SimpleText Task 1 Report

Ben Gaudreau and Gabrielle Akers

University of Southern Maine, Portland ME 04101, USA

Abstract. Given the task of text simplification and passage retrieval as outlined in the SimpleText@CLEF 2024 Lab, we summarize the challenges of text simplification. We then discuss previous models submitted to earlier iterations of the SimpleText Lab, and analyze their structure, along with their strengths and weaknesses. With this information, we then propose a model that is both capable of providing decent results while also being technically feasible as relatively new students of machine learning topics.

1 The Task At Hand

The SimpleText@CLEF 2024 Lab is focused on the problem of simplifying academic resources for general audiences. In a time where the Internet has allowed vast quantities of information to reach the public, the complex language and concepts of scientific research continues to be a limiting factor[1]. This lab explores the methods by which these texts may be simplified and rewritten for greater accessibility.

Task 1 of the SimpleText Lab presents a deceptively tricky question: **What is in (or out)?** Given a topic and a query, the model must provide a relevant summary from abstracts in the corpus[1]. The accuracy, complexity, and credibility of each referenced resource plays a factor in the overall quality of the result.

The SimpleText Lab has been iterated upon several times over the last few years, providing information that will be of use when constructing a model to perform the task. These past examples will be discussed in Section 2, and our approach with respect to previous iterations will be detailed in Section 3.

2 Previous Approaches

Many models from previous iterations of this lab have utilized natural language processing (NLP) models such as BERT[2, 3]. While these have historically performed rather well, other submissions in previous years have opted to use other models, to varying levels of success. Most notably, the models provided by Elsevier[4] in 2023, utilizing generative pseudo-labeling (GPL), showed stronger results compared to those that did not.

Other NLP approaches not specific to the SimpleText Lab have demonstrated other methods of passage selection that may be of use when consider a new model. For example, techniques implementing control tokens in a model[5] can provide additional context to a retrieved passage's overall complexity, which could play a role in determining the better of two similarly relevant passages. Similarly, methods of dense passage retrieval (DPR) can be incorporated into a neural models to further improve the speed and accuracy of retrieval over traditional methods[6].

Table 1. Various strengths and weaknesses of models as they relate to the task.

Model	Strengths	Weaknesses
Bi-encoder models[7]	Easy to implement, numerous fine-tuned models for various tasks	Varied performance across different testing conditions
GPL[4]	Strong performance across different testing conditions	Generative inaccuracy present, more difficult to implement
ChatGPT[8]	Can perform simplifications and translations together, can be implemented on top of another model	Requires careful pre-processing, possibility of hallucinations

3 Our Approach

Our proposed model will attempt to incorporate elements of bi-encoder models alongside other techniques to counteract the previously mentioned limitations. Specifically, a ColBERT model[9] used alongside DPR should provide a solid foundation upon which further improvements may be made. Our decision comes from our relative inexperience in programming machine learning tasks, so we plan to use this as an opportunity to develop our skills in this area of computer science.

The methods by which we will compensate for the inaccuracies displayed by bi- and cross-encoder models over subsets of the testing data, as demonstrated in the results of last year's SimpleText submissions[1], is currently unclear. We will look into solutions as we begin our development of the model.

4 Conclusion

In this report, we have laid out the task required of our model as stated by the SimpleText@CLEF 2024 Lab. By analyzing prior attempts at this task, we have furthered our own understanding of the problem and our methods of solving it. Ultimately, we have decided to build off existing work with bi- and cross-encoders and apply them in different structures in the hopes of achieving a better result. Regardless whether or not we succeed in this endeavor, the process of working through this lab will have developed our understanding of machine learning topics overall.

References

[1] Eric SanJuan et al. *Overview of the CLEF 2023 SimpleText Task 1: Passage Selection for a Simplified Summary.* 2023.

[2] Roos Hutter et al. *University of Amsterdam at the CLEF 2023 SimpleText Track.* 2023.

[3] Behrooz Mansouri et al. *AIIR and LIAAD Labs Systems for CLEF 2023 SimpleText.* 2023.

[4] Artemis Capari et al. *Elsevier at SimpleText: Passage Retrieval by Fine-tuning GPL on Scientific Documents.* 2023.

[5] Sweta Agrawal and Marine Carpuat. *Controlling Pre-trained Language Models for Grade-Specific Text Simplification.* 2023. arXiv: 2305.14993 [cs.CL].

[6] Vladimir Karpukhin et al. *Dense Passage Retrieval for Open-Domain Question Answering.* 2020. arXiv: 2004.04906 [cs.CL].

[7] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.* 2019. arXiv: 1810.04805 [cs.CL].

[8] Björn Engelmann et al. *Text Simplification of Scientific Texts for Non-Expert Readers.* 2023.

[9] Carlos Lassance et al. *A Study on Token Pruning for ColBERT.* 2021. arXiv: 2112.06540 [cs.IR].

Project Part 1 Report

Finn Michaud

University of Southern Maine, 96 Falmouth St, Portland, ME 04103, USA

Abstract. This paper provides an analysis of the strategies and technologies that may be used for the upcoming CLEF 2024 SimpleText lab. The focus is on Task 2 ”Identifying and explaining difficult concepts”. The goal of this task is to design an algorithm to gather up to 5 difficult terms from a scientific research paper passage and rank them based on their difficulty. After ranking, a meaningful definition will then be provided to accurately explain the term. This task is based on a problem that can often arise when reading scientific papers, complex, discipline-specific terminology may be used that makes reading the research paper harder for a more broad audience. This task is designed in a way to explore a solution to this issue. This paper will analyze strategies used by past CLEF SimpleText task 2 participants and the technologies that could be effective for the problem at hand.

Keywords: SimpleText Task 2 · NLP · Complexity Spotting.

1 Summary of Related Research Papers for Task 2 of CLEF 2024 SimpleText lab and Strength/Weakness of The Approaches Visualized

1.1 Paper 1: *Overview of the CLEF 2023 SimpleText Task 2: Difficult Concept Identification and Explanation*

This paper gives an overview of “Task 2: What is unclear?” from the CLEF 2023 SimpleText lab [1]. Task 2 revolves around the problem of automatic text simplification in the domain of Computer Science Research papers, using Natural Language Processing to spot complex terms or concepts from these papers. Task 2 may be further split into 2 additional subtasks: Complexity spotting - extracting five complex terms from a passage and rating them on a scale from 0-2, and providing explanations - meaningful descriptions of the terms deemed complex. Complexity in this instance is defined as terms or concepts that require background knowledge to understand, making the goal of the task to not only extract and rate these complex terms but also to provide meaningful explanations.

The paper goes on to provide a summary of the types of approaches participants took in retrospect of the CLEF 2023 SimpleText lab. Of the twelve teams that participated, all the teams employed pretrained models for the complexity spotting such as *YAKE*, *GPT-3*, *BLOOM*, *BLOOMZ*, and others. Of the methods employed, LLM’s provided solid results in the evaluation phase. The evaluation phase involved using BLEU and other tools to evaluate the definitions provided and the complexity proposed by the teams’ approaches. The team with the best results used *GPT-3* with zero-shot and few-shot learning strategies on an auto-regressive version of *GLT-3* - employing prompt engineering as one of their main strategies. Teams with weaker results used *Wikipedia* models for complex term definition, which sometimes failed to define the terms that don’t have Wikipedia pages.

1.2 Paper 2: *CLEF 2023 SimpletText Tasks 2 and 3 Enhancing Language Comprehension Addressing Difficult Concepts and Simplifying Scientific Texts Using GPT, BLOOM, KeyBert, Simple T5 and More*

This paper discusses the types of approaches that could be taken for the CLEF 2023 SimpleText lab for tasks 2 and 3 [2]. For complexity spotting, employing pre-trained models such as *keyBERT*, *YAKE*, *Bloom*, and *Simple T5* are all possible choices that can be used for the task of identifying and extracting complex terms. The results section of this paper showed that *SimpleT5* was one of the most effective models at keyword extraction, scoring 90 percent for correctly identifying difficult terms. As for approaches that were to accurately rate these identified terms, the best approach was a Flesch Reading Ease formula paired with *RAKE* and *YAKE* complex term extraction procedures.

Moving on to effective approaches toward explaining the difficult terms, the approach that yielded the highest semantical match of 70 percent was *SimpleT5*. *SimpleT5* is made out to be an effective approach according to the results that this paper proposes. While these results are solid, the highest of them only account for single-word terms.

1.3 Paper 3: *CLEF2023 SimpleText Task 2, 3: Identification and Simplification of Difficult Terms*

This paper describes approaches taken for Task 2 of the CLEF 2023 SimpleText lab that involved using AI models such as *Bloom*, *GPT-3*, *YAKE*, and *TextRank* [3]. Of the models used by this team, the most success was achieved by using *GPT-3 text-davinci-003* (temperature set to 0.7 and maximum token length at 256) for complexity spotting, rating, and explaining the identified difficult terms. The team believed that *GPT-3* provided the best results by standards of meaningful explanations and accurate complexity spotting. The model that performed the worst for task 2 was *WIKI*, which struggled to provide meaningful definitions of complex terms, often paraphrasing or having gaps in the definitions. This paper also suggests the importance of explanatory data to avoid paraphrasing results. Another thing to note is the inclusion of sample prompts used by the team to gain a better understanding of how to work with these models for effective results.

1.4 Paper 4: *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*

This paper provides a technical explanation of Google's pre-trained *BERT* language representation model [4]. *BERT* is described in this paper as being a powerful tool in Natural Language Processing tasks. In particular, natural language inference, paraphrasing, and other kinds of sentence-level tasks are things that this model excels at. *BERT* uses bidirectional pre-training, which this paper describes as being a very important feature of *BERT* that allows it to be successful at many NLP tasks.

BERT was a model employed by many participants of past CLEF SimpleText lab participants to varying levels of success. *BERTS*'s strengths in sentence-level tasks make it an option for task 2 of the SimpleText lab. The ability to fine-tune *BERT* and customize it to a particular task could make it effective at this task if properly tuned.

1.5 Paper 5: *AIIR and LIAAD Labs Systems for CLEF 2023 SimpleText*

The *AIIR* lab from the University of Southern Maine employed *YAKE!* And *KBIR* keyword extraction models for their approach to complexity spotting in task 2 of the CLEF 2023 SimpleText lab. The *AIIR* lab proposed two approaches towards task 2: one that just uses *YAKE!* as a keyword extraction tool, and the other approach combines *YAKE!* scores with *IDF* scores. For explaining the identified difficult terms, *AIIR* lab uses *TF-IDF* to find the top-1000 relevant documents for each phrase and then uses fine-tuned *ALBERT* on *DEFT* corpus, containing 16,800 labeled sentences indicating whether a sentence contains a definition. Their fine-tuning approach involved using 5 epochs, electing the highest accuracy model on a 90-10 validation set.

The evaluation section of this paper suggests that *Cross-Encoder* models could be a superior choice for initial retrieval steps. The results of their approaches showed that *YAKE!* was effective at extracting terms, but not as effective in detecting term limits. For defining, *KBIR* had the highest semantic accuracy rating with a .50.

1.6 Paper 6: *GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints*

This paper suggests a technique for training generalized multi-query transformer models: single key-value heads [6]. The proposed benefit of this technique is a drastic improvement in decoder inference. Another benefit of this technique is a way to counteract memory bandwidth overhead which is a consequence of autoregressive decoder inference. The downsides to this technique are quality degradation and training instability, which could result in unexpected results.

When considering approaches to task 2 of the CLEF 2024 SimpleText lab and how this technique could be applied to that, efficiency would be one of the main benefits. However, it's important to consider the quality degradation and training instability, which could result in worse output. The main objective of this lab is to have meaningful results over increased efficiency.

1.7 Paper 7: *Assembly Models for SimpleText Task 2: Results from the Wuhan University Research Group*

This paper highlights the approach taken by a team for task 2 of the 2022 CLEF SimpleText lab [7]. This team's approach involved using *keyBERT* and filtering those results with *PhraseSimilarity*. They also used preprocessing techniques like removing certain words and punctuation. For complexity evaluation, they trained ensemble models using models like *LightGBM*, *CatBoost*, and *XGBoost* and employed a soft voting strategy. For this part of the task, their best results were achieved with an integrated model. Their approach resulted in good results compared to other participants. The highest-performing techniques involved the use of ensemble models. This team suggests that one of the areas of improvement for this task would be the term extraction process and using domain-specific pre-trained word embeddings.

1.8 Paper 8: *UBO Team @ CLEF SimpleText 2023 Track for Task 2 and 3 - Using IA Models To simplify Scientific Texts*

This paper highlights an approach taken for tasks 2 and 3 of the CLEF 2023 SimpleText lab [8]. For Task 2, this team used *FirstPhrases*, *TF-IDF*, *YAKE*, *TextRank*, *SingleRank*, *TopicRank*, and *PositionRank*. For complexity spotting and for ranking they used the *Wikipedia* API package for defining the difficult terms. They also used *nltk* to help retrieve an initial sentence from the *Wikipedia* pages that were found for the specific term. For the complexity spotting aspect of Task 2, all of the models they used had reasonable performance, with *YAKE* being the lowest performing, and for defining the difficult terms, they ran into issues of no *Wikipedia* pages being available for a term, which resulted in lack of defining in some instances.

1.9 Summary of Strengths/Weaknesses of each Paper's Approach

The last page of this paper contains a table to visualize the strengths and weaknesses of the approaches analyzed in the papers covered.

References

1. Ermakova, L., Azarbonyad, H., Bertin, S.: Overview of the CLEF 2023 SimpleText Task 2: Difficult Concept Identification and Explanation. DOI: `https://ceur-ws.org/Vol-3497/paper-239.pdf`
2. Dadic, P., Popova, O.: CLEF 2023 SimpletText Tasks 2 and 3 Enhancing Language Comprehension Addressing Difficult Concepts and Simplifying Scientific Texts Using GPT, BLOOM, KeyBert, Simple T5 and More. DOI: `https://ceur-ws.org/Vol-3497/paper-246.pdf`
3. Davari, D. R., Prnjak, A., Schmitt, K..: CLEF2023 SimpleText Task 2, 3: Identification and Simplification of Difficult Terms. DOI: `https://ceur-ws.org/Vol-3497/paper-247.pdf`

4. Devlin J., Chang, MW., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. DOI: https://doi.org/10.48550/arXiv.1810.04805
5. Mansouri, B., Durgin, S., Franklin, S.J., Fletcher, S.†, and Campos, R.: AIIR and LIAAD Labs Systems for CLEF 2023 SimpleText. DOI: https://ceur-ws.org/Vol-3497/paper-253.pdf
6. Ainslie, J., Lee-Thorp, J., de Jong, M., Zemlyanskiy, Y., Lebrón, F., Sanghai, S.: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints. DOI: https://doi.org/10.48550/arXiv.2305.13245
7. Huang, J., Mao, J.: Assembly Models for SimpleText Task 2: Results from the Wuhan University Research Group. DOI: https://ceur-ws.org/Vol-3180/paper-239.pdf
8. Dubreuil, Q.: UBO Team @ CLEF SimpleText 2023 Track for Task 2 and 3 - Using IA Models To simplify Scientific Texts. DOI: https://ceur-ws.org/Vol-3497/paper-248.pdf

Paper	**Strengths of Approaches**	**Weaknesses of Approaches**
Paper 1	SINAI achieved the best results using *GPT-3* with effective prompt engineering.	Teams using the Wikipedia model struggled to provide definitions for all complex terms.
Paper 2	*SimpleT5* was the most effective at complexity spotting and providing definitions.	Methods with the highest results were on single-word terms.
Paper 3	*GPT-3* was good for both complexity spotting and defining.	The dataset had to be split into abbreviations and non-abbreviations using a regular expression for optimal results.
Paper 4	Bidirectionality allows for effectiveness at sentence-level NLP tasks.	Clever fine-tuning is required for optimal performance.
Paper 5	*YAKE!* proved to be effective at extracting difficult terms.	*YAKE!* was not as effective at determining term limits.
Paper 6	*MQA* is effective at speeding up decoder inference.	*MQA* can cause quality degradation and/or training instability.
Paper 7	*keyBERT* proved to be effective at keyword extraction.	Pre-trained embedding is trained in the public domain; better results could come from pre-trained word embeddings on specific domains.
Paper 8	The models used in the keyword extraction phase gave solid results.	The Wikipedia API package often couldn't deduce definitions because some terms didn't have a corresponding Wikipedia page.

Table 1. Strengths and Weaknesses of Approaches in Different Papers

SimpleText Task 3 Report

Nicholas Largey[1][0525664]

University of Southern Maine, Portland, ME, 04101, USA
nicholas.largey@maine.edu

Abstract. Phase 1 of our project for COS470/570 is designed so that we the students may get a full understanding of the chosen task from this year's CLEF2024 conference. My group and I have chosen the SimpleText category where I have chosen task 3, Simplifying Scientific Texts, as my contribution to the project. Scientific texts can be extremely difficult for the average person to understand, so this task aims to create or utilize a Deep Learning model to provide accurate and digestible simplifications for a person without an extensive background in the sciences.

Keywords: CLEF2024 · SimpleText · LLMs · B. Mansouri.

1 The Task At Hand

The goal of the SimpleText lab for the 2024 CLEFlabs conference is to make scientific writings more accessible to the general public.

Task 1 - Retrieving passages to include in a simplified summary - has the goal of filtering out sections, either as little as sentences or as large as entire passages, in order to provide a summary that still gets the overarching purpose, findings and methods in a text across to the reader, without necessarily going into extensive detail.

Task 2 - Identifying and explaining difficult concepts - is meant to create a model that can extract difficult topics or concepts from a given text and provide an explanation that would be suitable for the "average" English speaking person.

Task 3 - Simplify Scientific Text - is a mix between the first 2 tasks. Given a scientific text as input, a model needs to be designed or fine-tuned to output a version of that input that is accurate and digestible for a person without and extensive background in the sciences. This will require both approaches taken for task 1 and task 2 since in order to provide a simplified version, the input text will need to be reduced to it's "essence" and which will still contain difficult to understand concepts or wording which will need to be identified and explained.

2 Previous Approaches

Many of the approaches taken by past participants (2022 and 2023) utilized a number of similar approaches and models. T5 and iterations of the GPT model

Table 1. Approaches previously taken by participants and other researcher

Paper	Strengths	Weaknesses
APEAtSTS[1]	T5,GPT4 - Refined the pipeline and computational resources needed with prompt engineering and were able to attain a 8.43 FKGL	SARI score only reached 47.98 out
ASoSTUPTLLMs[2]	T5, AI21 J2, BLOOM (GPT2) - An easy to use and approachable method	Final output wasn't very competitive as no hyper-parameter or fine-tuning was performed.
CSSUTL[3]	T5, COVID-SciBERT - An interesting approach to reduce time and resources.	The input text most likely required some of the masked out information for the T5 model to produce the finalized output. Paper was also a bit difficult to follow as many things relating to NLP but not utilized were mentioned.
STSaGA[4]	SimpleT5, GPT3 - Fairly decent FKGL and SARI results considering the approach	Approach needs more refinement and training
STSUBART[5]	BART - Good approach to data preprocessing saving time and resources	SARI score is given, but FKGL is not so determining how the work compares to others in more difficult.
TSoSTfNER[6]	T5, GPT4, PEGASUS - Very detailed reporting on the approach taken and which models and metrics were used	Results had lots of hallucinations and metrics reported didn't match the required metrics by CLEF
AIAYN[7]	Seminal work on Transformer Models with many excellent visual representations and mathematical explanations	None really.
L2:OFaFTCM[8]	Llama-2 is the SOTA in LLMs and is fully open-source	Paper could have been split into sections that were papers on their own.
Orca[9]	Amazing approach to increase efficiency and transparency of smaller models that are trained on "Large Foundational Models" (LFM's) such as ChatGPT.	Can be difficult to implement while fine-tuning models to outperform existing LFM's

were by far the most popular, with T5 being used by almost all of the entries[1–4, 6]. Many of the teams attempted to utilize prompt engineering with GPT(2-4 were used) [1, 2, 4, 6]with varying degrees of success. It did prove useful as a preprocessing step [6] but I believe that the same results could be achieved with other method of preprocessing and data cleaning which would save time and compute resources as it is generally faster and less resource intensive to clean the data rather than have a GPT model do it for you.

The approach that the team from E.T.S.I. Informática in Spain took where they utilized a masking layer in order to process sentences with ”simple concepts” in the masking layer to avoid having it run through the ”complex version” (T5 model) I thought was pretty clever[3]. Although, I do see some issues that could lead to loss of context of what the article is about, but would need to see the implementation to confirm if the ”simple concepts” are omitted entirely from the T5 model, or if they are simply flagged as not needing any further simplification.

3 Planned Approach

Looking through these past submissions, I was pretty amazed at how many of the teams utilized the same processes and models. I know that one of the best ways to increase efficiency is to experiment with hyperparameters, but it seemed like many of the entries just ran the given training data through ChatGPT and one other model without experimenting with other models or pipelines [1, 2, 4, 6].

My planned approach is to utilize the Llama-2 and Orca LLM's since they are both State-Of-The-Art, and open-source, which will allow for a better understanding of the processes current LLM's are taking to generate the new text. I will have to experiment with the best way to create a pipeline and how to preprocess the data. For data preprocessing, I would like to explore the use the spaCy NLP package to see if it's ”scispaCy” pipeline would help with fine-tuneing the previously mentioned LLM's since it has been trained on scientific and bio-medical text already. Ideally, I will take a different approach from the previous year's entries since many of their FKGL and SARI results clustered into similar ranges [1, 2, 4, 6]. I will need to read into and experiment with Llama-2, Orca and SciSpaCy in order to find which methods and approaches will produce desirable results.

4 Conclusion

Although there is still much to discover about approaches that will work and approaches that won't be as successful, I believe that I now have a clear understanding of what the task at hand is. Past entries relied heavily on pipelines that utilized multiple models in order to achieve the results they entered. I think this approach will prove useful, especially if I am use a similar process with the above mentioned LLM's. With the benefit of Llama-2 and Orca being open-source, I, hopefully, won't need to blindly experiment as much as if I were to substitute a model like ChatGPT for my chosen ones.

References

1. Wu, SH., Huang, HY.: A Prompt Engineering Approach to Scientific Text Simplification. CLEF2023, vol. 3497, pp. 3057–3064 (2023)
2. Anjum, A., Lieberum, N.: Automatic Simplification of Scientific Texts using Pretrained Language Models. CLEF2023, vol. 3497, pp. 2899–2907 (2023)
3. Menta, A., Garcia-Serrano, A.: Controllable Sentence Simplification Using Transfer Learning. CLEF2022, vol. 3180, pp. 2818–2825 (2022)
4. Ohnesorge, F., Gutiérrez, M.A., Plichta, J.: CLEF 2023: Scientific Text Simplification and General Audience1. CLEF2023, vol. 3497, pp. 3027–3032 (2023)
5. Rubio, A., Martínez, P: HULAT-UC3M at SimpleText@CLEF-2022: Scientific text simplification using BART. CLEF2022, vol. 3180, pp. 2845–2851 (2022)
6. Engelmann, B. et al: Text Simplification of Scientific Texts for Non-Expert Readers. CLEF2023, vol. 3497, pp. 2899–2907 (2023)
7. Ashish Vaswani et al. Attention is All You Need. 2017. arXiv:1706.03762 [cs.CL]
8. Hugo Touvron et al. Llama 2: Open Foundation and Fine-Tuned Chat Models. 2023. arXiv:2307.09288 [cs.CL]
9. Subhabrata Mukherjee et al. Orca: Progressive Learning from Complex Explanation Traces of GPT-4. 2023. arXiv:2306.02707 [cs.CL]