Scaling Up Summarization: Leveraging Large Language Models for Long Text Extractive Summarization

Anonymous ACL submission

Abstract

In an era where digital text is proliferating at an unprecedented rate, efficient summarization tools are becoming indispensable. While Large Language Models (LLMs) have been successfully applied in various NLP tasks, their role in extractive text summarization remains underexplored. This paper introduces EYEGLAXS (Easy Yet Efficient larGe LAnguage model for eXtractive Summarization), a framework that leverages LLMs, specifically LLAMA2-7B and ChatGLM2-6B, for extractive summarization of lengthy text documents. Instead of abstractive methods, which often suffer from issues like factual inaccuracies and hallucinations, EYEGLAXS focuses on extractive summarization to ensure factual and grammatical integrity. Utilizing state-of-the-art techniques such as Flash Attention and Parameter-Efficient Fine-Tuning (PEFT), EYEGLAXS addresses the computational and resource challenges typically associated with LLMs. The system sets new performance benchmarks on well-known datasets like PubMed and ArXiv. Furthermore, we extend our research through additional analyses that explore the adaptability of LLMs in handling different sequence lengths and their efficiency in training on smaller datasets. These contributions not only set a new standard in the field but also open up promising avenues for future research in extractive text summarization.

1 Introduction

In the era of information overload, text summarization has emerged as a critical tool for distilling essential information from expansive text documents. This paper focuses on automatic text summarization, which can be broadly categorized into two paradigms: abstractive and extractive methods. Abstractive methods, despite their ability to generate flexible and creative summaries, often grapple with issues of grammatical inaccuracy and factual inconsistencies, commonly referred to as "hallucinations" (Bishop et al., 2022; Ji et al., 2023; Zhang et al., 2023b). These challenges are exacerbated when summarizing long texts and can be particularly detrimental in critical applications such as healthcare, scientific research, and legislation. In contrast, extractive summarization offers a more reliable approach by selecting pertinent sentences directly from the source text, thereby ensuring grammatical and factual integrity. Traditionally, this task has been framed as a sentence classification problem and has predominantly employed encoder-only pre-trained models (Liu and Lapata, 2019; Cho et al., 2022; Bian et al., 2023). Despite the promising capabilities of Large Language Models (LLMs) in various NLP tasks, their potential in extractive summarization remains largely untapped. This oversight is partly due to the computational challenges and fine-tuning limitations associated with these sizable models. However, recent advancements in long sequence processing for decoder-only models offer a glimmer of hope for harnessing LLMs in this context. To bridge this gap, our paper introduces EYEGLAXS (Easy Yet Efficient larGe LAnguage model for eXtractive Summarization), a system that leverages the power of LLMs—specifically LLAMA2-7B(Touvron et al., 2023) and ChatGLM2-6B (Zeng et al., 2022). We employ Flash Attention 2 (Dao, 2023) and Parameter-Efficient Fine-Tuning (PEFT)(Lialin et al., 2023) techniques to mitigate some of the challenges associated with using LLMs. Our contributions are manifold: we not only propose a novel method for employing LLMs in long extractive summarization tasks but also demonstrate their competitive performance against state-of-the-art methods. We further explore the adaptability of LLMs in handling varying sequence lengths and investigate their training efficiency on smaller datasets. Lastly, we delve into the issue of position bias inherent in LLMs.

2 Litterature Review

2.1 Long Extractive Text Summarization

In the literature, the task of extractive text summarization is predominantly approached as a sentence classification problem. In this framework, models are trained to predict a label for each sentence in the input document to determine whether or not the sentence should be included in the generated summary. Most state-of-the-art methods leverage pre-trained transformer models that are adapted for natural language understanding tasks. One of the pioneering works in this area slightly modified BERT's architecture by incorporating a priori information on sentence splitting and adding layers of inter-sentence transformers before feeding them into the classifier for prediction (Liu and Lapata, 2019). To address the issue of limited context size, various transformer architectures have been proposed to mitigate the quadratic complexity problem associated with self-attention computation. For instance, Longformer (Beltagy et al., 2020) and Bigbird (Zaheer et al., 2020) employ attention sparse methods such as sliding windows to handle longer sequences. Building on these architectures, many systems introduce additional mechanisms that exploit the unique characteristics of documents to improve performance on long sequences. For example, the work by (Xiao and Carenini, 2019) generates different representations that consider both local and global contexts. The system described in (Ruan et al., 2022) explicitly incorporates hierarchical information by using section titles and the hierarchical position of sentences to enrich representations. Similarly, (Bian et al., 2023) represents the hierarchical structure of the text through a heterogeneous graph of sentences and sections, while integrating reinforcement learning with a graph neural network. The approach by (Cho et al., 2022) aims to discover the latent structure of the document by jointly optimizing a secondary task of section segmentation alongside sentence extraction. Moreover, (Xie et al., 2022) incorporates domain-specific knowledge into the model by using adapters to infuse medical expertise. The abstractive-extractive approach has also been explored. For instance, (Bishop et al., 2022) generates an abstractive summary that later guides the extraction of salient sentences, irrespective of document length. Most of these approaches employ pre-trained RoBERTa (Liu et al., 2019) as the backbone model. However, this comes with limitations, such as the complexity of learning new positional encoding tokens and the relatively small number of parameters and tokens encountered during the pre-training stage, especially when compared to larger language models.

2.2 Large Language Models

Over the past few years, pre-trained Large Language Models (LLMs) have transitioned from being virtually unknown to becoming pervasive in the realm of machine learning. Their widespread adoption is largely due to their proven effectiveness in addressing zero-shot and few-shot learning challenges. They have been successfully applied to tasks such as abstractive summarization and translation of short documents. While these models excel in generative tasks, their application to extractive tasks poses greater challenges. A common workaround is to transform an extractive task into a text generation task by utilizing cloze prompting templates (Liu et al., 2021). While this technique is well-suited for simpler tasks like sentiment detection, its complexity escalates for tasks with more intricate scoring systems, such as named entity recognition, and becomes nearly unfeasible for tasks like extractive summarization (Wang et al., 2023). Additionally, these models are prone to hallucination issues, limiting their applicability in critical fields like healthcare (Kaddour et al., 2023). Although zero-shot approaches to extractive summarization have been explored (Zhang et al., 2023a), to the best of our knowledge, no attempts have been made to evaluate the fine-tuning of these models specifically for extractive summarization. Furthermore, it is plausible that the representations learned by these LLMs are richer than those learned by encoder models, owing to their larger number of parameters and more extensive pretraining data (Ni et al., 2022). Finally, efforts have been made to significantly increase the context size of these LLMs, notably through the use of scalable positional encodings (Su et al., 2021), extending the initial pretraining context length (Chen et al., 2023), or optimizing attention computation at the GPU memory level (Dao et al., 2022; Dao, 2023).

2.3 Parameter-Efficient Fine-Tuning methods

Large Language Models (LLMs) are increasingly being utilized to achieve state-of-the-art performance in various NLP tasks, capturing the attention of both researchers and industry professionals (Bubeck et al., 2023). However, these models come

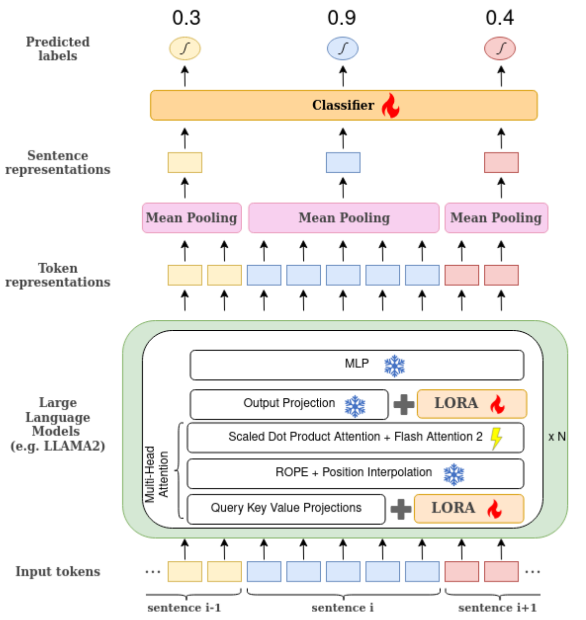

Figure 1: The overall framework of EYEGLAXS. Residual connections and normalizations do not appear for better readability. Snowflake logo means that weights are frozen, while Fire logo means that weights are trainable.

with significant computational and memory requirements for training from scratch. Recent iterations of these models often boast more than 70 billion parameters (Touvron et al., 2023; Zeng et al., 2022), making the fine-tuning process highly resource-intensive. To address this challenge, a new family of techniques known as Parameter-Efficient Fine-Tuning (PEFT)(Lialin et al., 2023) has been introduced. These methods advocate for the training of a relatively small number of additional parameters, which, in comparison to the overall size of the models, represent only a fraction. This approach substantially reduces both storage and computational costs. Among the PEFT techniques, prompt tuning, prefix tuning (Liu et al., 2022), and LoRA (Hu et al., 2022) have garnered significant attention. Models trained using PEFT methods have demonstrated performance levels comparable to those achieved through full fine-tuning (Lialin et al., 2023). In this article, we propose to fine-tune pre-trained LLMs using LoRA for the specific task of extractive summarization.

3 Method

3.1 Task Definition

We define extractive summarization as a sentence classification problem. Let note $D = \{s_1, s_2, \ldots, s_n\}$ the document D consisting of n sentences. The extractive summarization task aims to predict labels $\hat{y}_i \in (0,1)$ for each $i \in [0,n]$ where $\hat{y}_i = 1$ and $\hat{y}_i = 0$ means the sentence should be included , or not, in the summary respectively. As datasets only contain document-abstract pairs, in order to train these models in a supervised manner, reference extractive abstracts are generated using methods such as greedy algorithms with the objective to maximize a chosen metric, mostly ROUGE scores (Kedzie et al., 2018; Lin, 2004). This allows associating a label to each sentence of the original document based on their inclusion in these oracle summaries. These oracle summaries represent an upper bound for the expected performances of extractive models. Thus a label $y_i \in (0,1)$, is associated to each sentence of the text depending on its inclusion in the oracle.

We propose a system called EYEGLAXS (Easy Yet Efficient larGe LAnguage model for eXtractive Summarization), a system based on Large Language Models for the long text extractive summarization task described in Figure 1.

3.2 Choice of Large Language Models

Large Language Models (LLMs) are notably resource-intensive to train from scratch, making it common practice to leverage pre-existing architectures for new applications. To improve both reproducibility and evaluation, we employ fine-tuning techniques on models that have publicly available checkpoints. A crucial aspect of our selection process is the choice of models that feature extendable positional encoding mechanisms (Su et al., 2021; Press et al., 2021). We also ensure that the selected positional encoding can be efficiently parallelized to benefit from Flash Attention 2, which offers efficient GPU memory consumption (Dao et al., 2022; Dao, 2023). As a result, models relying on relative positional encoding—such as those based on the T5 architecture—are not suitable for our use case. Likewise, models with learned positional encoding, like XLM (Goyal et al., 2021), pose scalability challenges unless new positional encodings are retrained. Guided by these considerations, we have narrowed our evaluation to two distinct LLMs: first, LLAMA2 (Touvron et al., 2023), a decoder-only model that has gained widespread adoption; and second, ChatGLM2 (Zeng et al., 2022), a prefix decoder model that shows promise for superior information integration through the use of bidirectional attention mechanisms. More specifically, we assess the performance of the pre-trained long-sequence instruct-based models LLAMA2-7B-32K-Instruct[1] and ChatGLM2-6B-32K[2], both available on HuggingFace. The former has been fine-tuned for long-context summarization using the BookSum dataset, while the latter excels in conversational contexts. During our experiments, the focus was primarily on evaluating the feasibility of fine-tuning these models using LoRA. We did not place particular emphasis on the instructions used during training. Instead, we adhered to minimalist prompts, which were formed by concatenating the sentences of the input documents. This approach respected the format used by each model and did not include any additional instructions.

3.3 Transformer

Large Language Models (LLMs) are commonly built using transformer architectures. Architectures such as LLAMA2 (Touvron et al., 2023) and GLM (Zeng et al., 2022) consist of an initial embedding input layer followed by multiple decoder layers. Each of these decoder layers comprises Query-Key-Value (QKV) Projection Layers, Rotary Position Encoding, a self-attention module, an output projection, a multi-layer perceptron, residual connections, and normalization. For the sake of readability, we focus solely on our modifications to the original architecture in the areas of QKV Projection Layers, Rotary Position Encoding, self-attention module, and output projection.

3.4 Query Key Value Projection Layers with LoRA

This module transforms the input token x_m with position m into a trio of queries, keys, and values—represented as $\{q_m, k_m, v_m\}$ —through linear projection layers with corresponding weight matrices $\{Wq, Wk, Wv\}$. These matrices are later used to compute attention values. We apply LoRA on these specific matrices. More precisely, for a pre-trained matrix $W \in \mathbb{R}^{d \cdot p}$, we add low rank adapter δW such as :

$$\begin{aligned} q_m &= (W_q + \delta W_q)x_m = (W + B_q A_q)x_m \\ k_m &= (W_k + \delta W_k)x_m = (W + B_k A_k)x_m \\ v_m &= (W_v + \delta W_v)x_m = (W + B_v A_v)x_m \end{aligned} \quad (1)$$

where $B_{\{q,k,v\}} \in \mathbb{R}^{d \times r}$, $A_{\{q,k,v\}} \in \mathbb{R}^{r \times p}$, d is the dimension of x_m and with $r \ll min(d,p)$. The idea is that the weight updates in pre-trained models have a low intrinsic rank during adaptation. Thus, during training, $W_{\{q,k,v\}}$ weights are frozen and the number of trainable parameters (i.e. $B_{\{q,k,v\}}$ and $A_{\{q,k,v\}}$) are drastically reduced compared to full fine-tuning setting.

3.5 Rotary Positional Encoding

We employ architectures using a novel positional encoding scheme called Rotary Positional Embeddings (RoPE)(Su et al., 2021). RoPE is a distinctive form of positional embedding used in Transformer models to encode the absolute and relative positional information of tokens within a sequence. Moreover, RoPE is valued for its flexibility to expand to any sequence lengths (Su et al., 2021).

The mathematical intuition behind RoPE aims to devise a positional encoding function $f(x,m)$ for a token x at position m such that for a query vector q_m and a key vector k_l at positions m and l respectively, the inner product between $f(q_m, m)$ and $f(k_l, l)$ is sensitive only to the values of q_m, k_n, and their relative position $(m - l)$.

In practice, new vectors $\tilde{q_m}$ and $\tilde{k_l}$ are computed following this specific equation

$$\begin{aligned} \tilde{q_m} &= R^m_{\theta,d}(q_m) \\ \tilde{k_l} &= R^l_{\theta,d}(k_l) \end{aligned} \quad (2)$$

with

$$R^d_{\Theta,m}(x) = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \\ x_{d-1} \\ x_d \end{pmatrix} \otimes \begin{pmatrix} \cos m\theta_1 \\ \cos m\theta_1 \\ \cos m\theta_2 \\ \cos m\theta_2 \\ \vdots \\ \cos m\theta_{d/2} \\ \cos m\theta_{d/2} \end{pmatrix} + \begin{pmatrix} -x_2 \\ x_1 \\ -x_4 \\ x_3 \\ \vdots \\ -x_{d-1} \\ x_d \end{pmatrix} \otimes \begin{pmatrix} \sin m\theta_1 \\ \sin m\theta_1 \\ \sin m\theta_2 \\ \sin m\theta_2 \\ \vdots \\ \sin m\theta_{d/2} \\ \sin m\theta_{d/2} \end{pmatrix} \quad (3)$$

where m is the indice position and d is the dimension of the x.

While RoPE is in theory expandable to any sequence lengths, we found an exploding perplexity when directly extending a pre-trained model

[1]https://huggingface.co/togethercomputer/Llama-2-7B-32K-Instruct
[2]https://huggingface.co/THUDM/chatglm2-6b-32k

beyond the context length L of the pretraining process(Chen et al., 2023). In order to overcome this problem, we interpolate position indices from longer context length L' (i.e., $[0, L')$) to original pre-trained context length L (i.e., $[0, L)$) in order to match the original range of indices.

Formally, we replace $R^d_{\Theta,m}(x)$ by $R'^d_{\Theta,m}(x)$ function where

$$R'^d_{\Theta,m}(x) = R^d_{\Theta,m\cdot\alpha}(x) \tag{4}$$

with a parameterized scaling factor α defined as below:

$$\alpha = \frac{L}{L'}$$

3.6 Self-Attention Module and Output Projection

Once $\{\tilde{q}, \tilde{k}, v\}$ are computed, we compute the outputs o via a self-attention module as

$$o = softmax(\tilde{q}\tilde{k}^T)v \tag{5}$$

A significant challenge associated with self-attention is its computational burden when handling long sequences. Specifically, both computational and memory requirements increase quadratically with the length of the sequence. To address this limitation, we replace the original attention computation with Flash Attention 2(Dao, 2023), an Input-Output attention algorithm that scales memory consumption linearly and accelerates the training process. It's important to note that the attention computed using Flash Attention 2 is identical to that of the original operation. In practical terms, this allows us to process sequences of up to 12,000 tokens on a single A10 GPU card.

Subsequently, these outputs are projected by a linear layer with a weight matrix W_o. Similarly to the projection matrices $\{W_q, W_k, W_v\}$ We apply LoRA to the output projection matrix W_o.

3.7 Mean Pooling and Classification Layer

Unlike BERT models, LLMs such as LLAMA2 or ChatGLM2 do not use a 'CLS' token like in other models at the beginning of each sentence to get its representation. Nonetheless, we think that some knowledge is still well encoded within the token representations and a mean pooling across all input sentence tokens should provide a natural sentence representation(Ni et al., 2022). Therefore, we apply a mean pooling at the sentence level. More precisely, for each sentence s_i comprised of a list of M_i contextualized words processed by previous decoder layers of the LLM $\{w_{i,1}, w_{i,2} \ldots w_{i,M_i}\}$, we compute $\bar{s_i}$

$$\bar{s_i} = \frac{1}{M_i}\sum_{j=1}^{M_i} w_{i,j} \tag{6}$$

Once, $\bar{s_i}$ obtained, we pass it through a linear classification layer :

$$\hat{y_i} = \sigma(W_c\bar{s_i} + b) \tag{7}$$

where W_c is a weight matrix trainable, b is a biais term trainable and σ is the sigmoid function. The loss function used is the binary cross entropy between $\hat{y_i}$ and the oracle y_i.

4 Experiments

In this section, we present the results of our various experiments demonstrating the performances of our models in different settings compared to strong baselines of the state-of-the-art. We specify the detailed experimental settings of each experiment and share the models and used code on our repository [3].

4.1 Datasets

For our experiments, we evaluate our approach with two sources wildly used in summarization tasks, namely the arXiv and PubMed datasets(Cohan et al., 2018). They consist of rather long scientific papers, with PubMed focusing on the biomedical domain while arXiv includes articles from various scientific fields.

To train the model in a supervised manner for the extractive summarization task, sentence labels are needed. We use the already-computed labels from (Cho et al., 2022)[4]. They followed the methodology of (Kedzie et al., 2018) with the objective of maximizing the average of the R1 and R2 scores. Moreover, we derive from the original dataset two filtered datasets containing only documents shorter that a given sequence length, in order to evaluate EYEGLAXS trained on shorter documents when tested on longer documents. Characteristics of the resulting datasets are shown in the Table 1.

4.2 Experimental settings

We used and modified the implementation released on the TransformerSum[5]. Experiments have been

[3] anonymous
[4] https://github.com/tencent-ailab/Lodoss/tree/main
[5] https://github.com/HHousen/TransformerSum/tree/master

Set	PubMed			Arxiv		
	Length-4K	Length-12K	Length-16K	Length-4K	Length-12K	Length-32K
Train	70893	127192	131233	38532	153802	202648
Validation	3630	6442	6630	1124	5089	6435
Test	3682	6472	6657	1076	5085	6439

Table 1: The datasets we used in the experiments. cell values correspond to the number of documents for each dataset and split. Pubmed-16k and Arxiv-32k have been truncated from original datasets, contrary to the other datasets where longer documents have been filtered out.

carried out on 8 NVIDIA A10G GPUs. Aside from the experiments on low-volume data, models have been trained for 5 epochs, with a validation step occurring every fifth of an epoch. Models were saved based on the smallest validation loss achieved. We use a batch size of 1 with gradient accumulation every 32 steps and the adam8bit optimizer with a 3e-5 learning rate. Gradient-checkpointing and bf16-mixed precision are used. Deepspeed stage 1 is employed. No advanced hyperparameter search was performed. We use sequence lengths of 4k and 12k to train our models before testing them on the full length dataset. Results are obtained without trigram blocking by selecting the 7 and 5 sentences with the highest probability scores for the PubMed and arXiv datasets respectively as it is done in (Cho et al., 2022). The scaling factor α for RoPE is set to 8 to handle up to 32K length context. Rank r of LoRA is set to 8.

Models	R-1	R-2	R-L
Abstractive Models			
Bigbird-large	46.32	20.65	42.33
Long-T5	50.23	24.76	46.67
Extractive Models			
ORACLE	61.49	34.70	55.92
LEAD-10	37.45	14.19	34.07
SumBasic	37.15	11.36	33.43
LexRank	39.19	13.89	34.59
Sent-PTR	45.01	19.91	41.16
GenCompareSum	42.10	16.51	38.25
Histruct+	46.59	20.39	42.11
Lodoss-base (Longformer)	48.10	22.53	43.51
Lodoss-full-LG	49.38	23.89	44.84
GoSum	49.83	23.56	45.10
Our system EYEGLAXS (Extractive)			
CHATGLM2-6B (4K)	49.96	24.04	45.50
CHATGLM2-6B (12K)	50.17	24.41	45.66
LLAMA2-7B (4K)	49.48	23.64	45.08
LLAMA2-7B (12K)	**50.34**	**24.57**	**45.96**

Table 2: ROUGE results on the PubMed dataset

4.3 Evaluation Metrics

We use ROUGE scores to evaluate the model performance(Lin, 2004). More precisely, we report the F1 score of unigram, bigram overlap (ROUGE-1, ROUGE-2) and the longest common subsequence (ROUGE-L). We use the python implementation[6].

4.4 Baseline systems

Using the arXiv and PubMed datasets, which are two popular datasets in the domain of extractive summarization, allows us to easily assess the relevance of our approach. We can directly compare the results obtained with our models against previous systems of the state-of-the-art on the same ROUGE metrics. Among the baselines, we compare our approach with standard lexical methods like Sumbasic (Vanderwende et al., 2007) that is based on word frequencies or LexRank (Erkan and Radev, 2004) that uses a graph-based approach and centrality scoring of sentences. Taking advantage of what language models have to offer and building upon the BERTSUM framework (Liu and Lapata, 2019), we have strong extractive baselines: HiStruct+ (Ruan et al., 2022) explicitly exploits the hierarchical structure of the text, taking advantage of the position of sentences within sections. Among more recent models, Lodoss (Cho et al., 2022) represents a strong baseline that achieved great performances by being jointly trained for text summarization and segmentation in addition to using a novel regularizer to boost diversity among the selected summary sentences. GoSum (Bian et al., 2023) is another state-of-the-art model that showed some of the best results by exploiting graph neural networks and reinforcement learning. Alongside these extractive models that we directly compare ourselves to, we also show the performances of some popular abstractive baseline. Rather than measuring up against them, we simply add them as a reference to put the results into perspective.

5 Results and Analyses

Results on PubMed and arXiv are shown respectively in table 2 and table 3. For both datasets, the EYEGLAXS variants, specifically ChatGLM2-

[6]https://github.com/google-research/google-research/tree/master/rouge

Models	R-1	R-2	R-L
Abstractive Models			
Bigbird-large	46.63	19.02	41.77
Long-T5	48.35	21.92	44.27
Extractive Models			
ORACLE	59.41	30.05	52.34
LEAD-10	35.52	10.33	31.44
SumBasic	29.47	6.95	26.30
LexRank	33.85	10.73	28.99
Sent-PTR	42.32	15.63	38.06
GenCompareSum	39.66	12.30	35.38
Histruct+	45.22	17.67	40.16
Lodoss-base (Longformer)	47.64	19.73	41.71
Lodoss-full-LG	48.45	20.72	42.55
GoSum	48.61	20.53	42.80
Our system EYEGLAXS (Extractive)			
CHATGLM2-6B (4K)	46.87	18.96	41.37
CHATGLM2-6B (12K)	**49.02**	21.01	**43.33**
LLAMA2-7B (4K)	48.68	20.72	42.97
LLAMA2-7B (12K)	48.96	**21.07**	43.30

Table 3: ROUGE results on the arXiv dataset

6B (4K) and ChatGLM2-6B (12K), alongside the LLAMA variants, LLAMA2-7B (4K) and LLAMA2-7B (12K), showcase competitive performance compared to the state-of-the-art even if they are trained on smaller and shorter dataset. Models trained on a longer context (12K variant) exhibit superior performance compared to their counterpart trained on a shorter context. LLAMA2-7B (12K) and ChatGLM2-6B (12K) seem to have similar performances on both datasets. However, ChatGLM2-6B (4K) seems to underperform compared to LLAMA2-7B (4K) on arXiv, we hypothesize that since the arXiv dataset contains longer documents, LLAMA2 benefited more from his pre-training stage on long document compared to ChatGLM2 and could potentially need less amount of training data to converge (see section 5.3). Finally, we obtain new state-of-the-art results compared to other extractive methods on both datasets. We provide also in appendix the complete table results on filtered datasets (e.g. training LLAMA2-7B on the 4K dataset and testing it on the 12K dataset).

5.1 Evaluating LoRA's Impact on Fine-Tuning EYEGLAXS

To assess the contribution of LoRA and determine the relevance of the hidden representations provided by LLMs, we contrasted the performance of EYEGLAXS models trained on the 4K PubMed filtered dataset with a variant of same models with frozen weights, albeit with a trainable classifier head. The comparative outcomes are presented in Table 4. The notable enhancement in

Model	4K PubMed Dataset		
	R1	R2	RL
CHATGLM2-6B (4K) - Frozen	42.79	17.20	38.68
CHATGLM2-6B (4K) - LoRA	49.96	24.04	45.50
LLAMA2-7B (4K) - Frozen	42.38	17.12	38.42
LLAMA2-7B (4K) - LoRA	49.48	23.64	45.08

Table 4: Comparison of ROUGE Scores between Frozen Weights and EYEGLAXS Models on 4K PubMed Dataset

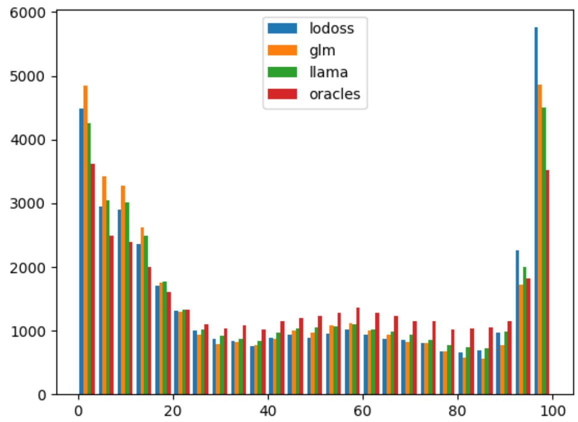

Figure 2: Number of sentences selected at each relative position by the EYEGLAXS models and baselines compared to the oracles

ROUGE scores underscores the necessity of employing Parameter-Efficient Fine-Tuning (PEFT) methodologies like LoRA to fine-tune LLMs, suggesting that the standalone hidden representations from LLMs may fall short of ensuring optimal performance.

5.2 Position Error Analysis

To get a better understanding of the strengths and weaknesses of our system, we further analyze the outputs of our models. This section is used to check the general behavior of the model, and can help verify some reported problems with bias when using LLMs for long sequences, including a natural bias at the beginning and end of each document (Liu et al., 2023). We choose to examine the extracted sentences and compare them to the ones forming the oracles. To achieve this, we first trace out the distribution of the selected sentences for both versions of our model as well as a Longformer baseline (Lodoss-base) (Cho et al., 2022) and the oracles. The lengths of the documents forming the datasets having a wide amplitude, we choose to use the relative positions of sentences to ensure an overall homogeneous comparison. To do so, we compare the absolute index of each extracted

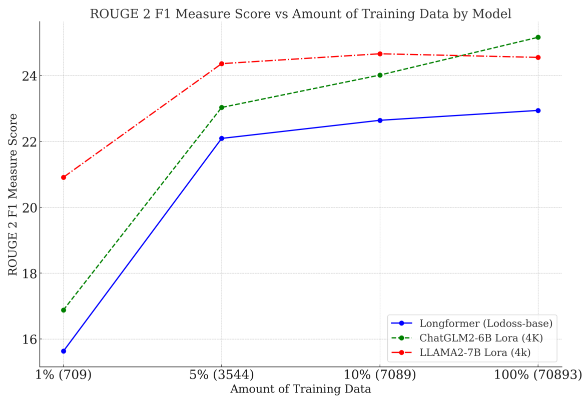

Figure 3: ROUGE-2 F1 Measure scores for Longformer (Lodoss-base), ChatGLM2-LoRA (4K) and LLAMA2-LoRA across varying training data sizes. The exact number of training instances is indicated in parentheses.

sentences against the document's total number of sentences, then plot the resulting histogram. This gives us a histogram showing the relative position of the sentences selected by each model. The results are showed for the PubMed dataset on the test split. From the Figure 2, we can see that the sentences chosen for the oracles or predicted by the models tend to be near the beginning and the end of the document. It is not surprising since both the introduction and the conclusion usually contain sentences that are representative of the document's subject. We can observe that the three models tend to choose sentences near both ends of the text in excess. On the subject of accessing relevant information located in the middle of inputs, even though all the models are lagging behind compared to the oracles, we notice that the EYEGLAXS models follow the oracle trend a little better.

5.3 Training on smaller datasets

In the medical field, obtaining a large database like PubMed is often challenging. Assessing the performance of Large Language Models (LLMs) on smaller databases compared to traditional methods becomes crucial. To this end, we conducted an experiment comparing the performance of ChatGLM2-6B (4K), Longformer (Lodoss-base) (Cho et al., 2022), and LLAMA2-7B LoRA (4k), on the filtered PubMed 4K dataset using varying portions of the training data. Specifically, we examined the performance when utilizing 1%, 5%, 10%, and 100% of the PubMed 4K dataset. The findings are illustrated in Figure 3. The results confirm a positive correlation between the size of the training dataset and the performance metrics for all three models, aligning with the intuitive expectation that larger training sets generally lead to improved model performance. Interestingly, the performance gap among the three models exhibits varying dynamics as the training data size increases. ChatGLM2-6B (4K) and LLAMA2-7B(4K) consistently outperform Longformer across all sizes of training data, validating the efficacy of LLMs even when limited data is available. LLAMA2-7B LoRA (4K) starts off with a strong performance at just 1% of the training data and maintains the lead as the dataset grows. However, its performance appears to converge, showing marginal gains as the dataset size increases compared to CHATGLM2-6B (4K) which seems to show improvements as the dataset size increases. This divergence in behavior may be attributed to the architectural differences between the two models. Specifically, ChatGLM2-6B (4K) employs bidirectional attention mechanisms, which could require a larger dataset to optimize but also offer a more favorable inductive bias for information extraction tasks.

6 Conclusion

This paper introduces EYEGLAXS, a novel system that leverages Large Language Models (LLMs) for long text extractive summarization. Our work challenges the traditional reliance on encoder-only models, showcases the adaptability of LLMs in managing different sequence lengths, and sets new performance standards on the PubMed and arXiv datasets. Despite these advancements, the use of LLMs comes with its own set of challenges, notably in computational resource requirements and the limitations of fine-tuning. Looking ahead, we aim to integrate sliding attention mechanisms in LLMs to further refine our system. Additionally, we plan to enrich the LLM backbone with existing techniques such as graph-based methods or Reinforcement Learning. Overall, our work paves the way for new research avenues in extractive text summarization and substantiates the utility of LLMs in this field.

7 Limitations

While EYEGLAXS demonstrates promising advancements in extractive text summarization, it is not without its challenges. The use of Large Language Models (LLMs) requires significant computational resources, making it less accessible for those with limited capabilities. Moreover, we report only a single run for each of our experiment due to the expensive training time, as can be seen in the Table 6 in the appendix showing the duration of the training epochs. Additionally, the sheer size of these LLMs restricts the possibility of full fine-tuning, thereby limiting further optimization and reporting upper limit of the full fine-tuning. The model's performance is also closely tied to the size of the training dataset, especially for CHATGLM2, which could be a constraint in fields where large, labeled datasets are not readily available. Lastly, the system's generalizability remains untested outside of scientific contexts like PubMed and arXiv. These limitations offer valuable avenues for future research to improve the system's robustness and applicability. Finally, we wanted to highlight again the security risk if such tool is used in sensitive applications (e.g., legal, medical), poor performance or errors could lead to serious consequences.

References

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

Junyi Bian, Xiaodi Huang, Hong Zhou, and Shanfeng Zhu. 2023. Gosum: Extractive summarization of long documents by reinforcement learning and graph organized discourse state.

Jennifer Bishop, Qianqian Xie, and Sophia Ananiadou. 2022. Gencomparesum: a hybrid unsupervised summarization method using salience. In *Proceedings of the 21st workshop on biomedical language processing*, pages 220–240.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. 2023. Extending context window of large language models via positional interpolation. *arXiv e-prints*, pages arXiv–2306.

Sangwoo Cho, Kaiqiang Song, Xiaoyang Wang, Fei Liu, and Dong Yu. 2022. Toward unifying text segmentation and long document summarization. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 106–118, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 615–621, New Orleans, Louisiana. Association for Computational Linguistics.

Tri Dao. 2023. FlashAttention-2: Faster attention with better parallelism and work partitioning.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems*.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479.

Naman Goyal, Jingfei Du, Myle Ott, Giri Anantharaman, and Alexis Conneau. 2021. Larger-scale transformers for multilingual masked language modeling. In *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*, pages 29–33, Online. Association for Computational Linguistics.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Jean Kaddour, Joshua Harris, Maximilian Mozes, Herbie Bradley, Roberta Raileanu, and Robert McHardy. 2023. Challenges and applications of large language models. *arXiv preprint arXiv:2307.10169*.

Chris Kedzie, Kathleen McKeown, and Hal Daumé III. 2018. Content selection in deep learning models of summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1818–1828, Brussels, Belgium. Association for Computational Linguistics.

Vladislav Lialin, Vijeta Deshpande, and Anna Rumshisky. 2023. Scaling down to scale up: A guide to parameter-efficient fine-tuning.

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

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *CoRR*, abs/2107.13586.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 61–68, Dublin, Ireland. Association for Computational Linguistics.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3730–3740, Hong Kong, China. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang. 2022. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1864–1874, Dublin, Ireland. Association for Computational Linguistics.

Ofir Press, Noah Smith, and Mike Lewis. 2021. Train short, test long: Attention with linear biases enables input length extrapolation. In *International Conference on Learning Representations*.

Qian Ruan, Malte Ostendorff, and Georg Rehm. 2022. HiStruct+: Improving extractive text summarization with hierarchical structure information. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1292–1308, Dublin, Ireland. Association for Computational Linguistics.

Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. 2021. Roformer: Enhanced transformer with rotary position embedding. *CoRR*, abs/2104.09864.

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

Lucy Vanderwende, Hisami Suzuki, Chris Brockett, and Ani Nenkova. 2007. Beyond sumbasic: Task-focused summarization with sentence simplification and lexical expansion. *Information Processing & Management*, 43(6):1606–1618.

Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, and Guoyin Wang. 2023. GPT-NER: Named Entity Recognition via Large Language Models. *arXiv e-prints*, page arXiv:2304.10428.

Wen Xiao and Giuseppe Carenini. 2019. Extractive summarization of long documents by combining global and local context. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3011–3021, Hong Kong, China. Association for Computational Linguistics.

Qianqian Xie, Jennifer Amy Bishop, Prayag Tiwari, and Sophia Ananiadou. 2022. Pre-trained language models with domain knowledge for biomedical extractive summarization. *Knowledge-Based Systems*, 252:109460.

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontañón, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big bird: Transformers for longer sequences. *CoRR*, abs/2007.14062.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. 2022. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*.

Haopeng Zhang, Xiao Liu, and Jiawei Zhang. 2023a. Extractive summarization via chatgpt for faithful summary generation. *arXiv e-prints*, pages arXiv–2304.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2023b. Siren's song in the ai ocean: A survey on hallucination in large language models.

Model	Training Context Length	Evaluation Context Length								
		ARXIV								
		4K			12K			32K		
		R1	R2	RL	R1	R2	RL	R1	R2	RL
CHATGLM2-6b	4K	45.18	18.07	39.73	47.55	19.64	41.97	46.87	18.96	41.37
	12k	46.11	19.10	40.59	48.96	21.11	43.26	49.02	21.01	43.33
LLAMA2-7b	4K	45.84	18.84	40.37	48.58	20.81	42.90	48.68	20.72	42.98
	12K	45.97	19.08	40.54	48.80	21.09	43.16	48.96	21.07	43.30
		PUBMED								
		4K			12K			16K		
		R1	R2	RL	R1	R2	RL	R1	R2	RL
CHATGLM2-6b	4K	49.43	25.17	45.35	50.08	24.24	45.64	49.96	24.04	45.50
	12k	49.40	25.21	45.26	50.21	24.52	45.72	50.17	24.41	45.66
LLAMA2-7b	4K	48.59	24.55	44.58	49.54	23.80	45.15	49.48	23.64	45.08
	12K	50.38	24.70	46.02	50.39	24.70	46.03	50.34	24.57	45.96

Table 5: ROUGE Metrics of EYEGLAXS Variants on ARXIV and PUBMED Datasets at Different Training and Evaluation Context Lengths.

Model	Training Context Length	Training Time	
		ARXIV	PUBMED
CHATGLM2-6b	4K	8h 08m	8h 06m
CHATGLM2-6b	12K	52h 54m	31h 14m
LLAMA2-7b	4K	8h 36mn	8h 33m
LLAMA2-7b	12K	51h 35m	32h 29m

Table 6: Training Time for One Epoch of CHATGLM2-6b and LLAMA2-7b Models on ARXIV and PUBMED Datasets at Different Context Lengths.

A Results on the different datasets

We provide in Table 5 the different results of all variants from EYEGLAXS tested on the different versions of datasets we have built. This table provides us a better idea about how model trained on short document perfom on a dataset containing longer documents.

B Model Training Time

We show in Table 6 the training time for one epoch for each model on both arXiv and PubMed. Hardware specifics and training parameters are specified in the Experimental Settings section.