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