Enhancing Scientific Document Simplification through Adaptive Retrieval and Generative Models

Artemis Capari, Hosein Azarbonyad, Zubair Afzal and Georgios Tsatsaronis

Elsevier, Amsterdam

Abstract

The CLEF SimpleText Lab focuses on identifying pertinent sections from an vast collection of scientific papers in response to general queries, recognizing and explaining complex terminology in those sections, and ultimately, making the sections easier to understand. The first task is akin to the ad-hoc retrieval task where the objective is to find relevant sections based on a query/topic, but it also requires ranking models to evaluate documents according to their readability and complexity, alongside relevance. The third task is centered around simplifying sentences from scientific abstracts.

In this paper, we outline our strategy for creating a ranking model to address the first task and our methods for employing GPT-3.5 in a zero-shot manner for the third task. To create the ranking model, we initially assess the performance of several models on a proprietary test collection built using scientific papers from various science fields. Subsequently, we fine-tune the top-performing model on a large set of unlabelled documents using the Generative Pseudo Labeling approach. We further experiment with generating new queries using the provided queries, topics, and abstracts to generate a search query. Our approach's primary contribution and findings indicate that a bi-encoder model, trained on the MS-Marco dataset and fine-tuned further on a vast collection of unlabelled scientific sections, yields the best results on the proprietary dataset, specifically designed for the scientific passage retrieval task.

For the third task, we aim to test the limits of a zero-shot Large Language Model (LLM), namely GPT-3.5, by experimenting with various zero-shot and few-shot prompts on both sentence-level and abstract-level. We find that few-shot prompting results in a higher performance on BLEU and SARI, but leads to a higher FKGL, as the simplified sentences in the provided test set have a higher FKGL as well. Conversely, lower FKGL can be obtained with zero-shot prompting, but will result in lower BLEU and SARI scores as well.

Keywords

Information Retrieval, Scientific Documents, Domain Adaptation, Scholarly Document Processing

1. Introduction

The scientific community often utilizes specialized and complex terminology, making scientific texts difficult to comprehend for the general audience [1]. With continuous developments in many disciplines, even researchers and scientists find it increasingly difficult to stay up to date with novel content and technical concepts. Studies have shown that the readability of scientific literature is declining over time [2]. This trend presents both challenges and opportunities for researchers and publishers to improve the readability of complex scientific information for a broader audience.

SimpleText Lab [3] is dedicated to addressing these challenges by making scientific content

CLEF 2024: Conference and Labs of the Evaluation Forum, September 9-12, 2024, Grenoble, France

CEUR Workshop Proceedings (CEUR-WS.org)

more accessible. The lab's primary objectives include identifying relevant passages in response to user queries [4], explaining complex terminology within these passages [5], and ultimately simplifying the text to improve readability [6]. The initial step in this process is a passage retrieval task known as "What is in (or out)?", where the goal is to retrieve all passages pertinent to a given query or topic, which can then be used to create a simplified summary. In addition to relevance, ranking models must also consider the complexity of passages, prioritizing those that are easier to understand.

Current state-of-the-art ranking models are based on semantic matching using either cross-encoder or bi-encoder architectures, or a combination of both [7]. These models are typically trained on publicly available datasets like MS-Marco [8], which do not include scientific documents. Since the SimpleText Lab's retrieval task and associated training/evaluation sets are centered around scientific literature, existing ranking models may underperform in this context due to the complexity and specialized terminology of scientific documents.

In this paper, we use our findings from our SimpleText participation of the previous year [9, 10] to further expand our experiments. In our previous participation, we fine-tuned pre-trained state-of-the-art ranking models [11, 12] on a set of unlabelled scientific documents using a domain adaptation technique known as *Generative Pseudo Labeling* (*GPL*) [13] to retrieve relevant documents for the SimpleText task. This method proved to be successful as our submissions dominated the top of the scoreboard in the 2023 SimpleText Task 1 [9, 10]. Therefore, we consider the models to be sufficiently fitted to the data, and we rather aim to improve the input given to the ranking model by generating new search queries using the provided queries, topics, and abstracts.

In addition to the first task, we also participated in the third task, "Rewrite this!", where the objective is to simplify passages from scientific abstracts given a query [6]. We aim to test the limits of prompt engineering on the text simplification task with GPT-3.5, developing prompts with instructions at varying levels of detail, comparing zero-shot versus few-shot prompting, and providing additional context for the sentence/abstract to be simplified within the prompt.

The remainder of this paper is organized as follows: Section 2 reviews related work, Section 3 details the technical aspects of our system, Sections 4 and 5 present our empirical evaluations, and Section 6 discusses the limitations of our current approach and suggests directions for future research.

2. Related Work

In this section, we review the related work on the passage retrieval (SimpleText Task 1 [4]) and text simplification (SimpleText Task 3 [6]) tasks.

2.1. Passage Retrieval

The field of information retrieval (IR) has seen significant advancements with the introduction of dense retrieval models. These models utilize fixed-length dense vector representations to depict both queries and documents [14]. This approach enables efficient and precise extraction of pertinent information from large text corpora, achieved by calculating the similarity score between vectors representing queries and documents. Compared to conventional sparse retrieval models like BM25 [15], dense retrieval models have exhibited superior performance in diverse tasks, including document ranking and open-domain question answering.

Bi-encoders and cross-encoders are two variants of dense retrieval models. Despite sharing the common objective of capturing the semantic meaning of queries and documents into dense vector representations, these two models differ in their neural network architecture. Bi-encoders operate by independently encoding the query and document with two separate encoders into dense vectors. These vectors are then compared using a similarity function, resulting in a relevance score. A prominent example of bi-encoders is the Dense Passage Retrieval (DPR) model [14]. DPR employs a two-stage retrieval process. Initially, a broad set of passages is retrieved using sparse techniques. Subsequently, each passage is represented as a dense vector using a pre-trained language model like BERT [16]. The query is mapped to a dense vector representation as well. The final ranking of the passages is determined by the cosine similarity between vectors representing the query and passage.

On the other hand, cross-encoders use a single encoder to encode the query and document into a shared embedding space. The documents are then ranked based on the similarity score computed between this shared embedding and the learned representation of the positive document. Cross-encoders can capture more intricate interactions between the query and the document. However, they are computationally more demanding since they require a unique embedding for each query-document pair. In contrast, bi-encoders separately encode queries and documents, thus requiring only a single document corpus for all queries [11]. Consequently, cross-encoders are typically used only as re-rankers [17, 18, 19, 20, 21, 22].

2.2. Text Simplification

When simplifying a text, multiple aspects are to be considered. One aspect is the Lexical Simplification (LS), where complex terminology is replaced with simpler synonyms or explanations. However, a sentence could still be grammatically complex, and therefore the Syntactic Simplification (SS) should also be considered [23].

The first attempts at automatic LS were rule-based approaches where texts were analyzed and complex terms were identified, after which they were replaced with their most frequently used synonyms [24]. Later rule-based LS approaches aimed to be more context-aware, using methods that better capture semantic meaning such as context vectors [25, 26] or employing a BERT model to generate and rank substitutions for complex words [27]. Data-driven LS uses scientific methods and machine learning techniques to learn LS rules from large datasets, such as English Wikipedia and Simple English Wikipedia. [28, 23, 29].

Syntactic simplification also consists of both rule-based and data-driven approaches. Early rule-based methods used handcrafted rules to split long sentences and simplify them, but often failed due to complexities such as crossed dependencies and ambiguities [30, 23]. Improvements were made by integrating a parser, Lightweight Dependency Analyzer (LDA), to learn simplification rules from a corpus of sentences and their simplified versions [30, 23]. Subsequent work focused on preserving text cohesion, syntactic dependencies, and multilingual applications, but struggled to generalize across different sentence structures and languages [31, 32, 33, 34]. Data-driven methods, using large corpora and statistical models, enhanced the flexibility and robustness of text simplification [35, 36].

Deep learning techniques have advanced the field of text simplification in recent years. For instance, Sequence-to-sequence (Seq2Seq) models with attention mechanisms [37, 38], have been adapted for text simplification tasks. Nisioi et al. [39] demonstrated the effectiveness of neural models in generating simplified text by training on large-scale datasets. These models can capture complex linguistic patterns and produce more fluent and coherent simplified sentences compared to traditional methods. Advancements in pre-trained language models [40] have further improved automatic text simplification. These models, pre-trained on vast amounts of text data, can be fine-tuned for specific tasks, including text simplification Martin et al. [41].

3. Methodology

In this section, we outline the specific methodologies employed in this paper for the passage retrieval and text simplification tasks in the context of SimpleText lab. In the first task [4], our aim is to adapt a passage retrieval model to the scientific domain and improve its performance. For this purpose, we construct a validation dataset using a selection of scientific texts. We also experiment with the Generative Pseudo Labeling (GPL) approach for unsupervised domain adaptation. Finally, we focus on creating effective search queries using GPT-3.5 to improve the performance of the retrieval model. For the text simplification task, we explore various prompt-engineering techniques on GPT-3.5 to simplify a given text. The following sections provide an in-depth description of these tasks and the rationale behind our chosen methods.

3.1. Task 1

In order to fine-tune and test our models, we initially construct a validation dataset utilizing a selection of scientific texts annotated by subject matter experts. Subsequently, we use part of this dataset, along with a large collection of scientific documents to fine-tune a dense-retrieval model. This serves to make the model more suitable for scientific passage retrieval. We also experiment with using LLM-generated search queries using abstracts as context.

3.1.1. Test Collection

The test collection [42, 43] is created using 100 queries dispersed through 20 distinct scientific disciplines [1]. Each query is specifically chosen to represent a recognized scientific concept, thus enabling the collection of credible and pertinent passages. Following the selection of queries, we employ the well-established pooling technique to retrieve candidate documents for annotation for each query. Five distinct models (comprising two lexical matching, two bi-encoders, and one cross-encoder) are selected for pool construction. These models are chosen based on their performance on a small subset, or to ensure a variety of models, which in turn guarantees a diversity of documents within the pool. We select 50 documents per query using the pooling

[1] Including Genetics and Molecular Biology, Computer Science, Economics, Agricultural and Biological Sciences, Biochemistry, Econometrics and Finance, Toxicology and Pharmaceutical Science, Chemical Engineering, Veterinary Science and Veterinary Medicine, Chemistry, Materials Science, Earth and Planetary Sciences, Engineering, Food Science, Immunology and Microbiology, Mathematics, Nursing and Health Professions, Medicine and Dentistry, Neuroscience, Pharmacology, Psychology, Physics and Astronomy, Social Science

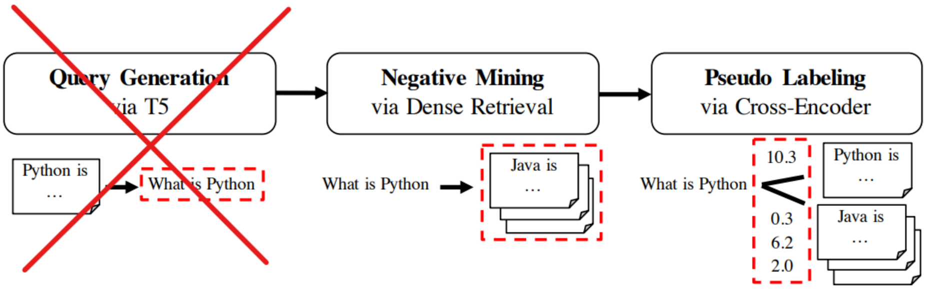

Figure 1: Generative Pseudo Labeling (GPL) for training domain-adapted dense retriever [13]

approach. These documents are then classified by domain experts as "relevant", "partially relevant", or "non-relevant". This curated dataset serves as the benchmark for assessing the performance of various ranking models [2].

3.1.2. GPL

The Generative Pseudo Labeling (GPL) approach, originally introduced in [13], is an unsupervised domain adaptation technique. This framework harnesses the architecture of a pre-trained generative model to generate pseudo labels for unlabeled data within the target domain, creating a training set suitable for supervised learning. This method has shown superior performance compared to other unsupervised domain adaptation methods across various benchmark datasets, and it has achieved state-of-the-art performance in the unsupervised domain adaptation of dense retrieval.

The importance of large data sets in training dense retrieval methods has been frequently emphasized in previous research [16, 14, 7]. Given our manually annotated dataset, comprised of only 5,000 snippets stemming from a set of 100 queries, we face a potential limitation. However, we possess a vast reservoir of unlabeled scientific documents, including research articles, that could provide an abundance of snippets and potential queries. These could be labeled through GPL, based on their relevance, to fine-tune and adapt the extant ranking models to the task of scientific document retrieval.

We adapt the GPL framework to suit our specific needs by firstly eliminating the query generation component (See Figure 1). Instead, we select a known set of scientific concepts per domain, and subsequently identify all passages that refer to each concept within the documents. This approach is predicated on the idea that the explicit mention of a scientific concept within a document is a strong indicator of the document's relevance to the concept. In this context, each document that mentions a specific concept is considered a positive example. A bi-encoder is then employed to determine negative examples for each query. The GPL framework employs a cross-encoder as a 'teacher model' to fine-tune the underlying bi-encoder model using the collated positive and negative documents. This process enables the adaptation of the bi-encoder model for our specific application - the ranking of scientific documents.

[2] The benchmark set can be found here https://github.com/acapari/KAPR

Table 1
Details on fine-tuning of various models

Model Name	Bi-Encoder	Queries	Documents	Batch Size	Training Steps	Epochs
MS-DB-v4-GPL-CS	msmarco-distilbert-base-v4	218 (10 golden)	23670	16	15000	1
MS-DB-tas-b-GPL-CS	msmarco-distilbert-base-tas-b	218 (10 golden)	23670	16	15000	1
MS-DB-v4-GPL-all	msmarco-distilbert-base-v4	4637 (80 golden)	893110	32	280000	1
MS-DB-tas-b-GPL-all	msmarco-distilbert-base-tas-b	4637 (80 golden)	893110	32	280000	1

For our use-case, we have fine-tuned two different bi-encoders *msmarco-distilbert-base-v4*[11] (MS-DB-v4) and *msmarco-distilbert-base-tas-b*[12] (MS-DB-tas-b) using a subsection of our benchmark set, spanning 20 different scientific domains, consisting of 4 queries each. We augmented the training set with a large set of unlabeled passages. When testing performance with the remainder of our benchmark set, we found that *msmarco-distilbert-base-tas-b* was most suitable for tasks that require understanding of a wide range of domains. However, as the SimpleText task aims at finding references in Computer Science, we have also fine-tuned the aforementioned models on queries and articles from just the Computer Science and Mathematics domains. Naturally, these models were fine-tuned on far less data (See Table 1). Each of the models were fitted on pseudo labels created with *ms-marco-MiniLM-L-6-v2*, using the Adam Optimizer [44] with a learning rate of $2e-5$ and 1000 warm-up steps.

3.1.3. Generated Search Queries

Several aspects are of importance when aiming to retrieve passages that are relevant for creating a simplified summary around a given topic. One aspect, as described above, is the model used to retrieve the passages. However, even with a high performing model, it is important *how* you ask the model to retrieve those passages. We therefore employ GPT-3.5 to generate search queries in two different set-ups. We first generate new *topics* using the provided abstracts only (See Figure 2). We also generate new *queries* using the provided queries and abstracts with the objective of finding a better search query to highlight a certain aspect of the article (See Figure 3). The generated search queries [3] are then used to retrieve a corpus of ElasticSearch top-k documents, after which the same queries are used to re-rank the corpus with our fine-tuned models.

3.2. Task 3

For the text simplification task, we explore the advantages and limitations of various prompt-engineering techniques on GPT-3.5. We first design several prompts where we simply ask the model to simplify a given sentence/abstract (See Figure 4, 6, and 7). Subsequently, we augment these prompts with more detailed instructions on how to simplify a given input (see Figure 5, 9, and 11).

We also apply few-shot prompting, a frequently used technique that enables in-context learning without the need to update model parameters [45, 46] by adding examples of the desired input and output to the prompt. We use sentences/abstracts and their simplified versions

[3] The queries can be found on https://github.com/acapari/SimpleText_24_T1/

```
Goal:
I have a task to retrieve passages that help understand a given article.

Request:
Your task is to help me write the best possible search query to retrieve
articles that would help understand the provided article.
This query should be concise and focus on the provided topic.
Only provide ONE search query.

Article:
"{abstract}"

Search Query:
```

Figure 2: Topic prompt

```
Goal:
I have a task to retrieve passages that help understand a given article.
We dissect the content of the article into key-topics, and retrieve
passages for those topics.

Request:
I need your help to create the best possible search query for a given
topic in the context of the provided article.
This query should be concise and focus on the provided topic.
Only provide ONE search query.

Topic:
{query_text}

Article:
"{abstract}"

Search Query:
```

Figure 3: Query prompt

from the provided test set as sample input-output pairs in our few-shot prompts. As models are biased by the order of the in-context examples [47], we randomly take n samples from the test set and ensure that the selected samples are not from the same abstract as the input sentence (see Figure 5, 8, 9, and 10).

Finally, we explore two different methods for adding background information that can potentially be used to help identify essential information that should be included in the simplified text. The first method provides additional context to sentence-level simplification prompts by simply including the abstract that the sentence to be simplified is extracted from. This can potentially aid in avoiding overly simplified sentences as it shows the role of the sentence in a bigger context (see Figure 8, 9, 10, and 11). Simplifying a text often involves breaking down and explaining complex concepts. Our second method for adding background information therefore involves a two-step process, where we first design a prompt where the task is to identify key concepts in a given abstract and provide their definitions or more generally known synonyms. We expect this method to aid with lexical simplification, in particular, [48] (see Figure 8). We thus explore and combine the following methods:

- Simple zero-shot prompting (Figure 1, 3, and 4)
- Zero-shot prompting with detailed instructions (Figure 2, 6, and 8)
- Few-shot prompting (Figure 2, 5, 6, and 7)
- Adding background information to the prompt
  - Adding abstract for sentence-level simplification (Figure 5, 6, 7, and 8)
  - Generating and providing definitions/synonyms of key concepts (Figure 5)

4. Experiments

In this section, we describe the details of the runs and the specific models used to produce the results per run for Task 1 and Task 2.

4.1. Task 1

We have applied our models in several settings before selecting the final 10 submitted runs. The selection was made based on the performance on the provided qrels and successes of submissions from the previous year [10]. As shown in Table 2, the rankings were retrieved by taking the top-k documents found for each query from he 2024 SimpleText Task 1 Train Qrels by the Elastic Search API (top-100 for runs 1, 4, 8, and 10, top-500 for runs 2, 5, 7, and 9, and top-1000 for runs 3 and 6). These were then re-ranked using our fine-tuned models. Runs 2 and 7 were obtained with MS-DB-v4-GPL-CS, a *msmarco-distilbert-base-v4* model that was only fine-tuned on Computer Science and Mathematics data, while we used MS-DB-tas-b-GPL-al, a *msmarco-distilbert-base-tas-b* model fine-tuned on all Science Direct Domains, for the remaining runs.

For runs 1, 3, and 7, the top-k documents were retrieved by searching for "query", and then re-ranked using one of our fine-tuned models, again using "query" as the query input, while run 9 simply uses "topic" as query input. A combination of the two inputs, namely "query, topic", was used as the query input for ranking and re-ranking runs 2, 4, 5, and 6. Finally, we included two runs with generated query inputs with run 8 at query-level and run 10 at topic-level, both using a top-100 ElasticSearch corpus and the *MS-DB-tas-b-GPL-all* model.

4.2. Task 3

Experiments conducted for Task 3 revolve around testing the limits of prompt engineering, by comparing performance between simple and very detailed or multi-step prompts, zero-shot and few-shot prompting, and sentence-level and abstract-level.

Table 2
Configurations of official submissions for Task 1

Run	Query Input	Corpus	Model
1	query	ES Top-100	MS-DB-tas-b-GPL-all
2	query, topic	ES Top-500	MS-DB-v4-GPL-CS
3	query	ES Top-1000	MS-DB-tas-b-GPL-all
4	query, topic	ES Top-100	MS-DB-tas-b-GPL-all
5	query, topic	ES Top-500	MS-DB-tas-b-GPL-all
6	query, topic	ES Top-1000	MS-DB-tas-b-GPL-all
7	query	ES Top-500	MS-DB-v4-GPL-CS
8	gen query	ES Top-100	MS-DB-tas-b-GPL-all
9	topic	ES Top-500	MS-DB-tas-b-GPL-all
10	gen topic	ES Top-100	MS-DB-tas-b-GPL-all

Table 3
Configurations of official submissions for Task 3

Run	Prompt	Few-Shot	Level	Two-Step	Uses Abstract
1	1	False	Sentence	False	False
2	2	False	Abstract	False	-
3	3	False	Sentence	False	False
4	4	False	Sentence	False	False
5	2	True	Abstract	False	-
6	5	False	Sentence	True	True
7	6	False	Sentence	False	True
8	7	True	Sentence	False	True
9	8	True	Sentence	False	True
10	6	True	Sentence	False	True
11	8	False	Sentence	False	True
12	5	True	Sentence	True	True

Table 3 presents the configurations of the submitted runs. We will also discuss the results of two additional runs for further comparisons between zero-shot and few-shot prompting. Each prompt can be found in Appendix A. The submission consists of 2 abstract-level runs, namely runs 2 and 5, using the same prompt, but comparing the performance between zero-shot and few-shot prompting. The remaining 8 runs are at sentence level, using prompts with varying levels of complexity, and using few-shot prompting for runs 8, 9, and 10. For sentence-level runs 6-10, the entire abstract is also provided as additional context. We include run 6, which involves a two-step process where we first request GPT-3.5 to identify and explain complex terms found in the abstract. Subsequently, we provide the completion of the first step as additional context in the second prompt, where the task is to simplify a given sentence.

5. Results

In this section, we describe the details of different runs and the results for the passage retrieval (Task 1) and text simplification (Task 3) tasks.

Table 4
Performance of Official Runs on the 2024 SimpleText Task 1 Train Qrels

Run	P@10	R@10	RR@10	nDCG@5	nDCG@10	nDCG@50	nDCG@100
1	**0.612**	**0.103**	**0.799**	**0.584**	**0.555**	0.399	**0.407**
2	0.584	0.088	0.727	0.566	0.550	**0.401**	0.364
3	0.552	0.091	0.761	0.547	0.511	0.369	0.352
4	0.500	0.076	0.666	0.487	0.468	0.356	0.330
5	0.508	0.079	0.657	0.500	0.461	0.353	0.335
6	0.472	0.072	0.697	0.471	0.439	0.337	0.327
7	0.344	0.044	0.470	0.373	0.340	0.227	0.210
8	0.340	0.042	0.502	0.328	0.321	0.236	0.227
9	0.312	0.040	0.451	0.324	0.298	0.205	0.191
10	0.244	0.026	0.309	0.253	0.234	0.160	0.138

5.1. Task 1

5.1.1. Submitted Runs

For Task 1, we have partially selected the runs based on the performance of our submissions of the previous year, where most of our submitted runs were produced using the *MS-DB-v4-GPL-CS* model as it showed higher performance on the provided qrels [10]. However, official results indicated otherwise, as our top-performing runs were obtained using the *MS-DB-tas-b-GPL-all* model [49]. We also found that "query, topic" was the best query input. However, the evaluation on the provided qrels indicates otherwise. Following Table 4, run 1 ranks highest across most metrics, which uses "query" as input and a corpus of top-100 ES documents. Last year's best-performing run, run 2, ranks second on the provided qrels. Increasing the number of ES documents from 100 to 500 has a negative impact on RR@10, $nDCG$@50, and $nDCG$@100, while increasing the corpus to 1000 ES documents worsens performance across *all* metrics. Finally, the results suggest that using "query" as query input performs best, followed by "query, topic", while the generated inputs perform as some of the worst at rank 8 and 10. Query inputs at topic level are the lowest-performing query inputs at rank 9 and 10.

5.1.2. Official Results

As per Table 5, where the results are sorted on the primary measure, nDCG@10, we see that the rankings do not correspond with those of the Train qrels presented in Table 4. While the generated query inputs, run 8 and 10, were ranked as some of the lowest, we see that they perform as some of the best on the Test qrels. We further observe that run 4 performs slightly better than run 10, both using the ES Top-100 corpus and *MS-DB-tas-b-GPL-all*, but using "query, topic" and "gen topic" as query inputs respectively. While the generated topic still obtains better results than most of our other submissions, we thus conclude that the "query", both generated in run 8 and original in run 4, provide an additional level of detail required when retrieving relevant documents. "gen query" results in highest performance, followed by"query, topic" and "gen topic", while "query" and "topic" produce the lowest performing results. Furthermore,

Table 5
Results for CLEF 2024 SimpleText Task 1 on the Test qrels (G01.C1-G10.C1 and T06-T11).

runid	**MRR**	**P@10**	**P@20**	**NDCG@10**	**NDCG@20**	**Bpref**	**MAP**
AIIRLab_Task1_LLaMABiEncoder	0.9444	0.8167	0.5517	0.6170	0.5166	0.3559	0.2304
AIIRLab_Task1_LLaMAReranker2	0.9300	0.7933	0.5417	0.5943	0.5004	0.3495	0.2177
AIIRLab_Task1_LLaMAReranker	0.8944	0.7967	0.5583	0.5889	0.5011	0.3541	0.2200
LIA_vir_title	0.8454	0.6933	0.4383	0.5013	0.3962	0.3594	0.1534
AIIRLab_Task1_LLaMACrossEncoder	0.7975	0.6933	0.5100	0.4745	0.4240	0.3404	0.1970
LIA_vir_abstract	0.7683	0.6000	0.4067	0.4207	0.3504	0.3857	0.1603
UAms_Task1_Anserini_rm3	0.7878	0.5700	0.4350	0.3924	0.3495	0.4010	0.1824
UAms_Task1_Anserini_bm25	0.7187	0.5500	0.4883	0.3750	0.3707	0.3994	0.1972
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
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
AIIRLAB_Task1_CERRF	0.7264	0.5033	0.4000	0.3584	0.3239	0.2204	0.1309
Arampatzis_1.GPT2_search_results	0.6986	0.5100	0.2550	0.3516	0.2462	0.0742	0.0577
UBO_Task1_TFIDFT5	0.7132	0.4833	0.3817	0.3474	0.3197	0.2354	0.1274
LIA_bool	0.7242	0.5233	0.3633	0.3381	0.2891	0.2661	0.1199
Elsevier@SimpleText_task_1_run8	0.7123	0.4533	0.3367	0.3146	0.2752	0.1582	0.0906
Elsevier@SimpleText_task_1_run4	0.6162	0.4300	0.3217	0.3063	0.2681	0.1642	0.1005
Elsevier@SimpleText_task_1_run10	0.5117	0.4067	0.2767	0.2885	0.2365	0.1236	0.0729
AB_DPV_SimpleText_task1_results_FKGL	0.6173	0.3733	0.2900	0.2818	0.2442	0.1966	0.1078
LIA_elastic	0.6173	0.3733	0.2900	0.2818	0.2442	0.3016	0.1325
Ruby_Task_1	0.5470	0.4233	0.3533	0.2756	0.2671	0.1980	0.1110
LIA_meili	0.6386	0.4700	0.2867	0.2736	0.2242	0.2377	0.0833
Elsevier@SimpleText_task_1_run6	0.5333	0.3833	0.3117	0.2633	0.2430	0.1841	0.0973
Tomislav_Rowan_SimpleText_T1_2	0.5444	0.3733	0.2750	0.2443	0.2183	0.0963	0.0601
Elsevier@SimpleText_task_1_run5	0.4867	0.3533	0.2883	0.2408	0.2232	0.1834	0.0943
Elsevier@SimpleText_task_1_run1	0.5589	0.3000	0.3300	0.2247	0.2399	0.1978	0.1018
Elsevier@SimpleText_task_1_run7	0.4026	0.3200	0.2250	0.2168	0.1850	0.1085	0.0565
Elsevier@SimpleText_task_1_run9	0.3868	0.3300	0.2283	0.2105	0.1829	0.1103	0.0590
Elsevier@SimpleText_task_1_run3	0.4733	0.2367	0.2033	0.1853	0.1703	0.1587	0.0714
Elsevier@SimpleText_task_1_run2	0.4193	0.2233	0.2433	0.1803	0.1865	0.1768	0.0820
Sharingans_Task1_marco-GPT3	0.6667	0.0667	0.0333	0.1149	0.0797	0.0107	0.0107
Tomislav_Rowan_SimpleText_T1_1	0.0217	0.0233	0.0150	0.0121	0.0106	0.0062	0.0025
Petra_Regina_simpleText_task_1	0.0026	0.0000	0.0050	0.0000	0.0035	0.0031	0.0007

MS-DB-tas-b-GPL-all always outperforms *MS-DB-v4-GPL-CS*, and the optimal corpus appears to be ES Top-100.

5.2. Task 3

5.2.1. Submitted Runs

When comparing the results in Table 6, we see a correlation between BLUE and SARI, but a trade-off with FKGL. This can be attributed to the fact that BLUE and SARI are metrics that reflect similarity between the predicted output and the reference, while FKGL reflects the required education level to understand the text [50]. As the FKGL score for the provided simplified sentences and abstracts is relatively high, i.e. 13.62 at sentence level and 13.38 at abstract level, a lower FKGL score would naturally lead to a lower performance in BLUE and SARI.

Table 6
Performance of Official Runs on the 2024 SimpleText Task 3 Test Set

FKGL	BLEU	SARI	Run	Prompt	Few-Shot	Level	Two-Step	Uses Abstract
11.54	0.15	36.63	**1**	1	False	Sentence	False	False
12.12	0.12	34.92	**2**	2	False	Abstract	False	-
13.09	**0.25**	**42.57**	**3**	3	False	Sentence	False	False
12.85	0.20	39.00	**4**	4	False	Sentence	False	False
13.26	0.14	36.39	**5**	2	True	Abstract	False	-
13.70	0.21	39.95	**6**	5	False	Sentence	True	True
13.80	0.20	39.31	**7**	6	False	Sentence	False	True
13.74	0.20	39.16	**8**	7	True	Sentence	False	True
13.68	0.21	39.12	**9**	8	True	Sentence	False	True
13.82	0.20	39.05	**10**	6	True	Sentence	False	True
13.70	0.20	38.92	**11**	8	False	Sentence	False	True
13.97	0.19	38.54	**12**	5	True	Sentence	True	True

This is further highlighted by the runs using few-shot prompting, where the provided samples were used. The examples used in the prompt were of a higher education level, and therefore the output sentences/abstracts are of a similar level. High FKGL scores were also obtained with certain zero-shot prompts, namely runs 5 and 11. Run 5 includes a two-step process, and run 8 has a very detailed prompt, while both include the entire abstract as additional context. We suspect that these were factors that contributed to the generation of relatively complex sentences instead of real simplifications. Shorter prompts containing fewer details and less additional context on the other hand, lead to simpler generated sentences. As we observe with runs 1, 3, and 4, which were generated with the simplest zero-shot prompts, have some of the lowest FKGL scores. Run 3 in particular has a relatively low FKGL with the highest BLEU and SARI scores.

If we compare the performance of zero-shot prompts with their few-shot counterparts, i.e. run 6 vs. run 12, run 7 vs. run 10, and run 8 vs run 11, we see that adding examples positively impacts all metrics. When comparing runs 2 and 5 however, we see that adding examples to the prompt negatively impacts FKGL on abstract level. This might indicate that the LLM is not able to generalize the simplification task on abstract level when multiple abstracts from potentially different articles are given as examples.

Overall, our results indicate that the off-the-shelf LLM performs well in simplifying scientific text, particularly at the sentence level. Interestingly, providing additional context around the sentences does not enhance its performance in this task. This could be because the additional context diverts the LLM's focus from the individual sentence. When more context is provided, the LLM may attempt to integrate information from multiple sentences, which can negatively affect the quality of the simplified sentences and lead to more complex sentences.

Table 7
Results for CLEF 2024 SimpleText Task 3.1 sentence-level text simplification (task number removed from the run_id) on the test set

run_id	**count**	**FKGL**	**SARI**	**BLEU**	**Compression ratio**	**Sentence splits**	**Levenshtein similarity**	**Exact copies**	**Additions proportion**	**Deletions proportion**	**Lexical complexity score**
References	578	8.86	100	100	0.7	1.06	0.6	0.01	0.27	0.54	8.51
Identity	578	13.65	12.02	19.76	1	1	1	1	0	0	8.8
Elsevier@SimpleText_Task3.1_run1	578	10.33	43.63	10.68	0.87	1.06	0.59	0.00	0.45	0.53	8.39
Elsevier@SimpleText_Task3.1_run4	577	11.73	43.14	12.08	0.85	1.00	0.63	0.00	0.37	0.50	8.54
Elsevier@SimpleText_Task3.1_run8	577	12.40	42.95	12.35	0.90	1.02	0.63	0.00	0.35	0.50	8.66
Elsevier@SimpleText_Task3.1_run6	577	12.65	42.88	11.76	0.95	1.00	0.64	0.00	0.38	0.47	8.63
Elsevier@SimpleText_Task3.1_run7	577	12.55	42.87	12.20	0.87	1.00	0.63	0.00	0.35	0.51	8.67
Elsevier@SimpleText_Task3.1_run9	577	12.53	42.61	12.15	0.87	1.00	0.63	0.00	0.35	0.50	8.67
Elsevier@SimpleText_Task3.1_run3	577	11.50	42.58	15.75	0.76	0.98	0.68	0.00	0.23	0.46	8.68
Elsevier@SimpleText_Task3.1_run10	577	12.57	42.49	11.91	0.91	1.02	0.63	0.00	0.34	0.50	8.67
AIIRLab_Task3.1_llama-3-8b_run1	578	8.39	40.58	7.53	0.90	1.37	0.56	0.00	0.48	0.58	8.45
AIIRLab_Task3.1_llama-3-8b_run3	578	9.47	40.36	6.26	1.17	1.52	0.53	0.00	0.53	0.56	8.51
AIIRLab_Task3.1_llama-3-8b_run2	578	10.33	39.76	5.46	1.03	1.19	0.51	0.00	0.60	0.56	8.34
UZH_Pandas_Task3.1_simple_with_cot	578	13.74	39.59	3.38	3.44	2.67	0.41	0.00	0.76	0.12	8.61
UZH_Pandas_Task3.1_simple	578	11.24	39.28	5.67	0.88	0.98	0.52	0.00	0.53	0.62	8.45
Sharingans_task3.1_finetuned	578	11.39	38.61	18.18	0.83	1.07	0.77	0.11	0.16	0.32	8.70
UZH_Pandas_Task3.1_selection_with_sle_cot	578	6.49	38.38	1.03	4.76	6.26	0.30	0.00	0.89	0.14	8.30
UZH_Pandas_Task3.1_simple_with_intermediate_definitions	578	21.36	38.29	3.13	1.93	0.99	0.46	0.00	0.69	0.33	8.86
UZH_Pandas_Task3.1_selection_with_lens_cot	578	6.74	38.16	1.10	4.54	5.88	0.32	0.00	0.87	0.14	8.32
UZH_Pandas_Task3.1_5Y_target_with_cot	578	6.39	37.95	0.97	4.73	6.25	0.30	0.00	0.89	0.14	8.30
UZH_Pandas_Task3.1_selection_with_lens	578	21.29	37.79	2.71	1.97	1.01	0.44	0.00	0.71	0.34	8.85
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
UZH_Pandas_Task3.1_selection_with_lens_1	578	7.79	36.72	3.65	0.72	0.98	0.46	0.00	0.54	0.73	8.25
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
UZH_Pandas_Task3.1_5Y_target_with_intermediate_definitions	578	19.30	36.53	2.27	1.76	1.01	0.45	0.00	0.70	0.41	8.87
UZH_Pandas_Task3.1_selection_with_sle	578	6.07	35.30	2.57	0.65	0.98	0.43	0.00	0.56	0.78	8.17
UZH_Pandas_Task3.1_5Y_target	578	5.94	34.91	2.29	0.66	0.99	0.43	0.00	0.57	0.78	8.17
UBO_RubyAiYoungTeam_Task3.2	578	8.76	34.40	15.37	0.60	1.22	0.69	0.03	0.05	0.44	8.71
SONAR_Task3.1_SONARnonlinreg	578	13.14	32.12	18.41	0.97	1.01	0.93	0.13	0.11	0.13	8.73
UAms_Task3-1_GPT2_Check	578	11.47	29.91	15.10	1.02	1.23	0.87	0.14	0.17	0.14	8.68
UAms_Task3-1_GPT2	578	10.91	29.73	13.07	1.30	1.50	0.79	0.06	0.29	0.12	8.63
YOUR_TEAM_Task3.1_T5	578	13.18	28.92	10.66	1.12	1.10	0.72	0.03	0.34	0.37	9.06
UAms_Task3-1_Wiki_BART_Snt	578	12.13	27.45	21.56	0.85	0.99	0.89	0.32	0.02	0.16	8.73
YOUR_TEAM_Task3.1_DistilBERT	578	5.85	19.00	13.56	1.03	3.00	0.95	0.00	0.22	0.11	8.65
UAms_Task3-1_Cochrane_BART_Snt	578	13.22	18.45	19.21	0.95	0.99	0.96	0.59	0.02	0.07	8.77
YOUR_TEAM_Task3.1_METHOD	578	13.65	12.12	19.77	1.00	1.00	1.00	0.99	0.00	0.00	8.80

5.2.2. Official Results

As per Table 7, where the results are sorted on SARI, we see that our submitted runs (e.g. *Elsevier*) dominate the top of the scoreboard on sentence-level simplification. We observe that the results differ from our evaluation on the provided test set presented in Table 6, as run 1 ranked relatively low on SARI, while it ranks highest in the official results. However, it still obtains the lowest FKGL score.

The rankings show a correlation between the simplicity of the prompt, and the simplicity of the generated sentences considering the runs with the simplest prompts, i.e. runs 1, 3, and 4,

Table 8
Results for CLEF 2024 SimpleText Task 3.2 abstract-level text simplification (task number removed from the run_id) on the test set

run_id	count	FKGL	SARI	BLEU	Compression ratio	Sentence splits	Levenshtein similarity	Exact copies	Additions proportion	Deletions proportion	Lexical complexity score
References	103	8.91	100.00	100.00	0.67	1.04	0.60	0.00	0.23	0.53	8.66
Identity	103	13.64	12.81	21.36	1.00	1.00	1.00	1.00	0.00	0.00	8.88
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
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
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

obtain the lowest FKGL scores. Run 3 obtains the highest BLEU score at 15.75, indicating that prompt 6 produces the sentences most similar to the test set.

When comparing the performance of runs 7 and 10, where zero-shot and few-shot versions of prompt 6 were used respectively, we see that the zero-shot version of the prompt performs better. This indicates that the references used in the test set were of higher simplicity than the provided examples. The same observation can be made on abstract level in Table 8, where run 2 and run 5 are zero-shot and few-shot versions of prompt 2 respectively.

While few-shot prompting is typically used to boost performance, we can infer why our submissions using zero-shot prompting obtain higher performance, as Table 7 shows that the FKGL of the reference set is 8.86, while the FKGL of the references in the 2024 SimpleText Task 3 Test Set is 13.62. The examples used in our few-shot prompts were thus more complex than the ones used in the official evaluation, which is in turn reflected in the performance of the sentences generated with these few-shot prompts.

6. Conclusion

Building on the success of our participation in the 2023 SimpleText Task 1 [9, 10], where we fine-tuned ranking models on a large collection of unlabeled scientific documents using Generative Pseudo-Labeling (GPL) [13], our objective for the 2024 SimpleText Task 1 [4] was to enhance the search queries provided as input to these ranking models. We generated these search queries with GPT-3.5 on both query and topic level, using article abstracts as context.

While our submissions using this method achieved high rankings on the scoreboard in the 2023 SimpleText Task 1 [9, 10], our models did not outperform those of other teams in the 2024 SimpleText Task 1 [4]. We hypothesize that this is due to the fact that the pool of rankings used to create the reference set differs from the previous year, as it consisted of rankings from many lexical search methods, while there was an increased use of semantic search models and generative methods in the current year.

Furthermore, we observed that our submissions using generated search queries outperformed those utilizing traditional search queries. Specifically, the *distilbert-base-tas-b* model, fine-tuned via GPL on a vast collection of scientific documents and employed to re-rank the top 500 documents retrieved by an Elastic Search system, demonstrated superior performance when combined with query-level generated search queries.

Furthermore, we employed various prompt-engineering techniques for the SimpleText simplification task [6], resulting in the highest-ranking performances on the sentence-level simplification task. While the success of our submission can be largely attributed to the inherent capabilities of the GPT-3.5 model, it is important to explore what methods can be used to best exploit GPT-3.5 for text simplification tasks nonetheless. Our findings indicate that the simplest prompts, wherein we requested sentence simplification without additional instructions or examples, yielded the best FKGL and BLEU performances. Conversely, runs generated with few-shot prompts did not perform as well, particularly on FKGL. This can be attributed to the complexity of sentences in the provided test set, compared to the significantly simpler sentences in the reference set of the official evaluation, which had a lower FKGL. Consequently, using the provided set as few-shot examples led to the generation of more complex sentences.

References

[1] Y. Jin, M.-Y. Kan, J. P. Ng, X. He, Mining scientific terms and their definitions: A study of the acl anthology, in: EMNLP, 2013, pp. 780–790.

[2] P. Plavén-Sigray, G. J. Matheson, B. C. Schiffler, W. H. Thompson, The readability of scientific texts is decreasing over time, Elife 6 (2017) e27725.

[3] L. Ermakova, E. SanJuan, S. Huet, H. Azarbonyad, G. M. Di Nunzio, F. Vezzani, J. D'souza, S. Kabongo, H. B. Giglou, Y. Zhang, et al., Overview of CLEF 2024 SimpleText track on improving access to scientific texts, in: CLEF, 2024.

[4] E. SanJuan, et al., Overview of the CLEF 2024 SimpleText task 1: Retrieve passages to include in a simplified summary, CEUR Workshop Proceedings, 2024.

[5] G. M. D. Nunzio, et al., Overview of the CLEF 2024 SimpleText task 2: Identify and explain difficult concepts, CEUR Workshop Proceedings, 2024.

[6] L. Ermakova, et al., Overview of the CLEF 2024 SimpleText task 3: Simplify scientific text, CEUR Workshop Proceedings, 2024.

[7] N. Thakur, N. Reimers, A. Rücklé, A. Srivastava, I. Gurevych, Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models, arXiv preprint arXiv:2104.08663 (2021).

[8] T. Nguyen, M. Rosenberg, X. Song, J. Gao, S. Tiwary, R. Majumder, L. Deng, Ms marco: A human generated machine reading comprehension dataset, choice 2640 (2016) 660.

[9] L. Ermakova, E. SanJuan, S. Huet, H. Azarbonyad, O. Augereau, J. Kamps, Overview of the CLEF 2023 SimpleText Lab: Automatic simplification of scientific texts, in: CLEF, Springer, 2023, pp. 482–506.

[10] A. Capari, H. Azarbonyad, G. Tsatsaronis, Z. Afzal, Elsevier at SimpleText: Passage retrieval by fine-tuning GPL on scientific documents (2023).

[11] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese BERT-Networks, in: EMNLP, 2019.

[12] S. Hofstätter, S.-C. Lin, J.-H. Yang, J. Lin, A. Hanbury, Efficiently teaching an effective dense retriever with balanced topic aware sampling, in: SIGIR, 2021, pp. 113–122.

[13] K. Wang, N. Thakur, N. Reimers, I. Gurevych, Gpl: Generative pseudo labeling for unsupervised domain adaptation of dense retrieval, arXiv preprint arXiv:2112.07577 (2021).

[14] V. Karpukhin, B. Oğuz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, W.-t. Yih, Dense passage retrieval for open-domain question answering, arXiv preprint arXiv:2004.04906 (2020).

[15] X. Wang, C. Macdonald, I. Ounis, Improving zero-shot retrieval using dense external expansion, Information Processing & Management 59 (2022) 103026.

[16] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

[17] R. Nogueira, K. Cho, Passage re-ranking with bert, arXiv preprint arXiv:1901.04085 (2019).

[18] R. Nogueira, W. Yang, J. Lin, K. Cho, Document expansion by query prediction, arXiv preprint arXiv:1904.08375 (2019).

[19] R. Nogueira, Z. Jiang, J. Lin, Document ranking with a pretrained sequence-to-sequence model, arXiv preprint arXiv:2003.06713 (2020).

[20] S. MacAvaney, A. Yates, A. Cohan, N. Goharian, CEDR: Contextualized embeddings for document ranking, in: SIGIR, 2019, pp. 1101–1104.

[21] S. MacAvaney, F. M. Nardini, R. Perego, N. Tonellotto, N. Goharian, O. Frieder, Efficient document re-ranking for transformers by precomputing term representations, in: SIGIR, 2020, pp. 49–58.

[22] C. Li, A. Yates, S. MacAvaney, B. He, Y. Sun, Parade: Passage representation aggregation for document reranking, arXiv preprint arXiv:2008.09093 (2020).

[23] S. S. Al-Thanyyan, A. M. Azmi, Automated text simplification: a survey, ACM Computing Surveys (CSUR) 54 (2021) 1–36.

[24] J. Carroll, G. Minnen, Y. Canning, S. Devlin, J. Tait, Practical simplification of english newspaper text to assist aphasic readers, in: AAAI-98 workshop on integrating artificial intelligence and assistive technology, Madison, WI, 1998, pp. 7–10.

[25] S. Bott, L. Rello, B. Drndarević, H. Saggion, Can spanish be simpler? lexsis: Lexical

simplification for spanish, in: COLING, 2012, pp. 357–374.

[26] O. Biran, S. Brody, N. Elhadad, Putting it simply: a context-aware approach to lexical simplification, in: ACL, 2011, pp. 496–501.

[27] J. Qiang, Y. Li, Y. Zhu, Y. Yuan, X. Wu, Lexical simplification with pretrained encoders, in: AAAI, volume 34, 2020, pp. 8649–8656.

[28] C. Scarton, Horacio saggion, automatic text simplification. synthesis lectures on human language technologies, april 2017. 137 pages, isbn: 1627058680 9781627058681, Natural Language Engineering 26 (2020) 489–492.

[29] W. Coster, D. Kauchak, Simple english wikipedia: a new text simplification task, in: ACL, 2011, pp. 665–669.

[30] R. Chandrasekar, C. Doran, S. Bangalore, Motivations and methods for text simplification, in: COLING 1996 Volume 2: The 16th International Conference on Computational Linguistics, 1996.

[31] A. Siddharthan, Syntactic simplification and text cohesion, Research on Language and Computation 4 (2006) 77–109.

[32] A. Siddharthan, Text simplification using typed dependencies: A comparision of the robustness of different generation strategies, in: The 13th European Workshop on Natural Language Generation, 2011, pp. 2–11.

[33] D. Ferrés, M. Marimon, H. Saggion, A. AbuRa'ed, Yats: yet another text simplifier, in: NLDB, Springer, 2016, pp. 335–342.

[34] C. Scarton, A. P. Aprosio, S. Tonelli, T. M. Wanton, L. Specia, Musst: A multilingual syntactic simplification tool, in: IJCNLP, 2017, pp. 25–28.

[35] Z. Zhu, D. Bernhard, I. Gurevych, A monolingual tree-based translation model for sentence simplification, in: COLING, 2010, pp. 1353–1361.

[36] K. Woodsend, M. Lapata, Learning to simplify sentences with quasi-synchronous grammar and integer programming, in: EMNLP, 2011, pp. 409–420.

[37] I. Sutskever, O. Vinyals, Q. V. Le, Sequence to sequence learning with neural networks, Advances in neural information processing systems 27 (2014).

[38] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, arXiv preprint arXiv:1409.0473 (2014).

[39] S. Nisioi, S. Štajner, S. P. Ponzetto, L. P. Dinu, Exploring neural text simplification models, in: ACL, 2017, pp. 85–91.

[40] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

[41] L. Martin, B. Sagot, E. de la Clergerie, A. Bordes, Controllable sentence simplification, arXiv preprint arXiv:1910.02677 (2019).

[42] A. Capari, H. Azarbonyad, G. Tsatsaronis, Z. Afzal, J. Dunham, Sciencedirect topic pages: A knowledge base of scientific concepts across various science domains, in: SIGIR, 2024.

[43] A. Capari, H. Azarbonyad, G. Tsatsaronis, Z. Afzal, J. Dunham, J. Kamps, Knowledge acquisition passage retrieval: Corpus, ranking models, and evaluation resources, in: CLEF, 2024.

[44] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).

[45] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, Advances in neural information processing systems 33 (2020) 1877–1901.

[46] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al., Llama: Open and efficient foundation language models, arXiv preprint arXiv:2302.13971 (2023).

[47] Z. Zhao, E. Wallace, S. Feng, D. Klein, S. Singh, Calibrate before use: Improving few-shot performance of language models, in: International conference on machine learning, PMLR, 2021, pp. 12697–12706.

[48] P. Lal, S. Ruger, Extract-based summarization with simplification, in: ACL, London, 2002.

[49] S. Eric, S. Huet, K. Jaap, E. Liana, Overview of the clef 2022 simpletext task 1: passage selection for a simplified summary, CEUR Workshop Proceedings, 2022, pp. 2762–2772.

[50] R. Flesch, A new readability yardstick., Journal of applied psychology 32 (1948) 221.

Appendix

A. Task 3 Prompts

```
### TASK ###
Simplify the language used in this sentence from a scientific article so that it can be understood by the general audience.
Focus on simplifying the sentence structure and replacing scientific jargon with everyday language.

### REQUEST ###
- Sentence:
{row.source_snt}

- Simplified Sentence:
```

Figure 4: Prompt 1

```
### TASK ####
You are going to simplify a given abstract intended for an academic audience to a text that is understandable to the
general audience.


### INSTRUCTIONS ####
1. Identify Key Concepts: First, I would identify the main points or key concepts that the article is trying to convey.
This could include its purpose, its methods, its findings or any other relevant information.
2. Simplify Language: Scientific articles often use complex terminology that is specific to the field of study. I would
replace these terms with simpler, more common words that a general audience would understand.
3. Break Down Complex Ideas: If the article contains complex ideas or processes, I would break these down into smaller
parts and explain them one at a time.
4. Avoid Jargon: I would avoid using jargon, unless it’s necessary for understanding the concept. If it is, I would
provide a clear and simple definition.

{
### EXAMPLE 1 ###
- Abstract:
{ex1.abs_source}

- Simplified Abstract:
{ex1.simplified_abs}

### EXAMPLE 2 ###
- Abstract:
{ex2.abs_source}

- Simplified Abstract:
{ex2.simplified_abs}
}

### REQUEST ###
Remember: the goal is not to oversimplify or distort the scientific abstract, but to make it accessible and understandable
to more people.

- Abstract:
{row.abs_source}

- Simplified abstract:
```

Figure 5: Prompt 2

```
Your task is to simplify a given sentence.

### OUTPUT ###
Sentence:
{row.source_snt}

Simplified sentence:
```

Figure 6: Prompt 3

```
### TASK ###
Simplify a given sentence extracted from a scientific article to a sentence that is understandable to the general
audience.

### REQUEST ###
Remember, the goal is to retain the original meaning of the sentence while making it easier for a general audience to
understand.

- Original Sentence:
{row.source_snt}

- Simplified sentence:
```

Figure 7: Prompt 4

Step 1

```
### TASK ###
Identify complex and technical terms from a given scientific abstract that require simplification or explanation in order
to be understood by a general audience.
Provide the complex terms along with their simpler synonym or definition in list format.

Abstract:
{row.abs_source}

Complex Terms:
```

Step 2

```
### TASK ###
Simplify a given sentence extracted from a scientific article to a sentence that is understandable to the general audience.

### INSTRUCTIONS ###
1. **Identify Technical Terms:** Look for scientific or technical terms that may not be commonly understood by a general
audience. Replace these with simpler, more universally understood terms. For example, simplify "co-ingestion" to
"consumption"; "nonsteroidal anti-inflammatory drug (NSAID)" to "nonsteroidal anti-inflammatory drugs".
2. **Simplify Complex Phrases:** Replace complex phrases with simpler ones. For example, simplify "carried on experiments"
to "conducted experiments".
3. **Eliminate Unnecessary Details:** Remove any details or information that is not essential to the main point of the
sentence.
4. **Clarify Statistics and Measurements:** If a sentence includes statistical data or measurements, explain it in a way
that makes it easier to understand. For example, simplify "(0.23 vs 0.45 [F = 4.24, p u003c 0.05])"  to "(0.23 vs 0.45,
with statistical significance)".
5. **Make the Subject Clearer:** Make sure the subject of the sentence is clear. For example, simplify "intervention
participants" to "those who received the CDSS suite".
6. **Use Active Voice:** Try to use active voice instead of passive voice as it is easier to understand.
7. **Break Down Long Sentences:** If the sentence is too long, try to break it down into smaller sentences.
8. **Use Everyday Language:** Instead of scientific jargon, use everyday language whenever possible.
9. **Use context:** Simplify the given sentence, but use the provided 'Source Abstract' if additional context is needed.
10. **Explain Complex Terms:** Replace complex terms with simpler equivalents where possible or provide a definition for
concepts that are essential, but not commonly understood. These terms are provided in 'Complex Terms'.
11. **Simplify Sentence Structure**.


{
### EXAMPLES ###
## EXAMPLE 1 ##
- Source Abstract:
{ex1.abs_source}

- Original Sentence:
{ex1.source_snt}

- Complex Terms:
{ex1.complex_terms}

- Simplified Sentence:
{ex1.simplified_snt}


## EXAMPLE 2 ##
- Source Abstract:
{ex2.abs_source}
                                                        ...
- Original Sentence:
{ex2.source_snt}

- Complex Terms:
{ex2.complex_terms}

- Simplified Sentence:
{ex2.simplified_snt}


## EXAMPLE 3 ##
- Source Abstract:
{ex3.abs_source}

- Original Sentence:
{ex3.source_snt}

- Complex Terms:
{ex3.complex_terms}

- Simplified Sentence:
{ex3.simplified_snt}

}


### REQUEST ###
Remember, the goal is to retain the original meaning of the sentence while making  it easier for a general audience to
understand. Focus on replacing scientific jargon with everyday language and explaining complex, essential terms.

- Source Abstract:
{row.abs_source}

- Original Sentence:
{row.source_snt}

- Complex Terms:
{row.complex_terms}

- Simplified sentence:
```

Figure 8: Prompt 5

```
### TASK ###
Simplify a given sentence extracted from a scientific article to a sentence that is understandable to the general audience.
{

### EXAMPLES ###
## EXAMPLE 1 ##
- Source Abstract:
{ex1.abs_source}

- Original Sentence:
{ex1.source_snt}

- Simplified Sentence:
{ex1.simplified_snt}

## EXAMPLE 2 ##
- Source Abstract:
{ex2.abs_source}

- Original Sentence:
{ex2.source_snt}

- Simplified Sentence:
{ex2.simplified_snt}

## EXAMPLE 3 ##
- Source Abstract:
{ex3.abs_source}

- Original Sentence:
{ex3.source_snt}
                                                              ...
- Simplified Sentence:
{ex3.simplified_snt}


## EXAMPLE 4 ##
- Source Abstract:
{ex4.abs_source}

- Original Sentence:
{ex4.source_snt}

- Simplified Sentence:
{ex4.simplified_snt}


## EXAMPLE 5 ##
- Source Abstract:
{ex5.abs_source}

- Original Sentence:
{ex5.source_snt}

- Simplified Sentence:
{ex5.simplified_snt}}


### INSTRUCTIONS ###
1. **Identify Technical Terms:** Look for scientific or technical terms that may not be commonly understood by a general
audience. Replace these with simpler, more  universally understood terms. For example, "co-ingestion" was simplified to
"consumption"; "nonsteroidal anti-inflammatory drug (NSAID)" was simplified to "nonsteroidal anti-inflammatory drugs".
2. **Simplify Complex Phrases:** Replace complex phrases with simpler ones. For example, "carried on experiments" was
simplified to "conducted experiments".
3. **Eliminate Unnecessary Details:** Remove any details or information that is not essential to the main point of the
sentence. For example, "based on user and tweets characteristics" was removed as it was not essential to understand the
main point.
4. **Clarify Statistics and Measurements:** If a sentence includes statistical data or measurements, explain it in a way
that makes it easier to understand. For example, "(0.23 vs 0.45 [F = 4.24, p u003c 0.05])" was simplified to "(0.23 vs
0.45, with statistical significance)".
5. **Make the Subject Clearer:** Make sure the subject of the sentence is clear. For example, "intervention participants"
was clarified to "those who received the CDSS suite".
6. **Use Active Voice:** Try to use active voice instead of passive voice as it is easier to understand.
7. **Break Down Long Sentences:** If the sentence is too long, try to break it down into smaller sentences.
8. **Use Everyday Language:** Instead of scientific jargon, use everyday language whenever possible.
9. **Use context:** Simplify the given sentence, but use the provided 'Source Abstract' if additional context is needed.

### REQUEST ###
Remember, the goal is to retain the original meaning of the sentence while making it easier for a general audience to
understand.

- Source Abstract:
{row.abs_source}

- Original Sentence:
{row.source_snt}

- Simplified sentence:
```

Figure 9: Prompt 6

```
### TASK ###
Simplify a given sentence extracted from a scientific article to a sentence that is understandable to the general audience.


### INSTRUCTIONS ###
1. **Identify Technical Terms:** Look for scientific or technical terms that may not be commonly understood by a general
audience. Replace these with simpler, more universally understood terms. For example, "co-ingestion" was  simplified to
"consumption"; "nonsteroidal anti-inflammatory drug (NSAID)" was simplified to "nonsteroidal anti-inflammatory drugs".
2. **Simplify Complex Phrases:** Replace complex phrases with simpler ones. For example, "carried on experiments" was
simplified to "conducted experiments".
3. **Eliminate Unnecessary Details:** Remove any details or information that is not essential to the main point of the
sentence. For example, "based on user and tweets characteristics" was removed as it was not essential to understand the
main point.
4. **Clarify Statistics and Measurements:** If a sentence includes statistical data or measurements, explain it in a way
that makes it easier to understand. For example, "(0.23 vs 0.45 [F = 4.24, p u003c 0.05])" was simplified to "(0.23 vs 0.45,
with statistical significance)".
5. **Make the Subject Clearer:** Make sure the subject of the sentence is clear. For example, "intervention participants"
was clarified to "those who received the CDSS suite".
6. **Use Active Voice:** Try to use active voice instead of passive voice as it is easier to understand.
7. **Break Down Long Sentences:** If the sentence is too long, try to break it down into smaller sentences.
8. **Use Everyday Language:** Instead of scientific jargon, use everyday language whenever possible.
9. **Use context:** Simplify the given sentence, but use the provided 'Source Abstract' if additional context is needed.

{
### EXAMPLES ###
## EXAMPLE 1 ##
- Source Abstract:
{ex1.abs_source}

- Original Sentence:
{ex1.source_snt}

- Simplified Sentence:
{ex1.simplified_snt}


## EXAMPLE 2 ##
- Source Abstract:
{ex2.abs_source}

- Original Sentence:
{ex2.source_snt}

- Simplified Sentence:
{ex2.simplified_snt}

## EXAMPLE 3 ##
- Source Abstract:
{ex3.abs_source}

- Original Sentence:
{ex3.source_snt}

- Simplified Sentence:
{ex3.simplified_snt}
}

### REQUEST ###
Remember, the goal is to retain the original meaning of the sentence while making it easier for a general audience to
understand.

- Source Abstract:
{row.abs_source}

- Original Sentence:
{row.source_snt}

- Simplified sentence:
```

Figure 10: Prompt 7

```
### TASK ###
Simplify a given sentence extracted from a scientific article to a sentence that is understandable to the general audience.

{

### EXAMPLES ###
## EXAMPLE 1 ##
- Source Abstract:
{ex1.abs_source}

- Original Sentence:
{ex1.source_snt}

- Simplified Sentence:
{ex1.simplified_snt}

## EXAMPLE 2 ##
- Source Abstract:
{ex2.abs_source}

- Original Sentence:
{ex2.source_snt}

- Simplified Sentence:
{ex2.simplified_snt}


## EXAMPLE 3 ##
- Source Abstract:
{ex3.abs_source}

- Original Sentence:
{ex3.source_snt}

- Simplified Sentence:
{ex3.simplified_snt}
}

### INSTRUCTIONS ###
1. **Identify Technical Terms:** Look for scientific or technical terms that may not be commonly understood by a general
audience. Replace these with simpler, more universally understood terms. For example, "co-ingestion" was simplified to
"consumption"; "nonsteroidal anti-inflammatory drug (NSAID)" was simplified to "nonsteroidal anti-inflammatory drugs".
2. **Simplify Complex Phrases:** Replace complex phrases with simpler ones. For example, "carried on experiments" was
simplified to "conducted experiments".
3. **Eliminate Unnecessary Details:** Remove any details or information that is not essential to the main point of the
sentence. For example, "based on user and tweets characteristics" was removed as it was not essential to understand the
main point.
4. **Clarify Statistics and Measurements:** If a sentence includes statistical data or measurements, explain it in a way
that makes it easier to understand. For example, "(0.23 vs 0.45 [F = 4.24, p u003c 0.05])" was simplified to "(0.23 vs
0.45, with statistical significance)".
5. **Make the Subject Clearer:** Make sure the subject of the sentence is clear. For example, "intervention participants"
was clarified to "those who received the CDSS suite".
6. **Use Active Voice:** Try to use active voice instead of passive voice as it is easier to understand.
7. **Break Down Long Sentences:** If the sentence is too long, try to break it down into smaller sentences.
8. **Use Everyday Language:** Instead of scientific jargon, use everyday language whenever possible.


### REQUEST ###
Remember, the goal is to retain the original meaning of the sentence while making it easier for a general audience to
understand.

- Source Abstract:
{row.abs_source}

- Original Sentence:
{row.source_snt}

- Simplified sentence:
```

Figure 11: Prompt 8