Carnegie Mellon University

Scaling Transformer to 1M tokens and beyond with RMT

Vedant Bhasin, Pin Qian, Prince Wang, Xinyu Yang

Topic list

1. Motivation - Pin Qian
2. Related Work - Pin Qian
3. Recurrent Memory Transformer - Vedant
4. Memorization tasks
5. Learning Memory Optimizations - Xinyu
6. Natural and Formal Language modelling
7. Conclusion & Discussion

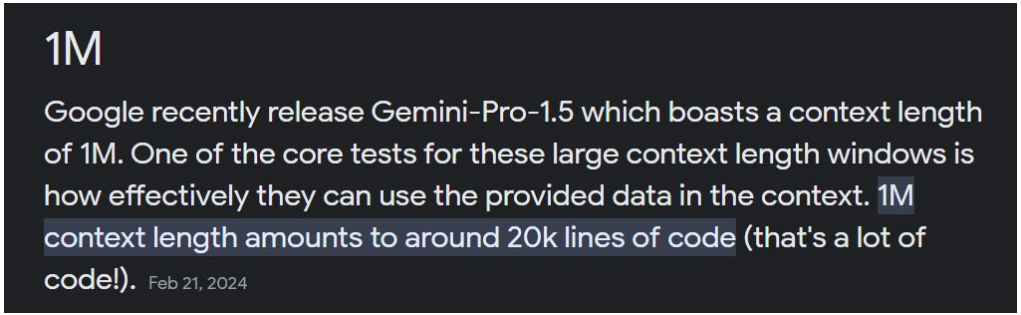
Topic list

1. **Motivation**
    a. **Why Long-Context LLMs?**
    b. **Challenges in Achieving Long-Context LLMs**
    c. **What does RMT aim to solve?**
2. Related Work
3. Recurrent Memory Transformer
4. Memorization tasks
5. Learning Memory Optimizations
6. Natural and Formal Language modelling
7. Conclusion & Discussion

Why Long-Context LLMs?

One of the primary limitations of transformers is their ability to operate on long sequences of tokens. For many applications of LLMs, overcoming this limitation is powerful.

- In Retrieval Augmented Generation (RAG), a longer context augments our model with more information.
- For Long-term planning tasks, such as chatbots, longer context means more tools and capabilities.
- ……



Challenges in Achieving Long-Context LLMs

1. Attention Complexity

- The length of an input sequence is limited by **quadratic computational complexity** of attention.

2. Max-Length Constraint

- During training, determining the **max-length** is necessary, which is commonly set based on the available computational resources.
- During inference, we also need to restrict the prompts length, since current Language Models exhibit noticeable performance degradation when handling input sequences exceeding max-length.

3. In-Context Memory

- LLMs lack an memory mechanism. Global and local information has to be stored in the same **element-wise** representations. They rely on the **KV cache** to store representations of all previous tokens.
- This statelessness offers computational advantages in terms of parallelism but presents challenges in tasks like chatbot applications, where **long-term memory retention** is essential.

Advancing Transformer Architecture in Long-Context Large Language Models: A Comprehensive Survey (Huang et al., 2024)

Recurrent Memory Transformer(RMT)

- **What does RMT aim to solve?**
  - Attention Complexity ✔
  - Max-Length Constraint ✔
  - In-Context Memory ✔
- **What is RMT?**
  - Based on special memory tokens similar to **Memory Transformer**, segment-level recurrence as in **Transformer-XL**.
  - **Memory** allows to store and process local and global information as well as to pass information between segments of the long sequence with the help of **recurrence**.

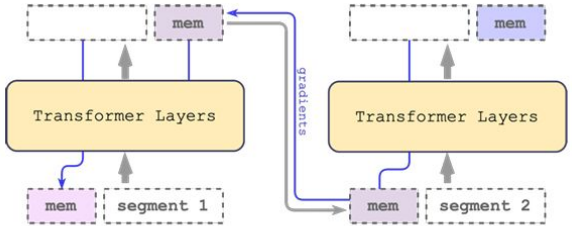


Figure 1: **Recurrent Memory Transformer.**

Recurrent Memory Transformer (Bulatov et al., 2022)

Carnegie Mellon University

Topic list

1. Motivation
2. **Related Work**
3. Recurrent Memory Transformer
4. Memorization tasks
5. Learning Memory Optimizations
6. Natural and Formal Language modelling
7. Conclusion & Discussion

Overview of Methods for Long-Context LLMs

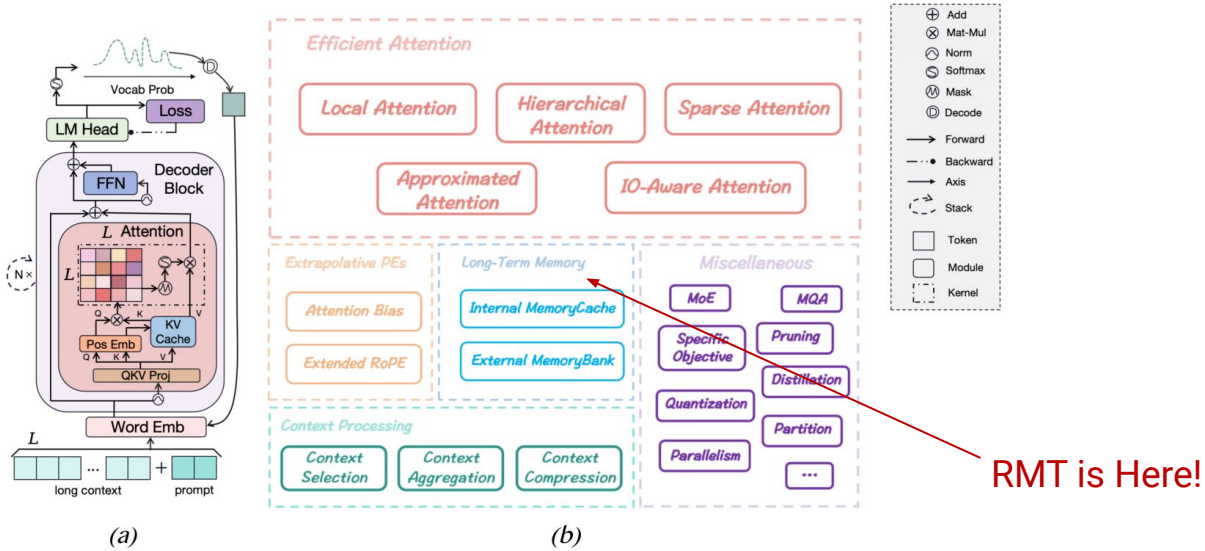


Fig. 1. The overview of the survey: ***(a)*** The typical architecture anatomy diagram of contemporary Transformer-based decoder-only LLMs, with the legend on the far top right; ***(b)*** The taxonomy of methodologies for enhancing Transformer architecture modules (corresponding to ***(a)*** by color): Efficient Attention (submodule of attention kernel), Long-Term Memory (targeting KV cache), Extrapolative PEs (against the positional embedding module), Context Processing (related to context pre/post-processing), and Miscellaneous (general for the whole Decoder Block as well as the Loss module).

Advancing Transformer Architecture in Long-Context Large Language Models: A Comprehensive Survey (Huang et al., 2024)

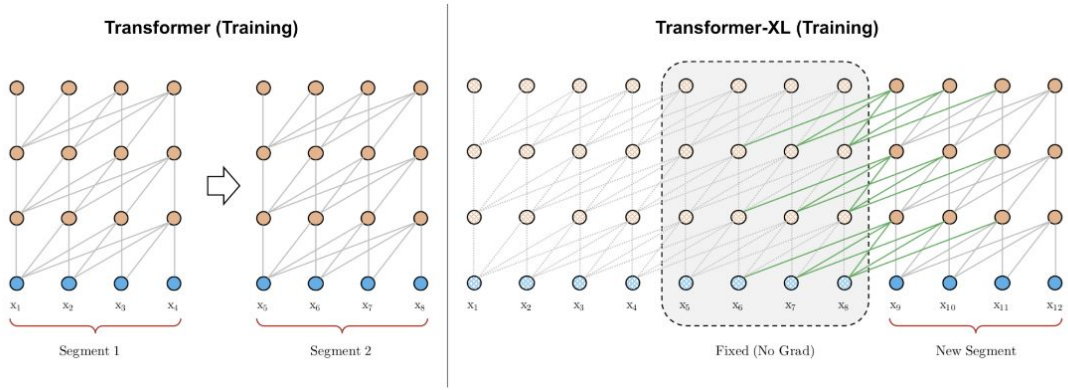
Carnegie Mellon University

Related Work: Transformers with Context Memory

Main Idea: Long inputs are divided into smaller segments, processed sequentially with memory to access information from past segments

Transformer-XL(Dai et al., 2019; "XL" stands for "extra long")

- Modifies the architecture to reuse hidden states between segments with an additional memory. The recurrent connection between segments is introduced into the model by continuously using the hidden states from the previous segments.



Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context (Dai et al., 2019)

Carnegie Mellon University

Related Work: Transformers with Context Memory

Main Idea: Long inputs are divided into smaller segments, processed sequentially with memory to access information from past segments

Compressive Transformer (Rae et al., 2020)

- Extends Transformer-XL by compressing past memories to support longer sequences.

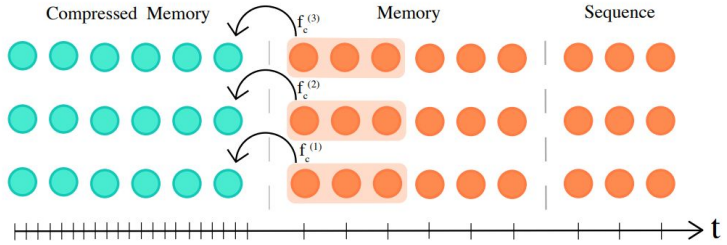


Figure 1: The Compressive Transformer keeps a fine-grained memory of past activations, which are then compressed into coarser *compressed* memories. The above model has three layers, a sequence length $n_s = 3$, memory size $n_m = 6$, compressed memory size $n_{cm} = 6$. The highlighted memories are compacted, with a compression function f_c per layer, to a single compressed memory — instead of being discarded at the next sequence. In this example, the rate of compression $c = 3$.

Limitations: A drawback of most existing recurrent methods is the need for architectural modifications that complicate their application to various pre-trained models.

What can RMT do: RMT can be built upon any model that uses a common supported interface (like Hugging Face Transformers).

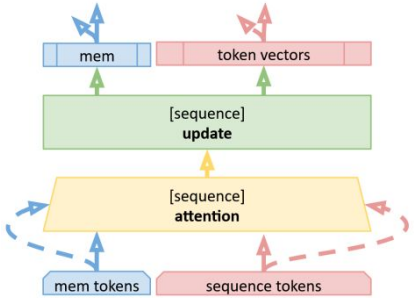
Compressive Transformers for Long-Range Sequence Modelling (Rae et al., 2020)

Related Work: Transformers with Context Memory

Main Idea: Long inputs are divided into smaller segments, processed sequentially with memory to access information from past segments

Memory Transformer(Burtsev et al., 2021)

- Extend the Transformer by adding [mem] tokens at the beginning of the input sequence and train the model to use them as universal memory storage



$$X^{mem+seq} = [X^{mem}; X^{seq}] \in \mathbb{R}^{(n+m)\times d}, X^{mem} \in \mathbb{R}^{m\times d}, X^{seq} \in \mathbb{R}^{n\times d}.$$

This modification can be applied independently to encoder and/or decoder. The rest of the Transformer stays the same with the multi-head attention layer processing the extended input.

Memory Transformer (Burtsev et al., 2021)

Related Work: Combination of Local and Global Context

Main idea: combine both local and global context when building an attention matrix.

- **ETC** (*Extended Transformer Construction*; Ainslie et al. 2019),
- **Longformer** (Beltagy et al. 2020)
- **Big Bird** (Zaheer et al. 2020)

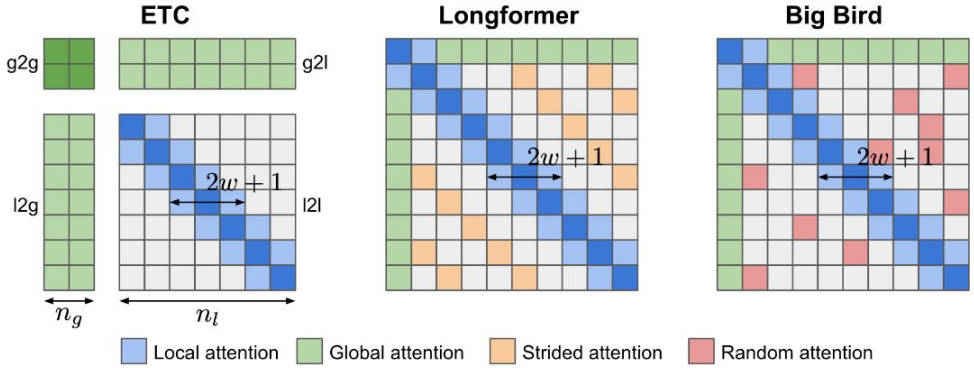


Fig. 19. Attention patterns of ETC, Longformer and Big Bird.

Limitations: A common constraint of these methods is that memory requirements grow with input size during both training and inference, inevitably limiting input scaling due to hardware constraints.

What can RMT do: In contrast, recurrent approaches have constant memory complexity during inference.

Carnegie Mellon University

Topic list

1. Motivation
2. Related Work
3. **Recurrent Memory Transformer**
4. Memorization tasks
5. Learning Memory Optimizations
6. Natural and Formal Language modelling
7. Conclusion & Discussion

Recurrent Memory Transformer(RMT)

- **What does RMT aim to solve?**
  - Attention Complexity ✔
  - Max-Length Constraint ✔
  - In-Context Memory ✔
- **What is RMT?**
  - Based on special memory tokens similar to **Memory Transformer**, segment-level recurrence as in **Transformer-XL**.
  - **Memory** allows to store and process local and global information as well as to pass information between segments of the long sequence with the help of **recurrence**.

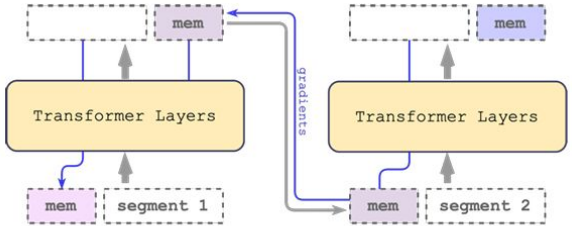
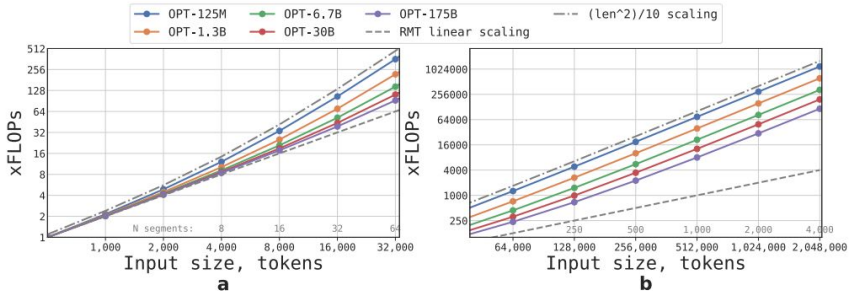


Figure 1: **Recurrent Memory Transformer.**

Recurrent Memory Transformer (Bulatov et al., 2022)

Carnegie Mellon University

Recurrent Memory Transformer(RMT)



- RMT scales linearly for any model size if the segment length is fixed
- Linear scaling is achieved by dividing a input sequence into segments and computing the full attention matrix only within segment boundaries
- RMT requires fewer FLOPs than non-recurrent models for sequence with more than one segment

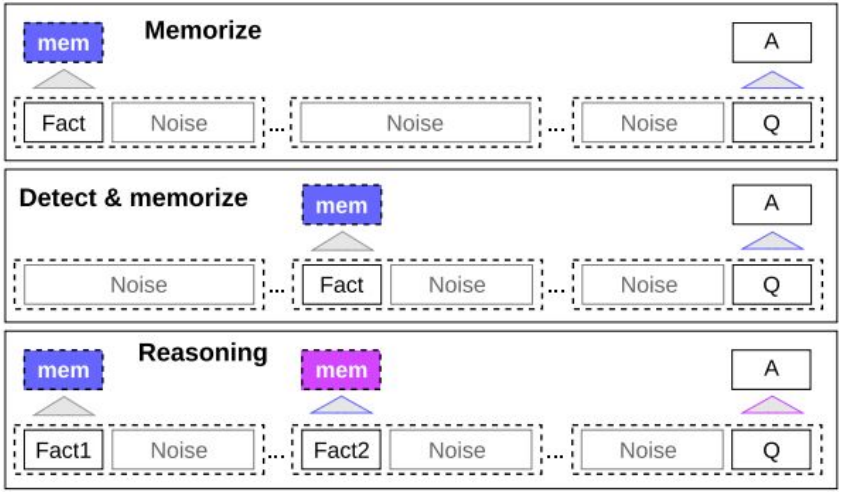
Carnegie Mellon University

Topic list

1. Motivation
2. Related Work
3. Recurrent Memory Transformer
4. **Memorization tasks**
5. Learning Memory Optimizations
6. Natural and Formal Language modelling
7. Conclusion & Discussion

Memorization Tasks

- Top: RMT's ability to write and store information in memory for an extended time.
- Middle: Fact detection task, which removes the fact to a random position in the input, requiring the model to first distinguish the fact from irrelevant text, write it to memory, and later use it to answer the question.
- Bottom: ability to operate with several facts and current context



Synthetics tasks are constructed to test memorization abilities

Recurrent Memory Transformer (Bulatov et al., 2022)

Carnegie Mellon University

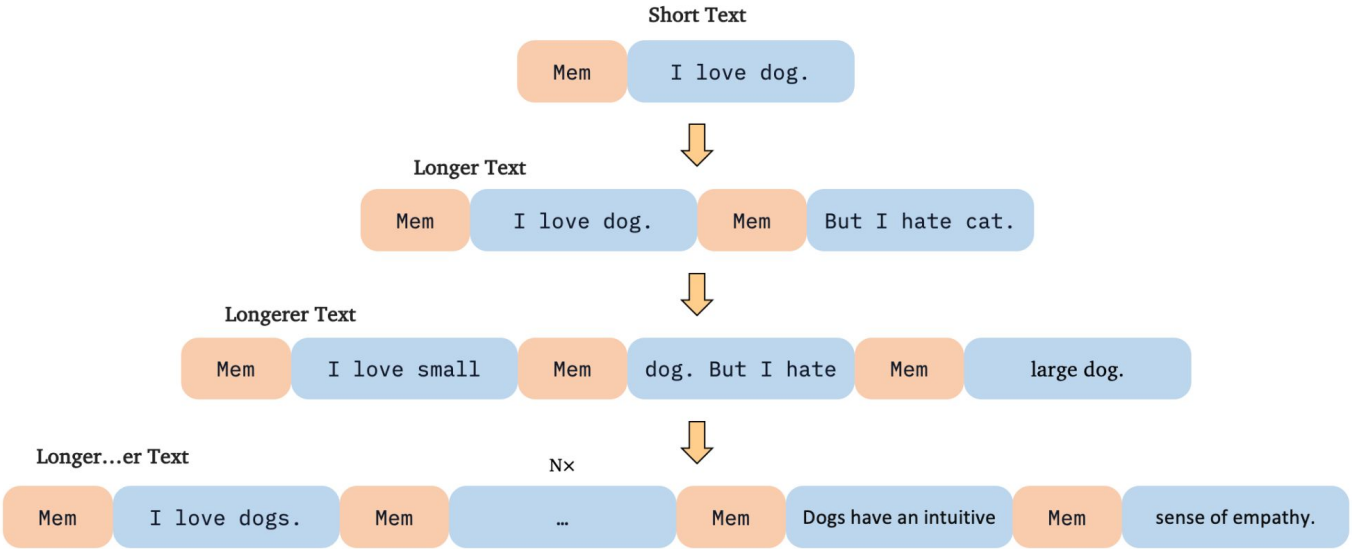
Topic list

1. Motivation
2. Related Work
3. Recurrent Memory Transformer
4. Memorization tasks
5. **Learning Memory Optimizations**
6. Natural and Formal Language modelling
7. Conclusion & Discussion

Learning Memory Operations

Question 1: How improve training stability of the original RMT ?

Answer: Curriculum Learning.

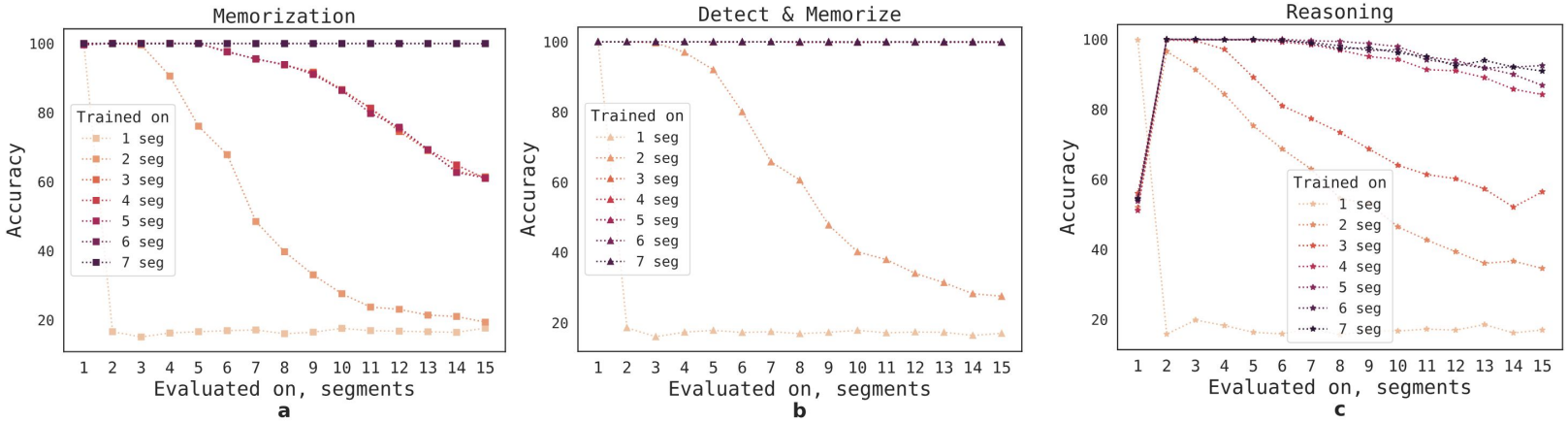


Carnegie Mellon University

Learning Memory Operations

Question 2: How well does RMT generalize to different sequence lengths?

Answer: The generalization ratio enlarges with increasing numbers of training segmentations.



Carnegie Mellon University

Learning Memory Operations

Question 2: How well does RMT generalize to different sequence lengths?

Answer: We observe a extrapolation of more than 500 times on a pre-trained BERT model.

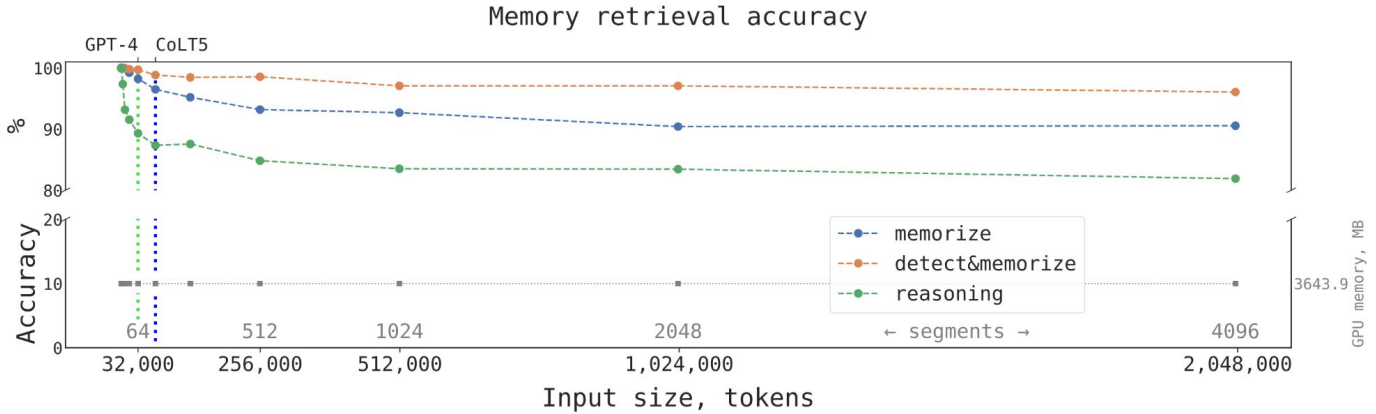
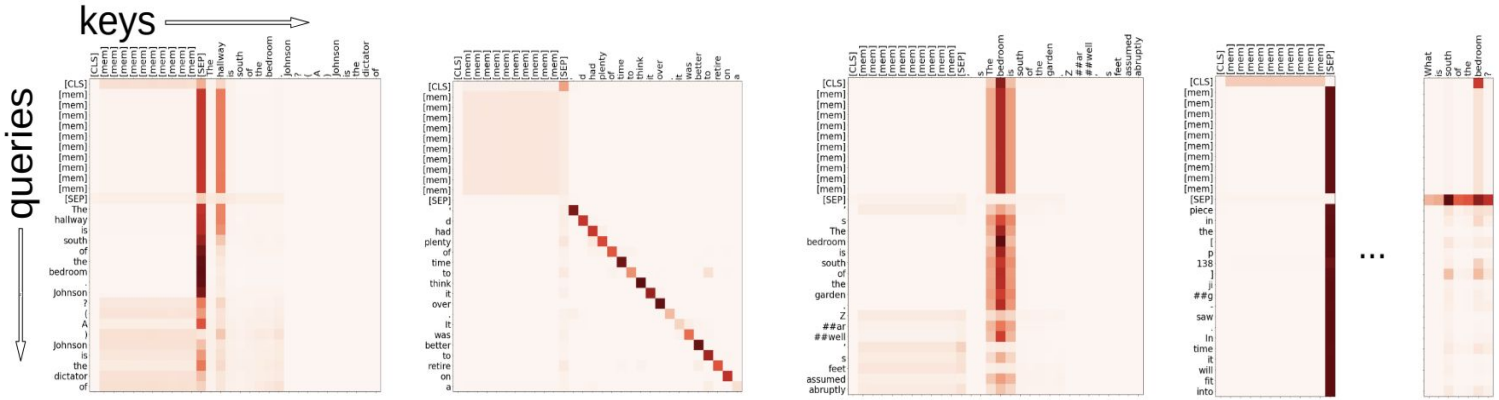


Figure 5: **Recurrent Memory Transformer retains information across up to 2×10^6 tokens**. By augmenting a pre-trained BERT model with recurrent memory (Bulatov, Kuratov, and Burtsev 2022), we enabled it to store task-specific information across 7 segments of 512 tokens each. During inference, the model effectively utilized memory for up to 4,096 segments with a total length of 2,048,000 tokens—significantly exceeding the largest input size reported for transformer models (64K tokens for CoLT5 (Ainslie et al. 2023), and 32K tokens for GPT-4 (OpenAI 2023), and 100K tokens for Claude). This augmentation maintains the base model's memory size at 3.6 GB in our experiments.

Carnegie Mellon University

Learning Memory Operations

Question 3: Can RMT identify important facts and store them in memory



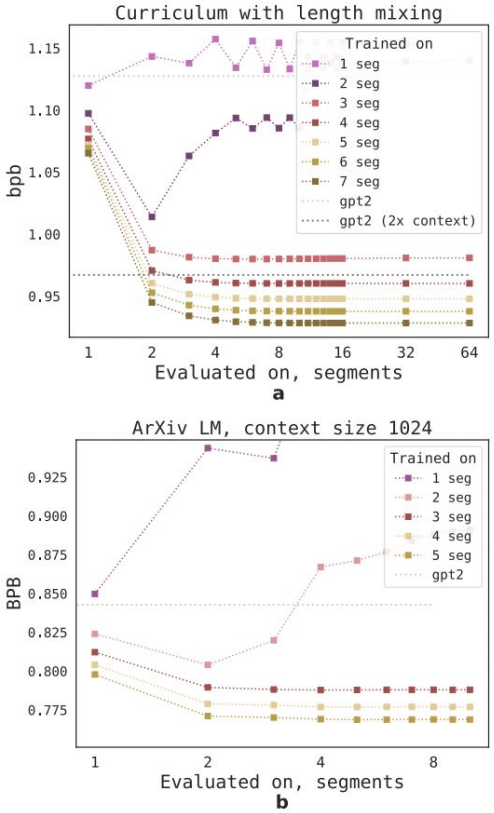
Carnegie Mellon University

Topic list

1. Motivation
2. Related Work
3. Recurrent Memory Transformer
4. Memorization tasks
5. Learning Memory Optimizations
6. **Natural and Formal Language modelling**
7. Conclusion & Discussion

Natural and Formal Language Modelling

- RMT trained for an equal number of steps as the baseline GPT-2 displays substantially lower perplexity values
- Increasing number of segments in training RMT exhibits better tolerance to longer history sizes.

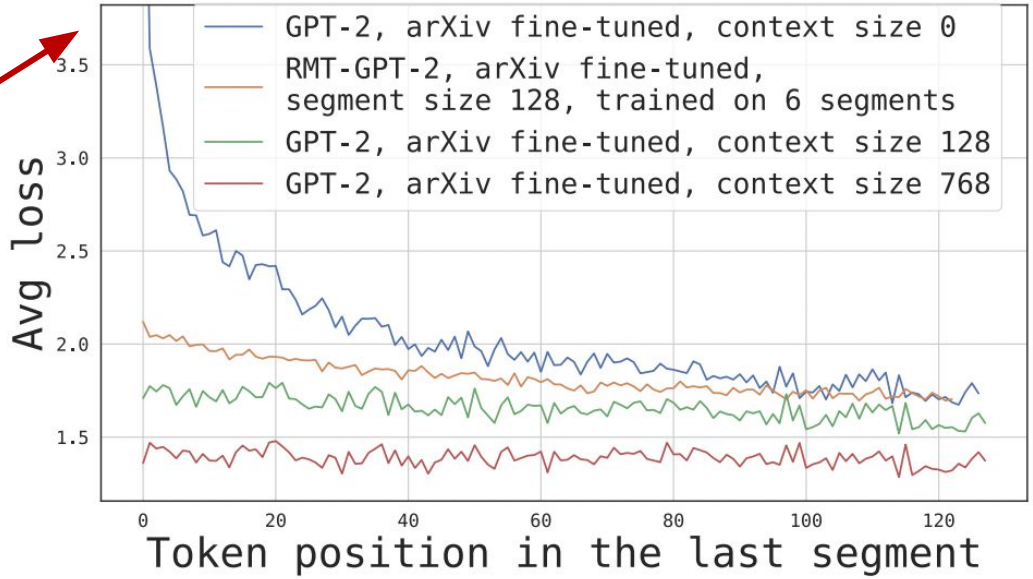


Carnegie Mellon University

Natural and Formal Language Modelling

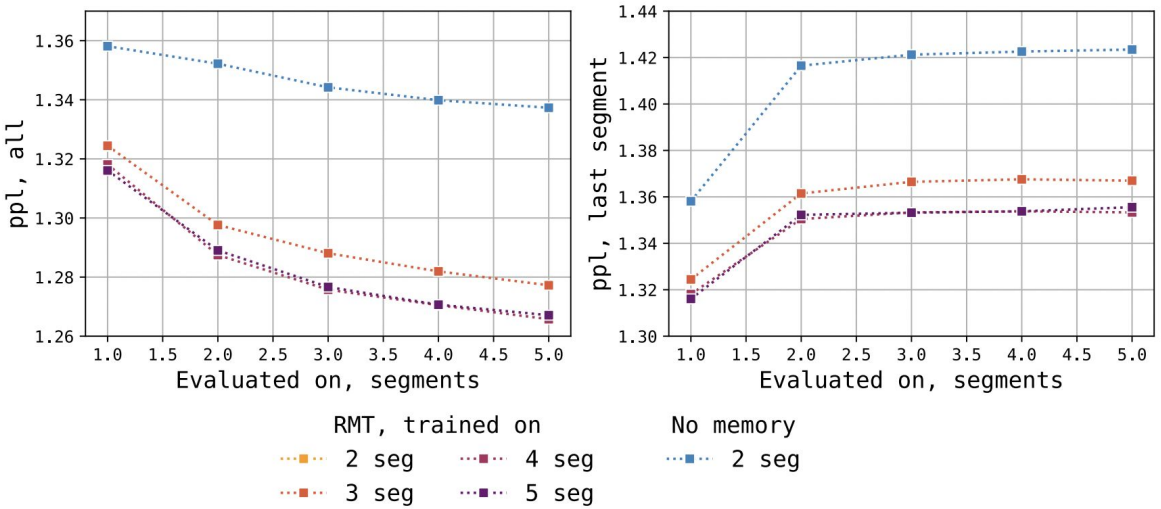
RMT ensures equally good prediction for all tokens due to carryover of information from the previous segment

High initial loss due to lack of context



Carnegie Mellon University

Natural and Formal Language Modelling



The RMT model improves perplexity compared to the memory-less model.

However, training with 4 or more segments does not enhance predictions for longer sequences

Carnegie Mellon University

Topic list

1. Motivation
2. Related Work
3. Recurrent Memory Transformer
4. Memorization tasks
5. Learning Memory Optimizations
6. Natural and Formal Language modelling
7. **Conclusion & Discussion**

Conclusion

- Problem: Long input scaling in Transformers (Encoder-only & Decoder-only)
- Solution: Segment-level recurrence using recurrent memory (RMT , a kind of compression)
  1. Linear Inference Complexity
  2. Limitation of Sequence Length
  3. In-context Memory
- Training method: Curriculum Learning (short → long → longer → ... → longer)
- Extrapolation: RMT can handle sequences exceeding 1 million tokens while only training on sentence no more than 5k tokens. Therefore, it can maintain computational complexity during training and inference.

Discussion

- Can RMT perform well in specialized memory-intensive tasks? Especially real-world tasks requiring long range dependency ?
- Comparing to other recurrent-based approaches, such as Mamba [1] and Griffin [2], what are the advantages and disadvantages of RMT ?
- Evaluated on BERT, Opt and GPT2, the effectiveness of RMT on recent LLMs remains unknown.

[1] Albert Gu, Tri Dao, Mamba: Linear-Time Sequence Modeling with Selective State Spaces.

[2] De *et al.* , Griffin: Mixing Gated Linear Recurrences with Local Attention for Efficient Language Models.