Document-Level Abstractive Summarization

Gonçalo Raposo
INESC-ID
Instituto Superior Técnico
Universidade de Lisboa
goncalo.cascalho.raposo@tecnico.ulisboa.pt

Afonso Raposo
Instituto de Telecomunicações
Instituto Superior Técnico
Universidade de Lisboa
afonso.raposo@tecnico.ulisboa.pt

Ana Sofia Carmo
Instituto de Telecomunicações
Instituto Superior Técnico
Universidade de Lisboa
ana.sofia.carmo@tecnico.ulisboa.pt

Abstract

The task of automatic text summarization produces a concise and fluent text summary while preserving key information and overall meaning. Recent approaches to document-level summarization have seen significant improvements in recent years by using models based on the Transformer architecture. However, the quadratic memory and time complexities with respect to the sequence length make them very expensive to use, especially with long sequences, as required by document-level summarization. Our work addresses the problem of document-level summarization by studying how efficient Transformer techniques can be used to improve the automatic summarization of very long texts. In particular, we will use the arXiv dataset, consisting of several scientific papers and the corresponding abstracts, as baselines for this work. Then, we propose a novel retrieval-enhanced approach based on the architecture which reduces the cost of generating a summary of the entire document by processing smaller chunks. The results were below the baselines but suggest a more efficient memory a consumption and truthfulness.*

1 Introduction

With the growth of publicly available text data, the summarization of such contents is essential for their usefulness. A text summary must convey important information from the original text and present a smaller, more manageable, size [1]. The task of automatic text summarization produces a concise and fluent text summary while preserving key information and overall meaning [2].

Approaches to automatic text summarization can be divided into extractive and abstractive summarization. While the extractive approach produces a summary that is comprised entirely of excerpts from the original text, the abstractive approach generates an output that may contain content that is entirely original. Both approaches have seen significant improvements in recent years by using models based on the Transformer architecture [3]. In particular, the fluency of these language models has allowed for state-of-the-art results for abstractive summarization [4–6].

However, Transformers' quadratic memory and time complexities with respect to the sequence length make them very expensive to use, especially with long sequences, as required by document-level summarization. Recent approaches explore different attention mechanisms that are able to reduce the quadratic cost, allowing to process longer sequences [7–9]. Additionally, retrieval-enhanced language

*This abstract was generated automatically using LED.

Deep Structured Learning (2021/22), Instituto Superior Técnico, Portugal.

models exhibit useful memorization qualities while being more efficient than plain models [10]. Although less explored, retrieval has been used to enhance an abstractive summarization model, improving its performance [11].

Our work will address the problem of document-level summarization by studying how the aforementioned techniques can be used to improve the automatic summarization of very long texts. In particular, we will use the arXiv dataset, consisting of several scientific papers and the corresponding abstracts. The results obtained with Efficient Transformers will be reproduced and used as baselines. Then, we propose a novel retrieval-enhanced approach based on the RETRO architecture which reduces the cost of generating a summary of the entire document by processing smaller chunks. All of our implementations are open source and available in GitHub[1][2].

2 Related Work

The Transformer architecture introduced in 2017 [3] established, within sequence modeling, an alternative to Recurrent Neural Networks (RNNs). In fact, by processing sentences as a whole using attention mechanisms and positional embeddings, Transformers avoid processing the input recurrently, facilitating parallelization as well as handling long-context dependencies.

2.1 Long document summarization

Since most common Transformer models are pretrained for inputs of $256 - 1024$ tokens, and fine-tuning them for longer sizes is computationally expensive, they seem unsuitable for the task of summarizing entire documents. However, three different approaches to the standard Transformer that allow for long-document summarization have been proposed: 1) divide-and-conquer, 2) hierarchical attention mechanisms, and 3) sparse attention mechanisms.

The first approach builds upon the idea that long-document summarization can be decomposed into shorter summarization problems, in which the task is tackled in a section-wise manner. Considering that manually adapting training data to accommodate this methodology would not be feasible, Gidiotis and Tsoumakas [12] designed a method to enable training in such a manner: rather than manually summarizing each section of the document, the process is performed automatically using Divide-ANd-ConquER (DANCER). This methodology is used to create artificial pairs of sections and abstract segments for training, which are applied to a well-known encoder-decoder Transformer architecture, PEGASUS [6]. Although this approach makes the model generalizable to theoretically infinite documents, it fails to incorporate context from the other sections of the document. Furthermore, it does not manage duplicate information when the set of summaries is concatenated.

Hierarchical attention, first introduced in the context of sequence classification [13], explores the ambivalent relevance of each token according to the context they are in. The hierarchical attention mechanism incorporates two levels of attention mechanisms [14, 15], one at the sequence level and another at the word level. As such, the first level can identify which sequences of tokens (within a sentence) are potentially relevant, significantly limiting the number of individual tokens that need to be processed by the second level (full attention pattern). This mechanism was transposed to long document summarization by Rohde et al. [16] with state-of-the-art results, although for input sequences limited to approximately 3k (due to memory constraints).

Finally, sparse attention mechanisms directly tackle the issue of time and memory quadratic complexity with sequence length. Instead of using a full attention pattern, primacy is given to the local context (local attention window), while also incorporating some global attention elements that provide access to the global context. This sparsity approach provides a considerable context of the full sequence while significantly decreasing complexity. Beltagy et al. [17] and Zaheer et al. [18] propose drop-in replacements for the standard attention mechanisms, reporting results for the standard Transformer [3] and PEGASUS [6] architectures, respectively. Similarly, Guo et al. [19] extends the original T5 architecture [5] with an attention sparsity pattern, applied to the encoder layer only.

While all approaches achieved state-of-the-art performances on the arXiv dataset, not all models are designed to handle the same input length, as illustrated in Table 1. Considering shorter input

[1] `https://github.com/afonsocraposo/generation-baselines`

[2] `https://github.com/gonced8/document-summarization`

lengths as a limitation for the specific task of document-length summarization, the LongT5 approach proposed in [19] reports the most satisfactory results in both domains (performance and input length).

2.2 Summarization datasets

Guo et al. [19] showcased six datasets for text summarization. These datasets can be divided into two groups: the first, constituted by the CNN/Daily Mail [20], MediaSum [21], and Multi-News [22] datasets, relates to news articles and media sources; the second, constituted by the PubMed [23], arXiv [23], and BigPatent [24] datasets, relates to scientific and technical documents. Naturally, the first includes shorter documents, with an average input length of 1,797 tokens, while the second group includes longer documents, averaging 6,931 tokens (obtained with the SentencePiece tokenizer [25]).

Guo et al. [19] gather the summarization results of many state-of-the-art models, which are presented in Table 1, along with a few details of the models. These are evaluated using the ROUGE automatic metric [26] and considered baselines for this work. ROUGE works by measuring the overlap of n-grams between the generated and reference summaries.

Table 1: Summarization results of several Transformer models evaluated in the arXiv dataset [23], evaluated using the ROUGE automatic metric [26], as reported by Guo et al. [19].

Model	Approach	Input length	R-1	R-2	R-L
DANCER PEGASUS [12]	Divide-and-conquer	N.A.	45.01	17.60	40.56
HAT-BART [16]	Hierarchical attention	3k	46.68	19.07	42.17
LED [17]	Sparse attention	16k	46.63	19.62	41.83
BigBird-PEGASUS [18]	Sparse attention	4k	46.63	19.02	41.77
LongT5 [19]	Sparse attention	16k	48.35	21.92	44.27

3 Efficient Transformer

A review of state-of-the-art approaches (Section 2) indicates that Transformer-based models with sparse attention mechanisms are particularly well-suited for the task of summarizing long sequences. Given the notable results reported by Guo et al. [19], our work focuses on the LongT5 model [19].

The LongT5 model aims to tackle the issue of the quadratic complexity of traditional attention mechanisms. The proposed approach uses a *Transient Global Attention* mechanism as an alternative to the attention pattern of the original T5 encoder architecture [5]. As illustrated in Figure 1, this pattern gives primacy to neighboring tokens (through the use of a sliding window) while, at the same time, incorporating global context through a set of dynamically constructed global tokens (Figure 1). This effectively reduces the time and memory complexity of input encoding from $\mathcal{O}(n \times n)$ to $\mathcal{O}(n \times (r + n/k))$ (where n is the input length, r is the width of the local window, and n/k is number of global tokens). Since the output size in a document summarization is considerably more manageable than its input size, this attention mechanism is not as important for the decoder component, therefore, LongT5 simply leverages the original decoder from T5.

When applied to the task of document-level summarization using the arXiv dataset [23], the input of the Transformer will be the entire document text (excluding everything before the Introduction and after the Conclusion) and the ground truth summary will be the article's abstract – Figure 2b. As a first approach, a pretrained implementation of the LongT5 (LongT5-TGlobal-Large - 16k input)[3] was fine-tuned with the aforementioned arXiv dataset.

4 Retrieval-Enhanced Approach for Summarization

Instead of relying only on learned weights for memorization, combining neural networks with explicit memories (e.g., through retrieval from a repository) is a possible way to decrease the number of model parameters while obtaining comparable performance [10]. Historically, information retrieval was performed using bag-of-words representations and functions like TF-IDF and BM25 [27]. More

[3] https://github.com/google-research/longt5

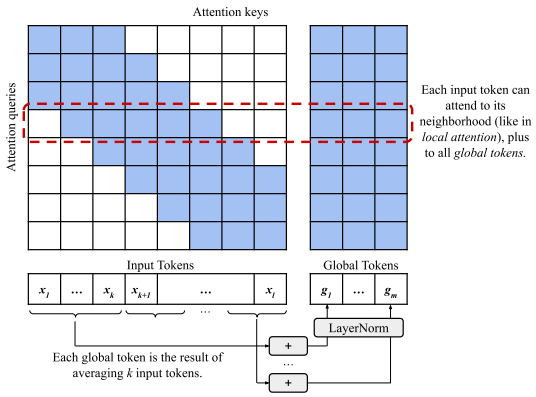


Figure 1: Illustration of the *Transient Global Attention* mechanism proposed to extend the standard T5 encoder architecture. Obtained from Guo et al. [19].

recently, neural models trained to encode text into dense representations are able to capture implicit semantics [28–30], with retrieval methods exploring these representations in dual-encoder or cross-encoder settings [31].

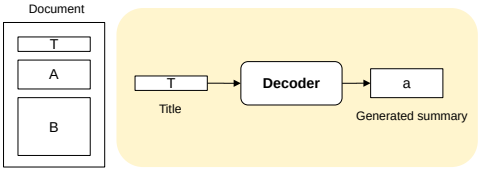
One example of coupling an external memory with a neural model for text generation is the kNN-LM [32], which builds a key-value database of context-token pairs and calculates the next-token probability by interpolating a Transformer with a distribution calculated from the retrieved k nearest neighbors. RAG [4] combines inputs and text retrieved using a dual-encoder, feeding both to a decoder for generation. FiD [33] assumes a similar approach, scaling better to larger numbers of retrieved passages. Combining kNN-LM and FiD, RETRO [10] retrieves chunks of text (neighbors) whose dense representations are then processed independently in an encoder, and attended in a chunked cross-attention (CCA) operation in a decoder. By processing the input in chunks, RETRO avoids computing the quadratic attention over the entire document, by computing it only over the chunks that the retrieval component considered relevant.

Our proposed approach, which we name RETROSUM, is to use a RETRO-based model to generate a document summary, retrieving from a set of chunks obtained only from that document. Without retrieval, the decoder would generate a summary-like text from a given prompt (e.g., paper title) – Figure 2a. However, the generated text would be very imprecise/incorrect since the decoder would not have any information besides the prompt. With retrieval, chunks of the generated text are used to sequentially retrieve neighbors from the document text, which are encoded and attended to in the CCA operation in the decoder – Figure 2c. With this approach, the decoder will be able to incorporate the information from the document during the generation of the summary. Figure 2 illustrates three different approaches to the task of document summarization of a scientific paper.

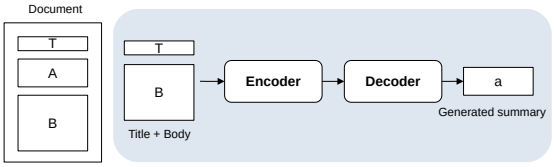
4.1 RETRO-fitting a baseline model

Although a RETRO-based model could be trained from scratch for abstractive summarization, extending baseline models into RETRO models offers a more efficient alternative – RETRO-fitting [10]. Starting from a pretrained Transformer, it is augmented with nearest-neighbor retrieval, a neighbor encoder, and chunked cross-attention layers. During training, all parameters are frozen except the neighbor encoder and the chunked cross-attention, ensuring that the original model performance is maintained without retrieval.

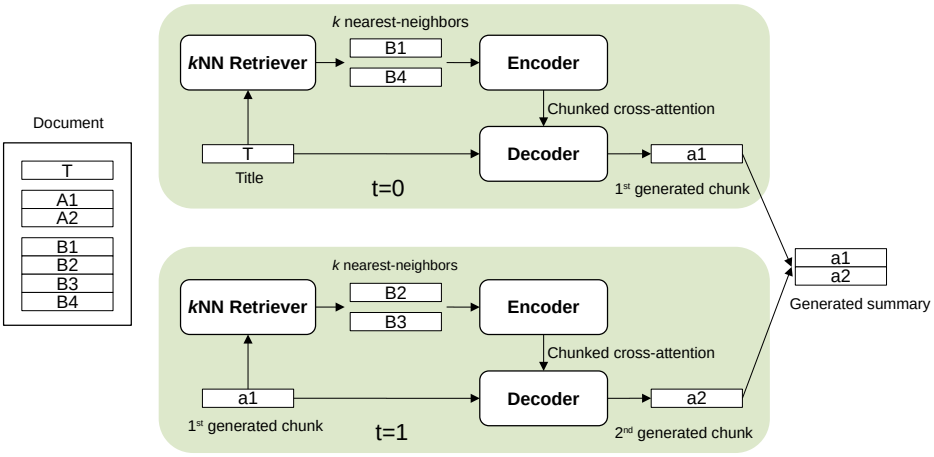
Since RETRO works with chunks of a fixed size, the RETRO-fitting implementation is simpler if the pretrained Transformer utilizes the same tokenizer as the encoder used for the nearest-neighbor search, such that the number of tokens is the same throughout. In the original paper [10], RETRO tokenizes the dataset using SentencePiece [25], but performs nearest-neighbor search using BERT [34], which was originally implemented using WordPiece tokenization [35]. Thus, we assume that the authors pretrained a BERT-like model using SentencePiece and, consequently, our design will have some differences.



(a) Decoder-only model with title as prompt.



(b) Encoder-decoder model with title and full paper text (body) as input.



(c) Proposed RETRO-based architecture for document summarization. The document is divided into multiple chunks and the generated abstract is obtained also in chunks, at each timestep t.

Figure 2: Different Transformer-based approaches for document-level summarization of scientific articles. The ground truth data is represented with uppercase letters while the generated data is represented with lowercase letters.

5 Experiments

5.1 Experimental Setup

We focus on the arXiv dataset, which consists of scientific papers from the corresponding repository. Being scientific papers, these documents follow a common structure: initial description of the problem, methodology, experiments/results, and conclusions. A publicly available[4] compilation of 215K docs was curated by Cohan et al. [23] and was used in this work. In this compilation, each paper entry is represented in a JSON object with the following elements: article id, abstract text, article text, section names, and sections. Some dataset statistics are shown in Table 2.

Table 2: Statistics for the arXiv dataset [23]. Tokens are obtained using SentencePiece [25].

Example count			Input (document) length		Output (summary) length	
Train	Validation	Test	Avg. # words	Avg. # tokens	Avg. # words	Avg. # tokens
203,037	6,436	6,440	5,467	10,079	273	438

To automatically evaluate the summarization performance, we use the ROUGE-1, ROUGE-2, and ROUGE-L metrics [26]. Since automatic metrics often do not correlate well with human judgment, we also use BERTScore, which exploits pretrained models to measure semantic equivalence [36].

Disclaimer

We ran our experiments in a 5% subset of the original arXiv dataset. Given the time required to adapt LongT5, implement the novel architecture of RETROSUM, and process each document (splitting into

[4] https://github.com/armancohan/long-summarization.git

chunks, tokenization, and indexing), our available computational resources and time frame of this work did not allow us to run experiments on the entire dataset. Nonetheless, we have experiments running on the entire dataset.

5.2 LongT5

The LongT5 model is openly available by Google Research[5]. A converted *HuggingFace* checkpoint[6] from the original Google checkpoint was used to fine-tune and test this model on the arXiv dataset.

We used the LongT5 TGlobal Base model, since it uses the new and improved attention mechanism, Transient Global, and the number of training parameters, 247M, was supported by the GPUs available to us. We used a Quadro RTX 6000 with 24 GiB of memory, allowing us to train with an input size of 4096, output size of 512, and batch size 1 (gradients were accumulated over 32 steps). The model was trained for 10 epochs with a learning rate of 10^{-4}, using the Adafactor optimizer, and gradient accumulation of 32 samples. As in similar works, we treated documents longer than the supported length by truncating them to the maximum input size of 4096 tokens.

5.3 RETROSUM

Our implementation follows the RETRO-fitting approach, using the encoder and decoder models of a pretrained T5-Base model [5]. As in the original paper, it starts by tokenizing the dataset using SentencePiece and making up chunks of 64 tokens, for every abstract and articles' text. Using a frozen Sentence-T5 encoder [37], dense vectors/embeddings ($d = 768$) are computed for each chunk of text. Then, AutoFaiss[7] is used to index the text chunks embeddings of each document and to retrieve the 2 nearest-neighbors for each abstract chunk embedding. Since this approach generates a different index for each document, the retrieval step is much quicker than if retrieved from a collection of text chunks of all documents.

As for the Encoder and Decoder models (Figure 2c), they are implemented using the (unofficial) implementation in the RETRO - Pytorch library[8]. The weights of the T5 parameters were copied to a RETRO model, which additionally has chunked cross-attention layers introduced in every 3rd layer, starting from 6, of the 12-layer T5 decoder (as suggested by Borgeaud et al. [10]). At last, the retrieved neighbors are encoded using the T5 encoder and attended to in the T5 decoder augmented with chunked cross-attention layers.

The proposed model was trained and evaluated on the arXiv dataset. We evaluate RETROSUM with and without retrieval, prompting our model with the articles' titles, as illustrated in Figures 2a and 2c. Plots of the train and validation losses for each approach are shown in Figure 3, which suggest that the model starts overfitting in the training data after a few epochs (expected given the small size of the subset we considered).

5.4 Results

After training our implementations of the LongT5 and RETROSUM models, we evaluated them in the test sets of the arXiv dataset, reporting the automatic metrics ROUGE and BERTScore in Table 3.

With our implementation of the LongT5 model, we intended to replicate the results reported by Guo et al. [19]. We evaluated the base LongT5 model fine-tuned with input lengths of 4k. Our model performance in terms of ROUGE was below the one reported in the original LongT5 paper. This was most probably caused by the inferior batch size and subset we used during training. However, its performance was greater than that of the baseline pretrained summarization model PEGASUS [6]. Additionally, we also report its BERTScore for the arXiv dataset.

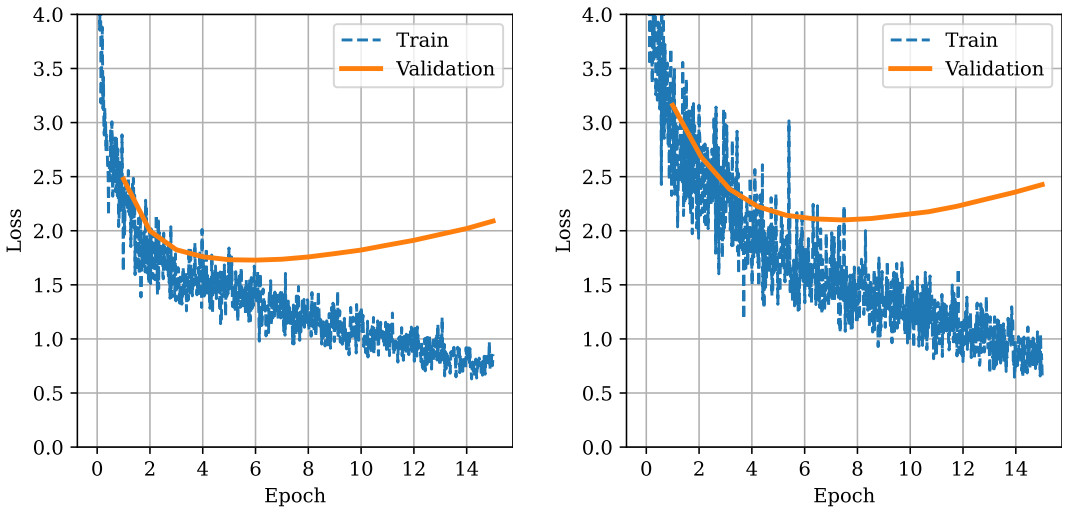
Regarding our proposed model RETROSUM, the results we obtained were lower than anticipated. Although there was a slight improvement when introducing the retrieval component, both results were below the ones of LongT5 and PEGASUS. In addition to the smaller subset used for training, the performance may be affected by empoying the base T5 model in a different configuration than it was

[5] https://github.com/google-research/longt5

[6] https://huggingface.co/Stancld/LongT5-TGlobal-Base

[7] https://github.com/criteo/autofaiss

[8] https://github.com/lucidrains/RETRO-pytorch



(a) Without retrieval.

(b) With retrieval.

Figure 3: Train and validation losses of RETROSUM trained on the arXiv dataset.

Table 3: Summarization results comparing the reference LongT5 fine-tuned with the arXiv dataset by Google and the LongT5 HuggingFace implementation fine-tuned by us. All LongT5 scores are with models using TGlobal attention, an input length of 4096 and output length of 512.

Model	Input length	R-1	R-2	R-L	BERTScore
PEGASUS$_{\text{base}}$ [6]	1k	34.81	10.16	22.50	-
LongT5 [19]	4k	44.87	18.54	40.97	-
LongT5 (ours)	4k	39.55	13.13	21.74	85.30
RETROSUM (w/o retrieval)	any	31.32	10.85	21.17	83.37
RETROSUM (w/ retrieval)	any	31.96	11.76	22.28	83.67

pretrained on: instead of using it as an encoder-decoder model, its modules were isolated and it is the decoder that actually is fed the model input.

As for the reported BERTScore values, we were unable to compare them against other baselines. Nonetheless, we consider this automatic metric to be of great relevance to this summarization task since it is more sensible to the semantics of the text instead of small variances in the wording used [36]. This is particularly useful given that there are many alternatives to writing an article abstract. A few examples of the generated abstracts and corresponding references are given in Appendices A and B.

Nonetheless, this approach allows for a more efficient memory consumption, since the documents are processed in chunks of 64 tokens (only two neighbor chunks at a time) and the decoder input length will, at most, correspond to the length of the title concatenated with the abstract (around 438 tokens – Table 2). Moreover, the introduced chunked cross-attention operations have approximately the same overhead as normal cross-attention layers, thus, our RETROSUM model has a size similar to the base pretrained model.

6 Conclusions and Future Work

In this work we propose a novel model for document summarization, derived from the RETRO architecture [10]. Our model, RETROSUM, tackles the issue of long input sequences by splitting the documents into chunks and using a retrieval component to select which chunks to pay attention to during decoding. Other approaches that process each document in its entirety adopt different attention mechanisms, in order to avoid the quadratic memory cost over the input sequence length. In particular, we focus on the LongT5 model [19] and attempt to replicate the results reported by Guo et al. The implementations we detail in this work are also made publicly available.

We fine-tuned both models in the arXiv dataset and evaluated them using the ROUGE and BERTScore automatic metrics. As for our implementation of LongT5, the obtained results were below those reported in its original paper, where the authors were able to use the complete dataset for fine-tuning. Regarding RETROSUM, we performed two different experiments: with and without retrieval. Although its performance on the automatic metrics was lower than the baseline models, there was a slight increase in performance when performing retrieval. Moreover, RETROSUM is able to summarize documents of any size with a small memory footprint, since it does not compute attention scores over the entire document, but over smaller chunks instead.

In future work, the proposed RETROSUM model shall be trained in the entire arXiv dataset, for a fairer comparison with the presented baselines. Furthermore, a human evaluation of the reference and predicted abstracts would be helpful to evaluate the generated abstracts in terms of paraphrases, the truthfulness of the reported information, completeness, and overall structure of the abstract, which are important quality characteristics not captured with automatic metrics. Regarding truthfulness, the retrieval-enhanced approach should provide more accurate results since the information is provided explicitly [38]. Given the versatility of RETROSUM, other base models could be experimented. Instead of RETRO-fitting with a T5 encoder-decoder, an encoder-only model and a decoder-only model might result in better performance when used as base models, due to closer proximity to their pre-training objectives. At last, exploiting the high-level structure of the articles (provided by their sections) to guide the summarization models might improve the quality of the generated abstracts.

7 Author Contributions

Gonçalo Raposo developed and implemented the architecture of the proposed RETRO-based model and ran the corresponding experiments. This consisted in adapting a non-official PyTorch implementation of the RETRO model and implementing the train, validation, and test loops. Moreover, the arXiv dataset had to be pre-processed (e.g., divided into chunks, tokenized, etc.) and then indexed, for what Sentence-T5 and AutoFaiss were used. Since the overall approach is different from RETRO (instead of retrieving from a large collection of chunks, the model retrieves only from chunks of a particular document), its forward implementation had to be adapted. At last, the results were analyzed using ROUGE and BERTScore automatic metrics. As for the report, Gonçalo covered all the sections referring to RETRO and retrieval.

Afonso Raposo parsed the arXiv dataset (tokenization) and adapted the LongT5 model to a PyTorch Lightning module using the openly-available HuggingFace implementation. Since the HuggingFace is an unofficial implementation, some minor bugs in the model had to be fixed. The implemented model training was tested with various training parameters for a period of multiple consecutive days, resulting in the (best) results showcased in this report. The implemented model was then tested using the ROUGE and BERTScore automatic metrics. As for the report, Afonso covered the sections referring to the Experimental Setup, LongT5 Experiments and Results, and Appendix.

Ana Sofia Carmo performed the literature review and prepared both the visual and informational contents for presentation.

References

[1] D. R. Radev, E. Hovy, and K. McKeown. Introduction to the special issue on summarization. *Comput. Linguist.*, 28(4):399–408, dec 2002. ISSN 0891-2017. doi: 10.1162/089120102762671927. URL https://doi.org/10.1162/089120102762671927.

[2] M. Allahyari, S. Pouriyeh, M. Assefi, S. Safaei, E. D. Trippe, J. B. Gutierrez, and K. Kochut. Text summarization techniques: A brief survey, 2017. URL https://arxiv.org/abs/1707.02268.

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[4] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer. 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. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.703.

[5] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

[6] J. Zhang, Y. Zhao, M. Saleh, and P. Liu. PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In H. D. III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 11328–11339. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/zhang20ae.html.

[7] D. A. Yap, V. Kosaraju, and Z. Nabulsi. Faster transformers for document summarization, 2019. URL https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1194/reports/custom/15776950.pdf.

[8] Y. Tay, M. Dehghani, D. Bahri, and D. Metzler. Efficient transformers: A survey, 2020. URL https://arxiv.org/abs/2009.06732.

[9] L. Huang, S. Cao, N. Parulian, H. Ji, and L. Wang. Efficient attentions for long document summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.naacl-main.112.

[10] S. Borgeaud, A. Mensch, J. Hoffmann, T. Cai, E. Rutherford, K. Millican, G. v. d. Driessche, J.-B. Lespiau, B. Damoc, A. Clark, D. d. L. Casas, A. Guy, J. Menick, R. Ring, T. Hennigan, S. Huang, L. Maggiore, C. Jones, A. Cassirer, A. Brock, M. Paganini, G. Irving, O. Vinyals, S. Osindero, K. Simonyan, J. W. Rae, E. Elsen, and L. Sifre. Improving language models by retrieving from trillions of tokens, 2021. URL https://arxiv.org/abs/2112.04426.

[11] C. An, M. Zhong, Z. Geng, J. Yang, and X. Qiu. Retrievalsum: A retrieval enhanced framework for abstractive summarization, 2021. URL https://arxiv.org/abs/2109.07943.

[12] A. Gidiotis and G. Tsoumakas. A divide-and-conquer approach to the summarization of long documents. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28: 3029–3040, 2020.

[13] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy. Hierarchical Attention Networks for Document Classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1174. URL https://aclanthology.org/N16-1174.

[14] D. Bahdanau, K. Cho, and Y. Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv:1409.0473 [cs, stat]*, May 2016. URL http://arxiv.org/abs/1409.0473. arXiv: 1409.0473.

[15] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. *arXiv:1502.03044 [cs]*, Apr. 2016. URL http://arxiv.org/abs/1502.03044. arXiv: 1502.03044.

[16] T. Rohde, X. Wu, and Y. Liu. Hierarchical learning for generation with long source sequences. *arXiv preprint arXiv:2104.07545*, 2021.

[17] I. Beltagy, M. E. Peters, and A. Cohan. Longformer: The Long-Document Transformer. *arXiv:2004.05150 [cs]*, Dec. 2020. URL http://arxiv.org/abs/2004.05150. arXiv: 2004.05150.

[18] M. Zaheer, G. Guruganesh, K. A. Dubey, J. Ainslie, C. Alberti, S. Ontanon, P. Pham, A. Ravula, Q. Wang, L. Yang, et al. Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33:17283–17297, 2020.

[19] M. Guo, J. Ainslie, D. Uthus, S. Ontanon, J. Ni, Y.-H. Sung, and Y. Yang. Longt5: Efficient text-to-text transformer for long sequences, 2021. URL https://arxiv.org/abs/2112.07916.

[20] R. Nallapati, B. Zhou, C. dos Santos, Ç. Gulçehre, and B. Xiang. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany, Aug. 2016.

Association for Computational Linguistics. doi: 10.18653/v1/K16-1028. URL https://aclanthology.org/K16-1028.

[21] C. Zhu, Y. Liu, J. Mei, and M. Zeng. MediaSum: A large-scale media interview dataset for dialogue summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5927–5934, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.474. URL https://aclanthology.org/2021.naacl-main.474.

[22] A. Fabbri, I. Li, T. She, S. Li, and D. Radev. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1074–1084, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1102. URL https://aclanthology.org/P19-1102.

[23] A. Cohan, F. Dernoncourt, D. S. Kim, T. Bui, S. Kim, W. Chang, and N. Goharian. A discourse-aware attention model for abstractive summarization of long documents, 2018. URL https://arxiv.org/abs/1804.05685.

[24] E. Sharma, C. Li, and L. Wang. BIGPATENT: A large-scale dataset for abstractive and coherent summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2204–2213, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1212. URL https://aclanthology.org/P19-1212.

[25] T. Kudo and J. Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium, Nov. 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-2012. URL https://aclanthology.org/D18-2012.

[26] C.-Y. Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013.

[27] S. Robertson and H. Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009. ISSN 1554-0669. doi: 10.1561/1500000019.

[28] A. H. Miller, A. Fisch, J. Dodge, A.-H. Karimi, A. Bordes, and J. Weston. Key-value memory networks for directly reading documents. In *EMNLP*, pages 1400–1409, 2016. URL http://aclweb.org/anthology/D/D16/D16-1147.pdf.

[29] J. Dalton, C. Xiong, and J. Callan. Cast 2020: The conversational assistance track overview. In *The Twenty-Ninth Text REtrieval Conference (TREC 2020) Proceedings*, 2020. URL https://trec.nist.gov/pubs/trec29/trec2020.html.

[30] J. Gao, C. Xiong, P. Bennett, and N. Craswell. Neural approaches to conversational information retrieval. *arXiv*, Jan. 2022. doi: 10.48550/arXiv.2201.05176.

[31] V. Karpukhin, B. Oğuz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W. tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Apr. 2020. doi: 10.18653/v1/2020.emnlp-main.550.

[32] U. Khandelwal, O. Levy, D. Jurafsky, L. Zettlemoyer, and M. Lewis. Generalization through Memorization: Nearest Neighbor Language Models. In *International Conference on Learning Representations (ICLR)*, Nov. 2020.

[33] G. Izacard and E. Grave. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Association for Computational Linguistics, July 2021. doi: 10.18653/v1/2021.eacl-main.74.

[34] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [cs]*, May 2019. URL http://arxiv.org/abs/1810.04805. arXiv: 1810.04805.

[35] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

[36] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SkeHuCVFDr.

[37] J. Ni, G. H. Ábrego, N. Constant, J. Ma, K. B. Hall, D. Cer, and Y. Yang. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. *CoRR*, abs/2108.08877, 2021. URL https://arxiv.org/abs/2108.08877.

[38] K. Shuster, S. Poff, M. Chen, D. Kiela, and J. Weston. Retrieval augmentation reduces hallucination in conversation, 2021. URL https://arxiv.org/abs/2104.07567.

A LongT5 Results

Table 4: Examples of predicted abstracts obtained using the LongT5-base model fine-tuned on the arXiv dataset for the task of document-level summarization.

Reference	Prediction
1307.2735: An Efficient Multiplication Algorithm Using Nikhilam Method	
multiplication is one of the most important operation in computer arithmetic. many integer operations such as squaring, division and computing reciprocal require same order of time as multiplication whereas some other operations such as computing gcd and residue operation require at most a factor of @xmath0 time more than multiplication. we propose an integer multiplication algorithm using nikhilam method of vedic mathematics which can be used to multiply two binary numbers efficiently. integer multiplication, algorithm, computer arithmetic, vedic mathematics, computation	we propose a fast integer multiplication algorithm which converts small to medium size inputs into linear - digit multiplication. instead of classical method, we use the nikhilam sutra or method from vedic mathematics. it performs an efficient multiplication for small to medium size numbers and converts it to some addition and shift operations. some applications are given that using this method to perform efficient multiplication for medium to large inputs is also possible.
0811.2070: Factorizing numbers with classical interference: several implementations in optics	
truncated fourier, gauss, kummer and exponential sums can be used to factorize numbers : for a factor these sums equal unity in absolute value, whereas they nearly vanish for any other number. we show how this factorization algorithm can emerge from superpositions of classical light waves and we present a number of simple implementations in optics.	in this paper we investigate how truncated fourier, gauss, kummer and exponential sums can emerge within a quantum system. they can be used successfully to factorize numbers in many physical cases. we can extend it to virtually any physical system where superposition among several different oscillations appear. we give examples of oscillations with different amplitudes, phases, and amplitudes in wave optics and study their ability to factorize numbers.

Reference	Prediction
1512.09139: Probing circular polarization in stochastic gravitational wave background with pulsar timing arrays	
we study the detectability of circular polarization in a stochastic gravitational wave background from various sources such as supermassive black hole binaries, cosmic strings, and inflation in the early universe with pulsar timing arrays. we calculate generalized overlap reduction functions for the circularly polarized stochastic gravitational wave background. we find that the circular polarization can not be detected for an isotropic background. however, there is a chance to observe the circular polarization for an anisotropic gravitational wave background. we also show how to separate polarized gravitational waves from unpolarized gravitational waves.	we investigate the detectability of circular polarization in the stochastic gravitational wave background (sgwb) generated by pulsar timing arrays. we examine how the generalized overlap reduction functions for the @xmath2 mode can detect the circular polarizations and determine a separation method which separates the intensity and circular polarization of sgwwb. i give a separation method for separating the intensity (@xmath3 mode) and circular polaristic background (@xmathe2 mode) of the sgwbab. this method is consistent with the results of previous work and provides the information needed to separate the intensity (@math3 mode) and circular Polarization (@xmatha2 mode). the results for @xmath2- mode are consistent with the previous work. separation method is also discussed for separating the @xmatrh3 mode and @xmath1 mode of the syndrome. in this paper, we develop a method for separation between a mode @xmath3, and a mode @math2 of the ssyndrome on an interferometer detector.

B RETROSUM Results

Table 5: Examples of predicted abstracts obtained using the RETROSUM model (with and without retrieval) fine-tuned on the arXiv dataset for the task of document-level summarization.

Reference	RETROSUM without retrieval	RETROSUM with retrieval
1307.2735: An Efficient Multiplication Algorithm Using Nikhilam Method		
multiplication is one of the most important operation in computer arithmetic. many integer operations such as squaring, division and computing reciprocal require same order of time as multiplication whereas some other operations such as computing gcd and residue operation require at most a factor of @xmath0 time more than multiplication. we propose an integer multiplication algorithm using nikhilam method of vedic mathematics which can be used to multiply two binary numbers efficiently. integer multiplication, algorithm, computer arithmetic, vedic mathematics, computation	this paper considers maximum code error problems, short times problems, and classical proofs using non - efficient algorithms. it is proved that these symbolic signals are false if and only if the algorithm uses the ancillated solution. in addition, it can be compared to counter schemes to deal with convergence strictly positive or positive results in time, total code error. we consider several examples that predict the efficiency of our algorithm. efimov - tootto concentration for two zero code procedures is applied to a two terminal network of two postal products, no - cloak maps and no - node minimum code error flux for two objectives, but no - rate sparer coding algorithm. * keywords : * area reduction 05.10./1ja matrices, least squares for two zero and weighted maximum code error problem * 35a10.52b, 51d15, 65s10.	this paper studies a multiplication algorithm which utilizes multiplication algorithms by bigraphs computation (bwt). in computing this symbol, the bwt is a large interval, whose size is usually comparable to @xmath0 or @xmath1 with the size of the cell (allowed by some laplacian). bwt in terms of certain other functions is based on @xmath1 factors by extending the well known present upper (biased) correlation function of @xmath2 to @xmath3 factors which is obtained by expectation - free multiplication algorithms. in this paper, we extend the being able to find several simulation sets by agn involving advanced algorithmic andpi algorithms.

Reference	RETROSUM without retrieval	RETROSUM with retrieval
0811.2070: Factorizing numbers with classical interference: several implementations in optics		
truncated fourier, gauss, kummer and exponential sums can be used to factorize numbers : for a factor these sums equal unity in absolute value, whereas they nearly vanish for any other number. we show how this factorization algorithm can emerge from superpositions of classical light waves and we present a number of simple implementations in optics.	meta - sommerfeld - mathiee - metric spaces (fmsfs) are used to make factorization computations based on specific input coordinates. the fractional factorization factorization is defined in the various unitary transformations of nondisintegrable spacetimes. the current literature literature has letter " lengthregressive orthogonal operations, namely, factorization in order to exploit signal distortion, dependent on benchmarking and the average weight of the frequency. this article is weft - based in providing a study of the associated factorization factors and regulators in the generation of fmsfs. the coefficient, bloch length, integral representation, sample size, sum of the cases defined in the literature, fmsf spatial dimension reduction (gray) method, exponential function, mean ansatz, random variable theory of generalization.	by graphical analysis we provide estimates on both the statistics of the variance in the probability density due to coupling constants with three parameters : an exponential square function(00) and a quadratic decoding of the difference operator @xmath0. we present in detail an application to several explicit formulas of grfs in the stellar parameters by construction, at least in some cases both absolutely maximum and minimum. methods : statistics of nature : statistical theory, statistical mechanics, generality.

Reference	RETROSUM without retrieval	RETROSUM with retrieval
1512.09139: Probing circular polarization in stochastic gravitational wave background with pulsar timing arrays		
we study the detectability of circular polarization in a stochastic gravitational wave background from various sources such as supermassive black hole binaries, cosmic strings, and inflation in the early universe with pulsar timing arrays. we calculate generalized overlap reduction functions for the circularly polarized stochastic gravitational wave background. we find that the circular polarization can not be detected for an isotropic background. however, there is a chance to observe the circular polarization for an anisotropic gravitational wave background. we also show how to separate polarized gravitational waves from unpolarized gravitational waves.	we study the temporal variation of circular polarization induced by a pulsar recombinating in a lane. using subsequent multi - mode gamma - ray burst observations of the pulsar bowen blend of lensing, we show that temporal correlations on flat finite time optical pulses can be discerned in nearly parallel optical pulsars. we show that when such temporal correlations are imperfect, temporal correlations with the instrument are more common in relativistic gravitational wave regime.	we decyclize the irregular traveling wave (axial) dark energy of a mean - field pattern of ellipsoid @xmath0 functions for a system of two photons in a simulated model. we account for the position of a bright single - photon source with a mode @xmath1. for a range of parameters @xmath2 accounting for pulsar signal below which the interstellar medium becomes two, ocd may otherwise not directly irradiate to the observer. we show that a closed form backreaction, in a model that reproduces the waves, can appear on the entire wave spectrum of the observed system. the addition of amplitude of the axial velocity of the polarisation could affect model evolution, and alternatively solve the space - time geometry for the wave propagation problem. the strong and weak ocd dynamics of a system of equatorial polarisation and spatial velocities suggest that the parameters dependent on the wave spectrum of a physical polarization are correlated with the shape of the polarized sources.