

# AIIR and LIAAD Labs Systems for CLEF 2023 SimpleText

Behrooz Mansouri<sup>1</sup>, Shea Durgin<sup>1,†</sup>, SJ Franklin<sup>1,†</sup>, Sean Fletcher<sup>1,†</sup> and Ricardo Campos<sup>2</sup>

<sup>1</sup>University of Southern Maine, Portland, Maine, USA

<sup>2</sup>University of Beira Interior, Covilhã, Portugal / Ci2 - Smart Cities Research Center, Polytechnic Institute of Tomar, Tomar, Portugal / INESC TEC, Porto, Portugal

## Abstract

This paper describes the participation of the Artificial Intelligence and Information Retrieval (AIIR) Lab from the University of Southern Maine and the Laboratory of Artificial Intelligence and Decision Support (LIAAD) lab from INESC TEC in the CLEF 2023 SimpleText lab. There are three tasks defined for SimpleText: (T1) What is in (or out)?, (T2) What is unclear?, and (T3) Rewrite this!. Five runs were submitted for Task 1 using traditional Information Retrieval, and Sentence-BERT models. For Task 2, three runs were submitted, using YAKE! and KBIR keyword extraction models. Finally, for Task 3, two models were deployed, one using OpenAI Davinci embeddings and the other combining two unsupervised simplification models.

## Keywords

Scientific text simplification, Keyword Extraction, Definition Extraction,

## 1. Introduction

The SimpleText lab at CLEF 2023 [1] has three tasks. The first task, **What is in (or out)?** involves searching a large database of academic abstracts and bibliographic metadata for passages that are relevant to a specific article. The topics for this task are a selection of press articles from two sources: the tech section of The Guardian newspaper (topics G01 to G20) and the Tech Xplore website (topics T01 to T20). Keyword queries are provided with each topic. Participants should find all passages from DBLP abstracts that are relevant to each query, and that could be used as citations in the paper associated with the topic. Each retrieved abstract is associated with a relevance score and a combined score indicating measures such as relevance, readability, and citation measures. For this task, the Artificial Intelligence and Information Retrieval (AIIR) lab from the University of Southern Maine (Maine, USA) and the Laboratory of Artificial Intelligence and Decision Support (LIAAD) lab from INESC TEC (Portugal) proposed five runs. One run, combine results from traditional Information Retrieval (IR) models, TF-IDF and PL2

---


*CLEF 2023: Conference and Labs of the Evaluation Forum, September 18–21, 2023, Thessaloniki, Greece*


\*Corresponding author.

†These authors contributed equally.

✉ behrooz.mansouri@maine.edu (B. Mansouri); shea.durgin@maine.edu (S. Durgin); sj.franklin@maine.edu (S. Franklin); sean.fletcher@maine.edu (S. Fletcher); ricardo.campos@ipt.pt (R. Campos)

ORCID 0000-0002-0400-9761 (B. Mansouri); 0000-0002-8767-8126 (R. Campos)

 © 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

[2]. The other runs take advantage of Sentence-BERT [3] cross-encoders and bi-encoder models, described in the next section.

The goal of the second task, **What is unclear?**, is to make the key concepts in a topic easy to understand by providing definitions, examples, and use cases. Participants should identify the most difficult terms in each passage and provide a ranked list of those terms. The difficulty score should be on a scale of 0 to 2, with 2 being the most difficult term and 0 being the easiest term (meaning can be guessed). Definition extraction for the terms is optional (second subtask). We participated in both subtasks, using three different approaches, followed by the same definition extraction pipeline for the second subtask, discussed in Section 3. For subtask 1, we used YAKE! [4], YAKE! combined with IDF score, and KBIR [5] to find difficult terms. Using the detected difficult terms, in subtask 2, we extracted their definitions from the SimpleText corpus, using TF-IDF model to find candidate definitions, followed by a classification model using a fine-tuned ALBERT [6] model.

And finally, the goal of the third task, **Rewrite this!**, is to create a simplified version of sentences that are extracted from scientific abstracts. For this task, our team considered two runs as explained in Section 4. One run chooses between two generated simplified versions of the text using existing techniques for text simplification, while the other uses OpenAI’s *text-davinci-003* model with a prompt to get the simplified text.

## 2. Task 1: What is in (or out)?

This section introduces our proposed models for task 1, along with the results and discussion.

### 2.1. Proposed Models

For the **What is in (or out)?** task, we proposed five systems: one based on traditional IR models, and the others based on Sentence-BERT [3]. Here, we provide the descriptions of these systems.

**TF-IDF combined with PL2.** For this run, we combine the results from two traditional IR models, TF-IDF and PL2 [2]. After retrieving the top-1000 results for each of the two systems, the retrieval results are combined using Modified Reciprocal Rank Fusion [7] as follows:

$$RRFscore(f \in F) = \sum_{m \in M} \frac{s_m(f)}{60 + r_m(f)} \quad (1)$$

where  $s_m$  is the similarity score given by the retrieval model and  $r_m$  is the rank of the retrieved passage among the top-1000 results. We used MinMax normalization to transform all the relevance scores to a common scale, which ranges from 0 to 1. For both systems, the input query is the given query concatenated with the topic text. For example, for query G01.1 with topic text as “Digital assistants like Siri and Alexa entrench gender biases says UN”, and query “Digital assistant”, the input query is “Digital assistant Digital assistants like Siri and Alexa entrench gender biases says UN”. We considered both the titles and abstracts of the DBLP papers in the collection for retrieval.

**Cross-Encoder.** For our first Sentence-BERT model, we use Cross-Encoder architecture<sup>1</sup>

<sup>1</sup><https://www.sbert.net/docs/pretrained-models/ce-msmarco.html>

with ‘ms-marco-electra-base’ as the pre-trained model without fine-tuning. This model was trained on the MS Marco Passage Ranking task [8]. In Cross-Encoder, query and candidate passage is passed to a BERT-based model and the relevance score is predicted. Each input query is represented as the concatenation of query and topic text (with white space), and the relevance is determined based on the abstracts of the papers. We combine the relevance scores from this model with scores from Elasticsearch model, using the Modified Reciprocal Rank Fusion as explained previously.

**Fine-tuned Cross-Encoder.** Our next two proposed approaches utilize fine-tuned cross-encoder models. For fine-tuning, we used the 29 assessed topics from the 2022 SimpleText lab [9]. As our pre-trained model, we employed ‘ms-marco-MiniLM-L-6-v2’. We considered 200 epochs and selected the best model based on the validation set, using a 90-10 split. Fine-tuning was performed with a maximum sequence length of 512 and a batch size of 4. Our approaches share a similar architecture but differ in the representation of the input query. In the first proposed model (Fine-Tuned Cross-Encoder (1)), we represented each input query as “query text + [QSP] + topic text”, while the papers are represented as “title + [TSP] + abstract”. Here, [QSP] is a special token used to separate the query text from the topic text, and [TSP] is another special token separating a paper’s title from its abstract. In the second approach (Fine-Tuned Cross-Encoder (2)), we just considered the query text as the input.

**Sentence-BERT Bi-Encoder.** Our final proposed approach uses the bi-encoder architecture, where sentences are independently passed through BERT models, and then their corresponding vectors are compared using cosine similarity. We adopted the same representation of query and topic text with [QSP] separator and title and abstract with [TSP] separator as the input for the model. For fine-tuning, we utilized a ‘distilroberta-base’ model with Triplet loss [10]. We incorporated all the training samples from the previous lab, with a 90-10 split for the training and validation sets. Positive samples were selected from abstracts with relevance scores of 1 and 2, and the negative samples were chosen from those with a score of 0. We generated all combinations to increase the number of training samples. We fine-tuned for 10 epochs with a batch size of 32 and a maximum sequence length of 512 and used the best model (lowest loss) on the validation set for ranking.

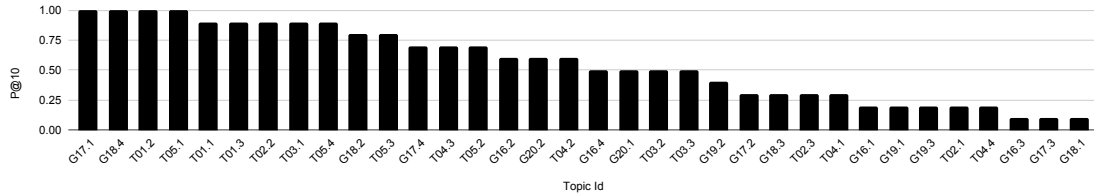
All the Sentence-BERT models use the top-100 results (for re-ranking or fusion) from Elasticsearch system provided by the organizers of the SimpleText lab. We chose 100 due to the efficiency of the Cross-Encoders models (being slower). Each team was allowed to retrieve up to 100 results per topic.

## 2.2. Evaluation

For evaluation, two aspects are considered: effectiveness and readability. Table 1 shows the effectiveness of our proposed approaches, provided by the organizers. As can be seen, our cross-encoder model (with special tokens and topic text), provides the highest MRR and BPref [11] values, while our proposed cross-encoder (with no fine-tuning) achieved the highest effectiveness considering other measures. To further analyze our proposed approaches, Figure 1 shows the P@10 values per test topic for the Fine-Tuned Cross-Encoder (1). As can be seen, for 11 out of 34 topics, this model has a P@10 > 0.8. For topics such as “T01.2” (light positioning with topic text as “Curve Light: A highly performing indoor positioning system”), this model

**Table 1**  
Evaluation of SimpleText Task 1 (Test Qrels).

Run	MRR	Precision		NDCG		Bpref	MAP
		10	20	10	20		
Fine-Tuned Cross-Encoder (1)	<b>0.734</b>	0.497	0.400	0.486	0.430	<b>0.344</b>	0.239
Cross-Encoder	0.731	<b>0.527</b>	<b>0.450</b>	<b>0.546</b>	<b>0.484</b>	0.334	<b>0.275</b>
Fine-Tuned Cross-Encoder (2)	0.708	0.471	0.393	0.462	0.409	0.326	0.225
Bi-Encoder (TripletLoss)	0.550	0.338	0.218	0.335	0.256	0.134	0.070
TF-IDf+PL2	0.563	0.418	0.281	0.401	0.322	0.216	0.136



**Figure 1:** Precision at 10 per test topic for our Fine-Tuned Cross-Encoder (1) model.

retrieves relevant abstracts such as “Visible Light Positioning (VLP) is emerging as a solution for indoor localization...” as the top document. Looking at the topics with lower effectiveness, for topic “G19.2” (kaspersky with topic text as International Space Station attacked by ‘virus epidemics’), this model has P@10 of 0.4. Documents retrieved as the top-10 results by this model are mostly related to individual terms in the topic text. For instance, documents such as “During the 2009 H1N1 influenza pandemic, there was rising concern about the potential contribution of international travel...” is related to the influenza pandemic and its effect on international travel, and not considered related to the topic, specifically International Space Station. Another example is topic G18.1 (Jay Gambetta with topic text as The next giant leap: why Boris Johnson wants to ‘go big’ on quantum computing), P@10 is 0.1. While the retrieved documents seem to be relevant, in our models we did not make use of meta-data for the abstract and the authors’ names are not considered. Abstracts with “Jay Gambetta” as the author (which are topically related to the query) are considered relevant for this topic.

Comparing the fine-tuned Cross-Encoder (1) with the Cross-Encoder (with no fine-tuning), there were cases that one model outperformed the other. For instance, for topic “G17.1” (quantum computing), the P@10 value for fine-tuned model is 1 while for the other is 0.3. With no fine-tuning, non-relevant documents discussing other related concepts to quantum computing such as transistors or CNFET cell were retrieved among the top-10 results. On the other hand, for topics such as “G16.2” (NLP application), the P@10 drops from 0.9 to 0.6 when using fine-tuned Cross-Encoder. The fine-tuned model retrieves more specific abstracts about NLP applications discussing tasks such as summarization while the model with no fine-tuning ranks the document on general NLP applications (such as evaluation approaches) higher.

The same pattern can be seen when comparing TF-IDF+PL2 and fine-tuned Cross-Encoder models. For topic “G18.1” (peer recommendations with topic text: New crowdsourced recruit-

**Table 2**

Text Analysis of SimpleText Task 1 output.

Run	Impact	#Refs	Length		FKGL	
			Mean	Median	Mean	Median
Fine-Tuned Cross-Encoder (1)	4.41	3.37	1003.75	988.00	15.01	14.80
Cross-Encoder	4.22	2.86	961.17	923.00	14.64	14.60
Fine-Tuned Cross-Encoder (2)	3.49	3.04	988.86	951.50	14.95	14.80
Bi-Encoder (TripletLoss)	4.76	3.29	969.09	973.50	14.69	14.60
TF-IDf+PL2	3.35	2.58	893.29	894.00	14.03	14.00

ment tool aims to get more women into tech), P@10 value with TF-IDF+PL2 model is 0.8 dropping to 0.1 when using fine-tuned Cross-Encoder. The first model retrieves abstracts that are mainly focused on women while with the Cross-Encoder non-related abstracts discussing minorities, or concepts such as peer-to-peer computing and recommenders are retrieved in the top-10 results. In contrast, for topics such as “T05.4” (empathy with topic text: New ‘emotional’ robots aim to read human feelings), P@10 value for TF-IDF+PL2 model is 0.2, increasing to 0.9 when fine-tuned Cross-Encoder is used. TF-IDF+PL2 model ranks non-relevant abstracts such as robot emotions where as the Cross-Encoder model focuses on empathy.

The second aspect of evaluation is readability. Table 1 shows the readability analysis provided on the top-10 results by the organizers. In this table, the impact indicates the impact factor based on ACM records, #Refs shows the average number of references per document, and FKGL(Flesch-Kincaid Grade Level) [12] is the readability score on a scale of 1 to 100. The higher the reading score, the easier a piece of text is to read. While all our proposed models had the highest impact and #Refs, our first fine-tuned cross-encoder model had the highest readability score among the participating teams.

Overall, our proposed approaches using Cross-Encoder architecture provided strong results for task 1. When fine-tuning the Cross-Encoder, we could achieve better results for more technical queries whereas with no fine-tuning, Cross-Encoder works better for general topics. Comparing traditional IR models with Cross-Encoders, Cross-Encoders could do better focusing on the query term when retrieving the documents than the topic text.

### 3. Task 2: What is Unclear?

Our team participated in both subtasks. For subtask 2.1, we proposed three approaches to detect difficult terms in the given sentences. For subtask 2.2, we used a fine-tuned BERT model to decide if a sentence with a difficult term contains a definition. Next, we elaborate on the specifics of our approaches.

#### 3.1. Proposed Models

**Subtask 2.1: Detecting difficult terms.** To detect difficult terms in the given sentences for this subtask, our first proposed model utilizes YAKE! [4] keyword extractor. With the default parameters and a window size of 3, we select the most relevant keyphrase (the one with the

lowest score). Our second approach combines YAKE! scores with IDF (Inverse Document Frequency) scores, similar to Chen et al.[13]. We first extracted the top-10 important phrases using YAKE!, and then we calculated the average IDF value for each phrase. The final phrase score is defined as  $YAKE!_{Score}/AVG_{IDF}$ . We selected the most important keyphrase (with the lowest score). For our last approach, we considered Keyphrase Boundary Infilling with Replacement (KBIR) [5] to extract difficult terms. KBIR is a pre-trained model that employs a multitask learning framework to optimize a combined loss function comprising Masked Language Modeling (MLM), Keyphrase Boundary Infilling (KBI), and Keyphrase Replacement Classification.

**Subtask 2.2: Providing an explanation for difficult terms.** To extract the explanations for each difficult phrase, we first retrieve the top-1000 relevant documents for each phrase. To do this, we use the TF-IDF model. This model is suitable for our task as we are aiming to find the exact matching of candidate phrases. Then, starting from the top retrieved document, we explored the sentences and check if they contain a definition.

To determine this, we fine-tune ALBERT [6] model on DEFT [14] corpus. This corpus contains 16,800 labeled sentences indicating whether the sentence contains a definition. For fine-tuning, we consider 5 epochs, choosing the model with the highest accuracy on the validation set (split of 90-10).

### 3.2. Evaluation

For subtask 2.1, two measures were considered for evaluation: the correctness of detected term limits and difficulty scores. The first metric reflects whether the retrieved difficult terms are well-limited or not. The second is a three-scale terms difficulty score, which reflects how difficult the term is in the context for an average user and how necessary it is to provide more context about the term. Table 3 shows the results of our proposed approaches to this subtask. Around 10% of our extracted terms were evaluated and among these, YAKE! did better compared to our other two models in extracting terms that should be defined, but KBIR provided better results in detecting the term limits (+Limits). For instance, in the sentence “This paper proposes a new Compressed Video Steganographic scheme.”, KBIR detects “Compressed Video Steganographic scheme” whereas YAKE! considers “Compressed Video Steganographic” as the terms that should be defined. On the other hand, YAKE! is able to extract correct terms in sentences such as “As a consequence, business processes that interact with large amounts of such data may easily cause GDPR violations, due to the typical complexity of such processes.”, extracting “GDPR violations” where KBIR extracts “business processes”.

For subtask 2.2, three measures are considered: 1) BLEU [15] score between the reference (ground truth definition) and the predicted definitions, 2) ROUGE-L F-measure [16] which measures the ROUGE F-measure based on the Longest Common Subsequence between the reference and the predicted definitions, and 3) Semantic match between the reference and predicted definitions measured using the “all-mpnet-base-v2” sentence transformer model. Table 4 shows the results of our runs for this subtask. Our models have similar effectiveness as a similar pipeline was used for definition extraction. We detected one major issue in our runs after submission; after retrieving a document with the TF-IDF model, the first sentence containing a definition is considered as the extracted term for the definition. The fix for this is to prioritize

**Table 3**

SimpleText Task 2.1: Results for the official runs.

Run	Total	Evaluated		Score	
			+Limits		+Limits
YAKE	4790	486	234	169	78
KBIR	4797	498	429	158	135
YAKEIDF	4790	465	241	154	75

**Table 4**

SimpleText Task 2.2: Results for the official runs.

Run	Evaluated	BLUE	ROUGE	Semantic
KBIR	556	1.62	0.15	0.50
YAKEIDF	179	1.13	0.14	0.41
YAKE	165	1.10	0.15	0.43

the sentences that contain definitions but also the target term for which we are looking for its definition. We also believe that based on the results from Task 1, the Cross-Encoder models might be a better suit for the initial retrieval steps.

## 4. Task 3: Rewrite This!

For task 3, rewriting scientific text, we propose two systems; one combining results from two systems based on how simplified the outputs are, and the other using *text-davinci-003* embeddings.

### 4.1. Proposed Models

In our first approach, we combined the simplification results from the models proposed by Laban et al. [17] (KiS), and Cripwell et al. [18]. The first model is an unsupervised simplification method, considering the problem as a reward maximization, rewarding simplicity, fluency, salience, and guardrails. The second model, however, considers four operations for simplification: whether to rephrase or copy and whether to split based on syntactic or discourse structure. For each sentence that we are aiming to simplify, we pass them to both models and then calculate the simplicity of the outcomes. Our simplicity score is defined based on the YAKE! and IDF scores introduced in the previous section. For each token in the sentence, we calculated the *YAKE!/IDF* score and averaged the scores to get the final simplicity score for a sentence. Our intuition is that a sentence with a higher YAKE! score and a lower IDF score is simpler. Therefore, a sentence with the highest score is then selected as the outcome of our proposed approach.

In the second approach, we used the OpenAI’s *text-davinci-003*, using a simple prompt as:

Simplify this sentence with simpler wording, explaining difficult terms:

**Table 5**  
SimpleText Task 3 Results.

Run	count	FKGL	SARI	BLEU	Compression ratio	Sentence splits	Levenshtein similarity	Additions proportion	Deletions proportion
Davinci	243	11.17	47.10	18.68	0.75	1.00	0.68	0.20	0.45
Combined	245	9.86	30.07	15.93	1.26	1.67	0.80	0.30	0.17

followed by the sentence to be simplified.

## 4.2. Evaluation

For Task 3 evaluation, several measures are considered, including the followings:

- Flesch-Kincaid Grade Level (FKGL) readability metric
- SARI [19] metric that compares the system’s output to multiple simplification references and the original sentence based on the words added, deleted, and kept by a system
- BLEU
- Compression ratio
- Sentence splits
- Levenshtein similarity that measures the number of edits (insertions, deletions, or substitutions) needed to transform one sentence into another
- Additions proportion
- Deletions proportion

Table 5 shows the results of our proposed approaches for this task. As indicated in this table, the Davinci model provides better effectiveness compared to our combined approach and better compresses the sentence with fewer additions and more deletions. Providing an example, for the source sentence “Abstract Novel technological advances in mobile devices and applications can be exploited in wildfire confrontation, enabling end-users to easily conduct several everyday tasks, such as access to data and information, sharing of intelligence and coordination of personnel and vehicles.”, the simplified version by this model is “Novel tech can be used to help with wildfire confrontation, allowing users to access data, share intelligence, and coordinate personnel and vehicles.”, closely similar to the provided simplified sentence by the organizers as “Novel mobile devices and applications can be used in wildfire confrontation by helping users to access data and information and coordinate personnel and vehicles.”.

## 5. Conclusion

This paper has described the AIIR and LIAAD labs submissions for SimpleText lab at CLEF 2023. Five runs were submitted for What is in (or out)? task. Our Cross-Encoder models provided better effectiveness compared to the model based on Bi-Encoder and the other model using traditional IR models. We participated in both subtasks of What is unclear?, proposing three approaches. However, our models seem to be less effective for this task. And finally, for the



Rewrite This! task, we considered two approaches, one based on the OpenAI Davinci model with a simple prompt for rewriting the text, and the other combining results from two simplification methods. The Davinci model for this task provided better effectiveness and compression. As this was our first attempt at the tasks in SimpleText lab, we aim to explore our current proposed approaches in detail, analyzing the results and removing potential errors as future work.

## Acknowledgments

Thanks to the organizers of SimpleText lab. We would like to thank Liana Ermakova, for giving a talk at the University of Southern Maine and discussing details of the SimpleText lab. Ricardo Campos was financed by National Funds through the FCT - Fundação para a Ciência e a Tecnologia, I.P. (Portuguese Foundation for Science and Technology) within the project StorySense, with reference 2022.09312.PTDC)

## References

- [1] L. Ermakova, E. SanJuan, J. Kamps, S. Huet, I. Ovchinnikova, D. Nurbakova, S. Araújo, R. Hannachi, E. Mathurin, P. Bellot, Overview of the CLEF 2022 SimpleText Lab: Automatic Simplification of Scientific Texts, in: *Experimental IR Meets Multilinguality, Multimodality, and Interaction: 13th International Conference of the CLEF Association, CLEF 2022, Bologna, Italy, September 5–8, 2022, Proceedings*, Springer, 2023.
- [2] G. Amati, C. J. Van Rijsbergen, Probabilistic Models of Information Retrieval based on Measuring the Divergence from Randomness, *ACM Transactions on Information Systems (TOIS)* (2002).
- [3] N. Reimers, I. Gurevych, Sentence-BERT: Sentence Embeddings using Siamese BERT-networks, *arXiv preprint arXiv:1908.10084* (2019).
- [4] R. Campos, V. Mangaravite, A. Pasquali, A. Jorge, C. Nunes, A. Jatowt, YAKE! Keyword Extraction from Single Documents using Multiple Local Features, *Information Sciences* (2020).
- [5] M. Kulkarni, D. Mahata, R. Arora, R. Bhowmik, Learning Rich Representation of Keyphrases from Text, *arXiv preprint arXiv:2112.08547* (2021).
- [6] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, R. Soricut, ALBERT: A Lite BERT for Self-supervised Learning of Language Representations, *arXiv preprint arXiv:1909.11942* (2019).
- [7] B. Mansouri, D. W. Oard, R. Zanibbi, DPRL Systems in the CLEF 2022 ARQMath Lab: Introducing MathAMR for Math-Aware Search, *Proceedings of the Working Notes of CLEF 2022* (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 (2016).
- [9] L. Ermakova, E. SanJuan, J. Kamps, S. Huet, I. Ovchinnikova, D. Nurbakova, S. Araújo, R. Hannachi, E. Mathurin, P. Bellot, Overview of the CLEF 2022 SimpleText Lab: Automatic Simplification of Scientific Texts, in: *Experimental IR Meets Multilinguality, Multimodality,*

- and Interaction: 13th International Conference of the CLEF Association, CLEF 2022, Bologna, Italy, September 5–8, 2022, Proceedings, Springer, 2022.
- [10] F. Schroff, D. Kalenichenko, J. Philbin, Facenet: A Unified Embedding for Face Recognition and Clustering, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015.
  - [11] C. Buckley, E. M. Voorhees, Retrieval Evaluation with Incomplete Information, in: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, 2004.
  - [12] R. Flesch, A New Readability Yardstick, *Journal of applied psychology* (1948).
  - [13] J. Chen, X. Zhang, X. Zhou, Y. Han, Q. Zhou, An Approach Based on a Cross-Attention Mechanism and Label-Enhancement Algorithm for Legal Judgment Prediction, *Mathematics* (2023) 2032.
  - [14] S. Spala, N. A. Miller, Y. Yang, F. Deroncourt, C. Dockhorn, DEFT: A corpus for definition extraction in free- and semi-structured text, in: Proceedings of the 13th Linguistic Annotation Workshop, 2019.
  - [15] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, BLEU: a Method for Automatic Evaluation of Machine Translation, in: Proceedings of the 40th annual meeting of the Association for Computational Linguistics, 2002.
  - [16] C.-Y. Lin, E. Hovy, Automatic Evaluation of Summaries Using N-gram Co-Occurrence Statistics, in: Proceedings of the 2003 human language technology conference of the North American chapter of the association for computational linguistics, 2003.
  - [17] P. Laban, T. Schnabel, P. Bennett, M. A. Hearst, Keep it Simple: Unsupervised Simplification of Multi-Paragraph Text, *arXiv preprint arXiv:2107.03444* (2021).
  - [18] L. Cripwell, J. Legrand, C. Gardent, Controllable Sentence Simplification via Operation Classification, in: Findings of the Association for Computational Linguistics: NAACL 2022, 2022.
  - [19] W. Xu, C. Napoles, E. Pavlick, Q. Chen, C. Callison-Burch, Optimizing Statistical Machine Translation for Text Simplification, *Transactions of the Association for Computational Linguistics* (2016).