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]