StreamingLLM with RAG

Cem Tepe
MIT
Cambridge, MA 01239
cemarda@mit.edu

Yanchen Liu
Harvard University
Cambridge, MA 01239
yanchenliu@g.harvard.edu

Abstract

Large language models (LLMs) are widely known to face challenges when dealing with long sequences due to computational limitations, such as memory constraints. Various approaches have been proposed to make the attention architecture sparse to address these limitations. However, these methods often struggle to capture the larger scope context of lengthy sequences effectively. In this work, we introduce a novel method called StreamingLLM with Retrieval Augmented Generation (StreamingLLMwRAG) and demonstrate its superior performance in long text summarization tasks. The StreamingLLMwRAG model leverages the power of retrieval-based techniques to enhance its understanding of the lengthy text, making it more effective at summarizing extensive documents. Our implementation and codebase for StreamingLLMwRAG are publicly available at `https://github.com/cmrtepe/StreamingLLMwRAG/tree/main`.

1 Introduction

Large language models (LLMs) [4, 17] encounter significant challenges when handling long sequences [6, 20], primarily due to computational limitations, including constraints on memory resources. The original self-attention mechanism [21], a fundamental component of many modern neural network architectures, exhibits a time complexity of $O(n^2)$, where n represents the input sequence length. This computational overhead imposes significant constraints on the scalability of models when processing longer input sequences. Consequently, there is a pressing need to devise sparse attention mechanisms that reduce the time complexity to linear, thereby enabling models to efficiently handle substantially longer sequences.

One notable advancement in this direction is StreamingLLM [22], a model that achieves linear complexity while maintaining competitive perplexity scores by adapting the window attention architecture. StreamingLLM has shown its prowess in addressing local tasks efficiently. However, a limitation of this model lies in its inability to effectively capture and utilize information beyond the defined window, which may prove undesirable in specific use cases.

To address these limitations, in this work, we combine StreamingLLM with recently highlighted retrieval-based techniques [9, 13, 3, 23, 16], resulting in StreamingLLM with Retrieval Augmented Generation (StreamingLLMwRAG). These enhancements empower the model to retain and effectively utilize information beyond the defined window, significantly broadening its applicability across various tasks and use cases.

Through an extensive series of experiments conducted on text summarization tasks, our findings underscore the remarkable capabilities of StreamingLLMwRAG. This variant leverages the power of retrieval-based techniques to augment its understanding of the lengthy text, resulting in significantly improved performance when summarizing extensive documents in comparison to its counterpart without retrieval mechanisms.

Preprint. Under review.

2 Related Work

Sparse Attention Sparse attention mechanisms have garnered significant attention in recent research efforts due to their potential to alleviate the computational challenges associated with self-attention models. [21] introduced the Transformer model, which relied on a full self-attention mechanism with quadratic complexity in sequence length, inspiring subsequent work on sparsity. Sparse Transformers [7] introduced structured patterns of attention that reduced complexity to linear or near-linear time with respect to the sequence length. Another significant advancement was made by Longformer [1], a model that leveraged global attention but with sparse attention patterns, achieving remarkable efficiency for long documents. Despite these efforts, retaining the ability to capture distant dependencies while maintaining computational efficiency remains a challenge. Our work builds upon these foundations and introduces StreamingLLM with Retrieval Augmented Generation (STREAMINGLLMWRAG), which combines the benefits of sparsity and retrieval-based techniques to address these limitations.

Retrieval-Augmented LMs Language Models (LMs) [4, 17] have demonstrated impressive capabilities across various natural language processing (NLP) tasks. However, these parametric LMs inherently lack adaptability over time [8, 12], often struggle to acquire extensive knowledge [18, 11], exhibit hallucinations [19], and may inadvertently disclose private data from the training corpus [5]. To address these limitations, retrieval-based LMs [9, 13, 3, 23, 16] incorporate a non-parametric datastore (e.g., text chunks from an external corpus) alongside their parametric counterparts. Retrieval-augmented LMs surpass LMs without retrieval by a significant margin while utilizing far fewer parameters [14]. Moreover, they have the capacity to update their knowledge by replacing their retrieval corpora [10] and offer citations, enabling users to easily validate and assess their predictions [15, 2].

3 StreamingLLM with RAG

The StreamingLLM model employs an attention mechanism divided into two crucial components: windowed attention and sink attention.

3.1 Windowed Attention

The windowed attention mechanism retains attention weights for tokens within a fixed distance from the current token, typically a constant size w. Given a query vector q, a set of key vectors k_i, and value vectors v_i for tokens i, where i ranges from 1 to n, the windowed attention can be described as follows:

$$\text{Attention}(q, k, v) = \sum_{i=1}^{n} \text{softmax}\left(\frac{q \cdot k_i}{\sqrt{d_k}}\right) \cdot v_i \tag{1}$$

Here, d_k represents the dimension of the key vectors, and the softmax operation computes the weighted sum of values based on the similarity between the query and key vectors.

3.2 Sink Attention

The sink attention mechanism focuses on the initial tokens throughout the entire sequence. It is calculated similarly to windowed attention, but it operates on a fixed set of key vectors k_{sink} and value vectors v_{sink}:

$$\text{Sink Attention}(q, k_{\text{sink}}, v_{\text{sink}}) = \sum_{i=1}^{n} \text{softmax}\left(\frac{q \cdot k_{\text{sink}}}{\sqrt{d_k}}\right) \cdot v_{\text{sink}} \tag{2}$$

This approach enhances the model's ability to capture essential information from the initial tokens, which remains important as sequence length grows.

3.3 StreamingLLM with Retrieval Augmented Generation (RAG)

In the context of StreamingLLM with RAG, the streaming process involves several well-defined steps that facilitate efficient information retrieval and attention adaptation. We formalize these steps here:

1. **Chunk Segmentation**: The current chat history H is segmented into non-overlapping chunks of a specified size C, creating a datastore D consisting of these chunks. This operation can be expressed as follows:

$$D = \text{Segment}(H, C)$$

2. **Top-k Chunk Selection**: To prioritize relevant information, a retrieval mechanism is employed to select the top-k most relevant chunks from the datastore D. This operation can be defined as:

$$D_{\text{selected}} = \text{Top-K}(D, k)$$

3. **Chunk Position Identification**: For each selected chunk in D_{selected}, the start and end indices within the original token sequence are determined. Let S and E represent sets of start and end indices, respectively, for all selected chunks:

$$(S, E) = \text{Identify}(D_{\text{selected}})$$

4. **Attention Mechanism Update**: The attention mechanism of the StreamingLLM is updated to incorporate the attention weights associated with each start-to-end interval identified in the previous step. This operation can be expressed as:

$$\text{Attention}(q, k, v) = \sum_{i=1}^{n} \text{softmax}\left(\frac{q \cdot k_i}{\sqrt{d_k}}\right) \cdot v_i$$

where q is the query vector, k_i represents the key vector for token i, v_i is the value vector for token i, and n is the number of tokens.

5. **Window Size Adjustment**: Finally, the attention window size is reduced by an amount proportional to the total number of tokens in the retrieved chunks. This ensures that the model continues to focus on the most relevant information while adapting to the retrieved context. The updated attention window size W' can be calculated as:

$$W' = W - \frac{\sum_{i=1}^{k} |E_i - S_i|}{\sum_{i=1}^{k} C_i}$$

where W is the original window size, k is the number of selected chunks, and C_i is the size of the i-th selected chunk.

Combining with RAG, StreamingLLMwRAG efficiently manages long sequences and adapts its attention mechanism during the streaming process. The choice of chunk size C is a critical parameter that depends on the specific use case. For tasks like long text summarization, larger chunk sizes are typically preferred to capture broader context, while for tasks like perplexity evaluation, smaller chunk sizes may be more appropriate to balance computational efficiency and memory constraints.

4 Experiments

4.1 Perplexity

First, we engage in a streaming process where a sequence of prompts is continuously fed to our models. Specifically, we evaluate the perplexity of two models: a vanilla StreamingLLM and a StreamingLLM with Retrieval Augmented Generation (StreamingLLMwRAG). Each stream consists of 50 prompts, and the average perplexity is computed after each stream. To ensure robustness, we repeat this process across 10 streams, and the results are subsequently averaged and compared in Fig. 1.

4.2 Long text summarization

To understand whether RAG improves the large-scope learning of the model, we further stream a novel and ask the model to generate a summary. In this case, the book is divided into 1500

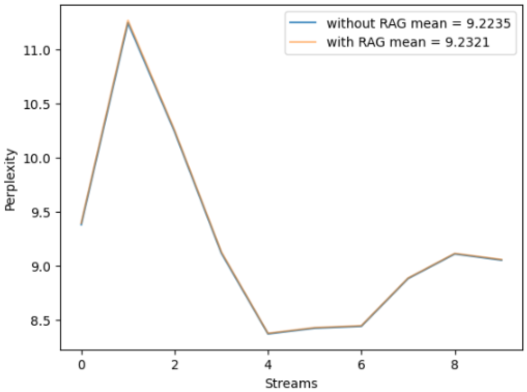
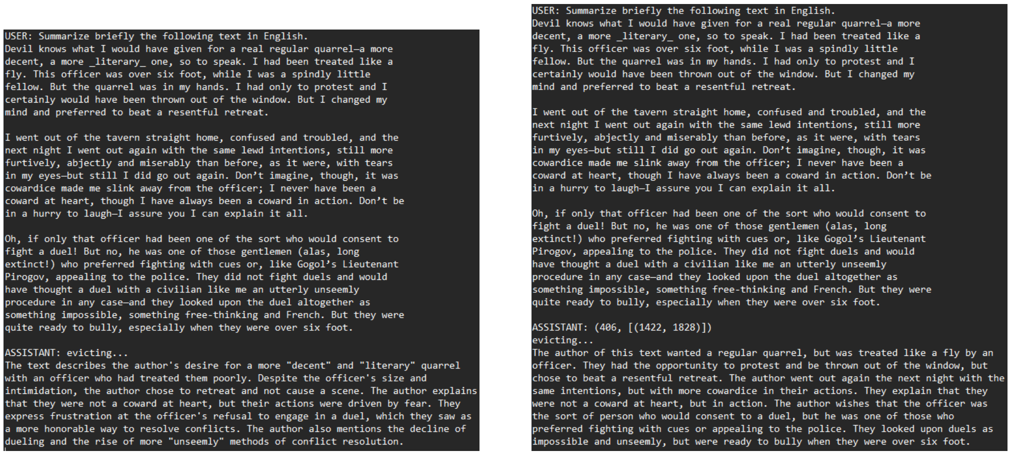


Figure 1: Perplexity values compared over 10 streams of 50 prompts. We observed that RAG does not seem to have an impact on perplexity, possibly because the task is localized.



```
USER: Summarize briefly the following text in English.
Devil knows what I would have given for a real regular quarrel—a more
decent, a more _literary_ one, so to speak. I had been treated like a
fly. This officer was over six foot, while I was a spindly little
fellow. But the quarrel was in my hands. I had only to protest and I
certainly would have been thrown out of the window. But I changed my
mind and preferred to beat a resentful retreat.

I went out of the tavern straight home, confused and troubled, and the
next night I went out again with the same lewd intentions, still more
furtively, abjectly and miserably than before, as it were, with tears
in my eyes—but still I did go out again. Don’t imagine, though, it was
cowardice made me slink away from the officer; I never have been a
coward at heart, though I have always been a coward in action. Don’t be
in a hurry to laugh—I assure you I can explain it all.

Oh, if only that officer had been one of the sort who would consent to
fight a duel! But no, he was one of those gentlemen (alas, long
extinct!) who preferred fighting with cues or, like Gogol’s Lieutenant
Pirogov, appealing to the police. They did not fight duels and would
have thought a duel with a civilian like me an utterly unseemly
procedure in any case—and they looked upon the duel altogether as
something impossible, something free-thinking and French. But they were
quite ready to bully, especially when they were over six foot.

ASSISTANT: evicting...
The text describes the author's desire for a more "decent" and "literary" quarrel
with an officer who had treated them poorly. Despite the officer's size and
intimidation, the author chose to retreat and not cause a scene. The author explains
that they were not a coward at heart, but their actions were driven by fear. They
express frustration at the officer's refusal to engage in a duel, which they saw as
a more honorable way to resolve conflicts. The author also mentions the decline of
dueling and the rise of more "unseemly" methods of conflict resolution.
```

(a) Without RAG

```
USER: Summarize briefly the following text in English.
Devil knows what I would have given for a real regular quarrel—a more
decent, a more _literary_ one, so to speak. I had been treated like a
fly. This officer was over six foot, while I was a spindly little
fellow. But the quarrel was in my hands. I had only to protest and I
certainly would have been thrown out of the window. But I changed my
mind and preferred to beat a resentful retreat.

I went out of the tavern straight home, confused and troubled, and the
next night I went out again with the same lewd intentions, still more
furtively, abjectly and miserably than before, as it were, with tears
in my eyes—but still I did go out again. Don’t imagine, though, it was
cowardice made me slink away from the officer; I never have been a
coward at heart, though I have always been a coward in action. Don’t be
in a hurry to laugh—I assure you I can explain it all.

Oh, if only that officer had been one of the sort who would consent to
fight a duel! But no, he was one of those gentlemen (alas, long
extinct!) who preferred fighting with cues or, like Gogol’s Lieutenant
Pirogov, appealing to the police. They did not fight duels and would
have thought a duel with a civilian like me an utterly unseemly
procedure in any case—and they looked upon the duel altogether as
something impossible, something free-thinking and French. But they were
quite ready to bully, especially when they were over six foot.

ASSISTANT: (406, [(1422, 1828)])
evicting...
The author of this text wanted a regular quarrel, but was treated like a fly by an
officer. They had the opportunity to protest and be thrown out of the window, but
chose to beat a resentful retreat. The author went out again the next night with the
same intentions, but with more cowardice in their actions. They explain that they
were not a coward at heart, but in action. The author wishes that the officer was
the sort of person who would consent to a duel, but he was one of those who
preferred fighting with cues or appealing to the police. They looked upon duels as
impossible and unseemly, but were ready to bully when they were over six foot.
```

(b) With RAG

Figure 2: Summaries of the same prompt from the middle of the stream with and without RAG. In (b), before the summary, it also prints the retrieved sequence length, and its start and end indices.

token-length prompts, and for each prompt, we ask the model to generate a summary in English. These summaries are combined and divided into new chunks, repeating the streaming process. A short section of the summaries from the first part of the process is shown in 2. At each step, of the reduction process described above, the model with RAG can retrieve prior summaries outside of its window depending on relevance to make the final output more cohesive. We demonstrated that using RAG makes the process of summarizing extensive documents more effective. However, this requires further testing to see if it works as intended.

5 Conclusion

In this study, we combined the Retrieval-Augmented Generation approach with the StreamingLLM model and assessed its performance based on perplexity and long text summarization. Our analysis revealed that the integration of RAG did not impact perplexity, which could be attributed to the task's localized nature. However, our investigation of long text summaries demonstrated that when RAG was applied in the later segments of the streaming process, the generated summaries exhibited greater alignment with the entire context of the text. It is important to note that our experiments were conducted under stringent memory constraints, permitting only a single retrieved chunk to be used. In future research, removing these constraints would be important for gaining a comprehensive understanding of this architecture's full summarization potential.

References

[1] Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer, 2020.

[2] Bernd Bohnet, Vinh Q. Tran, Pat Verga, Roee Aharoni, Daniel Andor, Livio Baldini Soares, Massimiliano Ciaramita, Jacob Eisenstein, Kuzman Ganchev, Jonathan Herzig, Kai Hui, Tom Kwiatkowski, Ji Ma, Jianmo Ni, Lierni Sestorain Saralegui, Tal Schuster, William W. Cohen, Michael Collins, Dipanjan Das, Donald Metzler, Slav Petrov, and Kellie Webster. Attributed question answering: Evaluation and modeling for attributed large language models, 2023.

[3] Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego De Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack Rae, Erich Elsen, and Laurent Sifre. Improving language models by retrieving from trillions of tokens. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 2206–2240. PMLR, 17–23 Jul 2022.

[4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[5] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting training data from large language models, 2021.

[6] Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. Extending context window of large language models via positional interpolation, 2023.

[7] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers, 2019.

[8] Nicola De Cao, Wilker Aziz, and Ivan Titov. Editing factual knowledge in language models. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6491–6506, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[9] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. Realm: Retrieval-augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

[10] Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. Atlas: Few-shot learning with retrieval augmented language models, 2022.

[11] Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. Large language models struggle to learn long-tail knowledge, 2023.

[12] Jungo Kasai, Keisuke Sakaguchi, Yoichi Takahashi, Ronan Le Bras, Akari Asai, Xinyan Yu, Dragomir Radev, Noah A. Smith, Yejin Choi, and Kentaro Inui. Realtime qa: What's the answer right now?, 2022.

[13] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021.

[14] Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories, 2023.

[15] Jacob Menick, Maja Trebacz, Vladimir Mikulik, John Aslanides, Francis Song, Martin Chadwick, Mia Glaese, Susannah Young, Lucy Campbell-Gillingham, Geoffrey Irving, and Nat McAleese. Teaching language models to support answers with verified quotes, 2022.

[16] Sewon Min, Weijia Shi, Mike Lewis, Xilun Chen, Wen-tau Yih, Hannaneh Hajishirzi, and Luke Zettlemoyer. Nonparametric masked language modeling. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2097–2118, Toronto, Canada, July 2023. Association for Computational Linguistics.

[17] OpenAI. Gpt-4 technical report, 2023.

[18] Adam Roberts, Colin Raffel, and Noam Shazeer. How much knowledge can you pack into the parameters of a language model? In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online, November 2020. Association for Computational Linguistics.

[19] Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. Retrieval augmentation reduces hallucination in conversation. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3784–3803, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[20] Szymon Tworkowski, Konrad Staniszewski, Mikołaj Pacek, Yuhuai Wu, Henryk Michalewski, and Piotr Miłoś. Focused transformer: Contrastive training for context scaling, 2023.

[21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[22] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks, 2023.

[23] Zexuan Zhong, Tao Lei, and Danqi Chen. Training language models with memory augmentation, 2022.