

基于记忆增强的长文本建模方法

刘炜杰, 汤泽成, 李俊涛

引用本文

刘炜杰, 汤泽成, 李俊涛. 基于记忆增强的长文本建模方法[J]. 计算机科学, 2025, 52(12): 231-238.

LIU Weijie, TANG Zecheng, LI Juntao. MemLong:Memory-augmented Retrieval for Long Text Modeling [J]. Computer Science, 2025, 52(12): 231-238.

相似文章推荐 (请使用火狐或 IE 浏览器查看文章)

Similar articles recommended (Please use Firefox or IE to view the article)

[大语言模型驱动的多智能体协同代码生成技术](#)

Multi-agent Collaborative Code Generation Technology Driven by Large Language Models

计算机科学, 2025, 52(11A): 241200033-9. <https://doi.org/10.11896/jsjcx.241200033>

[结合图检索与上下文排序的检索增强生成技术研究](#)

Research on Retrieval-augmented Generation Technology Combining Graph Retrieval and Contextual Ranking

计算机科学, 2025, 52(11A): 250100011-4. <https://doi.org/10.11896/jsjcx.250100011>

[基于深度学习的自然语言处理技术在智能翻译系统中的应用研究](#)

Research on Application of Deep Learning-based Natural Language Processing Technology in Intelligent Translation Systems

计算机科学, 2025, 52(11A): 241000037-6. <https://doi.org/10.11896/jsjcx.241000037>

[基于多语言嵌入图卷积网络的仇恨言论检测方法](#)

Multi-language Embedding Graph Convolutional Network for Hate Speech Detection

计算机科学, 2025, 52(11A): 241200023-8. <https://doi.org/10.11896/jsjcx.241200023>

[信息抽取技术在数字人文领域的应用研究综述](#)

Review of Application of Information Extraction Technology in Digital Humanities

计算机科学, 2025, 52(11A): 250600198-10. <https://doi.org/10.11896/jsjcx.250600198>

基于记忆增强的长文本建模方法

刘炜杰 汤泽成 李俊涛

苏州大学计算机科学与技术学院人工智能实验室 江苏 苏州 215031

(20224227039@stu.suda.edu.cn)

摘要 大语言模型(LLMs)近年来取得了显著进展,并在多个领域展现出卓越的性能。然而,由于注意力机制的二次时空复杂度以及生成过程中键值对缓存不断增长所带来的显存消耗,处理长文本相关任务仍然是LLMs面临的一大挑战。为了解决此问题,提出了基于记忆增强的长文本建模方法 MemLong,旨在利用外部检索器检索历史信息来增强长上下文语言建模的能力。MemLong 将一个非参数化的检索-记忆模块与一个部分可训练的大语言模型相结合,并引入了一种能够利用语义层面相关的文本块的细粒度可控的检索注意力机制。非参数化的检索-记忆模块负责从外部知识库中检索与当前输入相关的历史信息,而大语言模型则将检索到的信息和当前输入融合在一起并生成输出。细粒度可控的检索注意力机制允许模型在生成过程中动态地调整对检索信息的关注程度,从而实现更精准的文本生成。在多个长上下文语言建模基准测试上的综合评估表明,MemLong 方法始终优于其他先进的LLMs。此外,MemLong 显著提升了模型处理长文本的能力。在单卡 3090 GPU 上,MemLong 可以将上下文长度从 4000 扩展到 80000,提升了 20 倍。这一突破性的进展使得 MemLong 能够处理更长的输入文本,从而更好地理解 and 生成长文本内容,为处理超长文本任务提供了新的可能性,并为未来长文本语言建模的研究开辟了新的方向。

关键词: 检索增强语言建模;长文本生成;自然语言处理;长文本评估;检索增强生成

中图分类号 TP311

MemLong: Memory-augmented Retrieval for Long Text Modeling

LIU Weijie, TANG Zecheng and LI Juntao

School of Computer Science and Technology Artificial Intelligence Laboratory, Soochow University, Suzhou, Jiangsu 215031, China

Abstract Recent advancements in Large Language Models (LLMs) have yielded remarkable success across diverse fields. However, handling long contexts remains a significant challenge for LLMs due to the quadratic time and space complexity of attention mechanisms and the growing memory consumption of the key-value cache during generation. To address this issue, this paper proposes MemLong's a memory-augmented method for long-text modeling, which enhances long-context language modeling by leveraging an external retriever to access historical information. MemLong integrates a non-parametric retrieval-memory module with a partially trainable large language model, and introduces a fine-grained, controllable retrieval attention mechanism that effectively utilizes semantically relevant text blocks. The non-parametric module is responsible for retrieving relevant historical information from an external knowledge base, while the LLM generates outputs by fusing this retrieved information with the current input. The proposed attention mechanism allows the model to dynamically adjust its focus on the retrieved information during generation. Comprehensive evaluations on multiple long-context language modeling benchmarks demonstrate that MemLong consistently outperforms other state-of-the-art LLMs. Furthermore, MemLong significantly enhances the model's capacity to process long texts. On a single NVIDIA 3090 GPU, MemLong can scale the effective context length from 4000 to 80000 tokens, representing a 20-fold increase. This breakthrough enables MemLong to process longer input texts, leading to a better understanding and generation of long-form content. It provides new possibilities for tackling ultra-long text tasks and opens up promising new directions for future research in long-text language modeling.

Keywords Retrieval augment language modeling, Long context generation, Natural language processing, Long context evaluation, Retrieval augment generation

1 引言

大语言模型(Large Language Models, LLMs)在各个领域取得了显著的成功。然而,由于注意力机制^[1]的时空复杂

度呈二次方增长,扩展模型上下文具有一定的挑战性,这限制了涉及长序列任务的应用(如长文档摘要^[2]和多轮对话^[3])的发展。另一方面,人们普遍期望大语言模型能够拥有持久的工作能力(即长上下文大语言模型),从而高效

应对这些高要求的场景。

为了解决长文本问题,大量工作涌现。一类是高效注意力机制^[4-9]。其优点是实现简单,通过修改注意力机制以达到时间复杂度尽可能地靠近 $O(n)$;缺点是会损害一定的模型能力。因此,一些工作将注意力转移到记忆选择^[10-12]上,如词元级(Token-level)的记忆选择,但会导致语义信息的截断。另一项研究工作是检索增强语言建模^[13-15],通常通过引入检索机制来提升模型处理长文本的能力。然而,这些方法也伴随着一些显著的缺点。首先,由于训练过程中模型参数会动态变化,存储在记忆中的信息可能会出现分布偏移。其次,这些方法往往需要重新训练,这在当前大模型盛行的背景下显得不切实际。最后,这些模型在处理长文本输入时,往往以牺牲预训练模型的原始能力为代价。

为克服上述研究中存在的局限性,本文提出了一个研究问题:能否利用检索器的显式检索能力来近似模型内部的隐式检索过程?

本文提出了 MemLong,一种高效且轻量级的扩展 LLM 上下文窗口的方法。其关键思想是将过去的上下文和知识存储在一个不可训练的记忆库中,并利用这些存储的嵌入,从记忆库中检索块级键值对($K-V$ 对)输入模型。MemLong 通过结合一个额外的 ret-mem 组件用于记忆和检索,以及一个检

索因果注意力模块用于整合局部和记忆信息,从而适用于任何 decoder-only 的预训练语言模型。MemLong 的记忆与检索过程如图 1(b)所示。在生成过程中,超过模型最大处理长度的文本被存储在记忆中作为上下文信息。随后,给定一个长文档中最近生成的文本块,使用检索器显式地检索过去的信息,通过索引对齐获得额外的上下文信息。综合来看,MemLong 具有以下优势。

1)分布一致性:与之前的信息存储在记忆中存在分布偏移的模型方法不同,MemLong 能确保记忆的信息保持一致。

2)训练高效:本文通过冻结模型的底层,只微调上层,大大降低了计算成本。在实验过程中,仅需使用 0.5B 个词元对 MemLong 的 3B 参数版本进行微调,配备 8 张 3090 GPU,持续运行 8 小时即可完成。

3)扩展上下文窗口:由于只需要记忆单层的 $K-V$ 对,MemLong 能够在单个 3090 GPU 上轻松地将上下文窗口扩展到 8 万个词元。

大量实验表明,与其他领先的 LLMs 相比,MemLong 在多个方面表现出优异的性能。MemLong 在多个长上下文语言建模数据集上优于 OpenLLaMA^[16] 和其他基于检索的模型。在检索增强的上下文学习任务中,MemLong 相较于 OpenLLaMA 实现了高达 10.2 个百分点的提升。

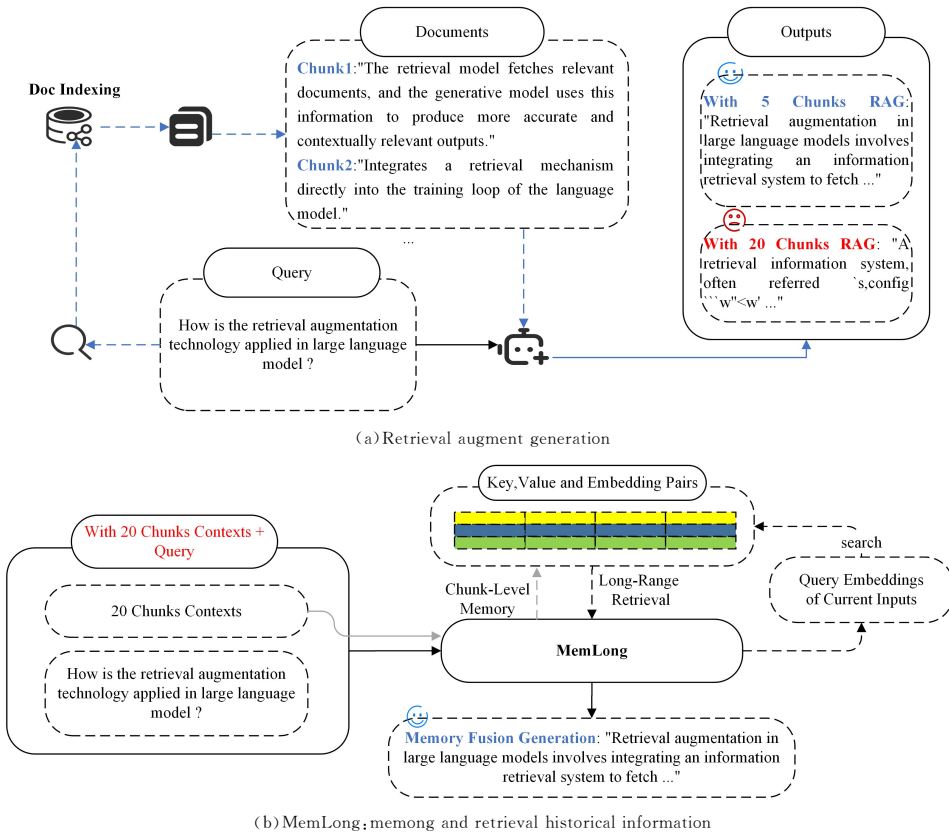


图 1 检索增强生成(RAG)与 MemLong 的记忆-检索流过程

Fig. 1 Illustration of Retrieval-Augment Generation and Memory-Retrieval flow of MemLong

2 准备工作

2.1 任务定义

语言模型旨在定义标记序列的概率分布,有效地预测给定语言中序列的可能性。给定一个序列 x_1, \dots, x_n , 将一个序

列的联合概率分解为一系列下一个令牌的预测问题,即计算:

$$p(x_1, \dots, x_n) = \prod_{i=0}^n p_{\theta}(x_i | x_{<i}), \text{ 其中 } x_{<i} := x_1, \dots, x_n \text{ 是第 } x_i \text{ 个}$$

令牌前的标记序列。与标准语言建模目标不同,本文不仅使用当前上下文进行下一个标记预测,还利用外部检索获取相

关信息并在模型的较高层进行知识融合。具体来说,给定一个由 l 个标记组成的序列和每个块的大小 τ ,将其划分为 $\gamma = \frac{l}{\tau}$ 个非重叠的块,表示为 $C = (c_1, \dots, c_\gamma)$ 。相应地,其文本形式被划分为 γ 个文本块,表示为 $T = (t_1, \dots, t_\gamma)$ 。在每一步中,在底层对每个块 c_i 进行因果语言建模;在上层,则对 t_i 进行细粒度可控检索以融合额外信息。之后,语言建模目标变为:

$$p(x_1, \dots, x_n) = \prod_{i=0}^n p_{\theta}(x_i | R(t_i), x_{<i}) \quad (1)$$

其中, $R(t_i)$ 表示相邻的检索 x_i 所在的 t_i 块。

2.2 模块和操作定义

如图 2 所示, Ret-Mem 模块包括一个用于信息交换的

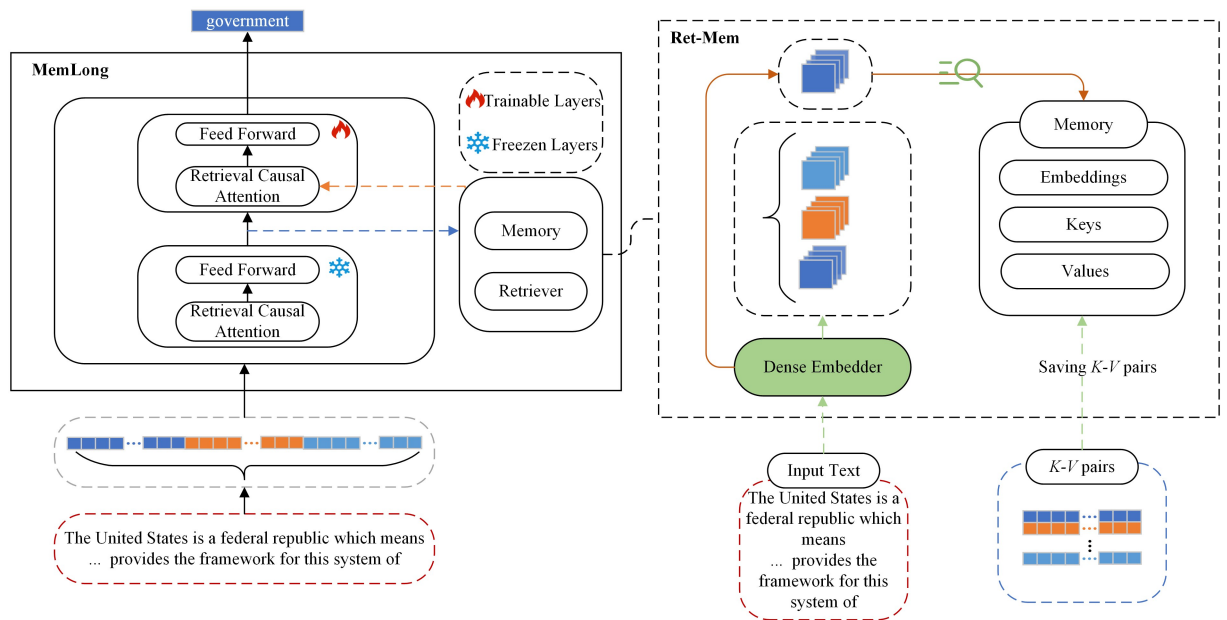


图 2 MemLong 标准工作流程(电子版为彩图)

Fig.2 Standard workflow of MemLong

3 MemLong

3.1 总览

如图 2 所示,每一步都需要输入一个文本块 c_i ,该文本块的原始文本为 t_i 。在被冻结住的模型底层中,标准的因果注意机制将应用于整个文本块 c_i 。模型底层的最后一层称作记忆层,每次向量化的数据经过记忆层之后,都会执行下面两个关键操作。第一个操作是检索,用红线表示,利用 t_i 来获取最相关的 $\mathbf{K-V}$ 对。在模型的上层中,检索到的 $\mathbf{K-V}$ 对与当前输入上下文进行整合,在训练阶段,训练模型参数对齐检索偏好。随后的章节将探讨 MemLong 框架的各个流程及其复杂性,包括检索和动态内存管理、注意力重塑,以及 MemLong 的推理过程。

3.2 检索和动态记忆规划

本节将对检索机制和动态记忆规划策略进行全面而系统的阐述。

3.2.1 检索过程

检索流程如图 2 中红线所示,为了实现用显示检索取代基于 $\mathbf{K-V}$ 键值对的传统 kNN 检索的目标,计划在每次输入

Retriever 和一个 Memory 组件。首先,将内存组件定义为 \mathbf{M} ,将检索器定义为 \mathbf{R} ,以及它们对应的操作 $\mathbf{M}(\cdot)$ 和 $\mathbf{R}(\cdot)$ 。此外,模型的维度设定为 d_{model} ,检索器的维度设定为 $d_{\text{retrieval}}$ 。记忆模块包括两个部分: $\mathbf{K-V}$ 对和相应的文本表征向量。键和值的维度都表示为 $R^{d_{\text{model}}}$,文本表征向量的维度表示为 $d_{\text{retriever}}$ 。值得强调的是,实际参与检索的是块级 (Chunk-level) 的检索表征,而不是 $\mathbf{K-V}$ 对。检索器本质上是一个预训练完备的稠密嵌入模型,具备优秀的文本表征能力。MemLong 利用该检索器,将每个文本块编码为文本表征向量 (Text Representation Embeddings)。由于它能为一个文本块生成一维文本表征向量,因此即使文本长度很长,也能保持极小的显存占用。

MemLong 前,尽可能地预先获取所需的信息。

具体来说,对于每个候选查询块 $c^q = c_i$ 及其对应的文本块 $t^q = t_i$,首先将 c^q 和 t^q 输入检索器,由此获得一个文本表征向量 $r^q = \mathbf{R}(t^q)$ 。随后,利用这个文本表征向量 r^q ,在记忆库 \mathbf{M} 中检索已存储的嵌入向量,以此获取所需的 k 个块级索引。检索过程中,计算文本表征向量 r^q 与记忆库 \mathbf{M} 中存储的嵌入向量之间的余弦相似度。最终,得到了 c^q 的 top- k 个索引 $z^q = \text{Top}k\{\text{Cos}(r^q)\}$ 。由于块内数据具有连续性,可以将这些索引进一步地扩展,以便全面覆盖相关的检索范围。最后,根据这些索引从记忆器中检索出相应的 $\mathbf{K-V}$ 对,随后与模型上层 (检索层) 进行信息融合等操作。特别值得一提的是,本文为记忆库内置了一套计数器机制,专门用于追踪每个索引的检索频次。MemLong 将这些频率数据作为内存动态调整的核心依据,并基于此优先处理高频检索信息。

3.2.2 记忆过程

记忆过程同步存储来自记忆层的 $\mathbf{K-V}$ 对,以及之前为了支持检索而计算的文本表征向量。为了确保每个块内所有 $\mathbf{K-V}$ 对与其表征向量精确对应(见图 2 右侧蓝线),对于每个可能的记忆块 $c^m = c_i$ 及其对应的文本块 $t^m = t_i$,将记忆过程分

为两部分:1)详细阐述 **K-V** 对记忆过程;2)解释每个块的文本表征向量的记忆方法。首先,将 c^m 输入 MemLong,然后从记忆层获得输出。值得注意的是,由于底层在训练过程中被冻结,因此可以确保输出 **K-V** 对的分布保持一致。这种一致性对于避免分布偏移问题至关重要,正如之前的 MemTrm 等工作中所指出的。MemLong 的记忆效率很高,因为它仅需要存储检索所需的文本表征向量,从而避免了冗余信息。在所有数据块对的检索完成后,记忆操作 $\mathbf{M}(k, \gamma; r^m)$ 会同步更新记忆库中的键值对及其对应的向量表示。

3.2.3 动态记忆更新

本文采用一个计数器对记忆器进行动态更新。在实验过程中,当内存溢出时,保留与当前输入内容时间上最近的 10%,因为最近信息具备潜在相关性;同时丢弃最旧的 10%,因为最旧信息可能已过时;对于中间的 80%,则依据检索频率进行优先处理,删除访问频率最小的记忆单元,直至记忆容量降至 50%。这种选择性过滤策略有效平衡了检索频率与相关性,既保留了有价值的信息,又去除了不太相关的数据。与传统的先进先出(First in First out, FIFO)策略相比,本文方法更注重检索频率,以高效剪除冗余信息,从而维持高质量的数据集。动态更新数据存储的决策,实质上是在有效性与效率之间进行权衡。对于需要长期依赖关系的任务,存储全部信息能提升综合处理能力;但对于短期任务,动态更新更为适宜。动态更新不仅能控制记忆空间大小,避免显存不足问题,还能丢弃过时信息,降低检索开销,在确保效率的同时,不会显著影响性能。此外,动态更新机制还能够实现数据的实时优化,确保存储的信息始终与当前任务需求保持高度相关。通过不断剔除无关紧要的旧数据,能够为新信息的加入腾出空间,从而确保记忆器能够持续捕捉并反映最新的数据趋势。

3.3 注意力重塑

为了能够将模型与检索偏好对齐,本文解冻了模型的上层,同时修改了注意力机制以融合长期记忆。

与传统 Transformer 解码层采用的多头注意力机制不同,本文提出了一种检索因果注意力,以扩展其功能,构建一个联合注意力机制,如图 3 所示。为了更有效地处理上下文信息,局部因果注意力(如图右侧所示)被应用于模型的底层。它专注于建模最近的局部上下文,确保模型处理当前输入时能够关注到紧邻的上文信息,并维持因果建模的特性,即仅允许模型关注过去的信息以避免信息泄露。与之相对,通过检索方法获得的块级 **K-V** 对(如图左侧所示),由于其历史性质,可以实现双向注意力而不存在信息泄露。此外,还设计了一种长期记忆融合过程,确保每个标记能够同时关注局部上下文以及具备完整连续语义的块级过去上下文。从前一层的头向隐藏状态输出 $\mathbf{H}^{l-1} \in R^{|\mathcal{X}| \times d_{\text{model}}}$ 和相应的检索到的键值对 $\mathbf{Z}_i^k = \{K_i, V_i\}_{i=1}^w \in R^{k \times \tau \times d_{\text{model}}}$,下一层的输出隐藏状态 \mathbf{H}^l 的计算式如下:

$$\mathbf{S}_a = \text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{d}\right) \quad (2)$$

$$\mathbf{S}_m = \text{Conc}\{\text{Softmax}(\mathbf{Z}_i^k)\}_{i=1}^w$$

为了避免在训练初期检索注意力分数 \mathbf{S}_m 导致训练不稳定,本文采用了一种遵循 LLaMA-adapter^[17] 的多头过滤注意

力机制方法:

$$\mathbf{S}_i^f = [(\mathbf{S}_m) \cdot g_i; (\mathbf{S}_a)]^\top \quad (3)$$

其中, g_i 是一个 0 到 1 的常数,用于平衡检索与当前输入的信息的比值。最后,将 \mathbf{V}_a 和 \mathbf{V} 连接起来以获得 \mathbf{H}^l :

$$\mathbf{V}_l = [\mathbf{V}_a; \mathbf{V}], \mathbf{H}^l = \mathbf{S}_i^f \mathbf{V}_l \quad (4)$$

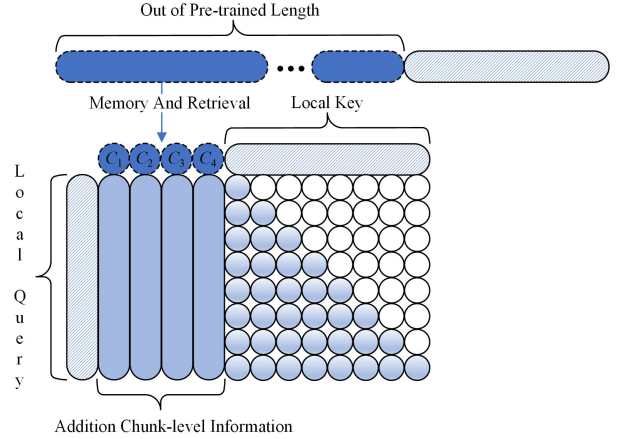


图 3 检索因果注意力

Fig. 3 Illustration of retrieval causal attention

3.4 推理过程

当 MemLong 遇到超过预训练长度的输入时,将输入分为两个部分:前缀输入和当前输入。下文将详细介绍推理阶段中长输入的编码生成过程。

首先,当 MemLong 处理较长的输入时,它会将前缀划分为多个不重叠的分块,并从其记忆层中计算出相应的信息。这一步骤确保了注意力机制所涉及的标记数等于分块大小,而分块大小远小于输入的总长度。值得注意的是,每个语块之间是相互关联的,例如,第 t 个块需要处理之前的 $t-1$ 个块。

接下来,根据块级表征向量,为当前输入选择最相关的 k 个块,并获取它们对应的 **K-V** 对所在的表征向量。对于上层检索注意力而言,检索的注意力窗口大小为 $k \times \tau$,这一窗口大小同样小于输入长度。通过限制长度的因果注意力机制与检索注意力机制,MemLong 能够有效地延长上下文窗口。最终,MemLong 将这些选定的块级表征向量对应的 **K-V** 对与当前输入部分一起输入 LLM。LLM 通过结合这些来自记忆的分块信息和当前输入部分的信息,生成每一个时间步的输出。通过这种方式,即使面对极长的输入,MemLong 也能够有效地捕捉到上下文信息,并生成连贯的输出。

4 实验

我们对提出的 MemLong 模型在需要长上下文处理的各项任务上进行评估:1)长上下文语言建模和检索增强的语言建模;2)可扩展的上下文学习。

4.1 实现细节

4.1.1 训练细节

实验中,使用 OpenLLaMA-3B 作为预训练的 backbone LLM,并采用了旋转位置编码^[18]。由于硬件限制,选择使用 LoRA^[19] 技术训练模型。具体而言,对于每一个非冻结层的 QKV 全连接层应用 LoRA,以缓解显存压力。本文将秩

(Rank)设置成 8, 缩放因子(Alpha)设置成 16, Dropout 设置成 0.1. backbone LLM 采用($L=26, H=32, d=100$)架构。除非另有说明,将第 13 层作为记忆层,将[14, 18, 22, 26]层作为检索增强层。对于检索增强调整的训练,本文只在 0.5B 个长度为 1024 序列的词元上进行迭代。MemLong 的可训练层数为第 14 到第 26 层,使用由 LLaMA-80k 采样的 slim-pajama 数据集^[20]作为训练语料。

4.1.2 位置编码重映射

在生成过程中,需要在记忆库 M 中检索多个块级的 $K-V$ 对。由于每一步检索的不确定性,需要将位置嵌入重新映射到检索到的数据块上。与 Focus Transformer^[21]一样,局部上下文(最多 2048 个词元)使用标准的旋转位置编码,而记忆键值对的位置信息则赋值为 0,这样它们就像位于本地上下文窗口的初始位置一样。

4.2 长文本语言建模

首先在长文本语言建模基准上对 MemLong 进行了基本的语言建模能力的评估,由于 $K-V$ 记忆提供了重要的背景和语境信息,MemLong 可以快速检索相关的 $K-V$ 记忆并充分利用它,从而增强了模型在长文本建模任务中的能力。

4.2.1 数据集

对各类模型在 4 个广泛的文本基准数据集上进行了全面评估,包括英文书籍数据集 PG-19^[22] 和 BookCorpus^[23]、维基百科文章数据集 Wikitext-103^[24],以及数学论文数据集 Proof-Pile^[25]。

4.2.2 实验设置

与 cepe^[26]一样,本文也计算每轮输入序列最后 2048 个词元的困惑度。这一实验设置旨在验证不同检索器大小对本文模型整体性能的影响。为了实现高效的细粒度检索,使用 faiss^[27]工具包在 GPU 上构建精确搜索索引,以记忆文本表征向量并执行高效检索。对于 MemLong,将微调长度超过 1024 上的 tokens 拆分并放入记忆单元中用于进一步的检索。

4.2.3 基准

本文实验使用 OpenLLaMA-3B 模型作为基准。为了确保比较的公平性,使用了与之前提到的相同的 LoRA 配置,

即对每个非冻结层的 QKV 全连接层应用 LORA。此外,还与 LongLLaMA-3B 进行了比较,该模型使用 Focused Transformer(FoT)方法和 5B tokens 进行了微调。为了进行更全面的比较,本文额外测试了两个 7B 模型 LLaMA-2-7B 和 LongLoRA-7B-32K^[28],以及两个位置编码模型 Yarn-7b-128k^[29] 和 Phi3-128k^[30]。此外,还对比了近期的 Self-RAG 检索框架,其引入反思机制来增强生成文本的可靠性。

4.2.4 结果

如表 1 所列,采用困惑度(Perplexity, PPL)作为语言模型的评估指标。PPL 越低,说明语言建模能力越强。所有实验均在一个 3090 GPU 上运行。表中标有 * 的 LongLLaMA-3B 和 MemLong-3B 表示在无内存的情况下评估,标有 † 的 LongLLaMA-3B 表示在无记忆的情况下进行评估。我们还在 4k/32k 内存情况下对 MemLong 进行了评估,“-/9.65”表示该模型在单 GPU 上会导致内存不足(OOM)错误。实验结果显示,在所有数据集上,与两个全量微调的模型 OpenLLaMA-3B 与 LLaMA-2-7B 相比,本文模型均实现了显著的困惑度下降。本文模型在从 1024 到 32768 个词元的长度范围内进行了严格测试。在所有数据集上,MemLong 通过有效利用外部检索器和记忆模块,在极低的显存开销下,展现了显著的性能提升。当测试长度在预训练范围内时,本文模型在多个数据集上表现持平。而一旦超出预训练长度,可以看到,尽管超出了 1024 的微调长度与 2048 的预训练长度,本文模型仍能够继续降低 ppl,这说明本文模型是可泛化的。相比之下,OpenLLaMA-3B 和 LLaMA-2-7B 的模型无法泛化超出预训练长度的输入;且由于 attention 的二次时空复杂度,显存开销显著增长。尽管 LongLoRA 提出的 Shifted Sparse Attention 能显著降低显存使用,却也降低了模型在短文本场景下的能力。而与本文方法类似的 LongLLaMA,其记忆空间是无限增长的,当测试长度过长时,会导致显存溢出(Out of memory, OOM)。基于位置编码的长文本模型具备很强的泛化性,但是这类方法的性能只能保证长文本生成性能不下降。然而,MemLong 能在利用更长输入 tokens 的同时获得更好的困惑度的下降,并且能控制记忆空间的大小避免显存不足。

表 1 不同上下文窗口扩展模型在 PG-19, Proof-pile, BookCorpus 和 Wikitext-103 上的滑动窗口困惑度

Table 1 Sliding window perplexity of different context window extension models on PG-19, Proof-pile, BookCorpus, Wikitext-103

Model	PG-19				Proof-Pile				BookCorpus				Wikitext-103				
	1000	2000	4000	16000	1000	2000	4000	16000	1000	2000	4000	16000	1000	2000	4000	16000	
7B Model	LLaMA-2-7B	10.82	10.06	8.92	—	3.24	3.40	2.72	—	8.73	7.91	6.99	—	10.82	6.49	5.66	—
	LongLoRA-7B-32k	9.76	9.71	10.37	7.62	3.68	3.35	3.23	2.60	14.99	12.66	11.66	6.93	7.99	7.83	8.39	5.47
	YARN-128k-7b	7.22	7.47	7.17	—	3.03	3.29	2.98	—	7.02	7.54	7.06	—	5.71	6.11	5.71	—
3B Model	OpenLLaMA-3B	11.60	9.77	>1000	—	2.96	2.70	>1000	—	8.97	8.77	>1000	—	10.57	8.08	>1000	—
	LongLLaMA-3B*	10.59	10.20	>1000	—	3.55	3.15	>1000	—	10.70	9.85	>1000	—	8.88	8.07	>1000	—
	LongLLaMA-3B†	10.59	10.25	9.87	—	3.55	3.22	2.94	—	10.14	9.62	9.57	—	10.69	8.33	7.84	—
	Phi3-128k	11.31	9.90	9.66	-/9.65	4.25	3.11	2.77	-/3.08	11.01	9.22	8.98	-/9.27	7.54	7.22	7.01	-/7.20
	Self-RAG	12.85	10.01	>1000	—	3.07	2.90	>100	—	9.99	9.38	>1000	—	9.47	9.03	>1000	—
	MemLong-3B*	10.66	10.09	>1000	—	3.58	3.18	>1000	—	10.37	9.55	>1000	—	8.72	7.93	>1000	—
w/ 4K Memory	10.54	9.95	9.89	9.64	3.53	3.16	3.15	2.99	10.18	9.50	9.57	9.61	8.53	7.92	7.87	7.99	
w/32K Memory	10.53	9.85	9.83	9.73	3.51	3.15	3.11	2.99	9.64	9.56	9.51	9.54	8.02	7.58	6.89	7.90	

4.3 长上下文学习

传统的上下文学习^[31](In-context learning, ICL)通常将少量非参数化的示例与查询一同输入模型。然而,这类方法

普遍受到模型输入长度的制约。在本实验中,鉴于 MemLong 能够在记忆中存储参数化的示例,重点探讨 MemLong 是否能够有效利用其记忆中存储的知识,以增强其涌现能力。实

验结果如表 2 所列。与完全依赖非参数知识的 OpenLLaMA 相比,在上下文演示数量相同的情况下,MemLong 可以利用存储在记忆中的更多演示。记忆中的演示数越多,性能就会进一步提高或保持一致。在与 LongLLaMA 的对比分析中发现,在保留记忆中演示的相同条件下,MemLong 在大多数

数据集上都优于 LongLLaMA。同时需要强调的是,与 LongLLaMA 相比,MemLong 的训练参数(200 M vs. 0.3 B)和微调数据量(0.5 B vs. 5 B)都要低得多。这同样印证了本文模型在利用外部检索器获取信息方面的效率,证明了它在大幅减少各类资源的情况下仍能有效综合和利用知识的能力。

表 2 在 4-shot 和 20-shot 条件下的 5 个自然语言理解任务中的 ICL 精确度

Table 2 Accuracy of 4-shot and 20-shot ICL on 5 NLU tasks

Model	In-Context # Demons.	In-Memory # Demons.						Avg.	
			SST-2 ACC ↑	MR ACC ↑	Subj ACC ↑	SST-5 ACC ↑	MPQA ACC ↑	(%)	
OpenLLaMA	4	—	90.7	84.0	58.2	41.0	70.5	68.9	
w. / RAG	4	4	90.9	90.5	61.6	39.2	64.2	69.1	
LongLLaMA	4	4	90.4	83.9	64.3	40.0	64.2	68.6	
MemLong	4	4	91.5	84.5	61.5	41.4	70.2	69.8	
LongLLaMA	4	18	91.4	87.1	59.1	41.0	64.5	68.7	
MemLong	4	18	91.0	89.6	61.7	43.5	69.4	71.0	
OpenLLaMA	20	—	93.6	91.2	55.4	38.2	66.4	69.0	
w. / RAG	20	18	92.2	91.3	75.8	39.8	57.6	71.3	
LongLLaMA	20	18	94.1	90.8	64.2	41.4	72.1	72.7	
MemLong	20	18	93.5	93.8	65.8	43.3	70.6	73.4	

5 消融实验

5.1 训练参数的设置对模型性能的影响

消融实验 1 检索层的设置对模型性能的影响

为探究检索层设置对模型性能的影响,进一步评估了本文方法在缓解 MemTrm 中所述分布偏移问题上的有效性。如前所述,本文旨在为分布偏移提供一种低成本的解决方案。如图 4 所示,与基于 MemLong 的多种不同训练策略相比,棕色曲线(图 4 顶部,代表一种基线训练策略,该策略复现了 MemTrm 的微调方式,即训练所有模型参数,并允许记忆层之后的所有层参数检索)在性能和收敛速度上均表现出显著的劣势,甚至逊于本文所采用的最简化的设置。后续章节将详细分析推理阶段的性能表现。

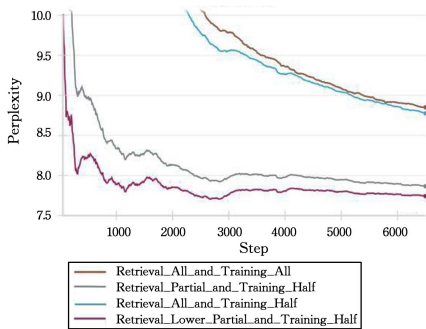


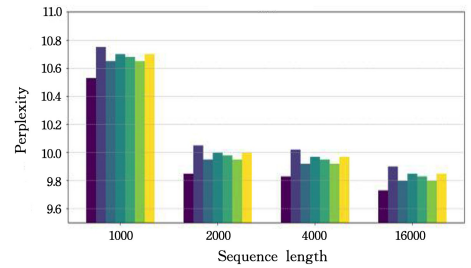
图 4 训练阶段的 PPL 情况(电子版为彩图)

Fig. 4 Degree of PPL during the training phase

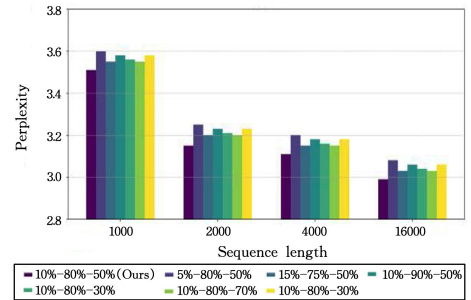
消融实验 2 不同比例的动态更新策略对模型性能的影响

在不同序列长度下,动态记忆更新策略(10%-80%-50%)的设置,即 10%的过时数据删除,80%的中间数据,依照计数器来删除其中 50%的不常用数据。如图 5 所示,本文提出的策略与多种参数变体的初步对比结果表明,动态记忆策略及其参数的选取对于最终效果至关重要。经验性认为,过时数据以及不常用数据的删除能够解放庞大的记忆库的

冗余性,并且一定程度上帮助提升检索效率。



(a) PG-19



(b) Proof-Pile

图 5 不同策略下 PG19 和 Proof-Pile 数据集上的 PPL

Fig. 5 PPL comparison of different strategies on PG19 and

Proof-Pile datasets

5.2 推理参数的设置对模型性能的影响

消融实验 3 记忆长度对模型性能的影响

对同一模型在不同记忆容量大小下的性能进行了研究,结果如图 6 所示,记忆容量与模型效率之间存在明显的相关性。这一趋势表明,记忆容量的增加会逐步提高性能。此外,在记忆容量为 65536 时会出现一个临界点,超过这个临界点,模型的性能就难以大幅跃升。这表明,虽然扩大记忆容量可以带来巨大的好处,但其有效性存在一个实际上限,这可能是受到数据分布的细微差别的影响。

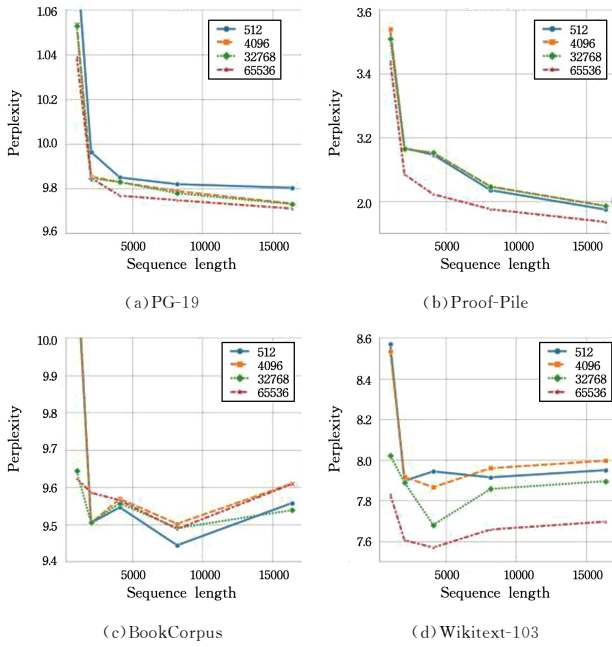


图6 评估不同数据集在不同记忆大小情况下的性能
Fig. 6 Evaluating different datasets at various memory sizes

消融实验4 检索增强记忆层数对模型性能的影响

如图4(粉红色线)和表3(RLP+TH)所示,检索层设置为较少时性能最佳,具体在实验中 MemLong 将检索层设置为[13,17,21,25]性能达到最优。经验表明,如果将检索信息引入模型的所有上层,会导致模型对局部上下文的关注度下降。因此,在适当的时间间隔内选择检索层,实际上可以增强模型的能力。

表3 不同的检索层对 MemLong 性能的影响

Table 3 Impact of different retrieval layers on MemLong performance

Method	PG19			Proof-Pile		
	2000	4000	8000	2000	4000	8000
MemLong*	10.09	>1000	—	3.18	>1000	—
w. RA+TA	11.43	11.40	10.36	3.51	3.26	3.14
w. RA+TH	10.57	10.48	10.36	3.30	3.26	3.15
w. RP+TH	10.28	10.15	10.12	3.21	3.13	3.08
w. RLP+TH	9.85	9.83	9.80	3.21	3.13	3.08

注:标记为*的 MemLong 表示不使用记忆进行评估。使用记忆的所有方法的大小设置为 32768;RA 表示跨所有上层进行检索;TA 表示训练所有参数而不冻结;RP 表示跨较少的上层进行检索;RLP 表示跨非常少的上层进行检索。

6 相关工作

6.1 长上下文语言建模

长上下文语言模型主要聚焦于长度扩展和上下文窗口扩展两个方面。长度扩展研究通常以广受欢迎的 RoPE 编码为研究对象,旨在将超过预训练长度的输入映射至预训练期间所涵盖的位置空间中。这些工作(如 RoPE, Alibi^[32], Pi, Yarn)使模型在推理过程中能够泛化至未见过的的位置编码,从而实现超越训练过程中所遇长度的外推能力。相比之下,本文方法无需对 PE 进行修改,仅需引入一个附加模块即可实现上下文的扩展。

上下文窗口扩展的核心问题在于,如何增加 LLM 单次

处理输入的上下文窗口大小。由于计算注意力的时间和空间复杂度均为二次方,扩展语言模型的输入长度是一项极具挑战性的任务。尽管稀疏注意力技术(如 Reformer, LongLoRA, Focus Transformer, Unlimiformer, Longformer)已取得显著进展,但本文的研究重点是通过检索增强方法,使 LLM 在更短的输入长度上高效获取相关信息,从而优化长距离语言建模的效果。

6.2 检索增强语言建模

在检索增强语言建模(Retrieval-Augmented Language Modeling)领域,研究者付出了巨大努力,提出了多种方法,如 RAG, FID^[34], In-Context RALM^[35], GenRead^[36] 和 Self-RAG^[37]。尽管这些方法中有一部分借助了外部检索器,但与修改模型内部参数的方法相比,非参数化的信息通常难以直接融合,而 MemLong 专注于将检索机制直接融入模型架构。REALM^[38] 认为单纯依赖内部模型知识是低效的,主张让模型学会检索和理解。kNN-LM^[39] 通过结合 LLM 的下一词预测与基于检索的预测,有效提升了语言建模能力。MemTrm 引入了记忆库概念,但由于参数调整,记忆分布可能发生变动。LongMEM 通过训练子网络来缓解这一问题,但随之带来了显著的计算开销。相较之下,本文方法采用固定的预训练模型,并通过与模型内部检索过程相一致的冻结检索器进行增强,从而避免了记忆分布和架构的变动。

结束语 本文提出了 MemLong——一种全新的利用外部检索器增强语言模型长文本能力的方法。利用一个具备完备检索能力的检索器,MemLong 能在仅少量显存代价的前提下快速获取远距离相关文本。长距离的 $K-V$ 外挂,使得 MemLong 能有效拓展模型的上下文窗口从 2000 到 80000。实验表明,MemLong 在长距离的建模与理解任务上表现出相当的竞争力,在上下文学习任务中,MemLong 对比全上下文的 OpenLLaMA 最多能提升 10.4 个百分点。然而,本文工作主要集中在 OpenLLaMA-3B 上,未来将探索和研究本文方法在各种规模模型中的应用。同时还发现,虽然单层 $K-V$ 对可以为上层提供额外的语义信息,但这种信息并不稳定。未来将提供一个更合理的框架来适应本文方法。此外,本文采用的检索器具有固定的 FlagEmbeddings,但研究更大范围的检索器将是有益的。

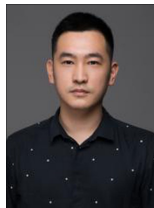
参考文献

- [1] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[C]// Proceedings of the 31st International Conference on Neural Information Processing Systems. 2017: 6000-6010.
- [2] KOH H Y, JU J X, LIU M, et al. An empirical survey on long document summarization: Datasets, models, and metrics [J]. ACM computing surveys, 2022, 55(8): 1-35.
- [3] WANG J, LEONG C T, WANG J S, et al. Instruct once, chat consistently in multiple rounds: An efficient tuning framework for dialogue[J]. arXiv:2402.06967, 2024.
- [4] BELTAGY I, PETERS M E, COHAN A. Longformer: The long-document transformer[J]. arXiv:2004.05150, 2020.
- [5] WANG S, LI B Z, KHABSA M, et al. Linformer: Self-attention

- with linear complexity[J]. arXiv:2006.04768,2020.
- [6] KITAIEV N, KAISER L, LEVSKAYA A. Reformer: The efficient transformer[J]. arXiv:2001.04451,2020.
- [7] XIAO G X, TIAN Y D, CHEN B D, et al. Efficient streaming language models with attention sinks[J]. arXiv:2309.17453,2023.
- [8] CHEN S Y, WONG S, CHEN L J, et al. Extending context window of large language models via positional interpolation[J]. arXiv:2306.15595,2023.
- [9] LU Y, ZHOU X, HE W, et al. Longheads: Multi-head attention is secretly a long context processor[J]. arXiv:2402.10685,2024.
- [10] DAI Z H, YANG Z L, YANG Y M, et al. Transformer-xl: Attentive language models beyond a fixed-length context[J]. arXiv:1901.02860,2019.
- [11] BERTSCH A, ALON U, NEUBIG G, et al. Unlimiformer: Long-range transformers with unlimited length input[C]// Advances in Neural Information Processing Systems. 2024.
- [12] YU H F, ZHANG Y, BI W, et al. Trams: Training-free memory selection for long-range language modeling[J]. arXiv:2310.15494,2023.
- [13] WU Y H, RABE M N, HUTCHINS D, et al. Memorizing transformers[J]. arXiv:2203.08913,2022.
- [14] WANG W Z, DONG L, CHENG H, et al. Augmenting language models with long-term memory[C]// Advances in Neural Information Processing Systems. 2024.
- [15] RUBIN O, BERANT J. Long-range language modeling with self-retrieval[J]. arXiv:2306.13421,2023.
- [16] TOUVRON H, LAVRIL T, IZACARD G, et al. Llama: Open and efficient foundation language models[J]. arXiv:2302.13971,2023.
- [17] ZHANG R R, HAN J M, LIU C, et al. Llama-adapter: Efficient fine-tuning of language models with zero-init attention[J]. arXiv:2303.16199,2023.
- [18] SU J L, AHMED M, LU Y, et al. Reformer: Enhanced transformer with rotary position embedding[J]. Neurocomputing, 2024,568:127063.
- [19] HU E J, SHEN Y L, WALLIS P, et al. Lora: Low-rank adaptation of large language models[J]. arXiv:2106.09685,2021.
- [20] FU Y, PANDA R, NIU X Y, et al. Data engineering for scaling language models to 128k context[J]. arXiv:2402.10171,2024.
- [21] TWOROKOWSKI S, STANISZEWSKI K, PACEK M, et al. Focused transformer: Contrastive training for context scaling[C]// Advances in Neural Information Processing Systems. 2024.
- [22] RAE J W, POTAPENKO A, JAYAKUMAR S M, et al. Compressive transformers for long-range sequence modelling[J]. arXiv:1911.05507,2019.
- [23] ZHU Y K, KIROUS R, ZEMEL R, et al. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books[C]// Proceedings of the IEEE International Conference on Computer Vision. 2015:19-27.
- [24] MERITY S, XIONG C M, BRADBURY J, et al. Pointer sentinel mixture models[J]. arXiv:1609.07843,2016.
- [25] AZERBAYEV Z, SCHOELKOPF H, PASTER K, et al. Llemma: An open language model for mathematics[J]. arXiv:2310.10631,2023.
- [26] YEN H, GAO T Y, CHEN D Q. Long-context language modeling with parallel context encoding[J]. arXiv:2402.16617,2024.
- [27] JOHNSON J, DOUZE M, JEEGOU H. Billion-scale similarity search withgpus[J]. IEEE Transactions on Big Data, 2019, 7(3):535-547.
- [28] CHEN Y K, QIAN S J, TANG H T, et al. Longlora: Efficient fine-tuning of long-context large language models[J]. arXiv:2309.12307,2023.
- [29] PENG B W, QUESNELLE J, FAN H L, et al. Yarn: Efficient context window extension of large language models[J]. arXiv:2309.00071,2023.
- [30] ABDIN M, JACOBS S A, AWAN A A, et al. Phi-3 technical report: A highly capable language model locally on your phone[J]. arXiv:2404.14219,2024.
- [31] BROWN T, MANN B, RYDER N, et al. Language models are few-shot learners[J]. Advances in neural information processing systems, 2020,33:1877-1901.
- [32] PRESS O, SMITH N A, LEWIS M. Train short, test long: Attention with linear biases enables input length extrapolation[J]. arXiv:2108.12409,2021.
- [33] LEWIS P, PEREZ E, PIKTUS A, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks[J]. Advances in Neural Information Processing Systems, 2020,33:9459-9474.
- [34] IZACARD G, GRAVE E. Leveraging passage retrieval with generative models for open domain question answering[J]. arXiv:2007.01282,2020.
- [35] RAM O, LEVINE Y, DALMEDIGOS I, et al. In-context retrieval-augmented language models[J]. Transactions of the Association for Computational Linguistics, 2023,11:1316-1331.
- [36] YU W H, ITER D, WANG S H, et al. Generate rather than retrieve: Large language models are strong context generators[J]. arXiv:2209.10063,2022.
- [37] ASAI A, WU Z Q, WANG Y Z, et al. Self-rag: Learning to retrieve, generate, and critique through self-reflection[J]. arXiv:2310.11511,2023.
- [38] GUU K, LEE K, TUNG Z, et al. Realm: Retrieval-augmented language model pre-training[J]. arXiv:2002.08909,2020.
- [39] KHANDELWAL U, LEVY O, JURAFSKY D, et al. Generalization through memorization: Nearest neighbor language models[J]. arXiv:1911.00172,2019.



LIU Weijie, born in 1999, postgraduate. His main research interests include RAG, long-context language model, NLP and LLMs.



LI Juntao, born in 1992, Ph.D, associate professor. His main research interest is natural language generation.