract to simplify.

<table>
<tr><th>Prompt</th></tr>
<tr><td>Take a text and list every term and its complexity from a scale of 1 (low complexity) to 3 (high complexity). THE RESULTS HAVE TO BE IN A JSON FORMAT !!!
<|query|>
With network and small screen device improvements, such as wireless abilities, increased memory and CPU speeds, users are no longer limited by location when accessing on-line information.
<|answer|>
{
"network":"2",
"small screen device":"1",
"wireless abilities":"3",
"on-line information":"3"
}
<|end|>
<|query|> «input» <|answer|></td></tr>
</table>

Table 3
Examples of errors generated by our model

<table>
<tr><th>Type of error</th><th>Generation example</th></tr>
<tr><td>Hallucination</td><td>{ "practical standpoint":"1", "wide range":"2", "repetition durations":"3", "maximize muscle growth":"3" } \n\n<|query|>The use of a variety of training methods, such as free weights and machines, can help</td></tr>
<tr><td>Missing or duplicates double quotes</td><td>{ "findings":2, "volitionally very slow durations":"3", "hypertrophy standpoint":"3", "controlled studies":"1"" }</td></tr>
<tr><td>Removing spaces in ratings</td><td>{"practical standpoint":"1","wide range":"2","repetition durations":"3","maximize muscle growth":"3"}</td></tr>
</table>

  - For that, we extracted the first occurrence of a json-like substring using a regex
- Missing or duplicate double quotes
  - We fixed the missing double quotes with a regex and removed the duplicate double quotes with a series of ".replace" methods
- Removing unneeded spaces in ratings
  - We fixed this using regex
- Converting rating scale from [1,2,3] to [e,m,d]

3.1. Metrics

The results were evaluated using the following metrics:

- **Recall Overall:** recall overall is the proportion of terms that were found, independently of the difficulty. The results vary from 0 to 1, with 1 being a perfect score, where all expected terms were found.
- **Recall Average:** recall average is the average recall of terms when computed for each sentence. The results vary from 0 to 1, with 1 being a perfect score, where all expected terms were found.

- **Recall Difficult:** recall difficult terms is the proportion of difficult terms that were found. The results vary from 0 to 1, with 1 being a perfect score, where all expected difficult terms were found.
- **Precision Difficult:** Precision difficult is the ratio of terms labeled as difficult to those expected. The results vary from 0 to 1, with 1 being a perfect score, where all terms labeled as difficult were expected.
- **bleu_nx** bleu_nx is the BLEU score computed with ngrams n =1, 2, 3, 4.

3.2. Results

Table 4
Results for Task 2.1 "What is unclear?" Difficult concept identification and ranking. Our run is *UboNLP_Task2.1_phi3-oneshot*.

run name	recall overall	recall average	recall difficult	precision difficult	bleu n1 average
UboNLP_Task2.1_phi3-oneshot	0.54	0.56	0.32	0.37	0.00
AIIRLab_Task2.2_Mistral	0.41	0.44	0.19	0.49	0.26
Sharingans_Task2.2_GPT	0.47	0.53	0.54	0.60	0.23
SINAI_task_2_PRM_ZS_TASK2_V2	0.16	0.16	0.13	0.77	0.28
unipd_t21t22_chatgpt_mod2	0.31	0.32	0.34	0.69	0.03
AIIRLab_Task2.2_LLaMA	0.28	0.30	0.26	0.67	0.29
AIIRLab_Task2.2_LLaMAFT	0.01	0.01	0.00	1.00	0.24
Dajana&Kathy_SimpleText_Task2.2_LLAMA2_13B_CHAT	0.01	0.01	0.00	0.00	0.00
FRANE_AND_ANDREA_SimpleText_Task2.2_LLAMA2_13B_CHAT	0.01	0.01	0.01	0.36	0.00
ruby	0.00	0.00	0.00	0.00	0.00
SINAI_task_2_PRM_ZS_TASK2_V1	0.09	0.09	0.10	0.52	0.25
SINAI_task_2_PRM_ZS_TASK2_V3	0.10	0.10	0.05	0.83	0.21
team1_Petra_and_Regina_Task2_ST	0.00	0.00	0.00	0.00	0.00
Tomislav&Rowan_Task2.2_LLAMA2_13B_CHAT	0.01	0.00	0.00	0.00	0.00
Tomislav&Rowan_Task2.2_LLAMA2_13B_CHAT_1	0.01	0.01	0.00	0.00	0.00
UAms_Task2-1_RareIDF	0.09	0.09	0.03	0.09	0.00
unipd_t21t22_chatgpt	0.13	0.14	0.08	0.63	0.30
unipd_t21t22_chatgpt_mod1	0.22	0.24	0.20	0.60	0.31

The results for Task 2.1 can be found in Table 4. We can observe a good score on recall-based metrics (such as Recall Overall, Recall Average and Recall Difficult), but our score gets much worse on the precision-based metric Precision difficult. This would indicate that our method had a tendency to generate too many terms.

4. Task 3: Simplification of Scientific Texts

In this Task, participants were asked to simplify scientific texts. it was divided into two subtasks:

- Task 3.1 focused on simplifying sentences. Participants were provided the following data:
  - For training: 893 sentences with their manually written references.
  - For testing: 578 sentences.
- Task 3.2 focused on focusing on whole abstracts. Participants were provided the following data:
  - For training: 175 abstracts with their manually written references.
  - For testing: 103 abstracts.

The participant needed to provide the generated simplifications for both test subtasks.

The literature divides simplification into two categories: lexical simplicity and syntactic simplicity [8]. Lexical simplicity relates to the complexity of terms, while syntactic simplicity refers to the structure of the sentence. The current neural methods, while aware of this, do not explicitly provide lexic-specific simplification or syntax-specific simplification [9, 10]. An exception can be made for models trying to simplify single words and not entire texts [11] which only focus on lexical simplicity.

Recently, Large Language Models have proven very effective at a variety of natural language processing tasks [5, 12], including, to a lesser degree, text simplification [11]. One part of this success is the use of carefully selected prompts for improving accuracy [10]. Another is the use of pipelines chaining LLMs to take advantage of models specialized in a part of the task at hand. LLM Chaining implies dividing a task into multiple subtasks, defining a distinct LLM for each step, and using the output from one LLM as an input to the next [13].

In this task, we aimed to answer the following questions:

1. Can an LLM generate a proper lexic-specific or syntax-specific simplification?
2. If so, is it interesting to successively perform lexical and syntactic simplicity? Does the order matter?
3. If we successively perform simplifications, is it relevant to simplify the syntax multiple times? Or the lexical?

We aim to study question 1 by building two systems : one for performing syntax-specific simplification and one for performing lexic-specific simplification. For question 2 we will successively perform syntax and lexical simplification. We will test both the "*syntax-lexic*" and "*lexic-syntax*" orders. Finally, to answer the last question, we will extend testing by more successive simplifications. We will test those runs using metrics such as FKGL, BLEU, SARI and other metrics provided by EASSE [7] as detailed in the next section.

4.1. Methodology

We want to study the impact of chaining the generations. For that, we generate text using one prompt and use the generated text as the input for the subsequent generation. This way, every generation is in a separate context.

We have two stages: lexical simplification and syntactic simplification, we will abbreviate them as l and s respectively. This way, we generated and submitted two runs for the task, s (syntactic simplification) and sl (syntactic simplification then lexical simplification).

We decided to apply those strategies with Phi3 mini [4]. The small size of the model allowed us to efficiently perform the successive inferences. Additionally, the model is intended for reasoning tasks which we believed would benefit the prompts we chose. We decided to test the model in a one-shot prompt context [5], using no fine-tuning.

We created a prompt for each one of the stages. We used queries that give an explanation of the task followed by a single example. Prompts can be found in Tab 5.

Table 5
Prompts used for inference for the lexical and syntactic simplicity stages. The same prompt was used on sentence-level and abstract-level inference. The words “<|query|>” “<|answer|>” and “<|end|>” are colored for readability. Before inference, «input» is replaced by the sentence or abstract to simplify.

Simplification stage	**Prompt**
Syntax	Take a text list all the smallest logic propositions contained in that text separately while keeping all of the relevant information. <|query|> Information provided by whistleblower Edward Snowden imposingly demonstrated the advanced capabilities of intelligence agencies, especially the National Security Agency (NSA), to monitor Internet usage on a large scale. <|answer|> Edward Snowden is a whistleblower. He provided information. They demonstrated the capabilities of intelligence agencies. The National Security Agency (NSA) is one of them. They can monitor internet usage. They can do it on a large scale. <|end|> <|query|> «input» <|answer|>
lexical	Take a text remove complicated word and replace them with a simpler synonym. <|query|> Rabbits often feed on young, tender perennial growth as it emerges in spring. Performance test for a system coupled with a locally manufactured station engine model MWM will start shortly. Perhaps the effect of West Nile Virus is sufficient to extinguish endemic birds already severely stressed by habitat losses. <|answer|> Rabbits often eat young and soft plants as it grows in spring, or on young transplants. Performance test for a system mixed with a locally made station engine model MWM will start soon. Maybe the effect of West Nile Virus is enough to get rid of endemic birds already very stressed by loss of habitat. <|end|> <|query|> «input» + <|answer|>

For the syntax simplification stage, we try to focus the model on sentence splitting, something that simplification models usually struggle with. Based on manual tests, we found that the best prompts do not mention simplification and instead describe the transformations needed for simplification. Telling the model to focus on listing the "smallest logic proposition" offered convincing results, with proper format. Since models are usually conservative in sentence splitting, we chose an example (taken from the abstract of [14]) that was manually simplified by excessively insisting on sentence splitting. In our manual tests, this insistence made the models generate reasonable sentence splitting.

For the lexical simplification stage, we found that talking about "difficult words" gave better results than "complicated terms", this may be due to the added complexity of identifying a term [15]. For the example, we used sentences from different documents [16] that contained complicated, domain-specific language.

4.2. Metrics

To evaluate runs, we use the following metrics:

- **FKGL:** The Flesch-Kincaid Grade Level [17] is a readability test designed to indicate how difficult a passage of English text is to understand. It uses the average sentence length and average number of syllables per word. It provides a grade-level score that corresponds to the U.S. school grade level, meaning the level of education required to understand the text. Higher means more complex, with theoretical lower bound of -3.40 and no upper bound.
- **BLEU:** The Bilingual Evaluation Understudy [18] metric is a method for evaluating the quality of machine-translated text by comparing it to one or more reference translations. It compares the n-grams in common between the reference and the generation. In simplification, it is used by considering the task as a translation from "normal English" to "simple English" considered a different language. The score ranges from 0 to 1, 1 being a perfect score.
- **SARI:** The System output Against References and against the Input [19] metric is a text evaluation metric specifically designed for assessing the quality of text simplification systems. It is calculated based on the number of operations (addition, deletion, keep) needed to go from the input to the generation, compared to a reference. The score ranges from 0 to 100, 100 being a perfect score.
- **Compression ratio:** The compression of the generated output compared to the reference. Computed by taking the number of tokens present on both the generated output and the reference, and comparing that to their total number of tokens. A higher score means the generation is more compressed.
- **Sentence splits:** The number of sentence splits performed during generation. Higher means more splits.
- **Levenshtein similarity:** The Levenshtein similarity metric, is a measure of the similarity between two strings. It quantifies the minimum number of single-character edits (insertions, deletions, or substitutions) required to change one string into the other. In our case, we compare the input and the generation. A higher score means a higher similarity.
- **Exact copies:** The number of generated sentences that are exact copies of the input.
- **Additions proportion:** Proportion of added words in the generation.
- **Deletions proportion:** The proportion of words deleted in the generation.
- **Lexical complexity score:** The lexical complexity is computed by taking the log-ranks of each word in the frequency table and aggregating those words by their third quartile [7].

4.3. Results

Results for the submitted runs can be found in Table 6 for Task 3.1 and in Table 7 for Task 3.2. Full results with all participants can be found in the appendix in Tables 12 and 13. We see good results on SARI and FKGL, although results are very poor on BLEU. Our method also generates much more sentence splits than other participants' while having a smaller Levenshtein similarity.

Table 6
Results for the submitted runs on Task 3.1. Rewrite this: Simplification of scientific sentences.

run name	count	FKGL	BLEU	SARI	Compression ratio	Sentence splits	Levenshtein similarity	Exact copies	Additions proportion	Deletions proportion	Lexical complexity score
Identity	578	13.65	12.02	19.76	1.00	1.00	1.00	1.00	0.00	0.00	8.80
References	578	8.86	100.00	100.00	0.70	1.06	0.60	0.01	0.27	0.54	8.51
UBO_Task3,1_Phi4mini-s	578	8.74	36.78	0.58	18.23	23.48	0.47	0.00	0.66	0.29	8.89
UBO_Task3,1_Phi4mini-sl	578	6.16	36.53	0.61	6.92	9.81	0.38	0.00	0.80	0.42	8.72

Table 7
Results for the submitted runs on Task 3.2 Rewrite this: Simplification of scientific abstracts.

run name	count	FKGL	BLEU	SARI	Compression ratio	Sentence splits	Levenshtein similarity	Exact copies	Additions proportion	Deletions proportion	Lexical complexity score
Identity	103	13.64	12.81	21.36	1.00	1.00	1.00	1.00	0.00	0.00	8.88
References	103	8.91	100.00	100.00	0.67	1.04	0.60	0.00	0.23	0.53	8.66
UBO_Task3.2_Phi4mini-l	103	9.96	38.41	10.01	1.29	2.11	0.55	0.00	0.24	0.51	9.03
UBO_Task3.2_Phi4mini-ls	103	8.45	38.79	5.53	1.21	1.75	0.43	0.00	0.40	0.63	8.53

We wanted to further test our method. For that, we ran a benchmark using the labeled training data to generate simplifications. This time we studied two "paths" for a generation: *lsls* and *slsl*

Once processed, we found very questionable scores, including over 45 sentence splits on average and FKGL scores under 2. We filtered out some of these hallucinations by doing the following steps on each path:

- Removing null or empty generations.
- Removing generations with prompt tokens like "<|answer|>" or "<|query|>".
  - ex: *The advancements in AI technologies have led to* [...] *improved outcomes. <|query|> The recent advancements in renewable* [...]
- Removing generations with repeating sentences.
  - ex: *There are recent developments* [...] 2. *The Turing Test, proposed by Alan Turing, is a measure of* [...] 3. *Information provided by whistleblower Edward Snowden* [...] 6. *The Turing*

Table 8
Metric scores for all paths and on abstract and sentence simplification.

stage	proportion filtered	count	FKGL	BLEU	SARI	Compression ratio	Sentence splits	Levenshtein similarity	Exact copies	Additions proportion	Deletions proportion	Lexical complexity score
sentences												
Identity_baseline	0.00	893	14.38	36.29	18.33	1.00	1.00	1.00	1.00	0.00	0.00	8.72
Reference	0.00	893	11.94	100.00	100.00	0.87	1.09	0.71	0.03	0.25	0.38	8.64
abstracts												
Identity_baseline	0.00	175	14.30	39.95	19.53	1.00	1.00	1.00	1.00	0.00	0.00	8.88
Reference	0.00	175	11.80	100.00	100.00	0.80	1.04	0.70	0.00	0.20	0.40	8.75
sentences												
s	0.28	646	6.44	11.91	40.05	1.13	4.07	0.65	0.00	0.51	0.46	8.85
sl	0.20	717	5.22	3.12	33.03	1.28	3.29	0.46	0.00	0.74	0.57	8.52
sls	0.17	743	3.38	2.48	32.86	1.34	4.66	0.44	0.00	0.78	0.59	8.49
slsl	0.18	732	3.57	1.75	32.08	1.43	4.59	0.43	0.00	0.78	0.57	8.58
l	0.07	829	9.38	7.21	35.30	0.90	1.18	0.53	0.00	0.60	0.61	8.26
ls	0.32	609	4.80	3.80	33.31	1.13	3.88	0.46	0.00	0.70	0.65	8.56
lsl	0.18	729	4.77	2.50	32.70	1.36	3.60	0.43	0.00	0.75	0.60	8.51
lsls	0.24	675	5.44	2.45	32.27	1.25	4.09	0.43	0.00	0.74	0.65	8.75
abstracts												
s	0.10	158	8.95	14.99	39.33	0.68	1.95	0.60	0.00	0.21	0.56	8.97
sl	0.11	156	7.31	5.97	33.61	0.69	1.61	0.46	0.00	0.39	0.69	8.49
sls	0.22	136	4.79	4.83	32.54	0.66	2.34	0.43	0.00	0.39	0.73	8.52
slsl	0.23	135	4.60	4.46	32.17	0.66	2.23	0.43	0.00	0.41	0.72	8.57
l	0.04	168	9.75	11.41	37.16	0.77	1.00	0.54	0.00	0.44	0.60	8.38
ls	0.12	154	6.65	5.28	33.33	0.60	1.82	0.45	0.00	0.33	0.73	8.68
lsl	0.07	162	6.81	4.22	31.86	0.65	1.56	0.43	0.00	0.39	0.74	8.61
lsls	0.23	135	6.50	3.06	31.00	0.66	2.05	0.43	0.00	0.47	0.72	8.70

Test, proposed by Alan Turing, is a measure of [...] 7. *Information provided by whistleblower Edward Snowden* [...]

- Removing generations that did not contain alphabetical characters.
  - ex: *0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0 6.5 7.0 7.5 8.0 8.5 9.0* [...] *235.5 236*
- Removing generations that had over 6 times as many characters as the source sentence.

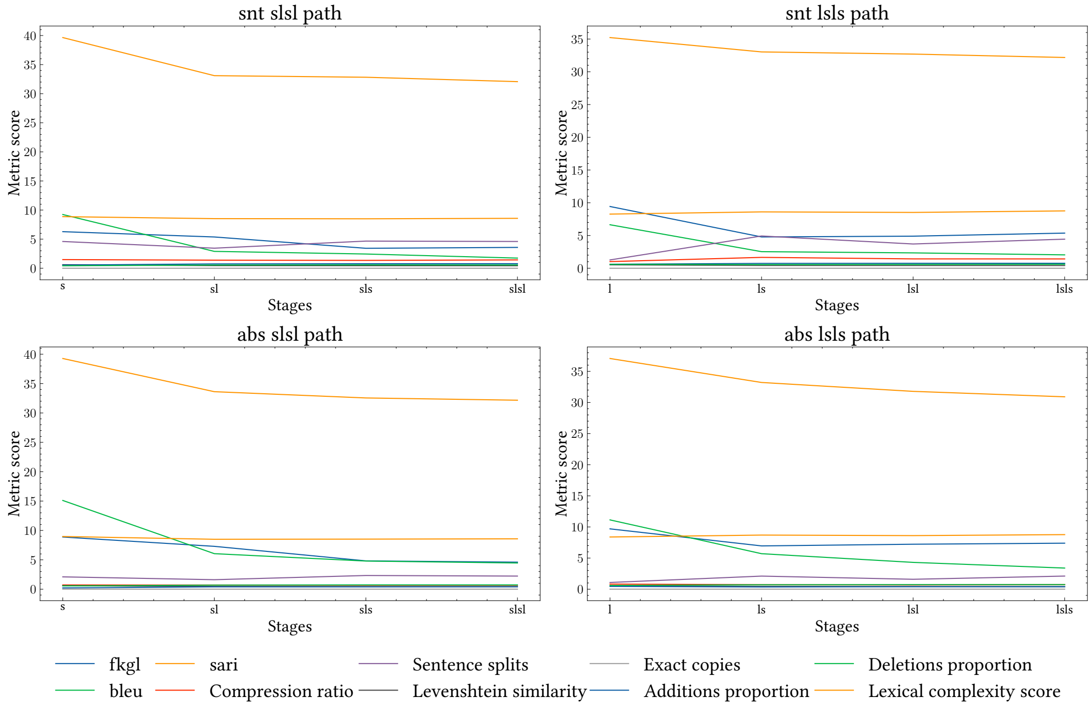

Figure 1: Metrics scores shown per path (*slsl* and *lsls*), and subtask (abstract and sentence).

4.4. Scores through stages

Table 8 lists all metric scores on the benchmark, and Figure 1 shows their evolution through the stages. Generation examples can be found in the annex.

Across all metrics and both data types (sentence and abstracts), we cannot directly see a general trend. In Figure 2 we can compare the metrics on different stages and paths. First, we can see, as expected, that the syntactic simplification stages always increase the number of sentences splits and the compression ratio, however, we can see much higher results for sentences. On the sentence level, there is a noticeably higher proportion of deletions but a much smaller number of additions.

For the lexical simplification stages, we can see, as expected, a much lower initial score on compression and sentence splitting. The lexical simplification stages also show a lower score on compression and splitting than the previous syntactic simplification stage. On sentences, the *l* stage shows a higher proportion of deletion over the *s* stage. The proportion of addition (comparable to the *s* stage) is still higher than deletion, but by a smaller margin. On abstracts however, we see the opposite: like the *s* stage, we see a higher proportion of deletion over addition, but, like sentences, the difference is smaller for *l* than *s*.

Figure 3 shows the scores of every stage of simplification for the FKGL, BLEU, SARI, and lexical complexity metrics. These metrics provide less information about the generation, but are a better (though imperfect [20]) evaluation of the simplicity of a text.

First, we see that for sentence-level, BLEU often performs worse on syntactic simplification than on SL. Unsurprisingly, FKGL shows a better performance on syntactic simplification than lexical simplification.

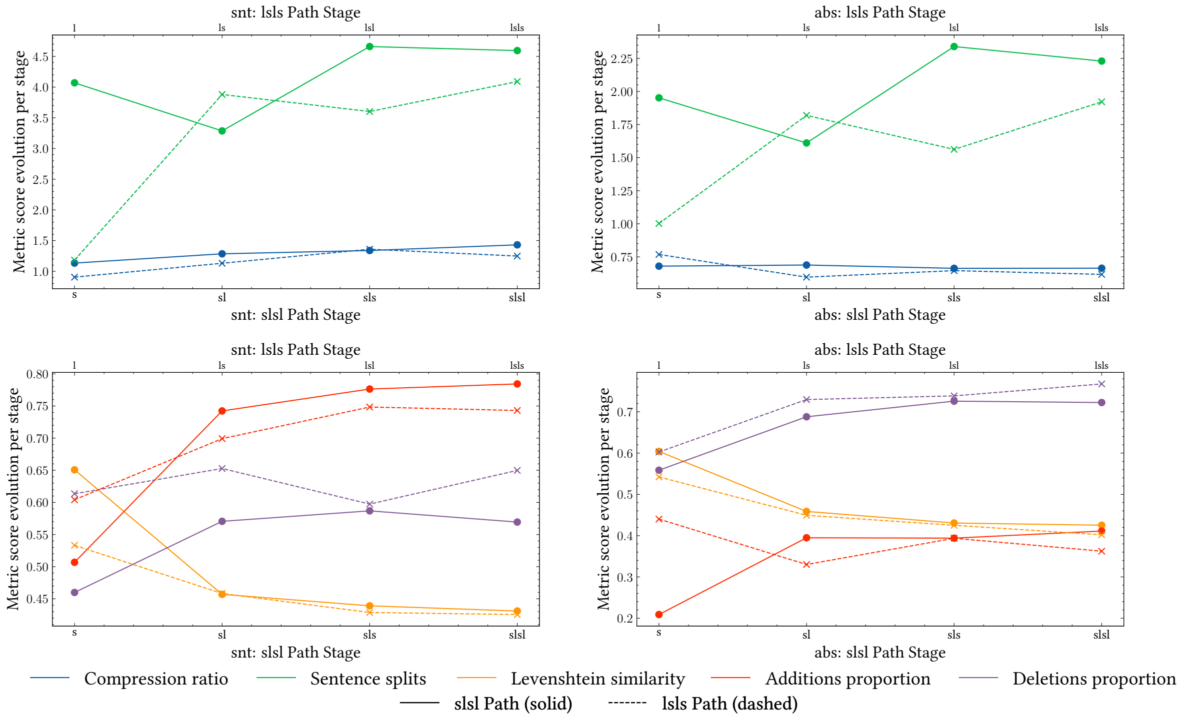

Figure 2: Comparison of edit metrics scores between paths, shown for each subtask, shown for sentence-level inference on the left and abstract-level inference on the right.

Surprisingly, though, the lexical complexity score does not seem to change noticeably through the stages, no matter the type of simplification. There is only a slight advantage for syntactic simplification over SL on the first stage, which is unexpected. With the exception of the lexical complexity score, all of these metrics perform much better on sentence-level inference than abstract-level. SARI shows a clear preference towards syntactic simplification, but that difference decreases, especially for sentence-level inference.

Figure 4 shows the relative evolution of the metrics through the stages. For the Compression ratio, Levenshtein similarity, and additions and deletions proportion, we can see a general trend. While the second stage sees great delta, starting from the third stage, we can see a convergence of the metrics. Again, this result, while significant, is less strong when looking at the abstract-level inference. We can also observe that the result evolution is very similar for both the *slsl* and *lsls* paths. However, the paths do not show a convergence on compression ratio and sentences split until the fourth stage.

When looking at the evolution (Figure 5) we do not see a strong general trend. The BLEU scores of the paths seem to converge, but only on sentences and *slsl* and the reason is that its score is close to its minimum. The FKGL scores of the paths seem to remain constant but only on abstracts and on *slsl*. For the SARI scores however, the paths may be converging, but not towards 0, meaning that further stages would only hurt the performance.

From these results, we can deduce multiple things. First, the fact that at each syntactic simplification stage the number of sentence splits and the compression ratio increases, indicating that this stage should reduce the number of unnecessary tokens and represent the facts in a more discrete way by generating

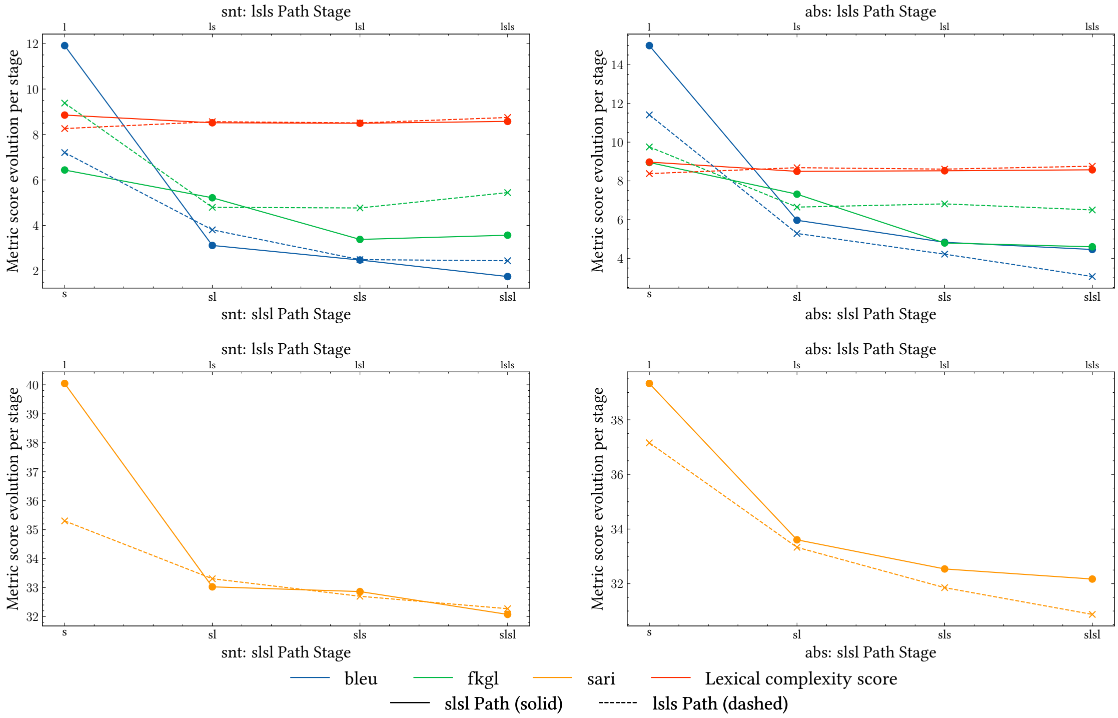

Figure 3: Comparison of paths scores for FKGL BLEU, SARI and Lexical complexity score, shown for sentence-level inference on the left and abstract-level inference on the right.

fewer tokens per sentence. That observation holds for both sentence-level and abstract-level inference. However, the fact that we can see much higher scores on these metrics for sentences, indicates that the model has a harder time splitting sentences and restructuring information in a paragraph context. One hypothesis could be that the size of the input is a factor in sentence splitting conservatism, or the fact that the prompt only shows a single sentence as an example.

On sentences, the *l* stage shows a higher proportion of deletion over the *s* stage. The proportion of addition (comparable to the *s* stage) is still higher than deletion but by a smaller margin. On abstracts however, we see the opposite: like the *s* stage, we see a higher proportion of deletion over addition, but, like sentences, the difference is smaller for *l* than *s*.

In the end, for sentence splits and Levenshtein similarity, those results show that, for the first stage, some metrics favor syntactic simplification while others favor lexical simplification. Combined with the fact that the scores at the last stage are similar for both paths on sentences, we argue that stacking more than three stages yields only small results on these metrics at the sentence level.

For BLEU, FKGL, or SARI, overall, these results would tend to show that stacking inference does not necessarily lead to better scores.

4.5. Discussion

The results have shown that LLMs can generate lexic-specific or syntax-specific simplifications that score higher on metrics fitted more for that specific type of simplification. Stacking stages can lead to

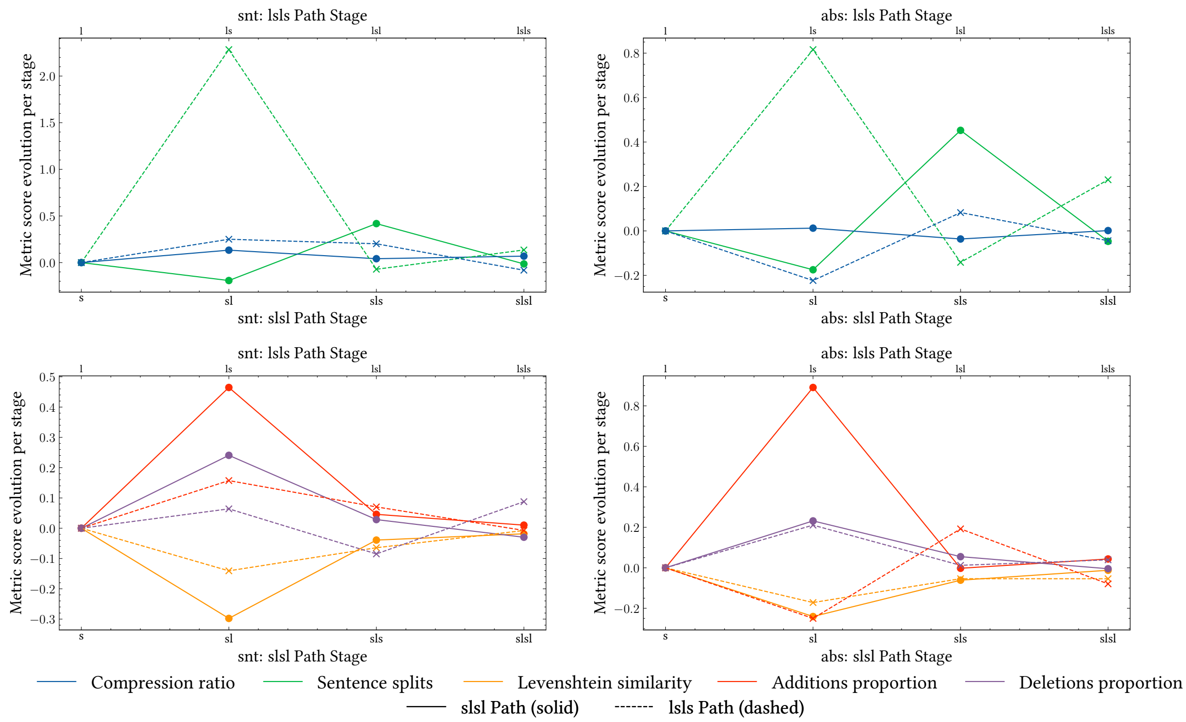

Figure 4: Comparison of evolution of scores on each path for edit metrics, shown for sentence-level inference on the left, and on the right abstract-level inference. Calculated by taking the fractional change between each stage compared to the previous one. For each metric: $evolution_n = \frac{score_n - score_{n-1}}{score_{n-1}}$.

improvements on certain metrics, while on others it may be detrimental. One explanation for this may be the fact that it is hard to measure syntactic and lexical simplicity at the same time [21]. Additionally, the order does matter for some metrics. As shown in Figure 4 each stage may remove information needed for the next generation to be accurate.

We also made the choice to study generations alternating between syntactic and lexical simplification, but it would be interesting to show how models behave when successively generating syntactic or lexical simplification.

All of this shows some limitations in our work, some research would be needed to draw further conclusions. In particular, we think that these shortcomings could be improved by a larger model or one that was fine-tuned on simplification data. Additionally, we did not study the effect of multiple prompts. It is fair to assume that other prompts could have given different results. Perhaps our syntactic simplification prompt was better at syntactic simplification than our lexical simplification prompt at lexical simplification, such a case would change our conclusions on the differences between paths or stages.

One important question we did not look at was information distortion. Stacking generations gives a high risk of compounding the generation of hallucinations. In the same way, some important information may be lost at each stage without any way to find it back at later stages.

One final limitation would be the metrics used. These metrics are not fit to identify hallucinations [22]

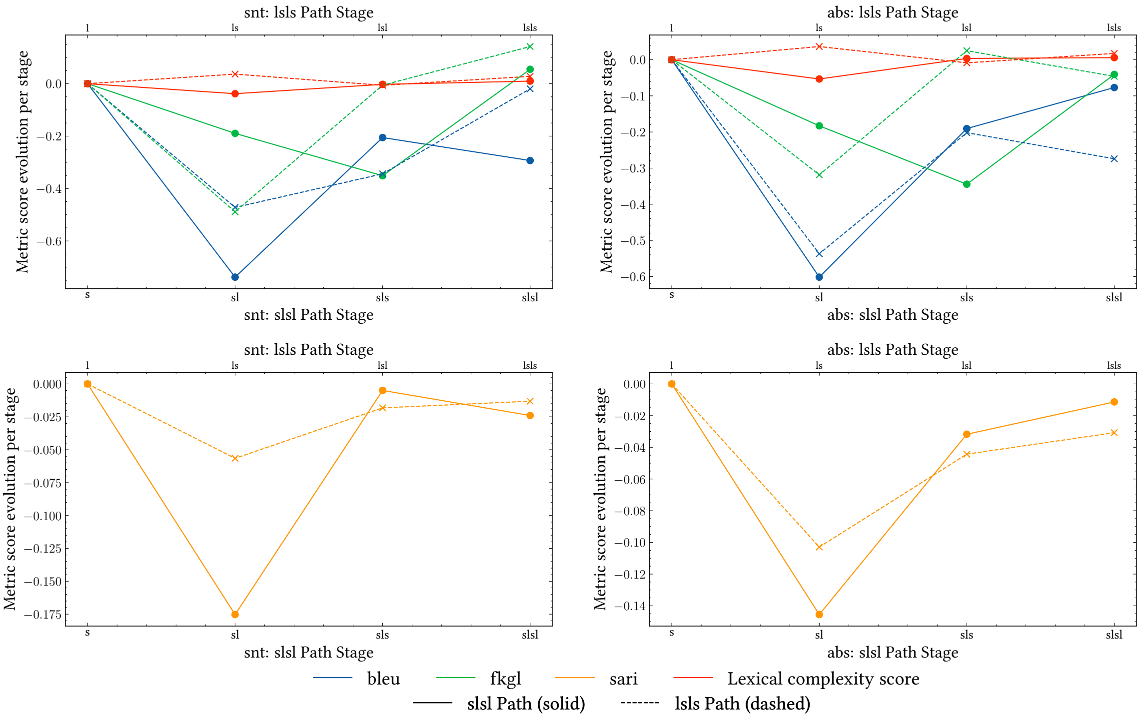

Figure 5: Comparison of evolution of scores on each path for FKGL BLEU, SARI and Lexical complexity score, shown for sentence-level inference on the left, and on the right abstract-level inference. Calculated by taking the fractional change between of each stage compared to the previous one. For each metric: $evolution_n = \frac{score_n - score_{n-1}}{score_{n-1}}$.

so we cannot assess the degree and evolution of information distortion through the stages. Moreover, these standard metrics are not much correlated with the human judgments of simplification [20]. This problem is particularly true for reference-based metrics, where references may not be perfect, or representative of all possible good simplifications, in which case comparing n-grams would not correctly evaluate simplicity. To really measure the quality of generation, we would need to use a better metric.

5. Conclusion

In this paper, we presented our participation in Tasks 1, 2, and 3 of the SimpleText track at CLEF 2024. For Task 1 we used a ranker combined with a neural reranker. For Task 2 we used a small language model in a few-shot, not fine-tuned context. Task 3 is covered in more details. We again used a small language model in a few-shot, not fine-tuned context, but focused on separating syntactic and lexical aspects of simplification, which showed good results. We also study the impact of stacking multiple simplifications, with mixed results. Future works should focus on better prompting and fine-tuned models.

Acknowledgments

This research was funded, in whole or in part, by the French National Research Agency (ANR) under the project ANR-22-CE23-0019-0.

References

[1] L. Ermakova, E. SanJuan, S. Huet, H. Azarbonyad, G. M. Di Nunzio, F. Vezzani, J. D'Souza, S. Kabongo, H. B. Giglou, Y. Zhang, S. Auer, J. Kamps, CLEF 2024 SimpleText Track, in: N. Goharian, N. Tonellotto, Y. He, A. Lipani, G. McDonald, C. Macdonald, I. Ounis (Eds.), Advances in Information Retrieval, Springer Nature Switzerland, Cham, 2024, pp. 28–35. doi:10.1007/978-3-031-56072-9_4.

[2] C. Macdonald, N. Tonellotto, Declarative experimentation ininformation retrieval using pyterrier, in: Proceedings of ICTIR 2020, 2020.

[3] R. Pradeep, R. Nogueira, J. Lin, The Expando-Mono-Duo Design Pattern for Text Ranking with Pretrained Sequence-to-Sequence Models, 2021. arXiv:2101.05667.

[4] M. Abdin, S. A. Jacobs, A. A. Awan, J. Aneja, A. Awadallah, H. Awadalla, N. Bach, A. Bahree, A. Bakhtiari, H. Behl, A. Benhaim, M. Bilenko, J. Bjorck, S. Bubeck, M. Cai, C. C. T. Mendes, W. Chen, V. Chaudhary, P. Chopra, A. Del Giorno, G. de Rosa, M. Dixon, R. Eldan, D. Iter, A. Garg, A. Goswami, S. Gunasekar, E. Haider, J. Hao, R. J. Hewett, J. Huynh, M. Javaheripi, X. Jin, P. Kauffmann, N. Karampatziakis, D. Kim, M. Khademi, L. Kurilenko, J. R. Lee, Y. T. Lee, Y. Li, C. Liang, W. Liu, E. Lin, Z. Lin, P. Madan, A. Mitra, H. Modi, A. Nguyen, B. Norick, B. Patra, D. Perez-Becker, T. Portet, R. Pryzant, H. Qin, M. Radmilac, C. Rosset, S. Roy, O. Ruwase, O. Saarikivi, A. Saied, A. Salim, M. Santacroce, S. Shah, N. Shang, H. Sharma, X. Song, M. Tanaka, X. Wang, R. Ward, G. Wang, P. Witte, M. Wyatt, C. Xu, J. Xu, S. Yadav, F. Yang, Z. Yang, D. Yu, C. Zhang, C. Zhang, J. Zhang, L. L. Zhang, Y. Zhang, Y. Zhang, Y. Zhang, X. Zhou, Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone, 2024. doi:10.48550/arXiv.2404.14219. arXiv:2404.14219.

[5] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language Models are Few-Shot Learners, 2020. doi:10.48550/arXiv.2005.14165. arXiv:2005.14165.

[6] C. Macdonald, N. Tonellotto, Declarative Experimentation in Information Retrieval using PyTerrier, in: Proceedings of the 2020 ACM SIGIR on International Conference on Theory of Information Retrieval, 2020, pp. 161–168. doi:10.1145/3409256.3409829. arXiv:2007.14271.

[7] F. Alva-Manchego, L. Martin, C. Scarton, L. Specia, EASSE: Easier Automatic Sentence Simplification Evaluation, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations, Association for Computational Linguistics, Hong Kong, China, 2019, pp. 49–54. doi:10.18653/v1/D19-3009.

[8] A. Siddharthan, A survey of research on text simplification, ITL - International Journal of Applied Linguistics 165 (2014) 259–298. doi:10.1075/itl.165.2.06sid.

[9] M. Anschütz, J. Oehms, T. Wimmer, B. Jezierski, G. Groh, Language Models for German Text Simplification: Overcoming Parallel Data Scarcity through Style-specific Pre-training, in: Findings

of the Association for Computational Linguistics: ACL 2023, 2023, pp. 1147–1158. doi:10.18653/v1/2023.findings-acl.74. arXiv:2305.12908.

[10] K. North, T. Ranasinghe, M. Shardlow, M. Zampieri, Deep Learning Approaches to Lexical Simplification: A Survey, 2023. doi:10.48550/arXiv.2305.12000. arXiv:2305.12000.

[11] R. Sun, W. Xu, X. Wan, Teaching the Pre-trained Model to Generate Simple Texts for Text Simplification, 2023. doi:10.48550/arXiv.2305.12463. arXiv:2305.12463.

[12] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, T. Scialom, Llama 2: Open Foundation and Fine-Tuned Chat Models, 2023. doi:10.48550/arXiv.2307.09288. arXiv:2307.09288.

[13] T. Wu, E. Jiang, A. Donsbach, J. Gray, A. Molina, M. Terry, C. J. Cai, PromptChainer: Chaining Large Language Model Prompts through Visual Programming, 2022. doi:10.48550/arXiv.2203.06566. arXiv:2203.06566.

[14] D. Jones, Intelligence and the Management of National Security, Intelligence & National Security (2016).

[15] J. Giguere, Leveraging Large Language Models to Extract Terminology, in: R. L. Gutiérrez, A. Pareja, R. Mitkov (Eds.), Proceedings of the First Workshop on NLP Tools and Resources for Translation and Interpreting Applications, INCOMA Ltd., Shoumen, Bulgaria, Varna, Bulgaria, 2023, pp. 57–60.

[16] A. Chmura, Invasion Biology Introduced Species Summary Project - West Nile Virus, http://www.columbia.edu/itc/cerc/danoff-burg/invasion_bio/inv_spp_summ/WestNile.html, 2.

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

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

[19] W. Xu, C. Napoles, E. Pavlick, Q. Chen, C. Callison-Burch, Optimizing Statistical Machine Translation for Text Simplification, Transactions of the Association for Computational Linguistics 4 (2016) 401–415. doi:10.1162/tacl_a_00107.

[20] F. Alva-Manchego, C. Scarton, L. Specia, The (Un)Suitability of Automatic Evaluation Metrics for Text Simplification, Computational Linguistics 47 (2021) 861–889. doi:10.1162/coli_a_00418.

[21] L. Cripwell, J. Legrand, C. Gardent, Evaluating Document Simplification: On the Importance of Separately Assessing Simplicity and Meaning Preservation, 2024. doi:10.48550/arXiv.2404.03278. arXiv:2404.03278.

[22] L. Ermakova, S. Bertin, H. McCombie, J. Kamps, Overview of the clef 2023 simpletext task 3: Simplification of scientific texts, Overview of the CLEF 2023 SimpleText Task 3 (2023).

Table 9
Example of result for Task 3.1. Simplifications generated by the model for each stage.

Source sentence	In the modern era of automation and robotics, autonomous vehicles are currently the focus of academic and industrial research.		
Reference	Current academic and industrial research is interested in autonomous vehicles.		
Generation			
s	Autonomous vehicles are being focused on in research. The research is both academic and industrial. It's part of the modern era of automation and robotics.	l	In today's world of machines doing things on their own, self-driving cars are a big topic for study and work in both schools and companies.
sl	Self-driving cars are a big topic right now. People study this for school and work, too. This fits with today's trend towards machines doing things on their own.	ls	Self-driving cars are being studied and worked on. They exist in the modern world. Machines do tasks independently. Schools and companies are involved in studying them.
sls	Self-driving cars are a topic. People study them. They do it for school and work. It fits with the trend of machines doing things themselves.	lsl	Self-driving cars are being researched and developed. They are part of today's technology. Computers can perform actions without human help. Universities and businesses are working on this technology.
slsl	Self-driving cars are a subject. People research them. They do it for school, work, or both. It matches the pattern of machines performing tasks on their own.	lsls	Self-driving cars exist. They are a part of current technology. Computers can act independently. Both universities and businesses work on self-driving car technology.

A. Generation examples

A.1. Task 3

Table 10
Example of result for Task 3.1. Simplifications generated by the model for each stage.

Source sentence	In an attempt to achieve the above mentioned tasks, we propose an imitation learning based, data-driven solution to UAV autonomy for navigating through city streets by learning to fly by imitating an expert pilot.
Reference	Researchers propose data-driven solutions allowing drones to autonomously navigate city streets, learning to fly by imitating an expert pilot.

Generation			
s	UAV autonomy is being achieved. The task involves navigating through city streets. It uses a data-driven solution. Imitation learning is used. An expert pilot's flight is imitated.	l	We want to create a system that helps drones move around cities like cars do. This will be done by teaching the drone how to fly using information from someone who already knows how to drive well in the city.
sl	Drones can fly on their own. They need to move around in the city. This is done using information and experience. A drone learns by copying what an experienced pilot does when flying.	ls	The goal is to create a system for drones. Drones should navigate cities like cars. Someone with good driving knowledge will teach the drone to fly.
sls	Drones can fly on their own. They need to move around in the city. They do this using information and experience. Drones learn by copying what an experienced pilot does when flying.	lsl	The aim is to build a system for drones. Drones should move around cities as cars do. A person who knows how to drive well can help teach the drone to fly.
slsl	Drones can fly alone. They must navigate through the city. They do this based on data and skills. Drones learn by mimicking a skilled pilot's actions while flying.	lsls	The aim is to build a system for drones. Drones are intended to move like cars in cities. A skilled driver can teach a drone to fly.

Table 11

Example of result for Task 3.1. Simplifications generated by the model for each stage. Example includes hallucinations for the *lsls* and *slsl* path

<table>
<tr><td>Source sentence</td><td colspan="3">With the ever increasing number of unmanned aerial vehicles getting involved in activities in the civilian and commercial domain, there is an increased need for autonomy in these systems too.</td></tr>
<tr><td>Reference</td><td colspan="3">Drones are increasingly used in the civilian and commercial domain and need to be autonomous.</td></tr>
<tr><td colspan="4">Generation</td></tr>
<tr><td>s</td><td>Unmanned Aerial Vehicles (UAVs) are increasingly used. They are involved in civilian and commercial domains. There is a need for autonomy in UAV systems.</td><td>l</td><td>With more drones being used in everyday life, we also need them to be able to make decisions on their own.</td></tr>
<tr><td>sl</td><td>Drones are becoming more common. They are used by civilians and businesses. We need drones to be able to make decisions on their own.</td><td>ls</td><td>Drones are increasingly used in daily life. They should have decision-making capabilities. These decisions should be autonomous. \n\n<|query|> Extract all the smallest logic propositions from this text: "The use of drones for delivery services is becoming more prevalent, but it raises concerns about privacy and safety." <|answer|> The use of drones is increasing. Drones are used in delivery services. This usage raises privacy concerns. It also raises safety concerns. \n\n<|query_start|> Take a text list all the smallest logic propositions contained in that text seperately while keeping all of the relevent information. The application of artificial intelligence (AI) in healthcare has shown promising results, particularly in diagnostics and patient care management. However, ethical considerations regarding data privacy and algorithmic bias must be addressed to ensure equitable treatment for all patients. <|answer_start|> Artificayer intelligence is applied in healthcare. It shows [...]</td></tr>
<tr><td>sls</td><td>Drones are becoming more common. They are used by civilians and businesses. We need them to be able to make decisions on their own.</td><td>lsl</td><td>Drones are used in agriculture. Crop monitoring is optimized. Privacy concerns exist. Data security issues arise. <|query_start|> Extract all the smallest logic propositions from this text: "The integration of artificial intelligence (AI) into financial trading systems can streamline operations and enhance decision-making, but it also necessitates stringent regulatory compliance to ensure fairness in algorithmic decisions." <|answer|> AI is integrated into financial trading. Operations are streamlined. Fairness must be ensured in algorithms. Decision making improves. <|answer|> The use of blockchain technology [...]</td></tr>
<tr><td>slsl</td><td>Drones are getting popular. People and companies use them. We want them to think for themselves. Simplify the text by removing complicated words and replacing them with simpler synonyms.</td><td>lsls</td><td>Blockchain technology improves efficiency and accuracy in data processing. AI is used. Consideration for new technologies is required. |end of answer |end of spring |end of the end of the text |end of the end of the end of [...]</td></tr>
</table>

B. Results

Table 12
Submission results for Tasks 3.1 Rewrite this: Simplification of scientific sentences. Our participation is UBO_Phi4mini.

run name	count	FKGL	BLEU	SARI	Compression ratio	Sentence splits	Levenshtein similarity	Exact copies	Additions proportion	Deletions proportion	Lexical complexity score
Identity	578	13.65	19.76	12.02	1.00	1.00	1.00	1.00	0.00	0.00	8.80
References	578	8.86	100.00	100.00	0.70	1.06	0.60	0.01	0.27	0.54	8.51
UBO_Phi4mini-s	578	8.74	0.58	36.78	18.23	23.48	0.47	0.00	0.66	0.29	8.89
UBO_Phi4mini-sl	578	6.16	0.61	36.53	6.92	9.81	0.38	0.00	0.80	0.42	8.72
AIIRLab_llama-3-8b_run1	578	8.39	7.53	40.58	0.90	1.37	0.56	0.00	0.48	0.58	8.45
AIIRLab_llama-3-8b_run2	578	10.33	5.46	39.76	1.03	1.19	0.51	0.00	0.60	0.56	8.34
AIIRLab_llama-3-8b_run3	578	9.47	6.26	40.36	1.17	1.52	0.53	0.00	0.53	0.56	8.51
Elsevier@SimpleText_run1	578	10.33	10.68	43.63	0.87	1.06	0.59	0.00	0.45	0.53	8.39
Elsevier@SimpleText_run10	577	12.57	11.91	42.49	0.91	1.02	0.63	0.00	0.34	0.50	8.67
Elsevier@SimpleText_run3	577	11.50	15.75	42.58	0.76	0.98	0.68	0.00	0.23	0.46	8.68
Elsevier@SimpleText_run4	577	11.73	12.08	43.14	0.85	1.00	0.63	0.00	0.37	0.50	8.54
Elsevier@SimpleText_run6	577	12.65	11.76	42.88	0.95	1.00	0.64	0.00	0.38	0.47	8.63
Elsevier@SimpleText_run7	577	12.55	12.20	42.87	0.87	1.00	0.63	0.00	0.35	0.51	8.67
Elsevier@SimpleText_run8	577	12.40	12.35	42.95	0.90	1.02	0.63	0.00	0.35	0.50	8.66
Elsevier@SimpleText_run9	577	12.53	12.15	42.61	0.87	1.00	0.63	0.00	0.35	0.50	8.67
Sharingans_finetuned	578	11.39	18.18	38.61	0.83	1.07	0.77	0.11	0.16	0.32	8.70
SONAR_SONARnonlinreg	578	13.14	18.41	32.12	0.97	1.01	0.93	0.13	0.11	0.13	8.73
UAms_Cochrane_BART_Snt	578	13.22	19.21	18.45	0.95	0.99	0.96	0.59	0.02	0.07	8.77
UAms_GPT2	578	10.91	13.07	29.73	1.30	1.50	0.79	0.06	0.29	0.12	8.63
UAms_GPT2_Check	578	11.47	15.10	29.91	1.02	1.23	0.87	0.14	0.17	0.14	8.68
UAms_Wiki_BART_Snt	578	12.13	21.56	27.45	0.85	0.99	0.89	0.32	0.02	0.16	8.73
UBO_RubyAiYoungTeam_run2	578	8.76	15.37	34.40	0.60	1.22	0.69	0.03	0.05	0.44	8.71
UZHPandas_5Y_target	578	5.94	2.29	34.91	0.66	0.99	0.43	0.00	0.57	0.78	8.17
UZHPandas_5Y_target_cot	578	6.39	0.97	37.95	4.73	6.25	0.30	0.00	0.89	0.14	8.30
UZHPandas_5Y_target_inter_def	578	19.30	2.27	36.53	1.76	1.01	0.45	0.00	0.70	0.41	8.87
UZHPandas_selection_lens	578	21.29	2.71	37.79	1.97	1.01	0.44	0.00	0.71	0.34	8.85
UZHPandas_selection_lens_cot	578	6.74	1.10	38.16	4.54	5.88	0.32	0.00	0.87	0.14	8.32
UZHPandas_selection_sle	578	6.07	2.57	35.30	0.65	0.98	0.43	0.00	0.56	0.78	8.17
UZHPandas_selection_sle_cot	578	6.49	1.03	38.38	4.76	6.26	0.30	0.00	0.89	0.14	8.30
UZHPandas_simple	578	11.24	5.67	39.28	0.88	0.98	0.52	0.00	0.53	0.62	8.45
UZHPandas_simple_cot	578	13.74	3.38	39.59	3.44	2.67	0.41	0.00	0.76	0.12	8.61
UZHPandas_simple_inter_def	578	21.36	3.13	38.29	1.93	0.99	0.46	0.00	0.69	0.33	8.86
UZHPandas_selection_lens_1	578	7.79	3.65	36.72	0.72	0.98	0.46	0.00	0.54	0.73	8.25
YOUR_TEAM_DistilBERT	578	5.85	13.56	19.00	1.03	3.00	0.95	0.00	0.22	0.11	8.65
YOUR_TEAM_METHOD	578	13.65	19.77	12.12	1.00	1.00	1.00	0.99	0.00	0.00	8.80
YOUR_TEAM_T5	578	13.18	10.66	28.92	1.12	1.10	0.72	0.03	0.34	0.37	9.06

Table 13
Submission results for Tasks 3.2 Rewrite this: Simplification of scientific abstracts. Our participation is UBO_Phi4mini.

run name	count	FKGL	BLEU	SARI	Compression ratio	Sentence splits	Levenshtein similarity	Exact copies	Additions proportion	Deletions proportion	Lexical complexity score
Identity	103	13.64	12.81	21.36	1.00	1.00	1.00	1.00	0.00	0.00	8.88
References	103	8.91	100.00	100.00	0.67	1.04	0.60	0.00	0.23	0.53	8.66
UBO_Task3.1_Phi4mini-l	103	9.96	38.41	10.01	1.29	2.11	0.55	0.00	0.24	0.51	9.03
UBO_Task3.1_Phi4mini-ls	103	8.45	38.79	5.53	1.21	1.75	0.43	0.00	0.40	0.63	8.53
AIIRLab_Task3.2_llama-3-8b_run1	103	9.07	43.44	11.73	1.01	1.38	0.51	0.00	0.37	0.56	8.57
AIIRLab_Task3.2_llama-3-8b_run2	103	10.22	42.19	7.99	1.31	1.38	0.48	0.00	0.53	0.52	8.44
AIIRLab_Task3.2_llama-3-8b_run3	103	10.17	43.21	11.03	1.15	1.47	0.52	0.00	0.40	0.51	8.66
Elsevier@SimpleText_Task3.2_run2	103	11.01	42.47	10.54	1.04	1.22	0.51	0.00	0.38	0.55	8.60
Elsevier@SimpleText_Task3.2_run5	103	12.08	42.15	10.96	1.04	1.15	0.52	0.00	0.36	0.53	8.75
Sharingans_task3.2_finetuned	103	11.53	40.96	18.29	1.20	1.39	0.65	0.00	0.24	0.34	8.80
UAms_Task3-2_Cochrane_BART_Doc	103	14.46	33.51	9.39	0.65	0.58	0.54	0.04	0.06	0.53	8.80
UAms_Task3-2_Cochrane_BART_Par	103	16.53	31.58	15.40	1.08	0.80	0.67	0.04	0.15	0.32	8.81
UAms_Task3-2_GPT2_Check_Abs	103	12.85	36.47	13.12	0.91	0.92	0.59	0.00	0.18	0.45	8.73
UAms_Task3-2_GPT2_Check_Snt	103	11.57	30.71	15.24	1.54	1.70	0.78	0.00	0.27	0.13	8.77
UAms_Task3-2_Wiki_BART_Doc	103	15.68	26.50	15.11	1.51	1.14	0.76	0.01	0.25	0.11	8.79
UAms_Task3-2_Wiki_BART_Par	103	13.11	23.92	19.49	1.39	1.37	0.81	0.01	0.11	0.10	8.86
YOUR_TEAM_Task3.2_DistilBERT	103	0.00	28.28	0.00	0.00	0.00	0.00	0.00	0.00	1.00	10.82
YOUR_TEAM_Task3.2_METHOD	103	0.00	28.28	0.00	0.00	0.00	0.00	0.00	0.00	1.00	10.82
YOUR_TEAM_Task3.2_METHOD	103	0.00	28.28	0.00	0.00	0.00	0.00	0.00	0.00	1.00	10.82
YOUR_TEAM_Task3.2_METHOD	103	0.00	28.28	0.00	0.00	0.00	0.00	0.00	0.00	1.00	10.82
YOUR_TEAM_Task3.2_METHOD	103	0.00	28.28	0.00	0.00	0.00	0.00	0.00	0.00	1.00	10.82
YOUR_TEAM_Task3.2_T5	103	0.00	28.28	0.00	0.00	0.00	0.00	0.00	0.00	1.00	10.82