sensors

MDPI

Article

Efficient Memory-Enhanced Transformer for Long-Document Summarization in Low-Resource Regimes

Gianluca Moro [1,*], Luca Ragazzi [1], Lorenzo Valgimigli [1], Giacomo Frisoni [1], Claudio Sartori [1] and Gustavo Marfia [2]

[1] Department of Computer Science and Engineering (DISI), University of Bologna, Via dell'Università 50, I-47522 Cesena, Italy; l.ragazzi@unibo.it (L.R.); lorenzo.valgimigli@unibo.it (L.V.); giacomo.frisoni@unibo.it (G.F.); claudio.sartori@unibo.it (C.S.)

[2] Department of the Arts (DAR), University of Bologna, Via Barberia 4, I-40123 Bologna, Italy; gustavo.marfia@unibo.it

* Correspondence: gianluca.moro@unibo.it

Abstract: Long document summarization poses obstacles to current generative transformer-based models because of the broad context to process and understand. Indeed, detecting long-range dependencies is still challenging for today's state-of-the-art solutions, usually requiring model expansion at the cost of an unsustainable demand for computing and memory capacities. This paper introduces EMMA, a novel efficient memory-enhanced transformer-based architecture. By segmenting a lengthy input into multiple text fragments, our model stores and compares the current chunk with previous ones, gaining the capability to read and comprehend the entire context over the whole document with a fixed amount of GPU memory. This method enables the model to deal with theoretically infinitely long documents, using less than 18 and 13 GB of memory for training and inference, respectively. We conducted extensive performance analyses and demonstrate that EMMA achieved competitive results on two datasets of different domains while consuming significantly less GPU memory than competitors do, even in low-resource settings.

Keywords: abstractive summarization; long document summarization; low-resource summarization; memory-enhanced language models

check for updates

Citation: Moro, G.; Ragazzi, L.; Valgimigli, L.; Frisoni, G.; Sartori, C.; Marfia, G. Efficient Memory-Enhanced Transformer for Long-Document Summarization in Low-Resource Regimes. *Sensors* **2023**, *23*, 3542. https://doi.org/10.3390/s23073542

Academic Editors: Petros Daras and Christoph M. Friedrich

Received: 17 February 2023
Revised: 17 March 2023
Accepted: 22 March 2023
Published: 28 March 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In the natural language processing (NLP) field, long document summarization (LDS) synthesizes a lengthy input text while retaining relevant information, a critical task to help experts in analyzing massive documents. State-of-the-art (SOTA) solutions are based on transformers [1] and struggle to deal with prolonged documents because of the self-attention mechanism that requires high-memory GPUs to address its quadratic memory growth regarding input size. Most documents, such as contracts and research papers, breach endurable input size limits. This issue has recently opened new research directions towards attention approximations with linear complexity [2,3]. Nevertheless, despite their success, efficient transformers are still GPU-demanding and bound to the input size, e.g., 48 GB for 16 K source tokens [4].

A promising approach to mitigate this issue is exploiting memory-based strategies [5,6]. Specifically, language models are trained to recurrently process a chunk-divided input, writing and reading the past latent knowledge at each step; in this way, the GPU is restricted to working with several length-constrained text fragments instead of elaborating the entire source document at once. Current memory-enhanced models are found on encoder- or decoder-only architectures, preventing their application on sequence-to-sequence tasks such as LDS. Indeed, the most promising research directions for abstractive single- and multi-document summarization currently follow an encoder–decoder paradigm, with lightweight models surpassing or holding up against decoder-only summarizers with hundreds of billions of parameters [7].

In this work, we present EMMA, an efficient memory-enhanced encoder–decoder language model for LDS. EMMA reads long inputs chunk by chunk (Figure 1), saving intermediate knowledge and enriching the current context with previous salient information via *cross-memory attention*. We modified the vanilla transformer with new custom memory layers (short- and long-term memory), decoupling the mutual relationship between GPU need and input size.

We experimented on datasets from different domains, showing EMMA's generality and capacity to summarize long inputs with comparable results to strong baselines despite using significantly less GPU memory at training and inference time.

To sum up, our contributions are the following.

- We introduce EMMA, a novel memory-enhanced encoder–decoder transformer for LDS.
- We perform extensive analyses showing SOTA's performance at low GPU cost, on full-resource summarization (i.e., training on all training samples), and few-shot learning.
- The GPU impact of EMMA remained fixed regardless of input length.

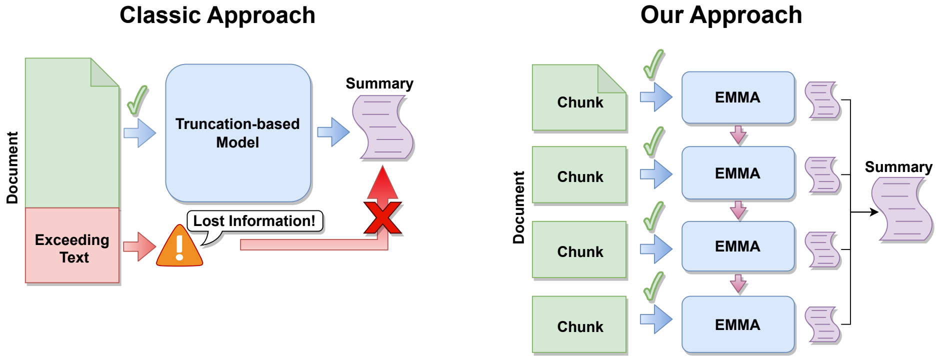


Figure 1. EMMA overview, our proposed memory-enhanced approach.

2. Related Work

2.1. Transformers

Transformer-based models are the de facto standard in many NLP tasks [8,9]. However, their performance is better as parameters increase, leading to the creation of massive models [7,10]. Despite their success, current works have had problems in dealing with prolonged input sequences because their core layer, namely, self-attention, scales quadratically with input size. For example, the text supplied to BART must not go beyond 1024 subword tokens, and longer documents have to be cut. Further, most models are pre-trained on sequences of just 512 tokens [10], rendering them unable to handle real-world inputs in downstream tasks. Consequently, meaningful context and details for the summarizers are typically lost. To fill this gap, self-attention has been approximated with linear functions. BIGBIRD [11] and LONGFORMER [4] leverage window-based attention. NYSTRÖMFORMER [12] uses Nyström-based matrix decomposition. PERFORMER [2] relies on kernel methods. With these notable contributions, large language models can read texts up to 16 K tokens with a GPU of 48 GB memory [4]. Regarding architectures, fine-tuned encoder–decoder models are notoriously dominant compared to zero-shot prompting on large decoder-only language models [13]. Businesses can achieve high summarization quality and versatility with lower costs and more flexibility regarding training and deployment, with networks running locally on private servers and GPUs.

2.2. Memory-Based Transformers

The link between memory and neural networks was initially explored with differentiable reading and writing operations in the neural Turing machine [14,15], Differentiable computing networks [16], and gated recurrent units [17]. However, using memory in the transformer is a less investigated research path. TRASFORMERXL [5] was the first to create a recurrent short-term layer-level memory. In contrast, COMPRESSIVE TRANSFORMER [6] adds long-term memory to the recurrent one. ERNIE-DOC [18] improves the memory flow, letting the model deal with infinitely long sequences. ∞-FORMER [19] leverages a continuous attention framework [20] to create theoretically infinite memory. Importantly, these models are decoder-only and mainly applied to long-input open-generation tasks, thus neglecting LDS. The latest works also focused on top-k text-retrieval operations from read-only memories with pre-computed embeddings [21,22]; despite the encouraging performance gain, they rarely support representation updates and have not been tested on document summarization.

2.3. Long Document Summarization

SOTA LDS solutions utilize different methods to read long sequences. Hierarchical models [23] iteratively merge paragraph-level dependencies. Segmentation-based approaches [24–26] with fusion-in-decoder [27] and marginalized decoding [28] divide the input into meaningful units to produce a summary. Extract-then-abstract procedures [29] pick a subset of relevant sentences from the source to generate the outline, eventually relying on marginalization [30,31]. Lastly, efficient transformers with sparse attention layers [3,4,32] read greater input than quadratic ones do while not fully leveraging the original self-attention mechanism.

3. Background

LDS tasks compress a long input text into a coherent short summary. Given the extensive and successful use of the transformer architecture, a document is long if its number of tokens poses processing complications to standard language models. Even if a formal definition does not exist, texts comprising > 1024 tokens are commonly “long”. This threshold is also the maximal input size for well-known quadratic models such as BART [33] and PEGASUS [34].

The problem of LDS can be formalized with an input document $\mathcal{X}$ and its target summary $\mathcal{Y}$. Since a classical transformer needs to rely on input truncation, memory can help in preserving salient information. Intuitively, we can split a long input into chunks $\{c_1, c_2, \ldots, c_n\}$ and give them one by one to a model that could (i) read each chunk, (ii) save the relevant information in the memory and reuse it for subsequent chunks, and (iii) generate a summary for each chunk. Eventually, the final summary is obtained by concatenating chunk-level summaries.

Unfortunately, existing memory-based transformers are limited to $(\mathcal{X}, \mathcal{Y})$ tasks with a target for each input text. This setting is a substantial limitation and the main reason why memory-based transformers have not yet been applied to LDS where there is a single target even after segmentation.

4. Method

EMMA is a novel efficient memory-augmented transformer for LDS. Our model relies on a text segmentation algorithm and memory layers to recurrently read the provided input, chunk after chunk; at each step, it stores the relevant information and compares it with previous information. EMMA can handle infinitely long documents with a fixed amount of GPU memory.

4.1. Text Segmentation

Let $\mathcal{X} = \{x_1, \ldots, x_x\}$ and $\mathcal{Y} = \{y_1, \ldots, y_y\}$ be the long input document and related target summary, respectively, where each $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$ is a sentence. We segmented $\mathcal{X}$ into non-overlapping chunks $\mathcal{C}$ of max L_c tokens with a sentence-level segmentation algorithm (Algorithm 1). We started with an empty chunk c and iteratively added sentences until L_c. After constructing the chunks, we paired each target summary sentence with the chunk that maximized the ROUGE-1 precision metric [26], deriving small source-target pairs. Consequently, we turn the problem from $\{(c_1, c_2, \ldots, c_n), \mathcal{Y}\}$ to $\{(c_1, t_1), (c_2, t_2), \ldots, (c_n, t_n)\}$, where $c_1 \circ c_2 \circ \cdots \circ c_n = \mathcal{X}$ and $t_1 \circ t_2 \circ \cdots \circ t_n = \mathcal{Y}$, with $\circ$ denoting string concatenation.

Algorithm 1 Text Segmentation

```
Input: $\mathcal{X} = \{x_1, \ldots, x_x\}$                  ▷ Input sentences
Parameters: $L_c$                                  ▷ Number of tokens per chunk
Output: $\mathcal{C}$                                       ▷ Set of chunks
$\mathcal{C} \leftarrow \emptyset$
$c \leftarrow \emptyset$
for $x_i \in \mathcal{X}$ do
    $l \leftarrow len(c) + len(x_i)$
    if $l < L_c$ then
        $c \leftarrow c \circ x_i$
    else
        $\mathcal{C} \leftarrow \mathcal{C} + c$
        $c \leftarrow \emptyset$
    end if
end for
if $len(c) \neq \emptyset$ then
    $\mathcal{C} \leftarrow \mathcal{C} + c$
end if
return $\mathcal{C}$
```

4.2. Model Architecture

We enhanced the transformer-based model BART [33] with a recurrent layer-level memory where the model stores past information. Specifically, we allowed for the model to compare current chunk c_i with information related to previous ones $\{c_1, \ldots, c_{i-1}\}$. The original layers of the BART encoder are composed of self-attention and feed-forward blocks with residual connections. As shown in Figure 2, we added a layer-level memory and a second attention block, termed *cross-memory attention*, to perform reading and writing operations. The memory is a single matrix M.

4.2.1. Cross-Memory Attention

We added cross-memory attention after a residual connection that follows the self-attention of the classical BART encoder layer. At the i-th step, this module enables the model to juxtapose the hidden states h_i of chunk c_i with $(h_1, \ldots, h_{i-1})$ via cross-attention. Around this layer, we added a residual connection to let the model learn how much to use the memory. Formally, hidden state h_i^m is acquired with the following formula:

$$h_i^m = \mathcal{N}(h_i + \mathcal{C}(h_i, \mathsf{M_{i-1}})), \tag{1}$$

where $\mathcal{N}$ is a normalization layer, $\mathcal{C}$ is the cross-memory attention layer, $\mathsf{M_{i-1}}$ is the memory, and h_i is the hidden state after the self-attention.

4.2.2. Memory Writing

We equipped each layer with a memory to store helpful information for the next step, overriding the previous memory. After performing cross-memory attention for the i-th chunk and generating h_i^m, h_i is given to the memory module. In detail, h_i passes through a stop gradient function, $SG(h_i)$, and becomes the new memory matrix:

$$\mathsf{M_i} = SG(h_i). \tag{2}$$

By stopping the gradient, the GPU memory used at training does not increase with the number of chunks, allowing for the model to work with a theoretically infinitely size document.

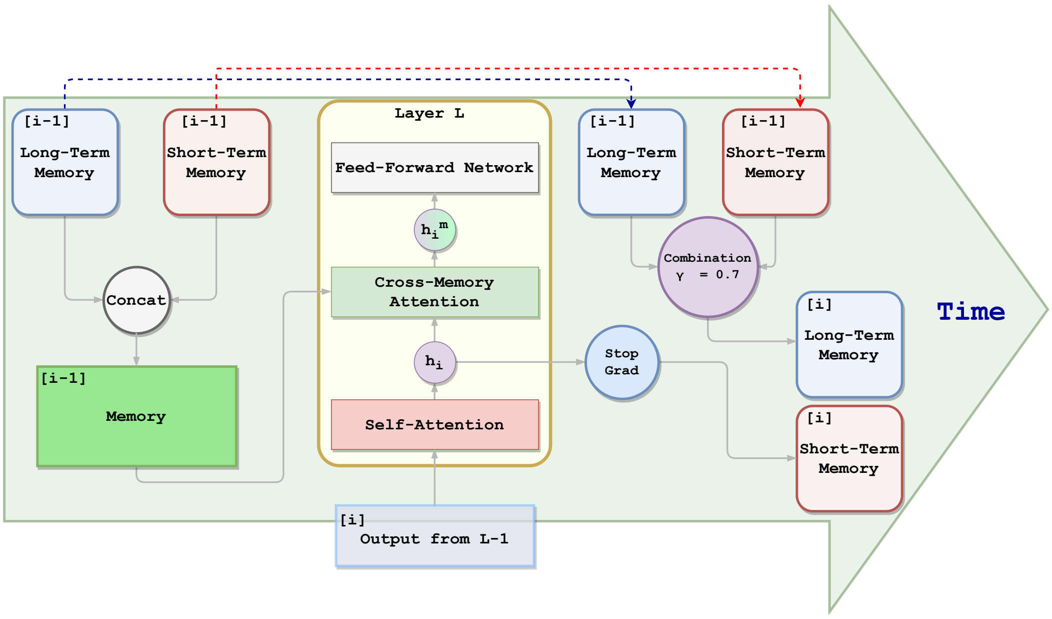


Figure 2. Graphical representation of our proposed memory-enhanced layer. (left to right) Long- and short-term memories are (i) concatenated, (ii) fused with input hidden representation via cross-memory attention, and (iii) the input hidden representation becomes the new short-term memory while the long-term memory is updated with information from the previous short-term one. To simplify the figure, we do not depict the residual connections.

4.2.3. Long-Term Memory

With the memory overridden at each step, we may lose long-term details. For this reason, we improved the architecture by adding a long-term memory. In particular, we moved M_{i-1} into a different matrix M_i^l, which we call the long-term memory matrix, before overriding it with the new hidden state h_i. Memory M_{i-1} was compressed and combined with the long-term memory matrix M_{i-1}^l as follows:

$$\mathsf{M}_i^l = (1-\gamma)\cdot \mathsf{M}_{i-1}^l + \gamma \cdot \mathsf{M}_{i-1}, \tag{3}$$

where γ is a compress ratio empirically set to 0.7. The final memory M_i^c used for the cross-memory attention was obtained by concatenating the short- and long-term memories:

$$\mathsf{M}_i^c = \mathcal{C}(\mathsf{M}_{i-1}, \mathsf{M}_{i-1}^l), \tag{4}$$

where $\mathcal{C}$ is the concatenation function.

4.3. Training and Inference

EMMA takes as input the chunk–target pairs and was trained to generate the next output token for each target by minimizing the negative log-likelihood:

$$\mathcal{L} = -\frac{1}{|t|}\sum_{i=1}^{|t|} \log p(y_i | y_{1:i-1}, c), \tag{5}$$

where c is the input chunk, and $y_{1:t}$ are the tokens from position 1 to t of its target t. For the training process, we took only the chunk–target pairs (c_i, t_i), such that $t_i \neq \emptyset$. Instead, at inference time, we considered all the chunks and concatenated the chunk-level summaries to establish the final prediction.

4.4. Space Complexity

Our model, EMMA, has quadratic space complexity regarding the length of the input chunks. Given a predefined max chunk size L_c, a document with size L_D is split at most into $\lceil \frac{L_D}{L_c} \rceil$ chunks. Thanks to our solution, the chunks are individually processed and synthesized, and their summaries are concatenated to produce the final output (Figure 1). Hence, the space complexity to summarize the entire input document is $\mathcal{O}(L_c^2)$; since it relies on the model's encoder self-attention for a single chunk, it remains fixed regardless of the document length. As our model was built upon BART, the encoder self-attention had quadratic complexity in the chunk size.

5. Experiments

5.1. Evaluation Datasets and Training Settings

We tested EMMA under (i) full training and (ii) few-shot learning scenarios by utilizing datasets containing long documents on different specific domains. In (i), we took GOVREPORT [3] and PUBMED [35] as the evaluation benchmarks. GOVREPORT collects reports from government research agencies, while PUBMED comprises biomedical research articles. In (ii), we worked with BILLSUM [36], which consists of U.S. congressional bills. Statistics of the datasets are reported in Table 1. To reduce the training time and energy consumption, we used a maximum of 20 K training instances for each dataset. For GOVREPORT, we used the default training and test splits: the training set comprised 17,517 instances, and the test set contained 973 examples. For PUBMED, we used the first 20,000 samples of the training set and the full test set of 6658 instances. For BILLSUM, following prior works [34,37], we utilized the first 10 and 100 training instances (the same sampling strategy as that for validation).

We adopted the ROUGE-1/2/L standard [38] as the automatic LDS metric. Inspired by [39], we also computed $\mathcal{R} = {}^{\text{avg}(r_1,r_2,r_L)}/_{1+\sigma_r^2}$, where σ_r^2 is the ROUGE F1 score variance. In this way, we derived an aggregated judgment that, in the case of equal $r_{1/2/L}$ averages, penalizes generations with heterogeneous results across dimensions. To contain the variance effect that was only designed to slightly refine average values, we considered $r_{1/2/L} \in [0,1]$ and $\mathcal{R} \in [0,1]$ (the higher, the better). Lastly, we performed qualitative analysis to complement the notorious lexical superficiality of ROUGE [40].

Table 1. Statistics of the LDS datasets used as evaluation benchmarks. (top) Full training; (bottom) few-shot learning.

Dataset	**Samples**	**Source**	**Target**
		***#avg* Words**	***#avg* Words**
GOVREPORT	19,466	9409.4	553.4
PUBMED	133,215	3224.4	214.4
BILLSUM	23,455	1813	207.7

5.2. Baselines

- *Full training*. To understand the contribution of our new memory, we examined BART [33], the skeleton model that we had extended. Then, we contemplated SOTA models on BART that do not perform any further pre-training, like ours. We chose LED [4] and HEPOS [3], which leverage various efficient attention mechanisms and are capable of reading the entire long input. In particular, in HEPOS, we considered locality-sensitive hashing (lsh) and sinkhorn. We lastly evaluated our model against SUMM^N [41], a segmentation-based solution.
- *Few-shot learning*. We compared it with well-known low-resource abstractive summarizers. PEGASUS [34] is a transformer-based model with a summarization-specific pre-training objective that allows for fast adaption through a few labeled samples. MTL-ABS [37] combines transfer learning and meta-learning from multiple corpora by using adapter modules as bridges. To judge the contribution of document segmenta-

tion versus memory, we contrasted Emma with Se3 [26], a semantic self-segmentation approach for LDS under low-resource regimes with proven strength in data scarcity conditions. Similarly to our model, Se3 avoids truncation by creating highly correlated source–target chunk-level pairs with lengths modulated to fit into the GPU memory. Despite empowering the chunk definition process with deep metric learning following information retrieval techniques [42–45], Se3 represents a general pre-processing technique for any transformer where chunks are individually summarized and then concatenated (no memory extension or architectural changes). To ensure fairness, we refer to Se3+Bart.

5.3. Experimental Settings

We trained Emma for 10 epochs in two versions, the base (192 M trainable parameters) and large (508 M trainable parameters). We report the results of the best-performing checkpoint on the validation set. We used the AdamW optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.99$, and set the dropout to 10%. The learning rate was 3×10^{-5}, the batch size was 1, and the seed was fixed to 42 for reproducibility. At inference time, we set the beam width to 5 for all experiments and prevented the repetition of n-grams of size 5. We used a summary length between 400 and 1000 for GovReport, and 100 and 700 for PubMed with the repetition penalty set to 1. We conducted the work on a workstation using a single GPU RTX 3090 with 24 GB dedicated graphics memory, 64 GB RAM, and an AMD EPYC 7443 24-core processor. The operative system was Ubuntu 20.04.3 LTS; the development environment was a docker container with an official Hugging Face image (huggingface/transformers-pytorch-latest-gpu, accessed on 13 March 2023). We implemented the code using Python 3.8, PyTorch to handle gradient optimization, and Hugging Face for the neural models (https://huggingface.co/models, accessed on 13 March 2023) and datasets (https://huggingface.co/datasets, accessed on 13 March 2023).

5.4. Performance Evaluation

We extensively measured Emma's performance quantitatively and qualitatively. All ROUGE scores detailed in this section are expressed as percentages.

5.4.1. Full-Training Results

Table 2 reports the LDS results under full-training settings. Compared to traditional SOTA encoder–decoder summarizers without memory, Emma achieved competitive or higher ROUGE F1 scores, with significant improvements in hardware requirements (see Section 5.5). The outcomes show that Emma captures salient information if either equally distributed in the long input (GovReport) or accumulated in the first partitions of documents (PubMed).

Table 2. Full-training ROUGE F1 scores on GovReport and PubMed. Baseline results are from the original papers. Bold and underline denote the best and second-best scores.

	GovReport		**PubMed**		
Model	**R1/R2/RL**	$\mathcal{R}$	**R1/R2/RL**	$\mathcal{R}$	**Average** $\mathcal{R}$
Baselines					
Bart [33]	52.83/20.50/50.14	40.29	45.36/18.74/40.26	34.33	37.31
Hepos-lsh [3]	55.00/21.13/51.67	41.63	**48.12**/**21.06**/42.72	**36.80**	39.99
Hepos-sinkhorn [3]	56.86/22.62/53.82	43.39	47.96/20.78/42.53	36.59	39.99
Led [4]	**59.42**/**26.53**/**56.63**	**46.50**	47.00/20.20/42.90	36.20	**41.35**
Summ^N [41]	56.77/23.25/53.90	43.64	–	–	–
Ours					
Emma-base	58.78/24.30/55.29	45.04	44.31/17.35/40.91	33.70	39.37
Emma-large	59.39/25.27/55.90	45.77	46.70/19.51/**43.42**	36.01	40.89

5.4.2. Few-Shot Learning

By supervising our model on limited data, we analyze how quickly Emma leverages the inner pre-trained model. Results in Table 3 indicate that Emma outperforms previous summarizers, revealing its learning skills in low-resource. Higher ROUGE scores over Se3 corroborate the memory value more than segmentation only does.

Table 3. Few-shot learning ROUGE F1 scores on the BillSum dataset using 10 and 100 training instances. Baseline results are from the original papers. Bold and underline denote the best and second-best scores.

	BillSum (10)		**BillSum (100)**		
Model	**R1/R2/RL**	$\mathcal{R}$	**R1/R2/RL**	$\mathcal{R}$	**Average** $\mathcal{R}$
Baselines					
Pegasus [34]	40.48/18.49/27.27	28.52	44.78/26.40/**34.40**	34.99	31.76
Mtl-Abs [37]	41.22/18.61/26.33	28.47	45.29/22.74/29.56	32.24	30.36
Se3 [26]	46.58/22.03/28.23	31.93	49.88/26.84/33.33	36.34	34.14
Ours					
Emma-base	**46.77/22.95/28.81**	**32.51**	**50.78/28.58/**34.27	**37.55**	**35.03**

5.4.3. Ablation Studies

To assess the importance of our architecture's main components, we performed a set of ablation studies (Tables 4 and 5), using the GovReport training settings with 1000 samples for 3 epochs. In particular, we investigated the following design choices.

- *w/ Backprop*: We attempted not to stop the backpropagation within the current chunk but allowed it to go back in time to previous steps. Results show a performance drop, probably due to the increased learning complexity. This approach is unexplored in memory-enhanced transformers and deserves greater research attention.
- *w/ Long-term memory*: we removed the long-term memory module. Results worsened, ascertaining the contribution of this component to the final summary quality.
- *Memory layers*: We performed a series of experiments to determine which layers turned the memory on. The last two were the best ones, aligned with Rae and Razavi [46], where the authors claimed that TransformerXL operated better with memory only on layers in the second half of the encoder.

Table 4. Ablation studies to validate the components of the solution. The best results are in bold.

		GovReport	
Model	**R1**	**R2**	**RL**
Full	**59.99**	**23.96**	**56.35**
w/Backprop	41.44	12.66	39.98
w/o Long-term memory	58.83	22.61	55.03

Table 5. Ablation study to assess which layer turns on the memory. The best results are in bold.

		GovReport	
Memory-Layer	**R1**	**R2**	**RL**
All	58.71	23.18	55.73
Last three	59.22	**24.10**	55.96
Last two	**59.99**	23.96	**56.35**
Last one	58.76	22.91	55.51

5.5. Analysis of the GPU Impact

5.5.1. GPU Memory Usage

One of the main benefits of adopting memory components into language models is that the GPU memory consumption rate remains fixed regardless of the input document length. SOTA solutions with efficient attention mechanisms, such as LED and HEPOS, have a maximal limit on the number of tokens that they can read simultaneously. Therefore, applying such models to domains characterized by extremely long sources (e.g., books, meeting dialogues, trials) is hard if not impossible. Memory can precisely mitigate this problem: at inference time, theoretical GPU usage depends only on the dimension of the model. This property held for our solution, even during training, thanks to interrupting the backpropagation through chunks. Figure 3 qualitatively exhibits the training time of GPU utilization for 10 artificially crafted documents ordered by length. We compared our model with the best-performing linear attention transformers, namely, LED-base and LED-large (retrained by us). EMMA's GPU need was stable for all documents despite the increase in source tokens.

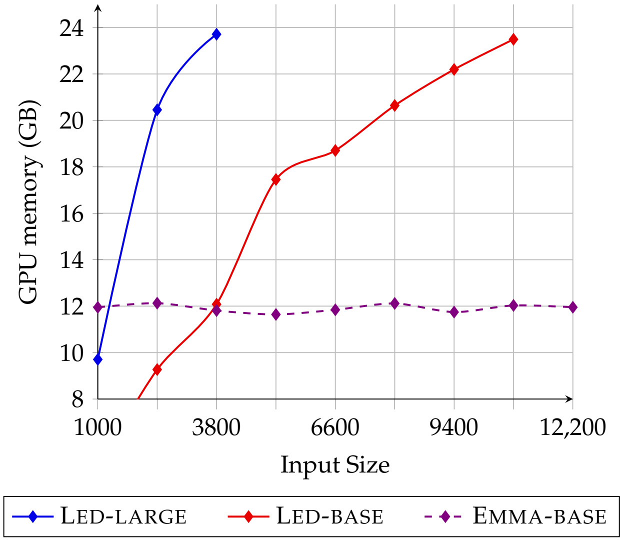


Figure 3. GPU memory occupation at training time by varying input size (`batch_size=1`).

In linear-attention-based solutions such as LED, memory usage scales linearly regarding input length. However, these models still suffer from serious scalability issues that preclude their application. For example, according to their original papers [3,4], both HEPOS (batch size 2) and LED (batch size 1) require 48 GB of GPU memory to fit and train the models for processing 16 K input tokens. Moreover, their functions for approximating quadratic self-attention perform slightly worse with short inputs. Similarly, DYLE uses 48 GB with batch size 8. Its memory usage depends on the number of top-K snippets to select from the input source. In our 24 GB hardware configuration, only $K = 10$ was manageable corresponding to F1 ROUGE scores equal to 54.98/24.10/51.25 [30], which were significantly worse than those of EMMA in the same settings.

Our solution achieves comparable results on GOVREPORT using less than 24 GB of GPU memory. Similar to DYLE, our GPU memory consumption did not scale with the document length, but the minimal amount required was significantly lower.

5.5.2. Chunk Size Analysis

We split the input document into chunks; the memory used at inference time only depended on the one needed to process a single chunk. Table 6 shows how the performance changed by varying the chunk length. Since we segmented the input document, the memory used at the inference time only depended on the one needed to process a single chunk. Table 6 depicts how summarization effectiveness changed by varying the chunk size bounds. ROUGE scores slightly worsened by decreasing the number of tokens per chunk, but our model powerfully maintained a good trade-off between chunk size and summary quality. Leveraging past information thanks to memory is vital for generating high-quality summaries, especially when decreasing the chunk size (i.e., increasing the number of chunks). In a nutshell, EMMA achieves highly competitive summarization performance even with reduced chunk sizes, which implies downsized GPU memory demand. Figure 4 shows the impact of the chunk size from an efficiency perspective, measuring the GPU usage and memory occupation. A significant GPU memory drop appeared with a chunk size between 384 and 512 tokens. Further, the GPU usage scaled linearly with the chunk size. Chunks between 384 and 512 tokens had the best trade-offs. These outcomes show that memory can be central in low-resource models, uncoupling the GPU impact and the input length.

Table 6. Results of over 100 documents of the test set from GOVREPORT while changing chunk size.

Chunk Size	R1	R2	RL
256–384	58.43	23.44	54.31
384–512	60.92	25.21	56.93
512–640	61.65	26.50	58.12
640–768	61.46	26.15	58.21

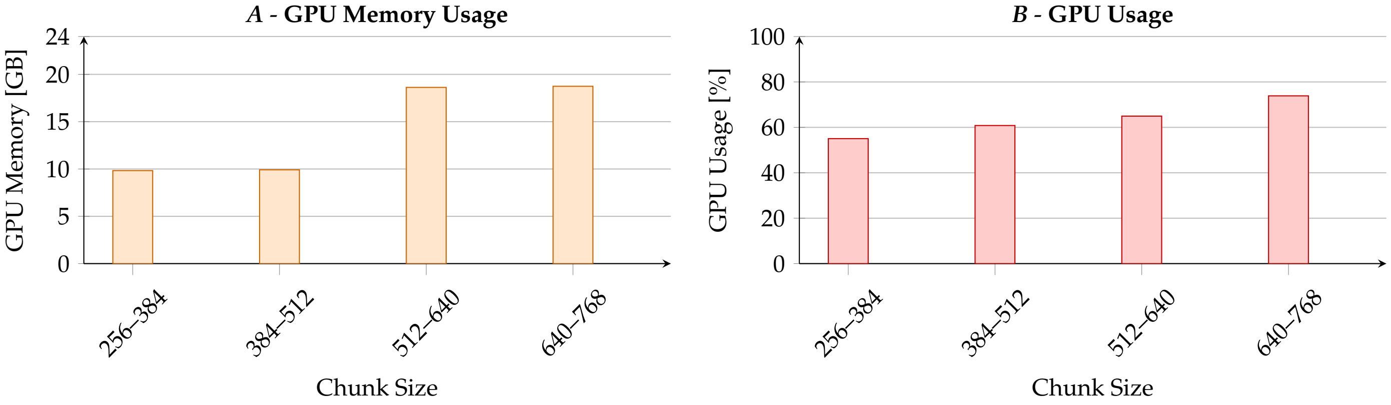


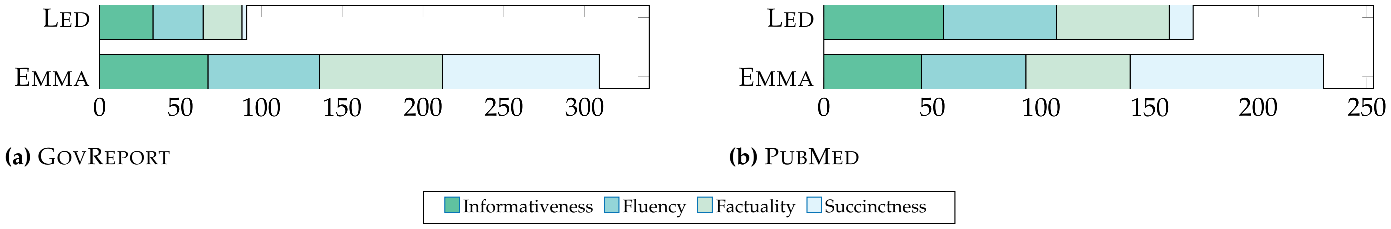
Figure 4. Graphs on how the chunk size hyperparameter of EMMA impacts the GPU. (**A**) GPU memory usage (0–24 GB); (**B**) GPU computational power usage (0–100%).

5.6. Human Evaluation

We conducted a comprehensive human evaluation study to better gauge the quality of the summaries produced by EMMA regarding LED (the main full-training competitor according to Table 2). We randomly selected 50 document–summary pairs from the test sets of GOVREPORT and PUBMED (25 from each source). We asked three evaluators who were proficient in English with legal and medical competencies to select their most and least preferred predictions according to informativeness, fluency, factuality, and succinctness, i.e., best–worst scaling [47,48]. We randomized the order of summaries within pairs to guard the rating against being gamed. Our setup with human instructions is illustrated in Figure A1. The annotation process took approximately 6 h per judge, 18 h in total. The average Kendall coefficient among all evaluators' inter-rater agreement was 0.60. All evaluation files were publicly released for transparency and reproducibility: https://github.com/disi-unibo-nlp/emma (accessed on 13 March 2023). Results are outlined in Table 7, showing the overall percentage of times that a particular system was the most preferred summary source. Additionally, we plot the distribution of dimension-specific votes in Figure 5. Across both quality dimensions and datasets, we observed a clear preference for EMMA. LED tended to be less abstractive and to have more extended outputs, often cut before reaching the `end-of-sentence` token, focused on the first part of the document. Instead, EMMA was much more concise, going straight to the point and covering all the relevant content mentioned in the document with high frequency and factuality. The overall advantage of our solution is strongly accentuated as the length of the target summary increased (GOVREPORT summaries were, on average, 2.58× longer than those of PUBMED).

Table 7. Percentage of times that a summarizer was selected as the best from all evaluators. Annotators preferred EMMA outputs over LED for approximately 70% of the sampled document–summary pairs. The best results are on a green background.

Model	GOVREPORT	PUBMED	OVERALL
LED	22.67	42.33	32.50
EMMA	77.33	57.67	67.50



(a) GOVREPORT

(b) PUBMED

Figure 5. Distribution of votes per quality dimension (cumulative best-selection percentages).

6. Conclusions

Although augmenting transformers with memory is receiving less attention and effort than efficient transformers, it can play a pivotal role in low-resource settings and domains with extremely long documents. In this work, we presented EMMA, the first memory-enhanced encoder–decoder transformer for long-document summarization. The proposed architecture leverages two fundamental elements: (i) a segmentation algorithm for splitting the input document into chunks and pairing them with the most related parts of the target summary, and (ii) a recursive memory module capable of storing information from past chunks. We tested our solution with multiple datasets of different domains, obtaining competitive results with state-of-the-art models under full-training conditions and outperforming prior works in few-shot learning. Exceptionally, depending only on the chunk size, the GPU need remained constant regardless of the whole document length. Compared to segmentation-only techniques, our memory component boosted summarization quality, avoiding treating each chunk independently, and better exploiting their semantic linkage. We also verified that the chunk size could be kept low without significant drops in summarization results, enabling SOTA performance on limited hardware. In-depth ablation studies support our architectural design choices. This study promotes novel research toward efficient memory-enhanced language models.

7. Limitations and Future Directions

Our document segmentation algorithm requires the length of the golden summary to not be too short; otherwise, paired targets are composed of a few tokens. According to our empirical tests in the ablation studies, enabling backpropagation through chunks led to worse results. Deeper investigations are needed to solve this issue.

Future works should explore memory writing/reading operations with structured information extracted from text, comparing unsupervised techniques for document metadata acquisition (e.g., classes [49,50] and entity relationships [51,52]) with advanced semantic parsing solutions such as event extraction [53,54] and abstract meaning representation, which was recently used for knowledge injection into deep neural networks [55,56]. The community should envisage novel graph representation learning methods [57–60] to densely represent multi-relational structured data following a Linked Open Data vision centered on the integration of several source knowledge graphs or relational databases via automatic entity matching [61]. Taking inspiration from biology [62,63] and communication networks [64–67], we underline the importance of managing dynamic scenarios, tracking knowledge refinements among sentences, and propagating information, which is pivotal when processing lengthy inputs. Segmentation strategies and memory-enhanced encoder–decoder transformers could be inspected in other downstream tasks with long documents and cross-dependencies among chunks, such as claim verification with evidence retrieval [68,69].

Author Contributions: Conceptualization, L.R., L.V. and G.M. (Gianluca Moro); methodology, L.R., L.V. and G.M. (Gianluca Moro); software, L.V.; validation, L.R., L.V. and G.F.; formal analysis, L.R., L.V. and G.F.; investigation, L.V.; resources, G.M. (Gianluca Moro); data curation, L.R. and L.V.; writing—original draft preparation, L.R., L.V., G.F. and G.M. (Gianluca Moro); writing—review and editing, L.R., L.V., G.F., G.M. (Gianluca Moro), C.S. and G.M. (Gustavo Marfia); visualization, L.R., L.V. and G.F.; supervision, G.M. (Gianluca Moro); project administration, G.M. (Gianluca Moro), C.S. and G.M. (Gustavo Marfia). All authors have read and agreed to the published version of the manuscript.

Funding: This research is partially supported by the project DARE (Digital Lifelong Prevention) initiative—(PNC-I.1 "Iniziative di ricerca per tecnologie e percorsi innovativi in ambito sanitario e assistenziale" D.D. 931, 06/06/2022—PNC0000002, CUP B53C22006450001 funded by the National Plan for NRRP Complementary Investments—Law Decree 6 May 2021, n. 59, converted and modified as to Law n. 101/2021 Research initiatives for technologies and innovative trajectories in the health and care sectors.

Institutional Review Board Statement: Depending on user intentions, the ability of language models to generate human-indistinguishable text can be dangerous, emphasizing the need for legislative regulations. Fake news production, automatic phishing, and sensible data extraction are possible misuses of these models. Nonetheless, the training of EMMA does not involve sensible or dangerous data.

Informed Consent Statement: Not applicable.

Data Availability Statement: All pre-trained models and corpora used in this work are publicly available (see Appendix A).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

LDS	Long document summarization
LSH	Locality-sensitive hashing

Appendix A. References to Models and Datasets

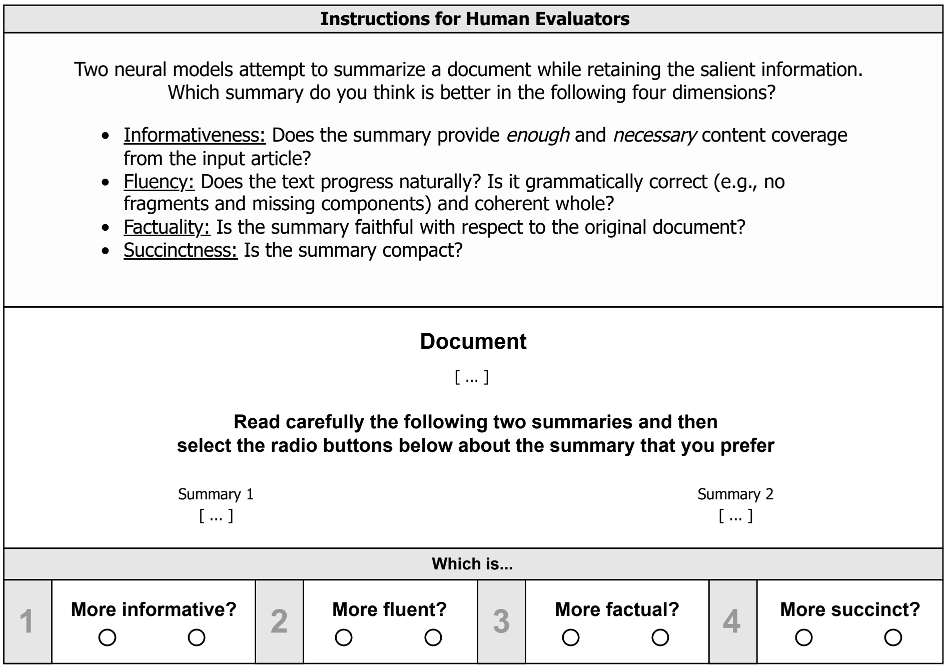
Table A1 enumerates all the trained models and datasets used in this study, linking to specific HuggingFace versions.

Table A1. List of the models and datasets used in this study. All the links have been last accessed on 13 March 2023.

Model	URL
LED-BASE	https://huggingface.co/allenai/led-base-16384
LED-LARGE	https://huggingface.co/allenai/led-large-16384
BART-BASE	https://huggingface.co/facebook/bart-base
BART-LARGE	https://huggingface.co/facebook/bart-large
Dataset	**URL**
GOVREPORT	https://huggingface.co/datasets/ccdv/govreport-summarization
PUBMED	https://huggingface.co/datasets/ccdv/pubmed-summarization
BILLSUM	https://huggingface.co/datasets/billsum

Appendix B. Human Evaluation Insights

The interface with human evaluation instructions is sketched in Figure A1.



Instructions for Human Evaluators

Two neural models attempt to summarize a document while retaining the salient information. Which summary do you think is better in the following four dimensions?

- Informativeness: Does the summary provide *enough* and *necessary* content coverage from the input article?
- Fluency: Does the text progress naturally? Is it grammatically correct (e.g., no fragments and missing components) and coherent whole?
- Factuality: Is the summary faithful with respect to the original document?
- Succinctness: Is the summary compact?

Document

[...]

Read carefully the following two summaries and then select the radio buttons below about the summary that you prefer

Summary 1	Summary 2
[...]	[...]

Which is...

1	**More informative?** ○ ○	2	**More fluent?** ○ ○	3	**More factual?** ○ ○	4	**More succinct?** ○ ○

Figure A1. Human assessment interface.

References

1. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All you Need. In Proceedings of the Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, Long Beach, CA, USA, 4–9 December 2017; Guyon, I., von Luxburg, U., Bengio, S., Wallach, H.M., Fergus, R., Vishwanathan, S.V.N., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2017; pp. 5998–6008.
2. Choromanski, K.M.; Likhosherstov, V.; Dohan, D.; Song, X.; Gane, A.; Sarlós, T.; Hawkins, P.; Davis, J.Q.; Mohiuddin, A.; Kaiser, L.; et al. Rethinking Attention with Performers. In Proceedings of the 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, 3–7 May 2021.
3. Huang, L.; Cao, S.; Parulian, N.; Ji, H.; Wang, L. 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, Online, 6–11 June 2021; Association for Computational Linguistics: Cedarville, OH, USA, 2021; pp. 1419–1436. [CrossRef]
4. Beltagy, I.; Peters, M.E.; Cohan, A. Longformer: The Long-Document Transformer. *arXiv* **2020**, arXiv:2004.05150.
5. Dai, Z.; Yang, Z.; Yang, Y.; Carbonell, J.G.; Le, Q.V.; Salakhutdinov, R. Transformer-XL: Attentive Language Models beyond a Fixed-Length Context. In Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, 28 July–2 August 2019; Korhonen, A., Traum, D.R., Màrquez, L., Eds.; Volume 1: Long Papers; Association for Computational Linguistics: Cedarville, OH, USA, 2019; pp. 2978–2988. [CrossRef]
6. Rae, J.W.; Potapenko, A.; Jayakumar, S.M.; Hillier, C.; Lillicrap, T.P. Compressive Transformers for Long-Range Sequence Modelling. In Proceedings of the 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, 26–30 April 2020.
7. Floridi, L.; Chiriatti, M. GPT-3: Its Nature, Scope, Limits, and Consequences. *Minds Mach.* **2020**, *30*, 681–694. [CrossRef]
8. Devlin, J.; Chang, M.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, 2–7 June 2019; Burstein, J., Doran, C., Solorio, T., Eds.; Volume 1 (Long and Short Papers); Association for Computational Linguistics: Cedarville, OH, USA, 2019; pp. 4171–4186. [CrossRef]
9. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv* **2019**, arXiv:1907.11692.
10. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.* **2020**, *21*, 140:1–140:67.

11. Zaheer, M.; Guruganesh, G.; Dubey, K.A.; Ainslie, J.; Alberti, C.; Ontañón, S.; Pham, P.; Ravula, A.; Wang, Q.; Yang, L.; et al. Big Bird: Transformers for Longer Sequences. In Proceedings of the Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, Virtual, 6–12 December 2020.
12. Xiong, Y.; Zeng, Z.; Chakraborty, R.; Tan, M.; Fung, G.; Li, Y.; Singh, V. Nyströmformer: A Nystöm-based Algorithm for Approximating Self-Attention. In Proceedings of the AAAI Conference on Artificial Intelligence, Online, 3 February 2021; National Institutes of Health (NIH) Public Access 2021; Volume 16, p. 14138.
13. Goyal, T.; Li, J.J.; Durrett, G. News Summarization and Evaluation in the Era of GPT-3. *arXiv* **2022**, arXiv:2209.12356. [CrossRef] .
14. Graves, A.; Wayne, G.; Danihelka, I. Neural Turing Machines. *arXiv* **2014**, arXiv:1410.5401.
15. Gülçehre, Ç.; Chandar, S.; Cho, K.; Bengio, Y. Dynamic Neural Turing Machine with Continuous and Discrete Addressing Schemes. *Neural Comput.* **2018**, *30*, 857–884. [CrossRef]
16. Graves, A.; Wayne, G.; Reynolds, M.; Harley, T.; Danihelka, I.; Grabska-Barwinska, A.; Colmenarejo, S.G.; Grefenstette, E.; Ramalho, T.; Agapiou, J.P.; et al. Hybrid computing using a neural network with dynamic external memory. *Nature* **2016**, *538*, 471–476. [CrossRef] [PubMed]
17. Moro, G.; Pagliarani, A.; Pasolini, R.; Sartori, C. Cross-domain & In-domain Sentiment Analysis with Memory-based Deep Neural Networks. In Proceedings of the IC3K 2018, Seville, Spain, 18–20 September 2018; SciTePress: Setúbal, Portugal, 2018; Volume 1, pp. 127–138. [CrossRef]
18. Ding, S.; Shang, J.; Wang, S.; Sun, Y.; Tian, H.; Wu, H.; Wang, H. ERNIE-Doc: A Retrospective Long-Document Modeling Transformer. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, Virtual Event, 1–6 August 2021; Zong, C., Xia, F., Li, W., Navigli, R., Eds.; Volume 1: Long Papers; Association for Computational Linguistics: Cedarville, OH, USA, 2021; Volume 1, pp. 2914–2927. [CrossRef]
19. Martins, P.H.; Marinho, Z.; Martins, A.F.T. ∞-former: Infinite Memory Transformer. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, 22–27 May 2022; Muresan, S., Nakov, P., Villavicencio, A., Eds.; Association for Computational Linguistics: Cedarville, OH, USA, 2022; pp. 5468–5485.
20. Martins, A.F.T.; Farinhas, A.; Treviso, M.V.; Niculae, V.; Aguiar, P.M.Q.; Figueiredo, M.A.T. Sparse and Continuous Attention Mechanisms. In Proceedings of the Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, Virtual, 6–12 December 2020.
21. Borgeaud, S.; Mensch, A.; Hoffmann, J.; Cai, T.; Rutherford, E.; Millican, K.; van den Driessche, G.; Lespiau, J.; Damoc, B.; Clark, A.; et al. Improving Language Models by Retrieving from Trillions of Tokens. In Proceedings of the International Conference on Machine Learning, ICML 2022, Baltimore, MA, USA, 17–23 July 2022; Chaudhuri, K., Jegelka, S., Song, L., Szepesvári, C., Niu, G., Sabato, S., Eds.; Proceedings of Machine Learning Research 2022; Volume 162, pp. 2206–2240.
22. Frisoni, G.; Mizutani, M.; Moro, G.; Valgimigli, L. BioReader: A Retrieval-Enhanced Text-to-Text Transformer for Biomedical Literature. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Abu Dhabi, United Arab Emirates, 7–11 December 2022; Association for Computational Linguistics: Cedarville, OH, USA, 2022; pp. 770–5793.
23. Rohde, T.; Wu, X.; Liu, Y. Hierarchical Learning for Generation with Long Source Sequences. *arXiv* **2021**, arXiv:2104.07545.
24. Zhang, Y.; Ni, A.; Mao, Z.; Wu, C.H.; Zhu, C.; Deb, B.; Awadallah, A.H.; Radev, D.R.; Zhang, R. Summˆ N: A Multi-Stage Summarization Framework for Long Input Dialogues and Documents. *arXiv* **2021**, arXiv:2110.10150.
25. Wu, J.; Ouyang, L.; Ziegler, D.M.; Stiennon, N.; Lowe, R.; Leike, J.; Christiano, P.F. Recursively Summarizing Books with Human Feedback. *arXiv* **2021**, arXiv:2109.10862.
26. Moro, G.; Ragazzi, L. Semantic Self-Segmentation for Abstractive Summarization of Long Documents in Low-Resource Regimes. In Proceedings of the Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Virtual Event, 22 February–1 March 2022; Association for the Advancement of Artificial Intelligence Press: Palo Alto, CA, USA, 2022; pp. 11085–11093.
27. Ivgi, M.; Shaham, U.; Berant, J. Efficient Long-Text Understanding with Short-Text Models. *arXiv* **2022**, arXiv:2208.00748. [CrossRef]
28. Liu, Y.; Ni, A.; Nan, L.; Deb, B.; Zhu, C.; Awadallah, A.H.; Radev, D.R. Leveraging Locality in Abstractive Text Summarization. *arXiv* **2022**, arXiv:2205.12476. [CrossRef]
29. Bajaj, A.; Dangati, P.; Krishna, K.; Ashok Kumar, P.; Uppaal, R.; Windsor, B.; Brenner, E.; Dotterrer, D.; Das, R.; McCallum, A. Long Document Summarization in a Low Resource Setting using Pretrained Language Models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Student Research Workshop*; Association for Computational Linguistics: Cedarville, OH, USA, 2021; pp. 71–80. [CrossRef]
30. Mao, Z.; Wu, C.H.; Ni, A.; Zhang, Y.; Zhang, R.; Yu, T.; Deb, B.; Zhu, C.; Awadallah, A.H.; Radev, D.R. DYLE: Dynamic Latent Extraction for Abstractive Long-Input Summarization. *arXiv* **2021**, arXiv:2110.08168.
31. Moro, G.; Ragazzi, L.; Valgimigli, L.; Freddi, D. Discriminative Marginalized Probabilistic Neural Method for Multi-Document Summarization of Medical Literature. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, 22–27 May 2022; Muresan, S., Nakov, P., Villavicencio, A., Eds.; Association for Computational Linguistics: Cedarville, OH, USA, 2022; pp. 180–189.
32. Tay, Y.; Dehghani, M.; Bahri, D.; Metzler, D. Efficient Transformers: A Survey. *ACM Comput. Surv.* **2023**, *55*, 109:1–109:28. [CrossRef]

33. Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; Zettlemoyer, L. 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, ACL 2020, Online, 5–10 July 2020; Jurafsky, D., Chai, J., Schluter, N., Tetreault, J.R., Eds.; Association for Computational Linguistics: Cedarville, OH, USA, 2020; pp. 7871–7880. [CrossRef]
34. Zhang, J.; Zhao, Y.; Saleh, M.; Liu, P.J. PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization. In Proceedings of the 37th International Conference on Machine Learning, ICML 2020, Virtual Event, 13–18 July 2020; Proceedings of Machine Learning Research 2020; Volume 119, pp. 11328–11339.
35. Cohan, A.; Dernoncourt, F.; Kim, D.S.; Bui, T.; Kim, S.; Chang, W.; Goharian, N. A Discourse-Aware Attention Model for Abstractive Summarization of Long Documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*; Association for Computational Linguistics: New Orleans, LA, USA, 2018; pp. 615–621. [CrossRef]
36. Kornilova, A.; Eidelman, V. BillSum: A Corpus for Automatic Summarization of US Legislation. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*; Association for Computational Linguistics: Hong Kong, China, 2019; pp. 48–56. [CrossRef]
37. Chen, Y.; Shuai, H. Meta-Transfer Learning for Low-Resource Abstractive Summarization. In Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, 2–9 February 2021; Association for the Advancement of Artificial Intelligence Press: Palo Alto, CA, USA, 2021; pp. 12692–12700.
38. Lin, C.Y. ROUGE: A Package for Automatic Evaluation of Summaries. In *Proceedings of the Text Summarization Branches Out*; Association for Computational Linguistics: Barcelona, Spain, 2004; pp. 74–81.
39. Moro, G.; Ragazzi, L.; Valgimigli, L. Carburacy: Summarization Models Tuning and Comparison in Eco-Sustainable Regimes with a Novel Carbon-Aware Accuracy. In Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Washington, DC, USA, 7–14 February 2023; Association for the Advancement of Artificial Intelligence Press: Palo Alto, CA, USA, 2023; pp. 1–9.
40. Frisoni, G.; Carbonaro, A.; Moro, G.; Zammarchi, A.; Avagnano, M. NLG-Metricverse: An End-to-End Library for Evaluating Natural Language Generation. In *Proceedings of the 29th International Conference on Computational Linguistics*; International Committee on Computational Linguistics: Gyeongju, Republic of Korea, 2022; pp. 3465–3479.
41. Zhang, Y.; Ni, A.; Mao, Z.; Wu, C.H.; Zhu, C.; Deb, B.; Awadallah, A.; Radev, D.; Zhang, R. Summ^N: A Multi-Stage Summarization Framework for Long Input Dialogues and Documents. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Dublin, Ireland, 22–27 May 2022; Association for Computational Linguistics: Dublin, Ireland, 2022; pp. 1592–1604. [CrossRef]
42. Moro, G.; Valgimigli, L. Efficient Self-Supervised Metric Information Retrieval: A Bibliography Based Method Applied to COVID Literature. *Sensors* **2021**, *21*, 6430. [CrossRef]
43. Moro, G.; Valgimigli, L.; Rossi, A.; Casadei, C.; Montefiori, A. Self-supervised Information Retrieval Trained from Self-generated Sets of Queries and Relevant Documents. In Proceedings of the Similarity Search and Applications—15th International Conference, SISAP 2022, Bologna, Italy, 5–7 October 2022; Skopal, T., Falchi, F., Lokoc, J., Sapino, M.L., Bartolini, I., Patella, M., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2022; Volume 13590, pp. 283–290. [CrossRef]
44. Moro, G.; Salvatori, S. Deep Vision-Language Model for Efficient Multi-modal Similarity Search in Fashion Retrieval. In Proceedings of the SISAP 2022, Bologna, Italy, 5–7 October 2022; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2022; Volume 13590, pp. 40–53. [CrossRef]
45. Meng, Z.; Liu, F.; Shareghi, E.; Su, Y.; Collins, C.; Collier, N. Rewire-then-Probe: A Contrastive Recipe for Probing Biomedical Knowledge of Pre-trained Language Models. In Proceedings of the ACL (1), Association for Computational Linguistics, Dublin, Ireland, 22–27 May 2022; pp. 4798–4810.
46. Rae, J.W.; Razavi, A. Do Transformers Need Deep Long-Range Memory? In Proceedings of the ACL, Association for Computational Linguistics, Online, 5–10 July 2020; pp. 7524–7529.
47. Louviere, J.J.; Woodworth, G.G. Best-worst scaling: A model for the largest difference judgments. In *Technical Report*; Working paper; University of Alberta: Edmonton, AB, Canada, 1991.
48. Louviere, J.J.; Flynn, T.N.; Marley, A.A.J. *Best-Worst Scaling: Theory, Methods and Applications*; Cambridge University Press: Cambridge, UK, 2015.
49. Domeniconi, G.; Moro, G.; Pagliarani, A.; Pasolini, R. Markov Chain based Method for In-Domain and Cross-Domain Sentiment Classification. In Proceedings of the KDIR, Lisbon, Portugal, 12–14 November 2015; SciTePress: Setúbal, Portugal, 2015; pp. 127–137.
50. Domeniconi, G.; Moro, G.; Pagliarani, A.; Pasolini, R. On Deep Learning in Cross-Domain Sentiment Classification. In Proceedings of the 9th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management—(Volume 1), Funchal, Portugal, 1–3 November 2017; Fred, A.L.N., Filipe, J., Eds.; SciTePress: Setúbal, Portugal, 2017; pp. 50–60. [CrossRef]
51. Frisoni, G.; Moro, G.; Carbonaro, A. Learning Interpretable and Statistically Significant Knowledge from Unlabeled Corpora of Social Text Messages: A Novel Methodology of Descriptive Text Mining. In Proceedings of the 9th International Conference on Data Science, Technology and Applications (DATA 2020), Online, 7–9 July 2020; SciTePress: Setúbal, Portugal, 2020; pp. 121–134.

52. Frisoni, G.; Moro, G. Phenomena Explanation from Text: Unsupervised Learning of Interpretable and Statistically Significant Knowledge. In Proceedings of the 9th International Conference on Data Science, Technology and Applications (DATA 2020), Online, 7–9 July 2020; Revised Selected Papers; Volume 1446, pp. 293–318. [CrossRef]
53. Frisoni, G.; Moro, G.; Carbonaro, A. A Survey on Event Extraction for Natural Language Understanding: Riding the Biomedical Literature Wave. *IEEE Access* **2021**, *9*, 160721–160757. [CrossRef]
54. Frisoni, G.; Moro, G.; Balzani, L. Text-to-Text Extraction and Verbalization of Biomedical Event Graphs. In *Proceedings of the 29th International Conference on Computational Linguistics*; International Committee on Computational Linguistics: Gyeongju, Republic of Korea, 2022; pp. 2692–2710.
55. Frisoni, G.; Italiani, P.; Salvatori, S.; Moro, G. Cogito Ergo Summ: Abstractive Summarization of Biomedical Papers via Semantic Parsing Graphs and Consistency Rewards. In Proceedings of the AAAI, Washington, DC, USA, 7–14 February 2023; pp. 1–9.
56. Frisoni, G.; Italiani, P.; Boschi, F.; Moro, G. Enhancing Biomedical Scientific Reviews Summarization with Graph—Based Factual Evidence Extracted from Papers. In Proceedings of the 11th International Conference on Data Science, Technology and Applications, DATA 2022, Lisbon, Portugal, 11–13 July 2022; pp. 168–179. [CrossRef]
57. Ferrari, I.; Frisoni, G.; Italiani, P.; Moro, G.; Sartori, C. Comprehensive Analysis of Knowledge Graph Embedding Techniques Benchmarked on Link Prediction. *Electronics* **2022**, *11*, 3866. [CrossRef]
58. Cao, J.; Fang, J.; Meng, Z.; Liang, S. Knowledge Graph Embedding: A Survey from the Perspective of Representation Spaces. *arXiv* **2022**, arXiv:2211.03536.
59. Frisoni, G.; Moro, G.; Carlassare, G.; Carbonaro, A. Unsupervised Event Graph Representation and Similarity Learning on Biomedical Literature. *Sensors* **2022**, *22*, 3. [CrossRef]
60. Chen, G.; Fang, J.; Meng, Z.; Zhang, Q.; Liang, S. Multi-Relational Graph Representation Learning with Bayesian Gaussian Process Network. In Proceedings of the AAAI, Virtual Event, 22 February–1 March 2022; pp. 5530–5538.
61. Singh, R.; Meduri, V.V.; Elmagarmid, A.K.; Madden, S.; Papotti, P.; Quiané-Ruiz, J.; Solar-Lezama, A.; Tang, N. Generating Concise Entity Matching Rules. In Proceedings of the SIGMOD Conference, Chicago, IL USA, 14–19 May 2017; pp. 1635–1638.
62. Domeniconi, G.; Masseroli, M.; Moro, G.; Pinoli, P. Cross-organism learning method to discover new gene functionalities. *Comput. Methods Programs Biomed.* **2016**, *126*, 20–34. [CrossRef] [PubMed]
63. Moro, G.; Masseroli, M. Gene function finding through cross-organism ensemble learning. *BioData Min.* **2021**, *14*, 14. [CrossRef] [PubMed]
64. Monti, G.; Moro, G. Multidimensional Range Query and Load Balancing in Wireless Ad Hoc and Sensor Networks. In Proceedings of the IEEE Computer Society Peer-to-Peer Computing, Aachen, Germany, 8–11 September 2008; pp. 205–214.
65. Lodi, S.; Moro, G.; Sartori, C. Distributed data clustering in multi-dimensional peer-to-peer networks. In Proceedings of the Database Technologies 2010, Twenty-First Australasian Database Conference (ADC 2010), Brisbane, Australia, 18–22 January 2010; Volume 104, pp. 171–178.
66. Moro, G.; Monti, G. W-Grid: A scalable and efficient self-organizing infrastructure for multi-dimensional data management, querying and routing in wireless data-centric sensor networks. *J. Netw. Comput. Appl.* **2012**, *35*, 1218–1234. [CrossRef]
67. Cerroni, W.; Moro, G.; Pirini, T.; Ramilli, M. Peer-to-Peer Data Mining Classifiers for Decentralized Detection of Network Attacks. In Proceedings of the Australasian Database Conference, Adelaide, Australia, 29 January–1 February 2013; Volume 137, pp. 101–108.
68. Kryscinski, W.; McCann, B.; Xiong, C.; Socher, R. Evaluating the Factual Consistency of Abstractive Text Summarization. In Proceedings of the EMNLP (1), Association for Computational Linguistics, Online Event, 16–20 November 2020; pp. 9332–9346.
69. Saeed, M.; Traub, N.; Nicolas, M.; Demartini, G.; Papotti, P. Crowdsourced Fact-Checking at Twitter: How Does the Crowd Compare With Experts? In Proceedings of the CIKM, Atlanta, GA, USA, 17–21 October 2022; pp. 1736–1746.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.