ELiRF-VRAIN at BioLaySumm: Boosting Lay Summarization Systems Performance with Ranking Models

Vicent Ahuir[†], **Diego Torres**[†,*], **Encarna Segarra**[†,§], **Lluís-F. Hurtado**[†]
[†]VRAIN: Valencian Research Institute for Artificial Intelligence
Universitat Politècnica de València, Spain
[§]ValgrAI: Valencian Graduate School and Research Network of Artificial Intelligence
[†]{vahuir,esegarra,lhurtado}@dsic.upv.es
[*]dtorber@etsinf.upv.es

Abstract

This paper presents our contribution to the BioLaySumm 2024 shared task of the 23rd BioNLP Workshop. The task is to create a lay summary, given a biomedical research article and its technical summary. As the input to the system could be large, a Longformer Encoder-Decoder (LED) has been used. We continuously pre-trained a general domain LED model with biomedical data to adapt it to this specific domain. In the pre-training phase, several pre-training tasks were aggregated to inject linguistic knowledge and increase the abstractivity of the generated summaries. Since the distribution of samples between the two datasets, eLife and PLOS, is unbalanced, we fine-tuned two models: one for eLife and another for PLOS. To increase the quality of the lay summaries of the system, we developed a regression model that helps us rank the summaries generated by the summarization models. This regression model predicts the quality of the summary in three different aspects: *Relevance*, *Readability*, and *Factuality*. We present the results of our models and a study to measure the ranking capabilities of the regression model.

1 Introduction

Nowadays, there is more information than ever at the disposal of the general public. In the specific domain of biomedical research, there is information that would be interesting to non-expert audiences, including journalists or even members of the public, such as what occurred during the recent COVID-19 global pandemic (Wang et al., 2020). However, the technical language is a barrier for the non-specialist public that may prevent them from accessing that information (Goldsack et al., 2022; Guo et al., 2021).

Abstract summarization models should be useful in reducing the gap in understanding information. Since the models can generate a concise summary of a given text and capture its most relevant information (Raffel et al., 2020; Lewis et al., 2020; Brown et al., 2020; Beltagy et al., 2020). It is possible to obtain new models that generate summaries adapted to a much wider audience; what is known as *lay summary*. In a lay summary, the text should contain the main ideas of the article that would be interesting for a non-expert audience, enhancing readability by adding background information and reducing (or avoiding) technical terminology.

In this paper, we present the results and analysis of our system in the participation at the BioLaySumm (Goldsack et al., 2024) at the 23rd BioNLP Workshop (Demner-Fushman et al., 2024).

2 Task Drescription

In the 2024 edition, the BioLaySumm poses a single shared task, rather than two, as in the previous edition (Goldsack et al., 2023). The task is to create a lay summary, given a biomedical research article and its technical summary (abstract section of the article).

The organization provides a biomedical dataset (Goldsack et al., 2022) that contains biomedical research articles from two sources: *eLife Sciences*[1] and *Public Library of Science* (PLOS)[2]. Each sample contains the text of the article, the technical summary, and the reference lay summary. The dataset is divided into three partitions: `train`, `val`, and `test`.

	train	val	test
eLife	4346 (91.9)	241 (5.1)	142 (3.0)
PLOS	24 773 (94.3)	1376 (5.2)	142 (0.5)

Table 1: Dataset samples distribution per partition and source. Additionally to the number of samples, the table also shows the percentage over the source.

Table 1 shows the sample distribution of each

[1] https://elifesciences.org/
[2] https://plos.org

Proceedings of the 23rd Workshop on Biomedical Language Processing, pages 755–761
August 16, 2024.

source. It can be observed that the number of samples is way unbalanced towards the PLOS source, even though `test` presents the same number of samples for each source. This kind of distribution would be challenging when someone would like to develop a single summarization model without prompting or instructions. The alternative would be to create separate summarization models, one for eLife and the other for PLOS. The BioLaySumm organizers invited the participants to present solutions indistinctly using one or two models.

To measure the performance of the systems, the organizers of the competition selected a set of measures that would help to evaluate the performance in three different aspects: *Relevance*, *Readability*, and *Factuality*. For ***Relevance*** the following scores were chosen: ROUGE (1, 2, L) (Lin, 2004), BERTScore (Zhang* et al., 2020). To measure the ***Readability*** aspect: Flesch-Kincaid Grade Level (FKGL) (Kincaid et al., 1975), Dale-Chall Readability Score (DCRS) (Dale and Chall, 1948), Coleman-Liau Index (CLI) (Coleman and Liau, 1975), and LENS (Maddela et al., 2023). Finally, to measure ***Factuality***, two scores were selected: AlignScore (Zha et al., 2023), SummaC (Laban et al., 2022).

3 Pre-trainined Model

For this task, we have used a Longformer Encoder-Decoder (LED) (Beltagy et al., 2020) since we were approaching summarizing long texts, such as the case of scientific articles. This lets us increase the amount of information available on the encoder side. We used as a starting point the LED base model from AI2[3], publicly available at the repository of HuggingFace (Wolf et al., 2020), and continuously pre-trained it with in-domain data.

For the continual pre-training phase, we followed the training methodology used in the News Abstractive Summarization models (NAS) work (Ahuir et al., 2021). This methodology combines multiple pre-training tasks to incorporate linguistic knowledge in the pre-training phase and enhance the abstract nature of the produced summaries. Incorporating those tasks in continuous pre-training should help the model to transfer knowledge specific to the summarization task and increase the performance of the downstream model after fine-tuning, just as it did in the original NAS work.

The data used for continuous pre-training was chosen specifically to adapt the model to the biomedical research domain. We collected text from three different sources: abstracts (technical summaries) from PubMed (National Center for Biotechnology Information (NCBI), 2024) (17M samples), PubMed articles and abstracts from the `scientific_papers`[4] dataset (Cohan et al., 2018) (240K). Also, articles and technical summaries from the dataset train partition used in this competition (eLife + PLOS) (29K).

Due to infrastructure limitations, we limited the encoder input to work with no more than 4096 tokens. Taking into account this restriction, and with the objective of maximizing the amount of data, we split text by lines, using a window of no more than 4000 words. We generated subsamples that contained at least a new line and filled the windows with as many words as possible. The final amount of samples went up to 59M samples.

When working with LongFormers, you have to select which tokens will receive global attention in addition to local attention. In the original work (Beltagy et al., 2020), the authors recommend setting `[CLS]` token with global attention. However, we hypothesized that adding landmarks across the input with global attention could increase performance. For this reason, we added a special token with global attention (`<sent>`) after a certain number of sentences. The number of sentences was not constant but dictated by a minimum number of words of separation between `<sent>` tokens. Thus, the special token was placed at the end of every number of sentences with a total length of at least k words. Previous experimentation was carried out to determine the number of words. The best results were obtained with at least $k = 20$ words of separation.

The base model was pre-trained for three epochs in our Research Institute's cluster with 8 NVIDIA A40 graphic cards with 48GB of VRAM were used for the process; which took a month. The main hyperparameters are: 128 samples per device, 4 gradient accumulation steps, a learning rate of 5×10^{-5} with a constant scheduler, gradient checking, and an 8-bit quantified optimizer.

4 Lay Summarization Models

We developed two different approaches for the competition. In the first approach (**M1**), the model re-

[3] `https://huggingface.co/allenai/led-base-16384`

[4] `http://tiny.cc/54x2yz/scientific_papers`

ceives the technical summary and adapts the text and information to a lay summary style. In the second approach (**M2**), additional text is included beside the technical summary, that was, the introduction and the discussion sections of the article, similar to (Poornash et al., 2023).

Since the distribution of samples is not well-balanced, we fine-tuned two models per approach: one for eLife and another for PLOS. The four models were fine-tuned for ten epochs each with an NVIDIA RTX 3090 with 24GB; each approximation took nearly 24 hours. The relevant hyperparameters are: 4 samples per device and a learning rate of 5×10^{-5} with a linear scheduler.

In our tests over validation, M1 outperformed M2 in the overall performance. The detailed results can be seen in Table 3 (Appendix A).

5 Ranking Model

In order to increase the quality of the lay summaries of the system, we developed a regression model to rank the summaries generated by the summarization models. This regression model predicts the quality of the summary in three different aspects: *Relevance*, *Readability*, and *Factuality*.

5.1 Dataset Creation and Model Development

We use a Longformer encoder already trained in the biomedical domain[5] to develop the regression model. The classification layer was modified from the default in HuggingFace. We use a mean-max function of the hidden states of the last attention layer to calculate the embedding that feeds the feedforward classification layer. In mean-max, the mean of the hidden states is concatenated with the max values of those hidden states.

To fine-tune the model, we needed first to find a way to obtain sample variability in the scores in the three aspects. For this reason, we employed data augmentation based on LLMs. For this purpose, we adapted to our needs the novel framework *TextMachina* (Sarvazyan et al., 2024) and generated new samples using four LLMs: Vicuna 13b (Chiang et al., 2023), Alpaca 13b (Taori et al., 2023), OpenChat 7.5b (Wang et al., 2023), and Llama2 13b (Touvron et al., 2023). Using the technical summary and the lay summary from randomly selected samples of both sources, we applied different prompts to gain diversity in the quality of the samples in the three aspects. With this data augmentation, we obtained $16\,236$ new samples for training and 4212 for validation.

To create the training and validation partitions for regression, we use the generated samples and the technical and lay summaries from the corresponding partition of the competition dataset. To obtain the reference scores, we computed *Readability*, *Relevance*, and *Factuality*, using the formulas shown in Appendix B. At this point, we should remark on two details: (a) it can be noticed that all the scores are in a range $[0, 1]$, and always correlate positively with the quality of the summary, (b) due to time constraints, the *Factuality* score is only measured with *AlignScore* in the regression dataset.

The regression model was trained for five epochs in VRAIN's cluster for two days with 4 NVIDIA A30 with 24GB of VRAM. The main hyperparameters are: 6 samples per device, 2 gradient accumulation steps, a learning rate of 5×10^{-5} with a lineal scheduler, gradient checking, and an 8-bit quantified optimizer.

5.2 Usage and Performance

To rank the samples, we first score them. For scoring the quality of a lay summary, we used the regression model to measure the quality regarding the *Relevance*, *Readability*, and *Factuality*. With those values, we compute a single score based on the harmonic mean of those three values. The harmonic mean would give better scores to summaries that simultaneously hold high quality on the three aspects. We will refer to this score as `hm-score` for clarity.

In order to measure the ranking capabilities of the regression model, we measured the Normal Discounted Cumulative Gain (NDCG) over the real `hm-score` of the score of the best summary available and the real score of the chosen summary, based on the predicted `hm-score`.

In Fig. 1, we observe the distribution of the NDCG scores when the model ranks one approach (M1 or M2) and when the model ranks a mix of both (M1+M2). It can be noticed that with M1, it has better ranking capabilities than with M2. However, in both approaches, the scores are mainly in range of $[0.95, 1.0]$, which means that most of the time, one of the best summaries is chosen. When we mix the sources, the regression model reduces its ranking capabilities, which could indicate that it would be less precise when the quality of sum-

[5] https://huggingface.co/kiddothe2b/biomedical-longformer-large

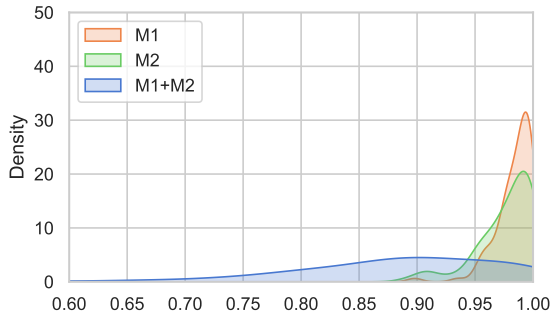

Figure 1: Distribution of the $NDCG_1$ scores obtained by the ranking model, when we consider both sources (eLife+PLOS). In `M1` and `M2`, the model ranks 10 summaries per sample; 20 summaries in `M1+M2`.

maries to choose from varies a lot.

The improvements in validation using the Ranking can be seen in Table 3 (Appendix A) can be seen for M1 M2, and M1+M2.

6 Results

For the competition, we sent a total of three submissions. ***S1*** that included lay summaries generated with M1 approach without any kind of ranking. ***S2*** that contained lay summaries generated with M1 and selected with the rank model (10 summaries per sample). Additionally, we sent a third submission (***S3***) that contained summaries from M1 and M2 and selected with the regression model (20 summaries per sample).

Table 2 shows official results for the test partition for the three submissions. It can be noticed that S2 provided the best results. Compared to S1, S2 increased the performance thanks to the ranking model. However, if the summarization model can not generate a wider variety of proposals, the ranking model will not help too much. Regarding S3, which includes the M1 and M2 summaries, we notice a lower quality of the final selection. Nevertheless, this submission increases the *Factuality* aspect, which could be attributed to the fact that M2 manages more information, reducing the factuality errors. Finally, regarding the relative performance (RP), our solution obtained more than 90% of performance in most of the scores, compared to the best overall submission. Further improvements need to be made, especially in the readability aspect.

	S1	S2	S3	RP(%)
Relevance				
↑ ROUGE-1	47.99	**48.15**	48.01	98.39
↑ ROUGE-2	13.61	**13.66**	13.60	87.06
↑ ROUGE-L	42.90	**43.09**	43.06	94.07
↑ BERTScore	85.94	**85.95**	85.91	99.05
Readability				
↓ FKGL	13.64	**13.61**	13.65	86.33
↓ DCRS	10.89	**10.86**	10.90	86.00
↓ CLI	14.71	**14.66**	14.70	91.13
↑ LENS	47.90	**48.02**	33.42	90.96
Factuality				
↑ AlignScore	78.37	78.21	**78.72**	97.71
↑ SummaC	60.91	60.66	**61.37**	82.67
`hm-score`	48.68	**48.69**	46.59	90.08

Table 2: Official results comparison for test partition for the three submissions (S1, S2, S3), and relative performance (RP) of S2 compared to the best overall system in the competition (UIUC_BioNLP). Bold values are the best values for each score. The up arrow (↑) indicates that the value of the score correlates positively with the quality of the lay summary, and the down arrow (↓) negatively. The `hm-score` is also included, which is not part of the official results.

7 Discusions

The results presented in Section 6 raise the benefits and constraints that must be taken into account when combining generation models with ranking models to choose which text will be presented to the end user.

Regarding the benefits, they are evident. With the ranking models, we can enhance the quality of the summaries presented to the user even though we use the same automatic summarization models. We use the ranking model to choose those summaries that obtained the best ranking scores since those texts will have better quality compared to other summaries generated by the same models. This selection should boost the overall performance of the system in most cases.

In relation to the constraints. The ranking model does not generate summaries or make texts better; it just rates summaries generated by the summarization models, and we select the best summaries based on those scores. Therefore, if summarization models have a bad performance and/or we can not provide enough variety to choose from, the benefits will be diminished. For this reason, we should combine the ranking models with summarization models that can complement each other depending on the text to summarize and offer variety in the generated summaries.

8 Conclusions

In this work, we have presented our contribution to the BioLaySumm 2024 shared task of the 23rd BioNLP Workshop. We used LED models to allow adding more text in the model input. Although we started from the same pre-trained model, different fine-tuned models were trained for the two sources of the competition: eLife and PLOS. Two different approaches were followed, one with just the technical summary as input, and another with additional text beside the technical summary. Our preliminary evaluation showed that the first approach performed better, but the second should be developed further since the larger input context improved the *Factuality* aspect. An additional contribution of our approach is the use of a regression-based ranking model that helped to boost the quality of the final summary by choosing the promising one from a set of summaries generated by the models. The model that obtained the best results in the competition was the one that combined the first approach and the ranking model.

Limitations

The data augmentation followed in this work to obtain the dataset for training the dataset is attached to the inner behavior of pre-trained LLMs. Those could present biases or limitations that we have not studied or detected. This could lead to limitations in the diversity and quality of the dataset, which could be inherited by the regression model.

Acknowledgments

This work is partially supported by MCIN/AEI/10.13039/501100011033, by the "European Union and "NextGenerationEU/MRR", and by "ERDF A way of making Europe" under grants PDC2021-120846-C44 and PID2021-126061OB-C41. It is also partially supported by the Generalitat Valenciana under project CIPROM/2021/023, and by the Spanish Ministerio de Universidades under the grant FPU21/05288 for university teacher training.

References

Vicent Ahuir, Lluís-F. Hurtado, José Ángel González, and Encarna Segarra. 2021. Nasca and nases: Two monolingual pre-trained models for abstractive summarization in catalan and spanish. *Applied Sciences*, 11(21).

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv:2004.05150*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*.

Meri Coleman and T L Liau. 1975. A computer readability formula designed for machine scoring. *Journal of applied psychology*, 60(2):283.

Edgar Dale and Jeanne S Chall. 1948. A formula for predicting readability. *Educational research bulletin*, 27(1):11–28.

Dina Demner-Fushman, Sophia Ananiadou, Mako Miwa, Kirk Roberts, and Jun-ichi Tsujii, editors. 2024. *The 23nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks*. Association for Computational Linguistics, Bangkok, Thailand.

Tomas Goldsack, Zheheng Luo, Qianqian Xie, Carolina Scarton, Matthew Shardlow, Sophia Ananiadou, and Chenghua Lin. 2023. Overview of the biolaysumm 2023 shared task on lay summarization of biomedical research articles. In *The 22nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks*, pages 468–477, Toronto, Canada. Association for Computational Linguistics.

Tomas Goldsack, Carolina Scarton, Matthew Shardlow, and Chenghua Lin. 2024. Overview of the biolaysumm 2024 shared task on the lay summarization of biomedical research articles. In *The 23rd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks*, Bangkok, Thailand. Association for Computational Linguistics.

Tomas Goldsack, Zhihao Zhang, Chenghua Lin, and Carolina Scarton. 2022. Making science simple: Corpora for the lay summarisation of scientific literature. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10589–10604, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yue Guo, Wei Qiu, Yizhong Wang, and Trevor Cohen. 2021. Automated lay language summarization of biomedical scientific reviews. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(1):160–168.

J. Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. *Research Branch report*, 8:75.

Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. 2022. Summac: Re-visiting nli-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Mounica Maddela, Yao Dou, David Heineman, and Wei Xu. 2023. LENS: A learnable evaluation metric for text simplification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16383–16408, Toronto, Canada. Association for Computational Linguistics.

National Center for Biotechnology Information (NCBI). 2024. Pubmed: A resource by the national center for biotechnology information. `https://pubmed.ncbi.nlm.nih.gov/`.

A.s. Poornash, Atharva Deshmukh, Archit Sharma, and Sriparna Saha. 2023. APTSumm at BioLaySumm task 1: Biomedical breakdown, improving readability by relevancy based selection. In *The 22nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks*, pages 579–585, Toronto, Canada. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Areg Mikael Sarvazyan, José Ángel González, and Marc Franco-Salvador. 2024. Textmachina: Seamless generation of machine-generated text datasets. *Preprint*, arXiv:2401.03946.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Guan Wang, Sijie Cheng, Xianyuan Zhan, Xiangang Li, Sen Song, and Yang Liu. 2023. Openchat: Advancing open-source language models with mixed-quality data. *arXiv preprint arXiv:2309.11235*.

Lucy Lu Wang, Kyle Lo, Yoganand Chandrasekhar, Russell Reas, Jiangjiang Yang, Doug Burdick, Darrin Eide, Kathryn Funk, Yannis Katsis, Rodney Michael Kinney, Yunyao Li, Ziyang Liu, William Merrill, Paul Mooney, Dewey A. Murdick, Devvret Rishi, Jerry Sheehan, Zhihong Shen, Brandon Stilson, Alex D. Wade, Kuansan Wang, Nancy Xin Ru Wang, Christopher Wilhelm, Boya Xie, Douglas M. Raymond, Daniel S. Weld, Oren Etzioni, and Sebastian Kohlmeier. 2020. CORD-19: The COVID-19 open research dataset. In *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020*, Online. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin

Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. 2023. AlignScore: Evaluating factual consistency with a unified alignment function. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11328–11348, Toronto, Canada. Association for Computational Linguistics.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

A Results in Evaluation (`val` partition)

	M1	M2	M1R	M2R	AR
Relevance score	48.23	45.02	**48.28**	45.26	47.26
↑ ROUGE-1	48.88	44.06	**48.97**	44.44	47.60
↑ ROUGE-2	14.52	11.20	**14.54**	11.38	13.30
↑ ROUGE-L	43.60	40.39	**43.68**	40.77	42.70
↑ BERTScore	85.90	84.41	**85.91**	84.44	85.42
Readability score	38.51	28.16	**38.64**	28.49	37.74
↓ FKGL	13.67	15.03	13.66	14.89	**13.58**
↓ DCRS	10.85	11.69	**10.82**	11.61	10.85
↓ CLI	14.47	15.53	14.43	15.43	**14.22**
↑ LENS	49.00	23.87	**49.11**	23.61	44.20
Factuality score	68.49	**81.35**	68.16	80.96	70.37
↑ AlignScore	77.00	86.65	76.64	**85.67**	77.36
↑ SummaC	59.97	76.04	59.68	**76.25**	63.38
`hm-score`	48.94	42.85	**48.97**	43.14	48.49

Table 3: Results comparison for validation partition for the two approaches without using ranking (M1 and M2), with ranking (M1R, M2R), and M1+M2 ranked (AR). Bold values are the best values achieved for each score. The up arrow (↑) indicates that the value of the score correlates positively with the quality of the lay summary, and the down arrow (↓) negatively.

Table 3 shows the results of the two model types when one summary is requested (columns M1 and M2). Or, when 10 summaries are requested per sample, rank with our ranking model and select the top-ranked summary for each sample (columns M1+R and M2+R).

B *Relevance*, *Readability* and *Factuality* scores.

We defined *Relevance* as the average of the following scores: ROUGE-1, ROUGE-2, ROUGE-L and BERTScore. *Factuality* is the average values of AlignScore and SummaC scores.

For defining *Readability*, we start first defining the function *Clamp and Complement (CC)*:

$$CC^z_f(x) = \frac{z - f(x)|_{[0,z]}}{z} \quad (1)$$

Eq. (1) shows that, given a function f, an integer number $z > 0$, and sample x. The sample x is evaluated with f. Then, the score is clamped in a range from $[0, z]$, complemented, and normalized.

Therefore, we define *Readability* as follows:

$$\begin{aligned} Readability(x) = (& \\ & CC^{20}_{FKGL}(x) + \\ & CC^{20}_{DCRS}(x) + \\ & CC^{20}_{CLI}(x) + \\ & \frac{LENS(x)}{100} \\ ) \cdot & \frac{1}{4} \end{aligned} \quad (2)$$

Eq. (2), shows that *Readability* is defined as the average of the following four scores: FKGL, DCRS, CLI, and LENS. For the three first scores (FKGL, DCRS, and CLI), the values below 20 are clamped since we consider that 20 is already a really high readability level for lay summarization purposes. Additionally, values are complemented and normalized when needed.