北京大学
PEKING UNIVERSITY

当大模型遇上长文本：可行路线探索与长上下文模型介绍

报告人：黄曲哲、陶铭绪、张晨

2023年12月2日

短文本 -> 长文本

问题

请根据以下文本回答，唐僧与猪八戒（猪刚鬣）在何处相遇？

短文本

却说三藏与那高老庄诸老谈今论古 … 行者道：“师父，那妖不是凡间的邪祟，也不是山间的怪兽。他本是天蓬元帅临凡，只因错投了胎，嘴脸象一个野猪模样，唤名猪刚鬣”

长文本

LLM

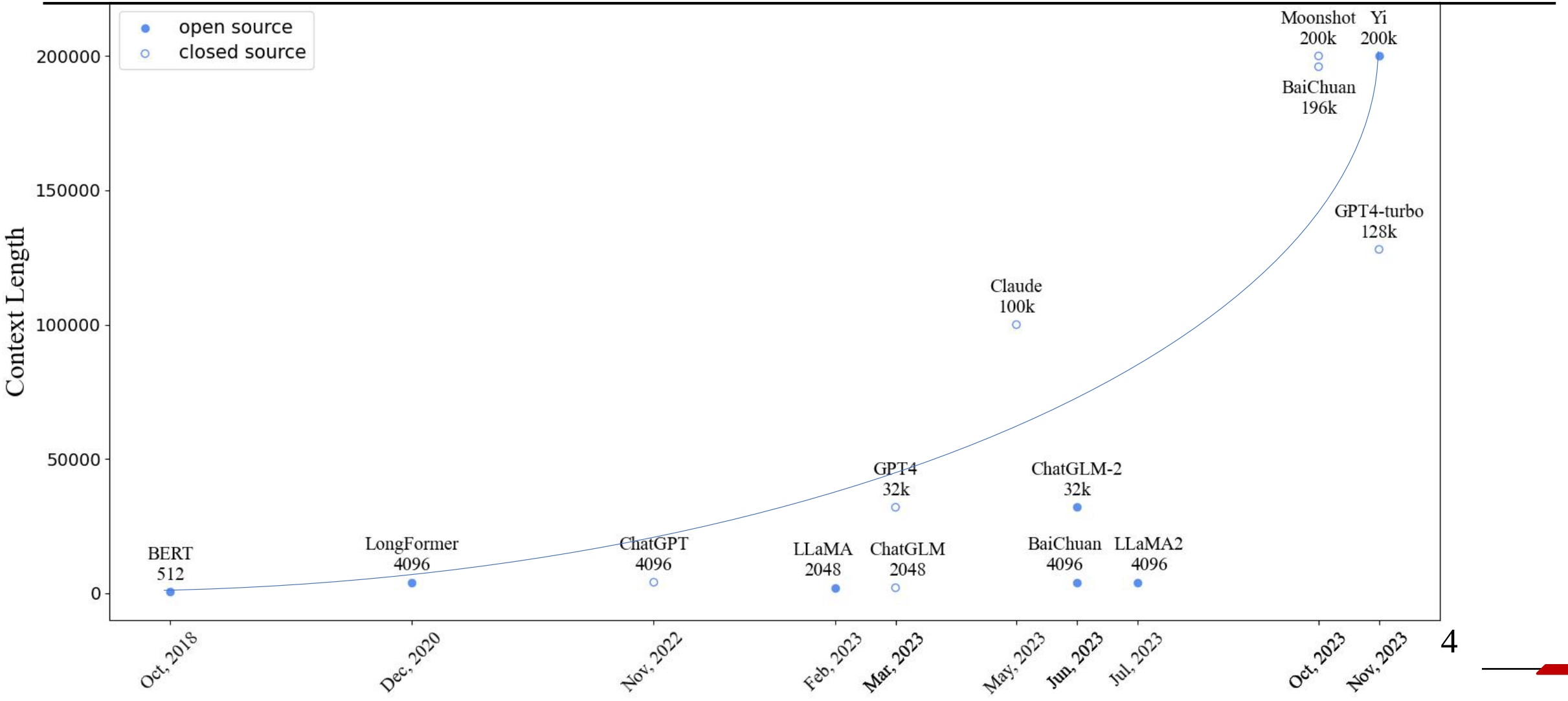
答案

高老庄

领域模型需要处理长文本

- 超长单文档：超过一百页的财报
- 多文档处理：相似案例整理

“长”文本有多“长”？


open source
closed source
Context Length
200000
150000
100000
50000
0
BERT
512
LongFormer
4096
ChatGPT
4096
LLaMA
2048
ChatGLM
2048
GPT4
32k
Claude
100k
ChatGLM-2
32k
BaiChuan
4096
LLaMA2
4096
Moonshot
200k
Yi
200k
BaiChuan
196k
GPT4-turbo
128k
Oct, 2018
Dec, 2020
Nov, 2022
Feb, 2023
Mar, 2023
May, 2023
Jun, 2023
Jul, 2023
Oct, 2023
Nov, 2023

直接训超长模型？

- 有限输入长度与长文本需求之间的不匹配
  - 开源模型上下文长度普遍在4K

上下文　: 2K -> 8K
训练速度: 降低20%

- 为什么不直接用超长的上下文训练 / 推理
  - Attention 复杂度和输入长度的平方成正比；显存开销大、训练时间长
  - 常见位置编码（RoPE）缺乏外延能力，难以处理超越训练长度的文本

如何处理长文本？

- 短文本训练，短文本推理 —— 将长文本变成短文本
  - e.g., Sliding Window (Xiao, 23), Retrieval Augmentation (Xu, 23)
- 长文本训练，长文本推理 —— 降低Attention的复杂度
  - e.g., Sparse Attention (Beltagy, 20)
- 短文本训练，长文本推理 —— 在文本长度上的泛化
  - e.g., Interpolation (Chen, 23)

目录

- 长上下文建模方法：
  - 短文本训练，短文本推理（黄曲哲）
  - 长文本训练，长文本推理（黄曲哲）
  - 短文本训练，长文本推理（陶铭绪）
- 长文本数据集与评测 （张晨）
- 总结与展望

目录

- 长上下文建模方法：
  - 短文本训练，短文本推理（黄曲哲）
  - 长文本训练，长文本推理（黄曲哲）
  - 短文本训练，长文本推理（陶铭绪）
- 长文本数据集与评测 （张晨）
- 总结与展望

滑动窗口

- 将长文本切成很多块，每块的长度可被模型接受
- 每次只处理一个窗口（块）内的词

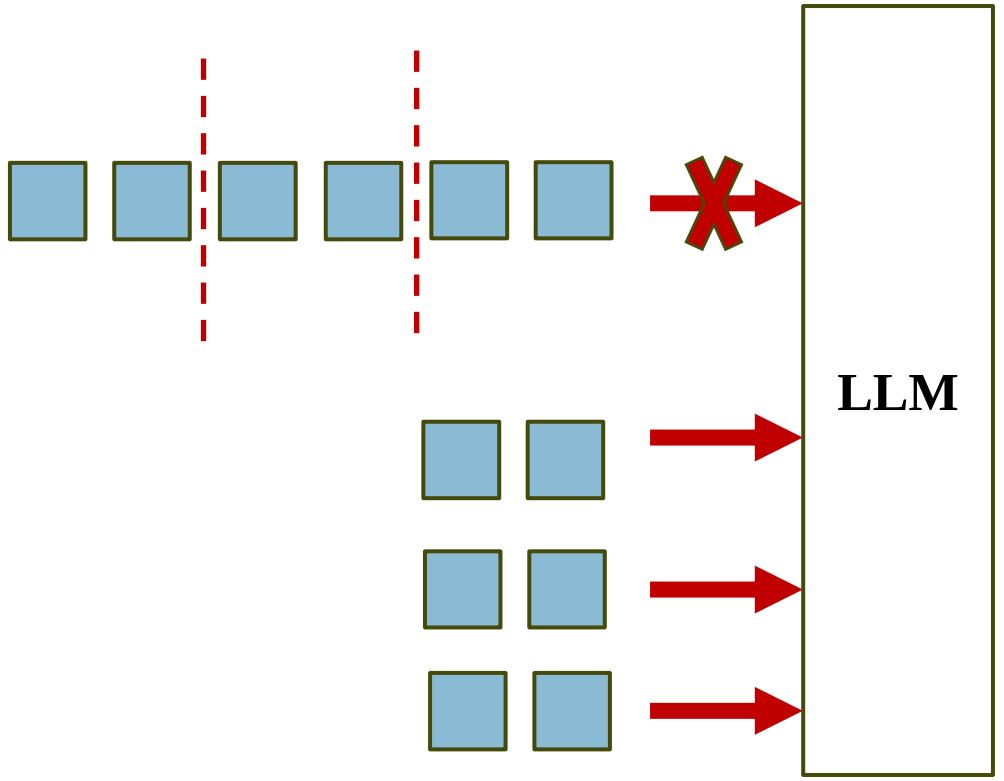
样例

原文：　成都有很多可爱的大熊猫。

窗口1：成都有很

窗口2:　　多可爱的

窗口3:　　大熊猫。



滑动窗口

- 将长文本切成很多块，每块的长度可被模型接受
- 每次只关注一个窗口（块）内的词

样例

原文：　成都有很多可爱的大熊猫。

窗口1：成都有很

窗口2:　　多可爱的

窗口3:　　大熊猫。

- 只能关注局部信息

哪有大熊猫？

滑动窗口＋重叠

- 文本块之间有重叠，通过重叠部分传递信息
- 每个token的表示计算后保留，用于后续计算
- 末尾 token 可以通过窗口内的其他token获取窗口外信息

样例

原文：　成都有很多可爱的大熊猫。

窗口1：成都有很多可爱

窗口2:　　很多可爱的大熊猫

Xiao et al. Efficient Streaming Language Models with Attention Sinks. 2023.

滑动窗口＋重叠

- 上下文太长时，Perplexity （PPL）爆炸，丧失语言建模能力

Xiao et al. Efficient Streaming Language Models with Attention Sinks. 2023.

滑动窗口 + 重叠 + First Token

- 上下文太长时，Perplexity （PPL）爆炸，丧失语言建模能力
- **Attention Sink：**Attention会对文本中的第一个词给予高关注
- 解决方案：在滑动窗口的同时，也加入第一个词

样例

原文： 成都有很多可爱的大熊猫。

窗口1：成都有很多可爱

窗口2: 成 很多可爱的大熊猫

Xiao et al. Efficient Streaming Language Models with Attention Sinks. 2023.

滑动窗口+摘要

- 切块，块之间无重叠
- 额外放入之前块的总结

样例

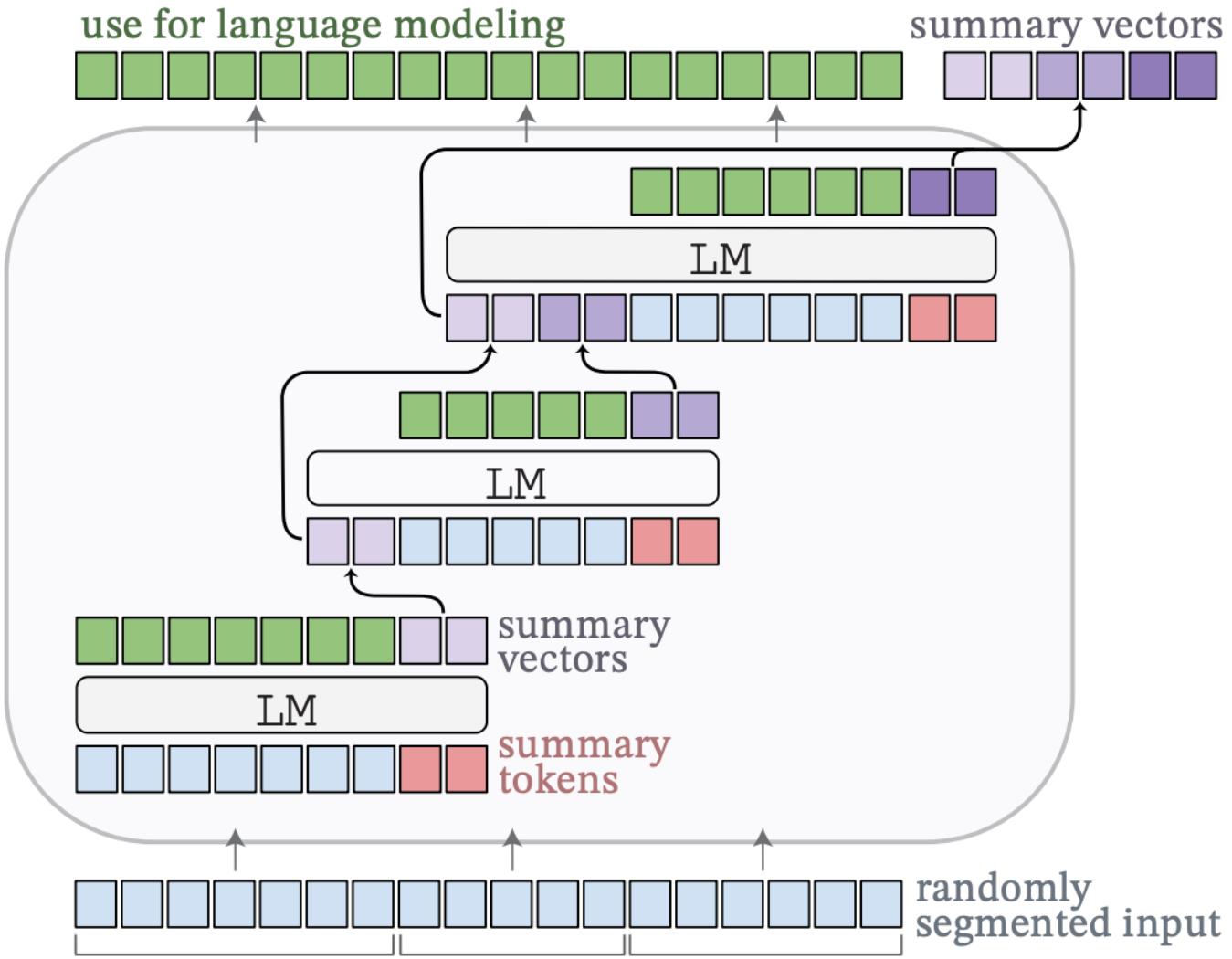
原文：　成都有很多可爱的大熊猫。

区块1：成都有很

区块2:　　[sum1]多可爱的

区块3:　　[sum1][sum2]大熊猫。

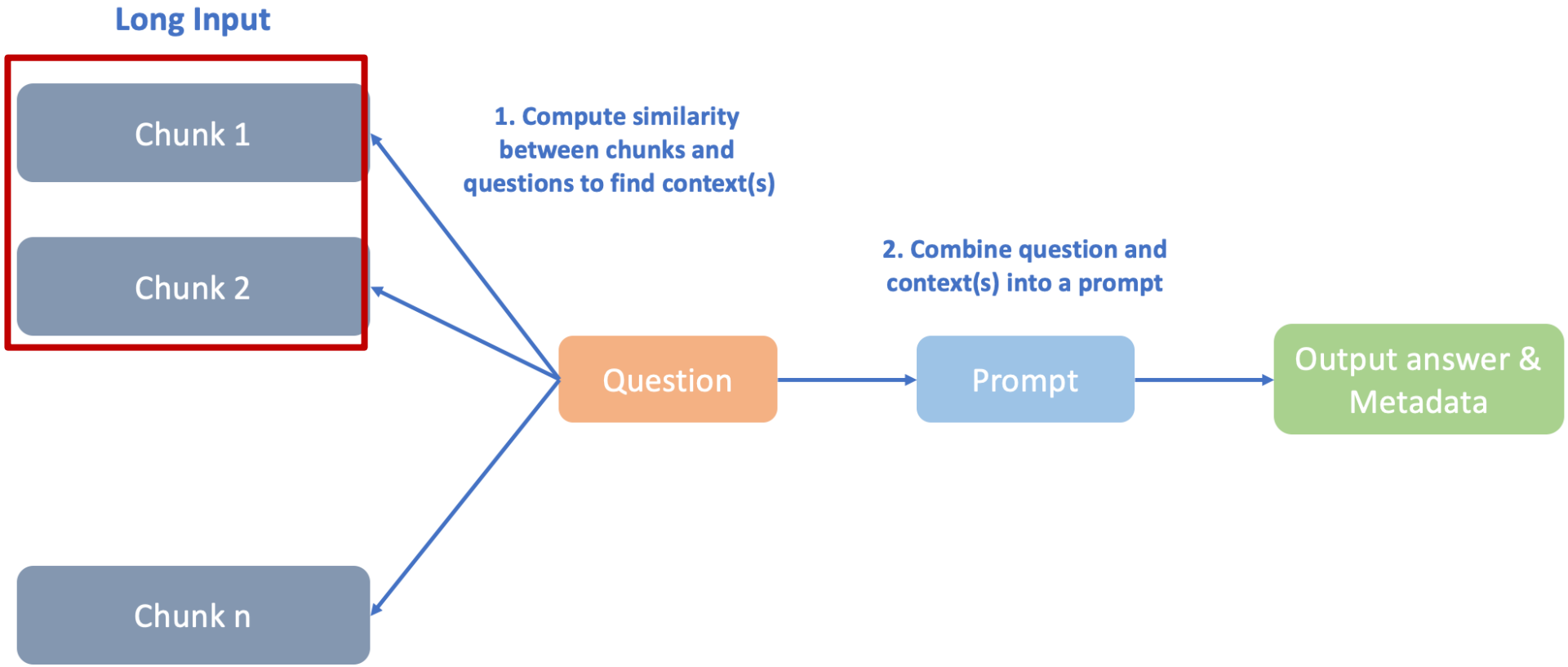
- 无法关注具体的词



Chevalier et al. Adapting Language Models to Compress Contexts. 2023

检索增强的方法

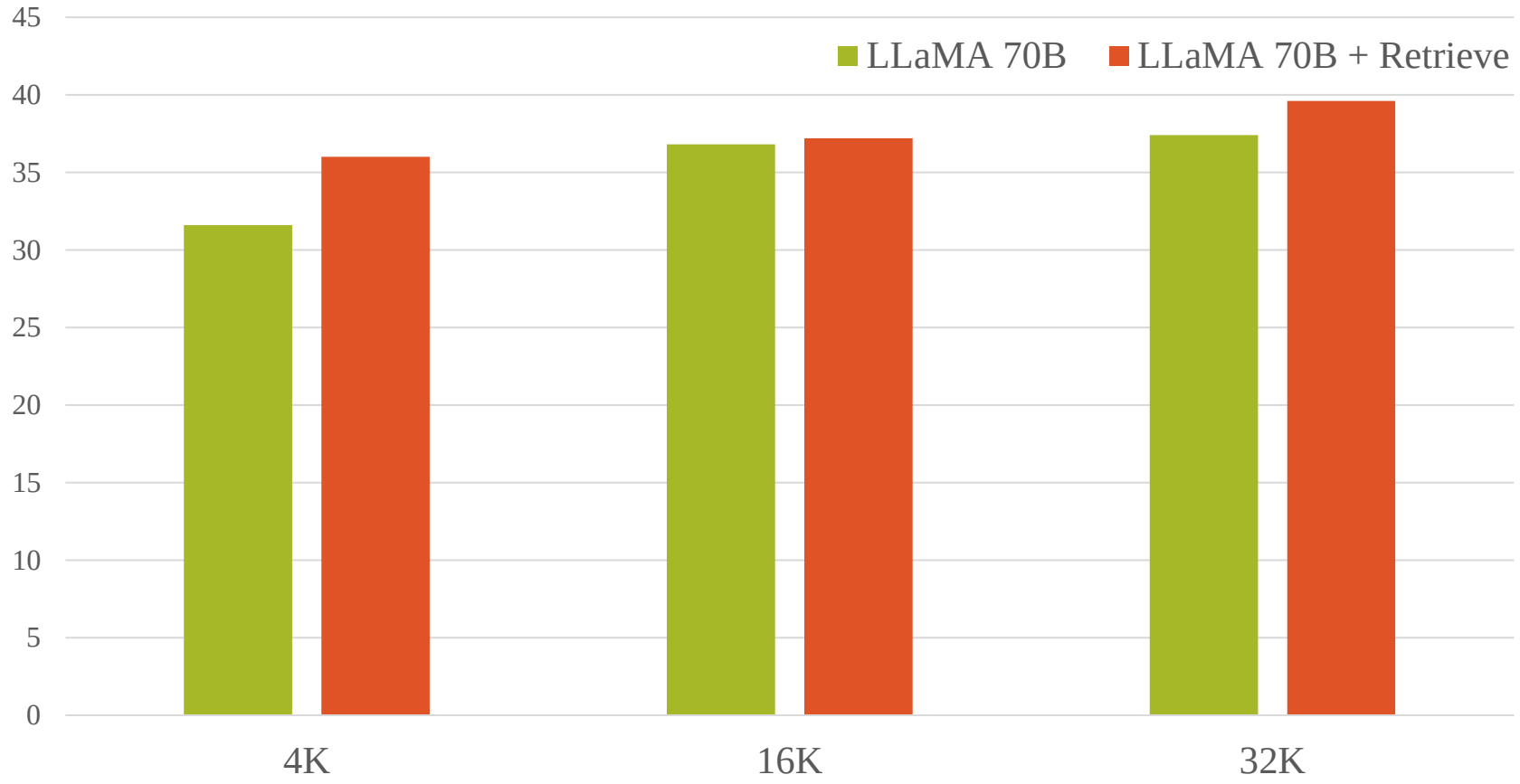
- 先从长文本中筛选重要信息，模型只处理筛选过的



Xu et al. Retrieval meets Long Context Large Language Models. 2023.

检索增强的方法

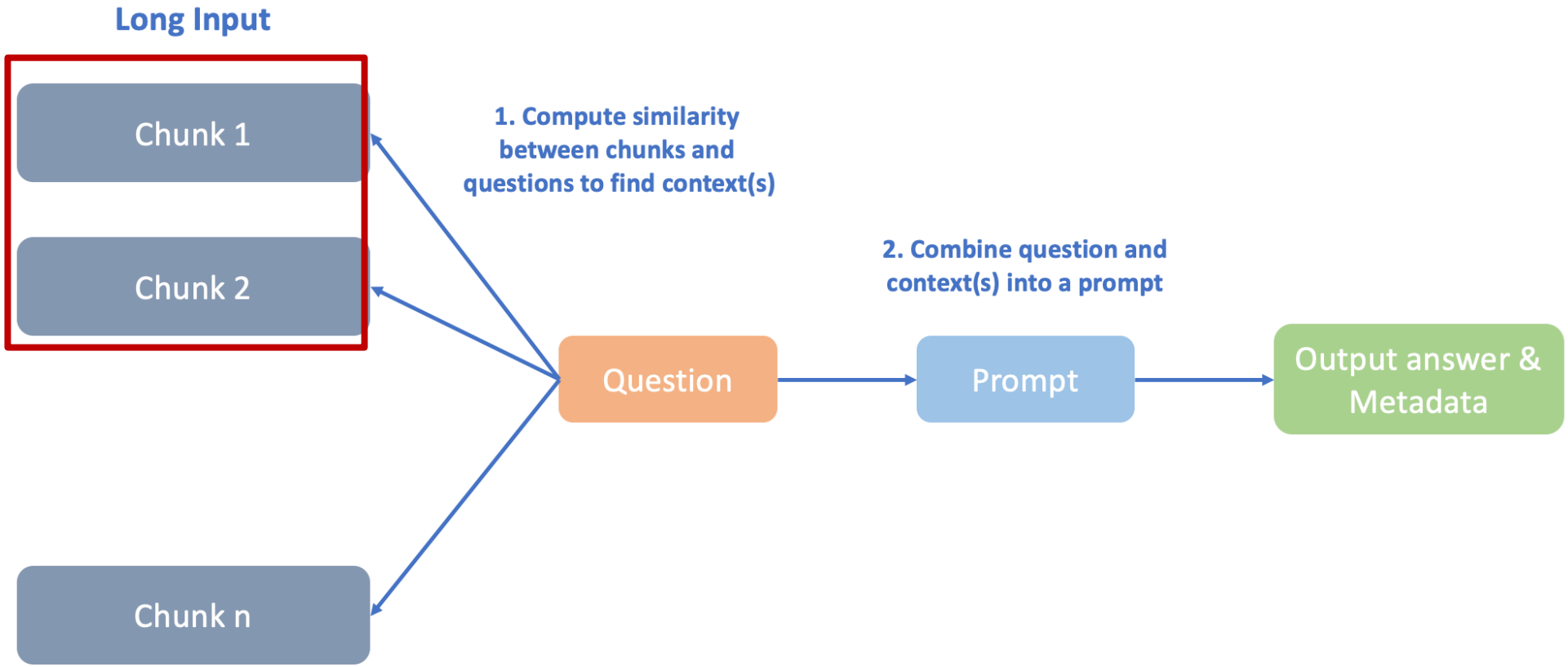
- 检索增强 + 短上下文模型能够和长上下文模型表现相当



Xu et al. Retrieval meets Long Context Large Language Models. 2023.

检索增强的方法

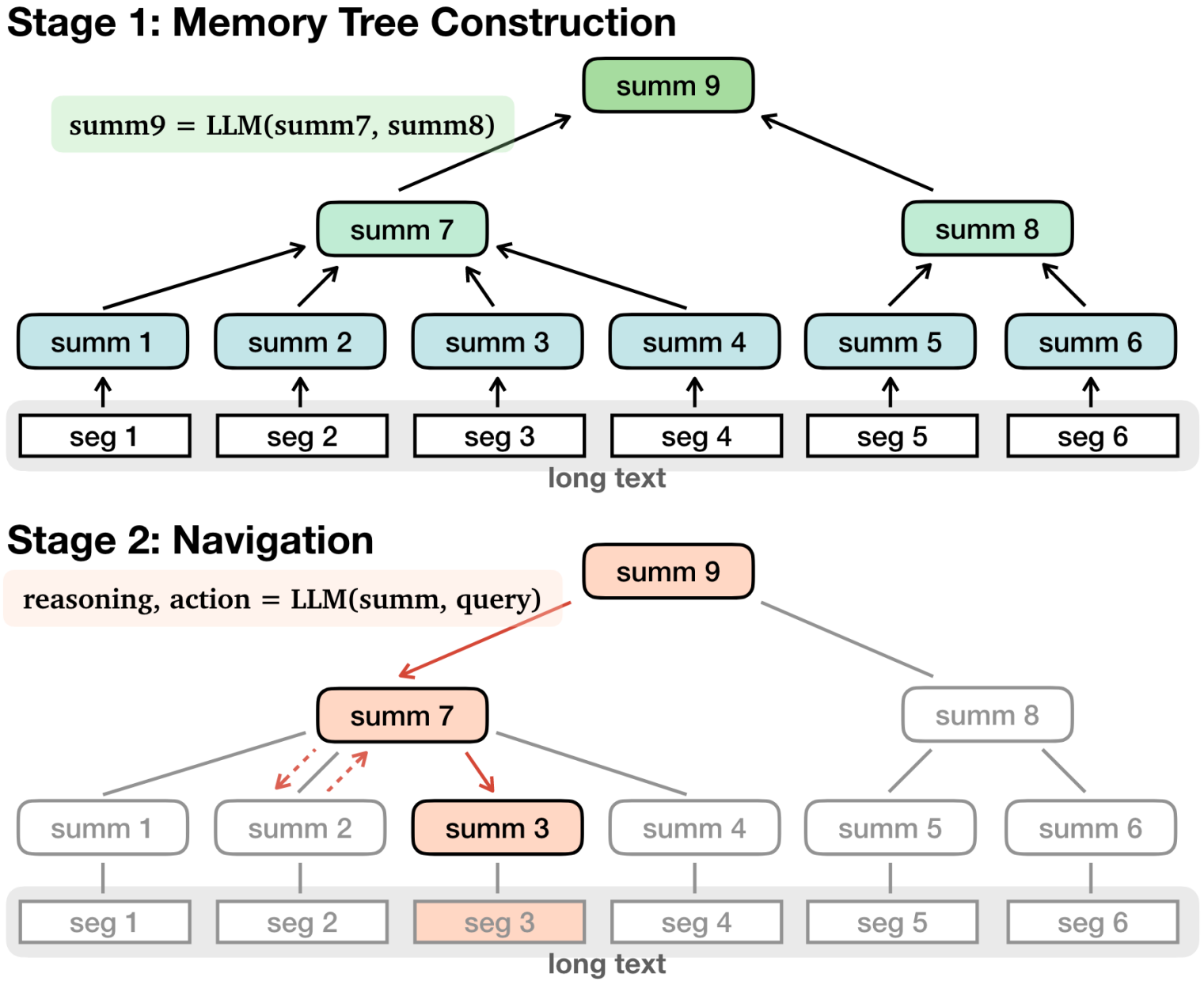
- 每个块的抽取彼此独立，没有考虑彼此之间的关联



树状检索

- 自底向上构建 summary tree
- 自顶向下搜索信息

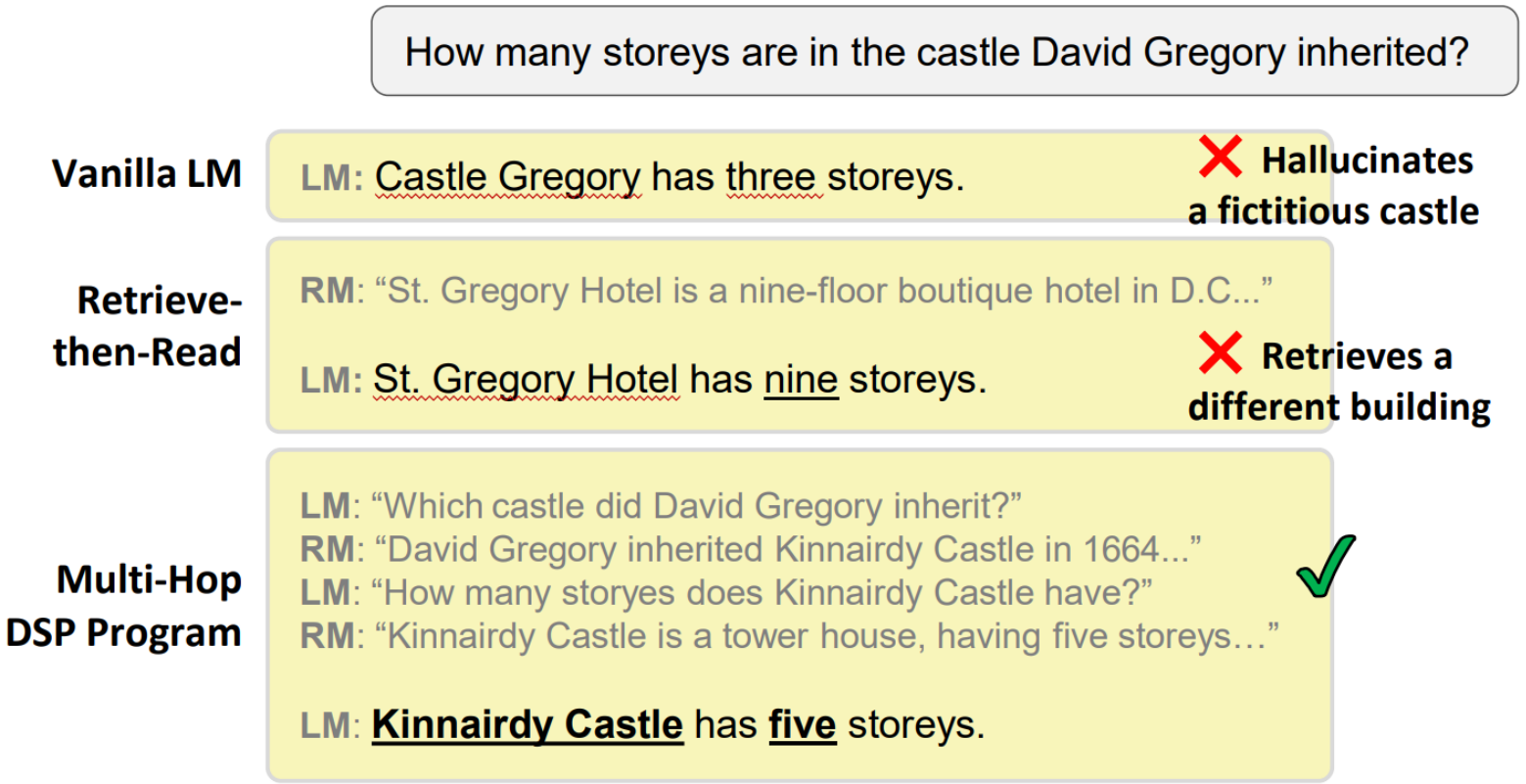
- 建模相邻块的联系
- 远距离仍不够好



Chen et al. Walking Down the Memory Maze: Beyond Context Limit through Interactive Reading. 2023.

迭代检索

- 根据前一步检索的信息，进行下一步检索
- 高昂的开销



Khattab et al. Demonstrate-Search-Predict. 2022

短文本训练，短文本测试

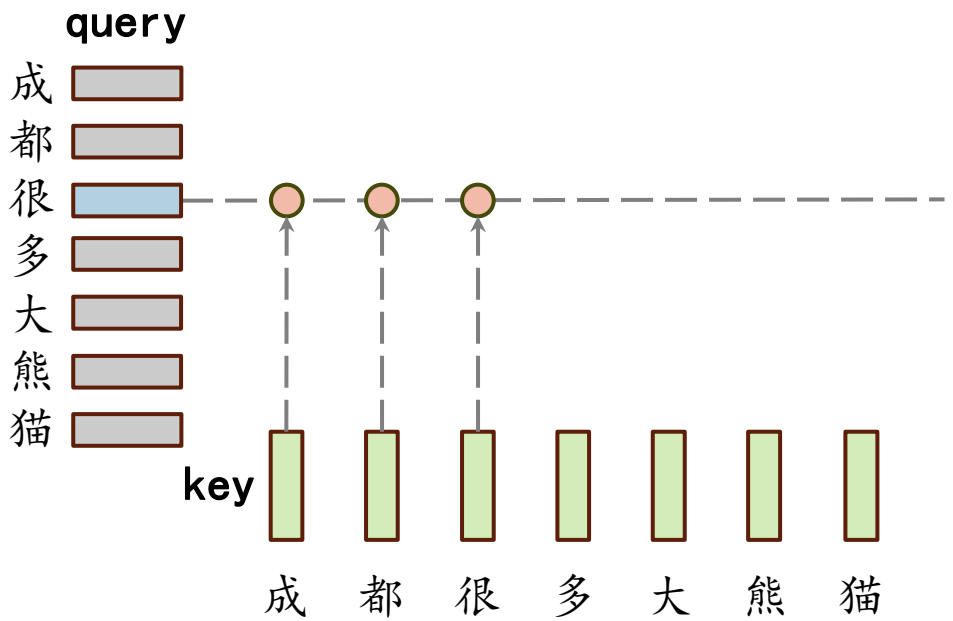
- 优点:
  - 不用对原始模型做大改动
  - 理论上可接受任意长度的输入
- 不足:
  - 远距离依赖建模能力不够好

目录

Attention原理回顾

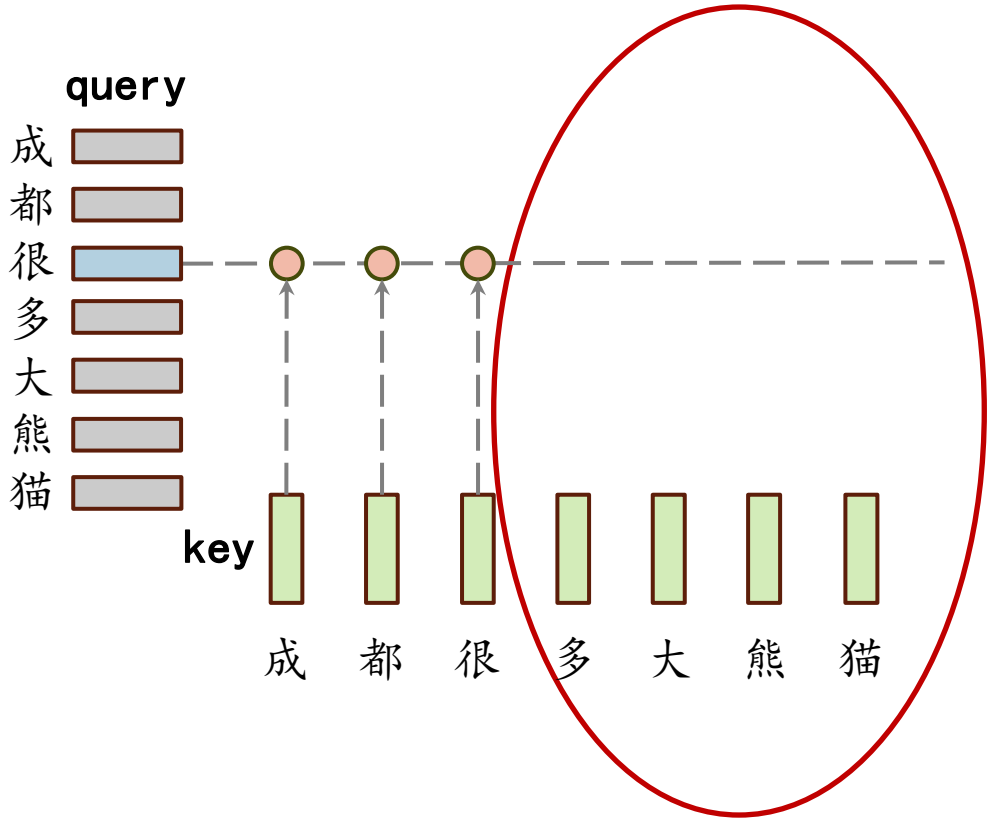
- Attention in Decoder Only Model
  - 融合上下文信息的方式
  - $h'_i = \sum_{j \in \{j<i\}} a_{ij} * h\_j$

第 i 个词和第 j 个词的相关性



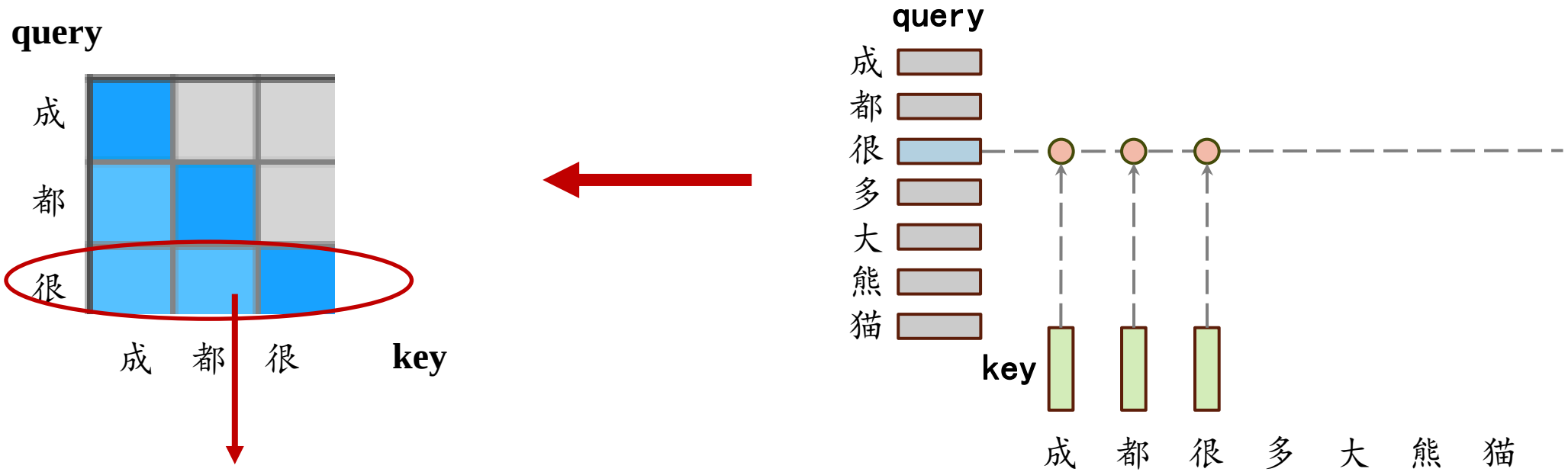
Attention原理回顾

- Attention in Decoder Only Model
  - 融合上下文信息的方式
  - $h'_i = \sum_{j \in \{j<i\}} a_{ij} * h\_j$



Attention原理回顾

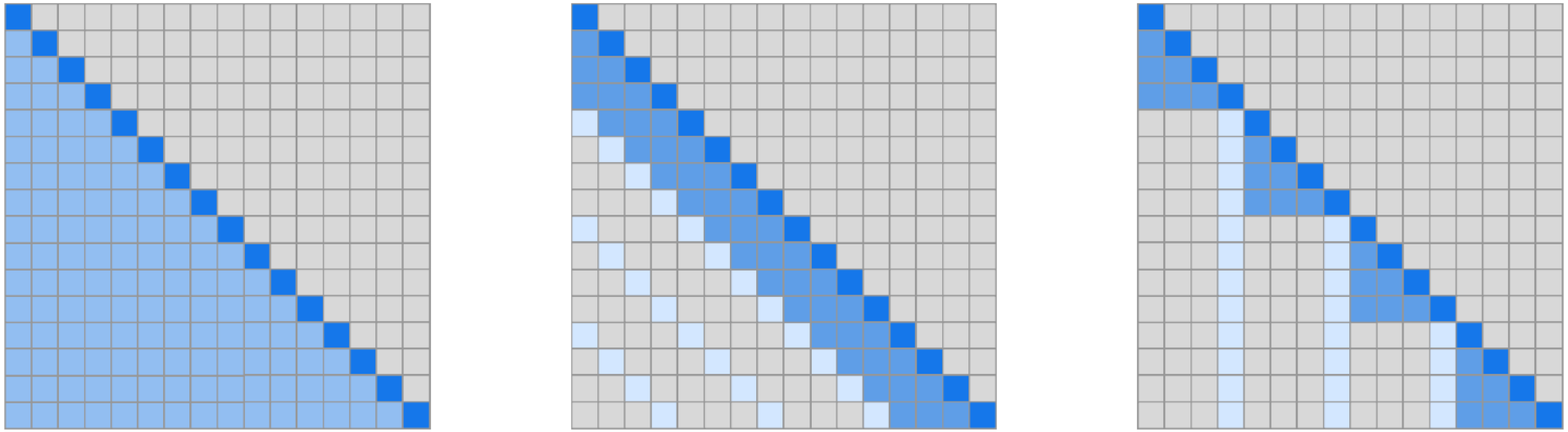
- Attention Matrix in Decoder Only Model



(i,j) 涂色表示第 i 行的词，关注 j 列的词

稀疏化注意力机制

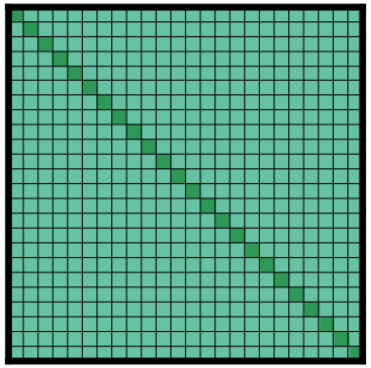
- 全连接注意力 -> 局部+全局
  - 局部：和滑动窗口类似，只关注特定窗口内的信息 （深蓝色）
  - 全局： 窗口以外的信息，选择性的关注 （浅蓝色）



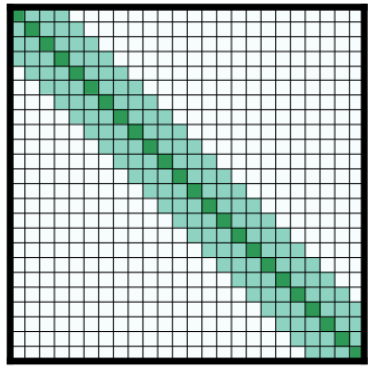
(a) Transformer (b) Sparse Transformer (strided) (c) Sparse Transformer (fixed)

稀疏化注意力机制

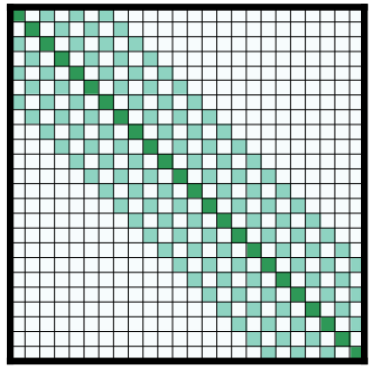
- 全连接注意力 -> 局部+全局
  - 局部：和滑动窗口类似，只关注特定窗口内的信息
  - 全局：启发式规则，例如Sparse Transformer、Longformer等



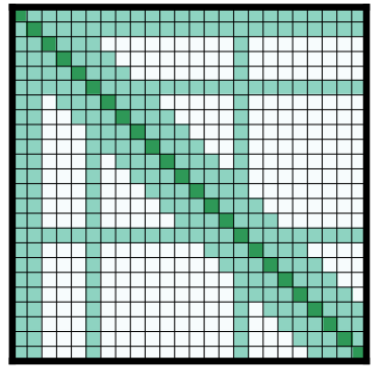
(a) Full n^2 attention



(b) Sliding window attention



(c) Dilated sliding window

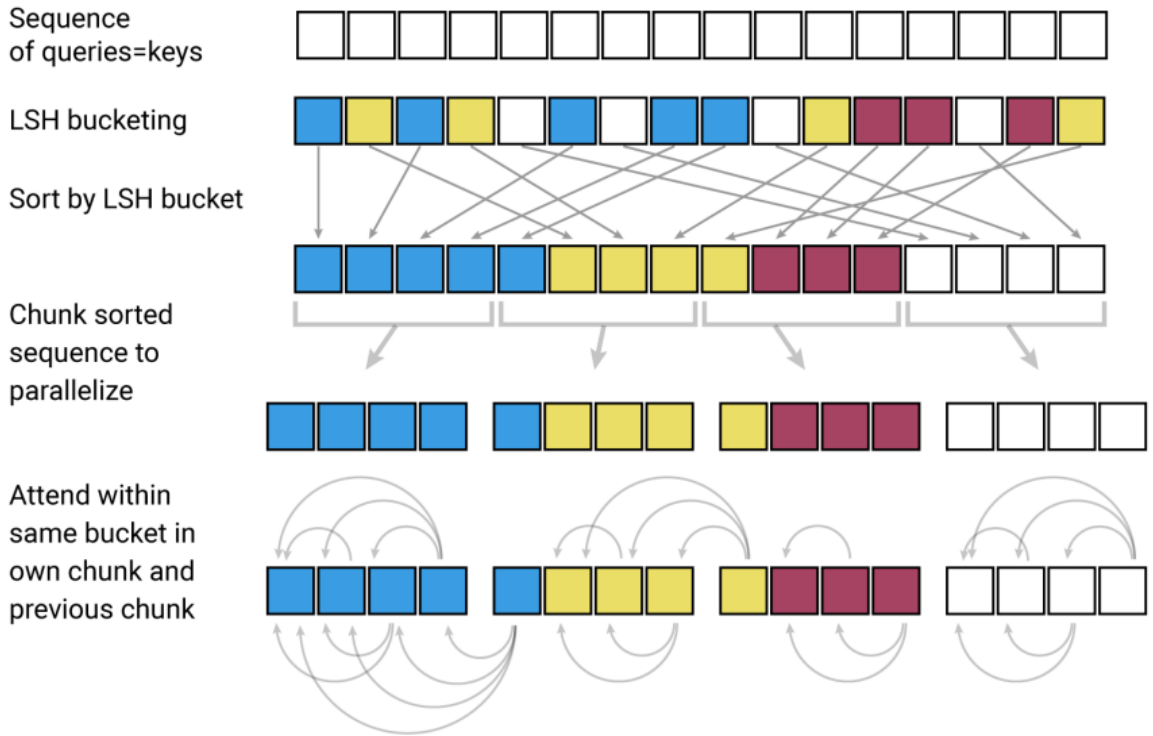


(d) Global+sliding window

Beltagy et al. Longformer: The Long-Document Transformer. 2020

稀疏化注意力机制

- 全连接注意力 -> 局部+全局
  - 局部：和滑动窗口类似，只关注特定窗口内的信息
  - 全局：启发式规则，自动学习模式
  - 例如： Reformer
    - 先对每个token做hash
    - hash值相近的相互关注



Kitaev et al. Reformer: The Efficient Transformer. ICLR 2020

稀疏化注意力机制

- 全连接注意力 -> 局部+全局
  - 局部：和滑动窗口类似，只关注特定窗口内的信息
  - 全局：启发式规则，自动学习模式

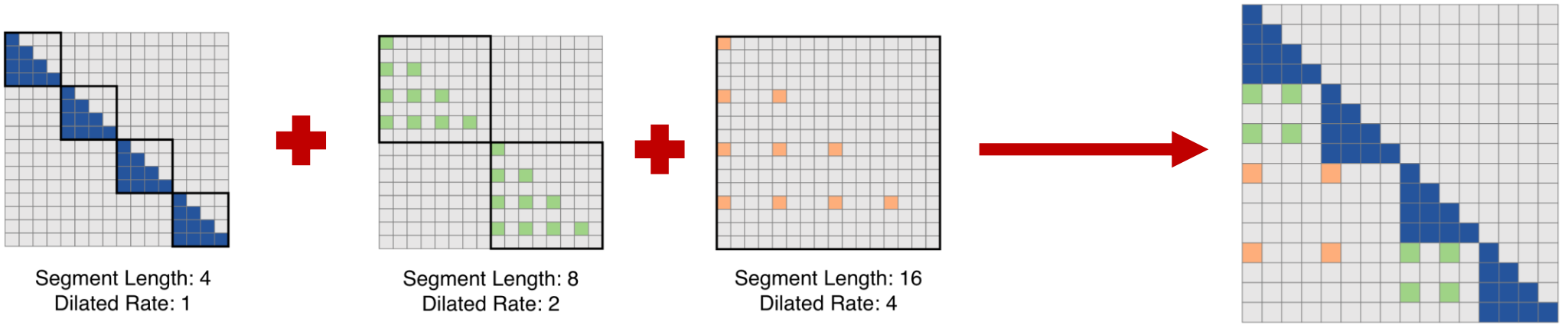
- 显著降低复杂度
  - $O(n^2) -> O(n\sqrt{n}d)\ \ or\ O(n\log n\ \ d)\ or\ O(nd)$
  - n 是总文本长度，d 是窗口大小

稀疏化注意力机制 —— LONGNET

- 注意力衰减假设：

  Attention allocation decreases exponentially as the distance between tokens grows.

- 复杂度 $O(nd)$，且理论上可扩展到1 billion



Ding et al. LONGNET: Scaling Transformers to 1,000,000,000 Tokens. 2023

长文本训练，长文本测试

- 优点：
  - 训练和推理代价均较低
  - 可训练超长上下文
- 不足：
  - 需要重新训练
  - 并非所有稀疏注意力机制都和全连接注意力兼容，无法继承优化
  - 效果是否和全连接等价，需要更多实验探索

目录

- 长上下文建模方法：
  - 短文本训练，短文本推理（黄曲哲）
  - 长文本训练，长文本推理（黄曲哲）
  - 短文本训练，长文本推理（陶铭绪）
- 长文本数据集与评测 （张晨）
- 总结与展望

基于位置编码的方法

- 原理回顾：RoPE
- 内插值法
- 外推法

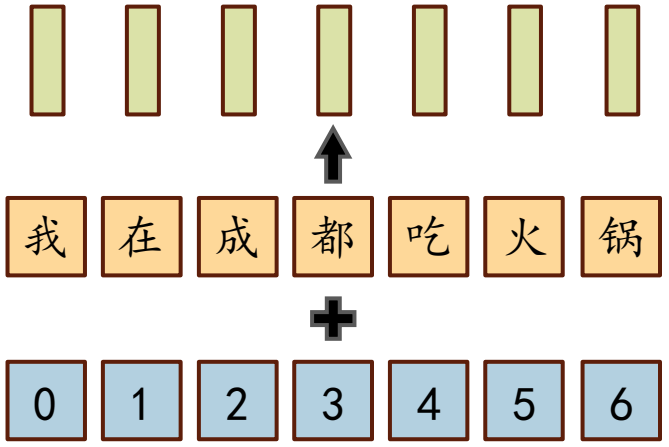
RoPE原理回顾

- RoPE是目前开源大语言模型广泛采用的位置编码算法：
  - LLaMA
  - Baichuan
  - ChatGLM
  - ……

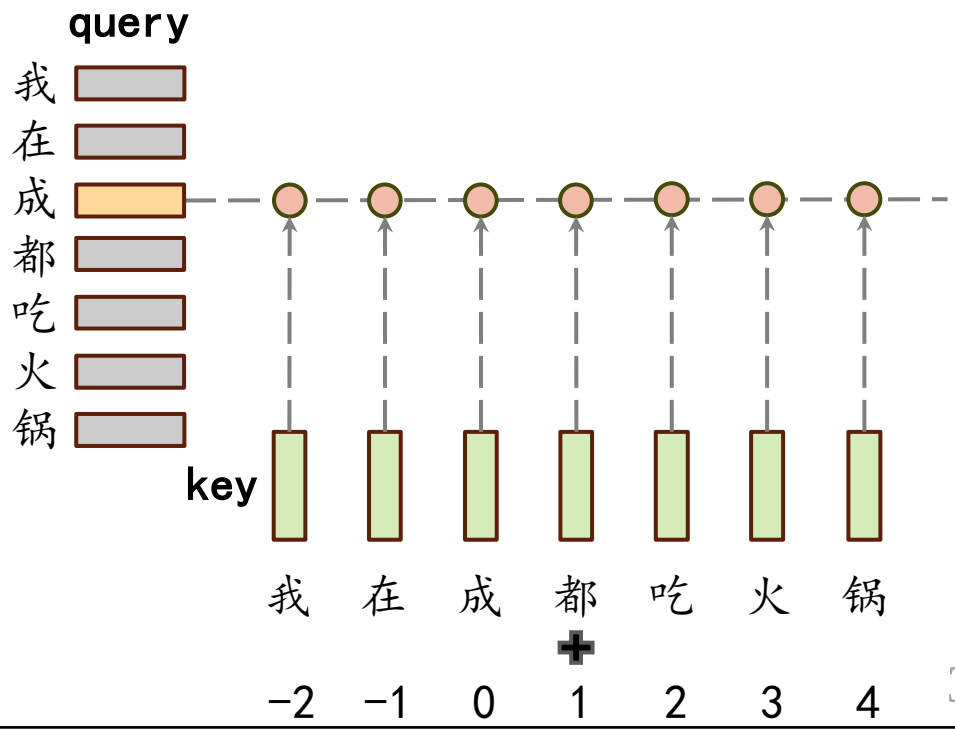
RoPE原理回顾

- 绝对位置编码 vs. 相对位置编码

编码：token在序列中的绝对位置



编码：token之间的相对距离



RoPE原理回顾

- 绝对位置编码 ✚ 相对位置编码 〓 ？

- 绝对位置编码
  - 显式引入token的绝对位置信息，但是相对位置建模只能依赖训练
  - 局部的表征能力可能不够好
- 相对位置编码
  - 清晰地建模了局部token的相对位置关系
  - 一般会对距离进行截断，无法建模远距离信息

RoPE原理回顾

【RoFormer论文】

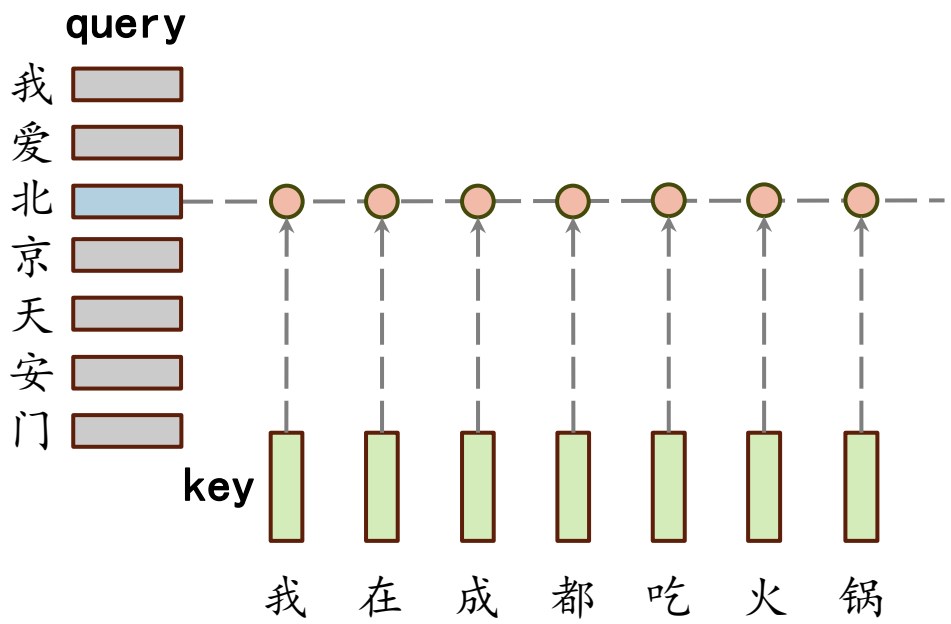
- 绝对位置编码 ✚ 相对位置编码 〓

- 旋转位置编码 RoPE（Rotary Position Embedding）
  - 通过token的绝对位置引入位置特征
  - 保证任意两个位置特征的“差值”只与相对距离有关，与各自的绝对位置无关

Su et al. RoFormer: Enhanced Transformer with Rotary Position Embedding. 2023

RoPE原理回顾

- Self-Attention（不考虑位置信息）
  - 初始特征：$\{x_0, x_1, \cdots, x_{L-1}\}$
  - 计算query：$q_m = W^Q x_m$
  - 计算key：$k_n = W^K x_n$
  - 计算相关性：$e_{mn} = \langle q_m, k_n \rangle$
  - 计算注意力分数：$a_{mn} = \frac{\exp(e_{mn})}{\sum_j \exp(e_{mj})}$



RoPE原理回顾

- Self-Attention
  - 初始特征：$\{x_0, x_1, \cdots, x_{L-1}\}$
  - 计算query：$q_m = W^Q x_m$
  - 计算key：$k_n = W^K x_n$
  - 计算相关性：$e_{mn} = \langle q_m, k_n \rangle$
  - 计算注意力分数：$a_{mn} = \frac{\exp(e_{mn})}{\sum_j \exp(e_{mj})}$

引入绝对位置

- Self-Attention
  - 初始特征：$\{x_0, x_1, \cdots, x_{L-1}\}$
  - 计算query：$q_m = f_q(W^Q x_m, m)$
  - 计算key：$k_n = f_k(W^K x_n, n)$
  - 计算相关性：

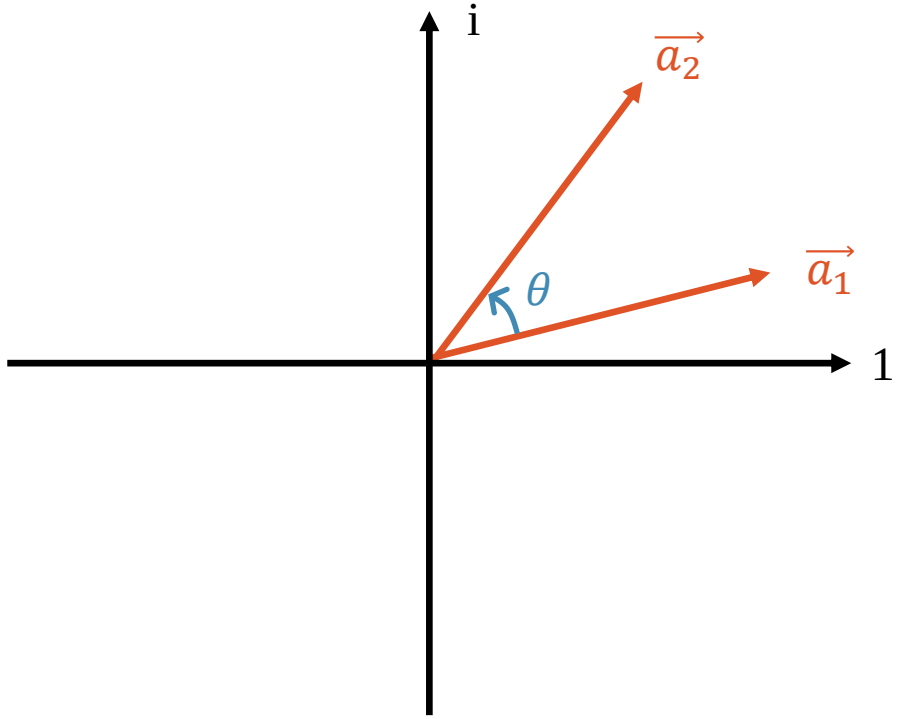
$$e_{mn} = \langle q_m, k_n \rangle \sim g(x_m, x_n, m-n)$$

确保相对差值与绝对位置无关

  - 计算注意力分数：$a_{mn} = \frac{\exp(e_{mn})}{\sum_j \exp(e_{mj})}$

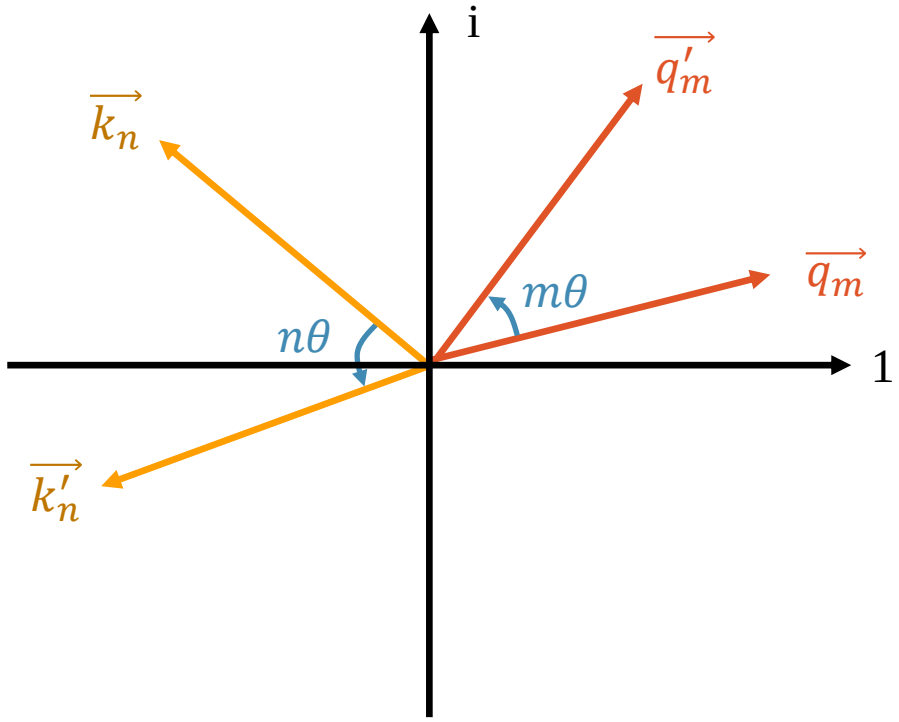
RoPE原理回顾

- 一个自然的例子：复平面
  - $a_2 = a_1 \cdot e^{\mathrm{i}\theta}$
- 性质
  - 模长不变：不改变注意力分数规模
  - 乘法变换：便于引入注意力计算



RoPE原理回顾

- 引入绝对位置
  - $q'_m = q_m \cdot e^{\mathrm{i}m\theta}$
  - $k'_n = k_n \cdot e^{\mathrm{i}n\theta}$
  - $e_{mn} = \langle q'_m, \overline{k'_n} \rangle = \mathrm{Re}\left(q_m \cdot k_n \cdot e^{\mathrm{i}(m-n)\theta}\right)$
  - $\mathrm{Re}(z)$表示复数z的实数部分

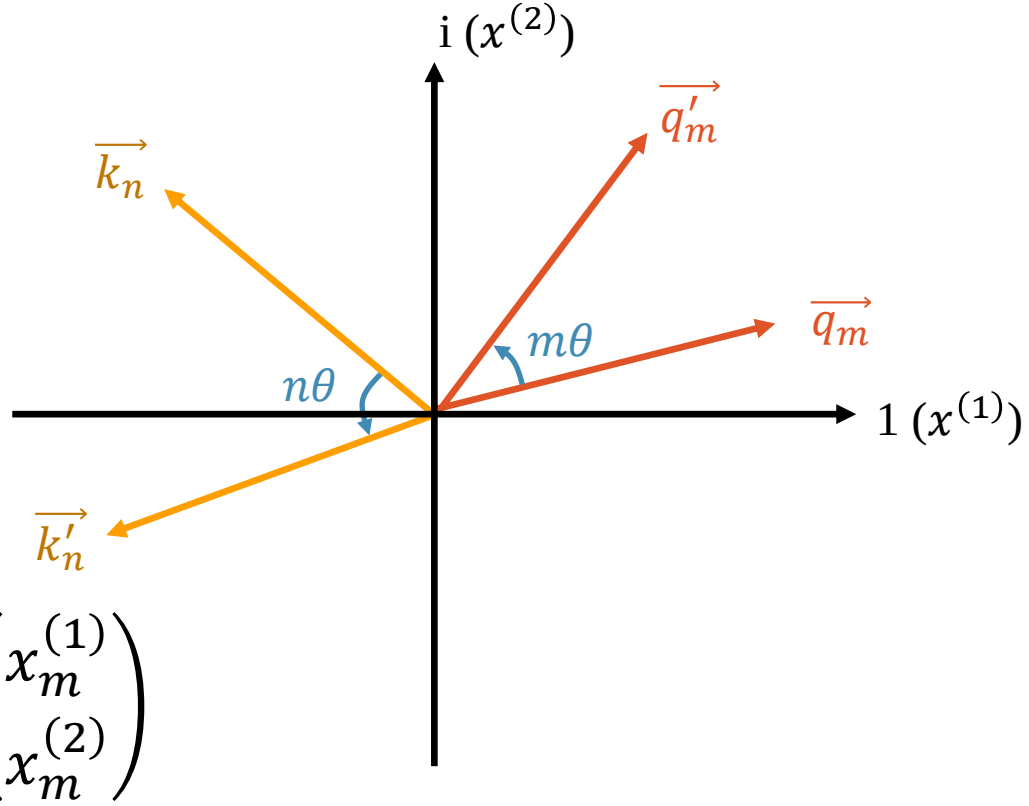


RoPE原理回顾

- 复平面 ➡ 二维欧氏空间
  - $q'_m = q_m \cdot e^{\mathrm{i}m\theta}$
  - $k'_n = k_n \cdot e^{\mathrm{i}n\theta}$

⬇ 欧拉公式：$e^{i\theta} = \cos\theta + \mathrm{i}\sin\theta$

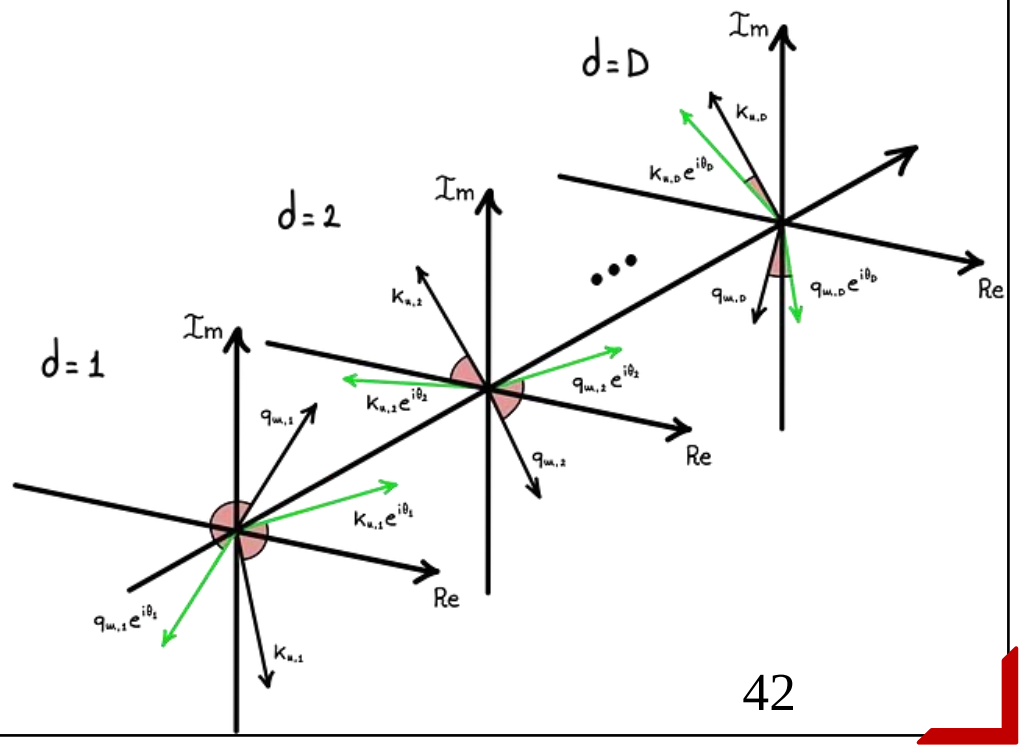
- $f(Wx_m, m) = \begin{pmatrix} \cos m\theta & -\sin m\theta \\ \sin m\theta & \cos m\theta \end{pmatrix} W \begin{pmatrix} x_m^{(1)} \\ x_m^{(2)} \end{pmatrix}$



RoPE原理回顾

- 扩展到任意偶数维空间
  - $\mathrm{R}^d = \mathrm{R}^2 \times \mathrm{R}^2 \times \cdots \times \mathrm{R}^2$ （d/2个）

$$\boldsymbol{R}^d_{\Theta,m} = \begin{pmatrix} \cos m\theta_1 & -\sin m\theta_1 & 0 & 0 & \cdots & 0 & 0 \\ \sin m\theta_1 & \cos m\theta_1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cos m\theta_2 & -\sin m\theta_2 & \cdots & 0 & 0 \\ 0 & 0 & \sin m\theta_2 & \cos m\theta_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \cos m\theta_{d/2} & -\sin m\theta_{d/2} \\ 0 & 0 & 0 & 0 & \cdots & \sin m\theta_{d/2} & \cos m\theta_{d/2} \end{pmatrix}$$

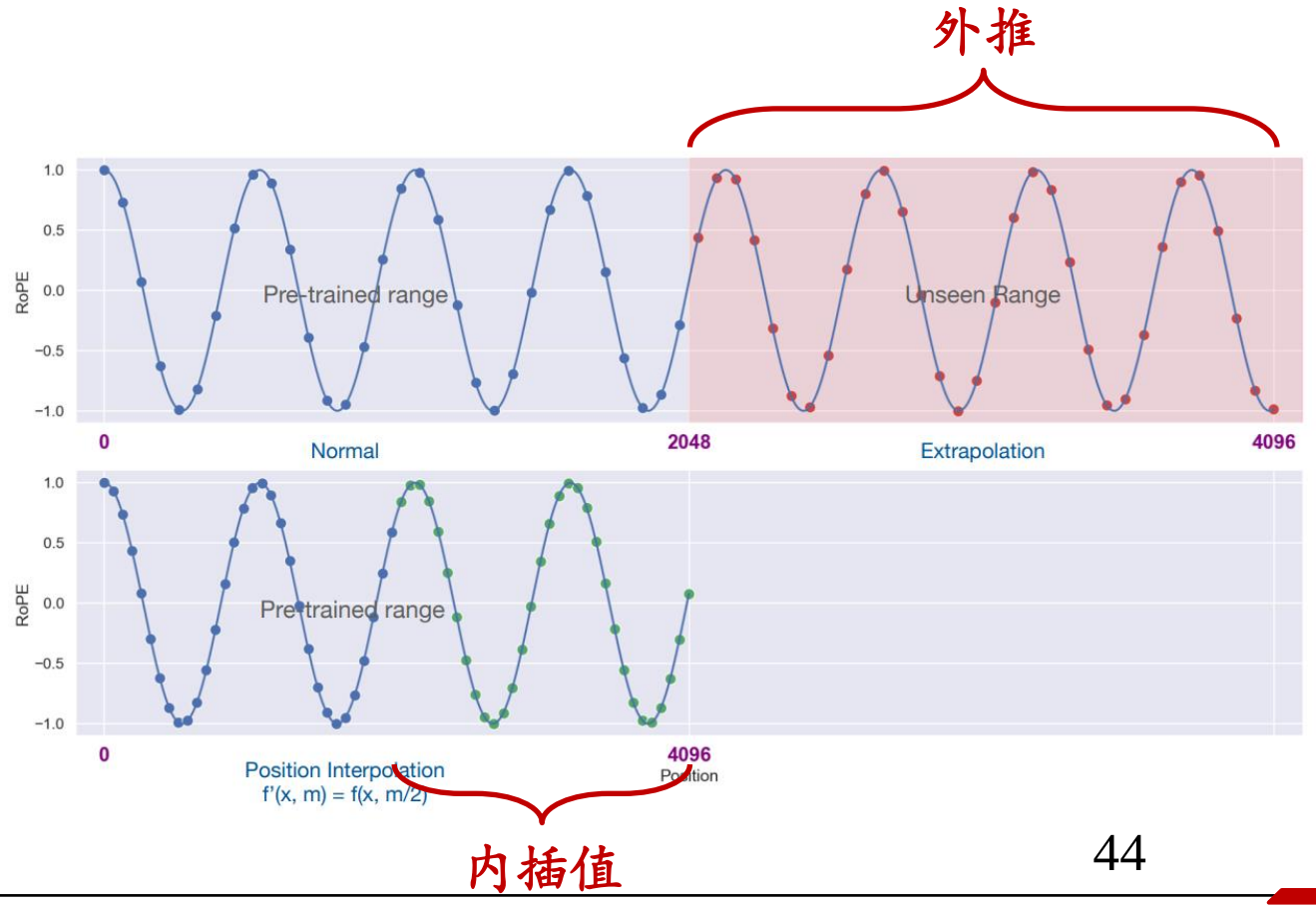


如何突破上下文长度限制？

- RoPE的内核是三角函数
  - 周期性
  - 第m个token的位置编码向量的第$2t$维(总共D维)：
    - $\sim\cos m\theta_t$
    - 其中，$\theta_t = b^{-\frac{t}{D/2}} = b^{-\frac{2t}{D}}$
    - b一般可以取较大值，例如10,000
    - 三角函数周期：$\frac{2\pi}{\theta_t} = \frac{2\pi}{b^{-\frac{2t}{D}}} = 2\pi b^{\frac{2t}{D}}$

如何突破上下文长度限制？

- RoPE的内核是三角函数
  - 周期性
  - 第$2t$维：$\theta_t, \cdots, m\theta_t, \cdots L\theta_t$
- 内插值法
- 外推法



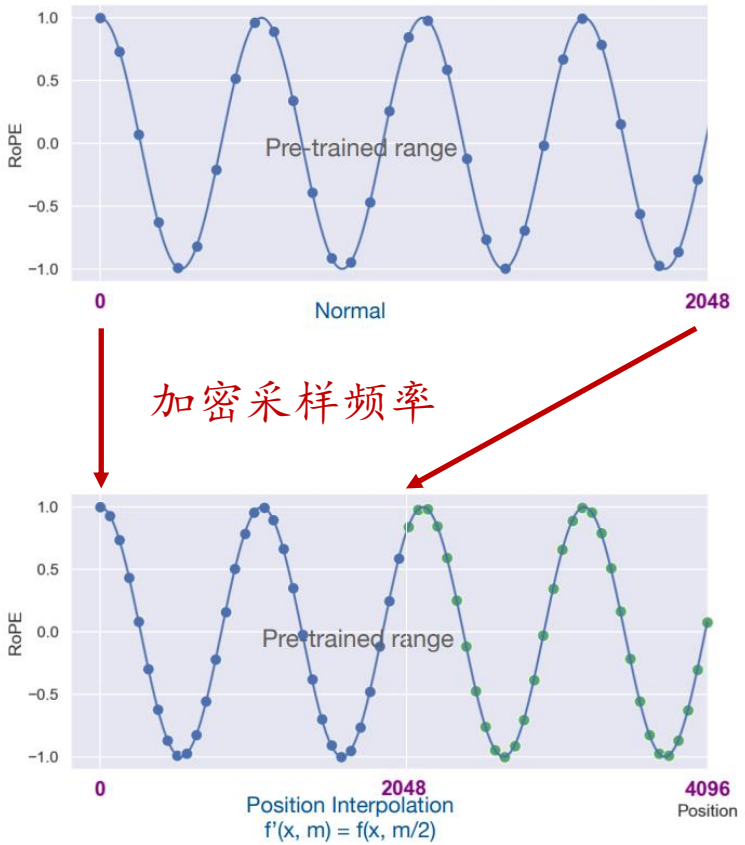
内插值法-线性法

【PI论文】

- 线性内插值法（Position Interpolation）
- 模型原本支持最大长度L
- 目标：支持在长度为L'的文本上推理

- 方法：对于任意位置m，其原本在第$2t$维的三角函数坐标是$m\theta_t$，现在将其变为 $\frac{L}{L'}m\theta_t$

- $m \leftarrow \frac{L}{L'}m$，$\theta_t$不变

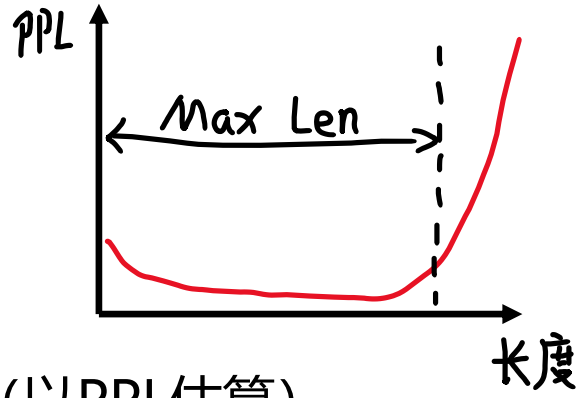


Chen et al. Extending Context Window of Large Language Models via Positional Interpolation. 2023.

内插值法-线性法

【PI论文】

- 线性内插值法（Position Interpolation）
  - 线性内插值法仅需少量训练，即可使LLM适应新的位置编码
  - 通常是数百步训练
  - 基于LLaMA-1的实验：原生模型支持的最大上下文长度是2048（以PPL估算）



Model			Fine-tuning steps					
Size	Context Window	Method	200	400	600	800	1000	10000
7B	8192	FT	1792	2048	2048	2048	2304	2560
33B	8192	FT	1792	2048	1792	2048	2304	-
7B	8192	PI	8192	8192	8192	8192	8192	-
7B	16384	PI	16384	16384	16384	16384	16384	-
7B	32768	PI	32768	32768	18432	32768	32768	-
33B	8192	PI	8192	8192	8192	8192	8192	-
33B	16384	PI	16384	16384	16384	16384	16384	-

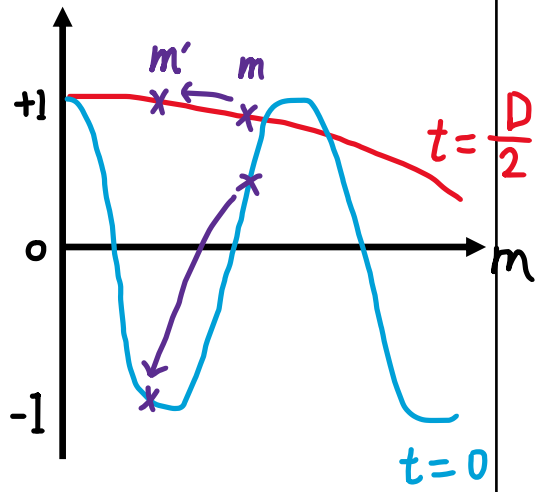
直接微调

先内插值，再微调

Chen et al. Extending Context Window of Large Language Models via Positional Interpolation. 2023.

内插值法-NTK

【Reddit - NTK】



- 线性内插值法的问题
  - 内插值之后，在短文本上表现显著变差
  - 潜在原因：靠前的维度周期更小，内插值导致cos/sin函数的值变动剧烈
- 第m个token的位置编码向量的第$2t$维(总共D维)：
  - $\sim \cos m\theta_t$，其中$\theta_t = b^{\frac{-2t}{D}}$
  - $t \to 0$时，周期→2π；$t \to \frac{D}{2} - 1$时，周期→20000π（$2b\pi$）

https://www.reddit.com/r/LocalLLaMA/comments/14lz7j5/ntkaware_scaled_rope_allows_llama_models_to_have/

内插值法-NTK

【Reddit - NTK】

- 线性内插值法的问题
  - 内插值之后，在短文本上表现显著变差
  - 潜在原因：靠前的维度周期更小，内插值导致cos/sin函数的值变动剧烈

$t \to 0$时，周期→2π；$t \to \frac{D}{2} - 1$时，周期→20000π（$2b\pi$）

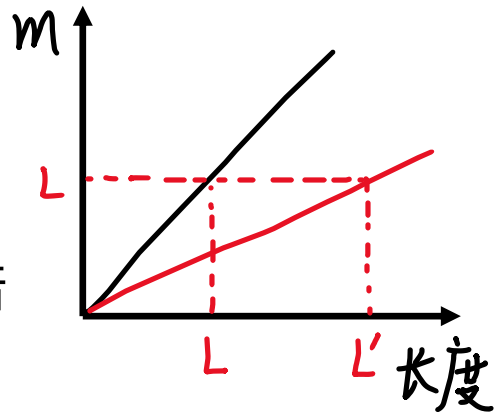
- 解决方案：
  - $t \to 0$时，尽量不改变周期；$t \to \frac{D}{2} - 1$时，适当增大周期
  - 不对m做缩放，改为调整b

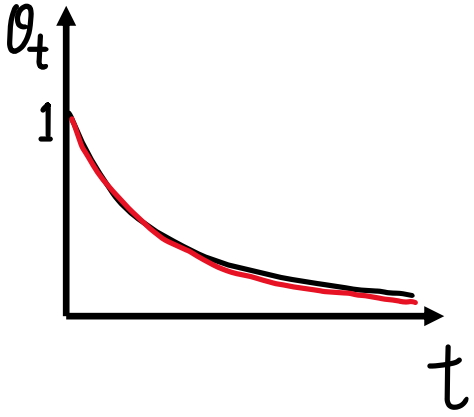
- $m \cdot \hat{b}^{-\frac{D-2}{D}} = \frac{L}{L'} m \cdot b^{-\frac{D-2}{D}}$
- $\hat{b} = b\left(\frac{L'}{L}\right)^{\frac{D}{D-2}}$
- m不变
- θ_t变为$b^{-\frac{2t}{D}}\left(\frac{L'}{L}\right)^{-\frac{2t}{D-2}}$

https://www.reddit.com/r/LocalLLaMA/comments/14lz7j5/ntkaware_scaled_rope_allows_llama_models_to_have/

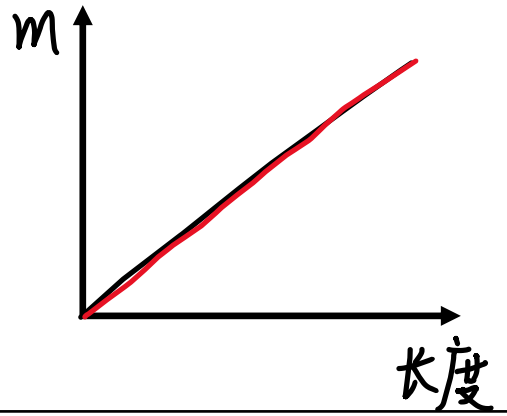
内插值法-总结

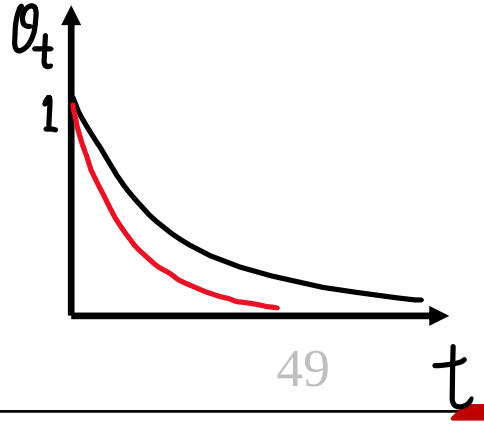
- 线性内插值法
  - 实现：等距离压缩位置m
  - 效果：将位置编码全部维度的周期延长L'/L倍





- NTK内插值法
  - 实现：对底数b施加延长系数
  - 效果：位置编码的高频维度的周期几乎不变；
    最低频维度的周期延长L'/L倍

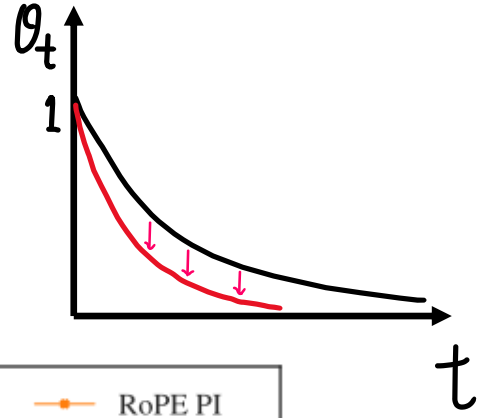


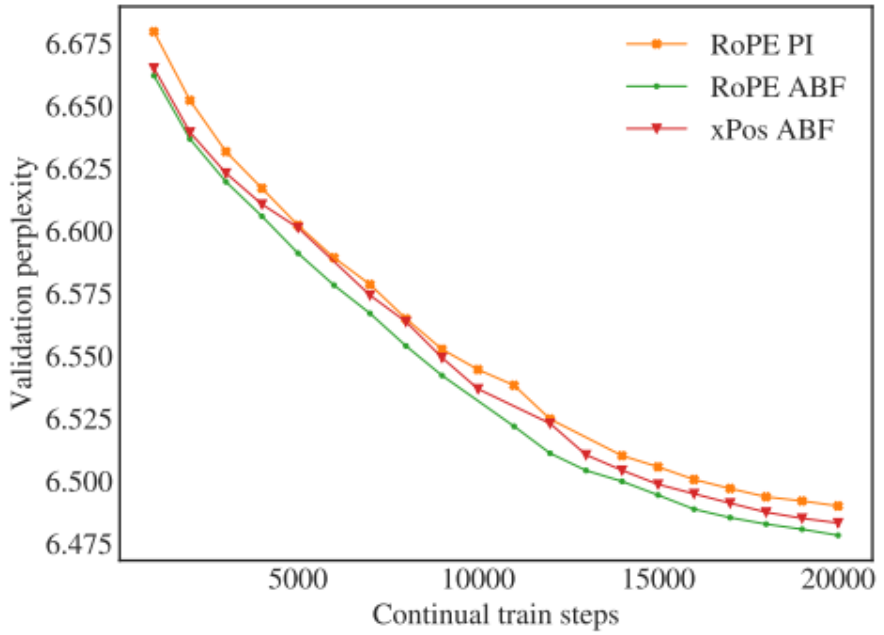


更为简单有效的实现——ABF

【Paper - ABF】

- Adjusted Base Frequency
- 核心：为底数b添加大于1的系数 → 延长三角函数周期
- 解决：直接增加底数b自身的值
- 例如：$b = 10{,}000 \rightarrow b = 500{,}000$

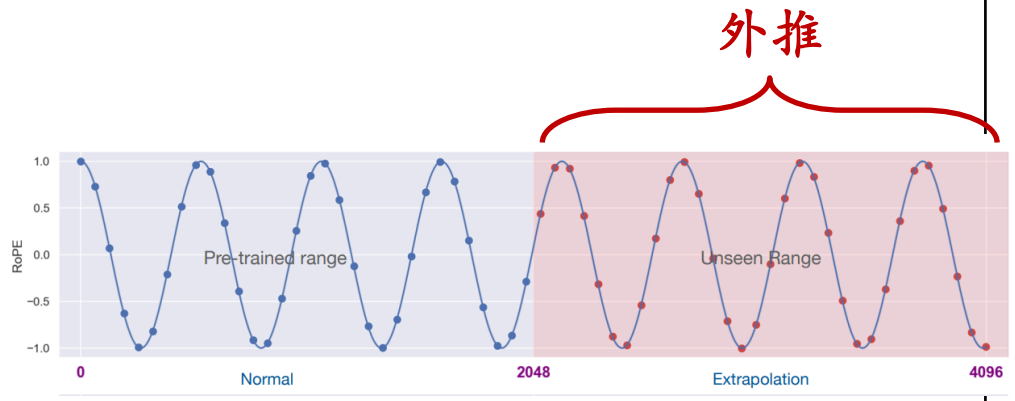




Xiong et al. Extending Context Window of Large Language Models via Positional Interpolation. 2023.

外推法

- 大语言模型有一定的外推能力
- 但是外推能力有限



- 核心原因：超越了模型训练时所见过的坐标区间
- 距离较远的两个token之间的注意力分数可能会异常升高

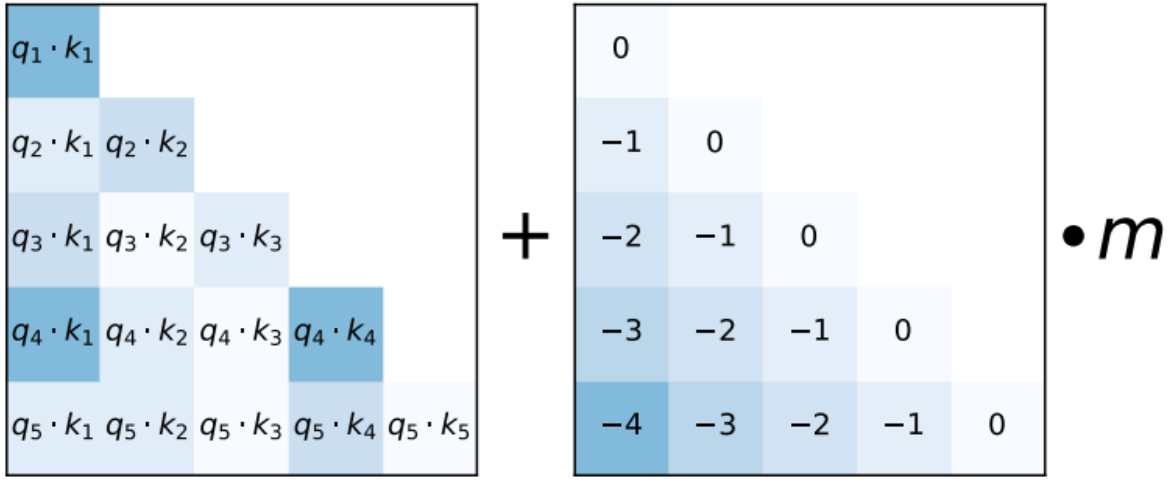
Xiong et al. Extending Context Window of Large Language Models via Positional Interpolation. 2023.

外推法-ALIBI

【ALiBi论文】

- 距离较远的两个token之间的注意力分数可能会异常升高

- 解决办法：在注意分数上添加距离惩罚，该惩罚值随着相对距离的增大而线性增加



Press et al. Train Short, Test Long: Attention with Linear Biases Enables Input Length Extrapolation. ICLR 2022.

外推法-ALIBI

【ALiBi论文】

- 好处：
  - 实现简单，计算代价小
  - 无需训练，即插即用

- 潜在缺陷：
  - 距离惩罚过大，导致模型无法关注远距离token

Press et al. Train Short, Test Long: Attention with Linear Biases Enables Input Length Extrapolation. ICLR 2022.

短文本训练，长文本推理总结

- 内插值法：
  - 更好的长序列建模能力；但是通常需要持续预训练
- 外推法：
  - 无需训练、直接部署；但有时会过多关注局部特征，丢失长序列信息

- 效果叠加：
  - 先使用内插值法持续训练，再使用外推法，可以将模型支持的上下文长度扩展到更长。

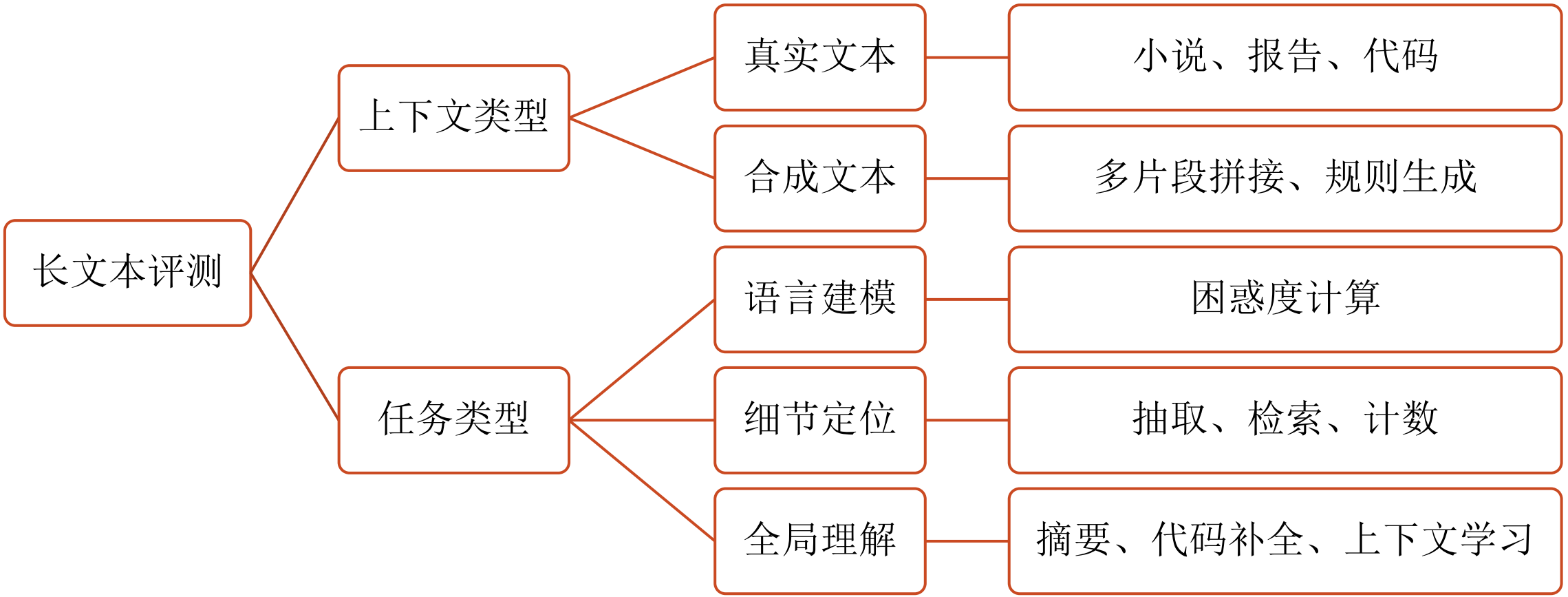
目录

长文本评测

- 长文本评测数据集分类
  - 上下文类型
  - 任务类型

- 长文本多任务评测套件

长文本评测数据集分类


长文本评测
上下文类型
任务类型
真实文本
合成文本
语言建模
细节定位
全局理解
小说、报告、代码
多片段拼接、规则生成
困惑度计算
抽取、检索、计数
摘要、代码补全、上下文学习

上下文类别1：真实长文本

- **真实长文本**：可满足人类真实需求的、自然存在的、长度较长的文本

- 包括但不限于
  - **情节性文本**（小说、剧本）
  - **记录性文本**（论文、报告、使用文档、会议记录）
  - **项目代码库**

上下文类别1：真实长文本

- **情节性文本**
  - 例如：小说、剧本
  - 人物众多、关系复杂
  - 前后情节联系紧密

- 常用数据集：NarrativeQA、QuALITY

Title: Ghostbusters II
Question: How is Oscar related to Dana?
Answer: her son
Summary snippet: . . . Peter's former girlfriend Dana Barrett has had a son, Oscar. . .
Story snippet:

DANA (setting the wheel brakes on the buggy)
Thank you, Frank. I'll get the hang of this eventually.

She continues digging in her purse while Frank leans over the buggy and makes funny faces at the baby, OSCAR, a very cute nine-month old boy.

FRANK (to the baby)
Hiya, Oscar. What do you say, slugger?

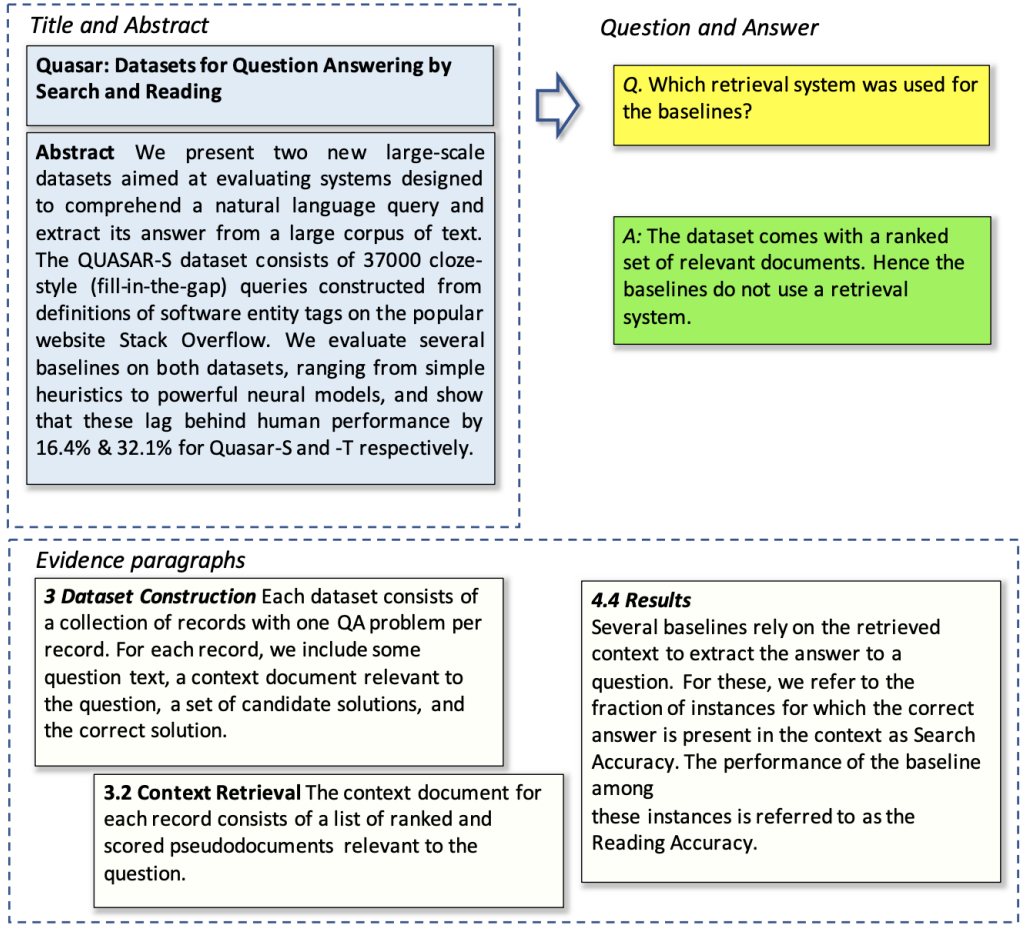
FRANK (to Dana)
That's a good-looking kid you got there, Ms. Barrett.

NarrativeQA样例

Kočiský et al. The narrativeqa reading comprehension challenge. TACL 2018

上下文类别1：真实长文本

- **记录性文本**
  - 例如：百科、非虚构书籍、论文、法律文书、公司财报、政府报告、会议记录
  - 事实性信息密集
  - 通常具有一定的结构性
- 常用数据集：Qasper、GovReport、QMSum



Qasper样例

Dasigi et al. A dataset of information-seeking questions and answers anchored in research papers. NAACL 2021

上下文类别1：真实长文本

- **项目代码库**
  - 写代码时，常需要模型基于整个项目代码库进行推理
  - 各种库、类、函数、文件自然地构成了长文本
- 常用数据集：LCC、RepoBench

format_sql/test_parser.py

```
# Path: format_sql/parser.py
# def _parse_identifier(toks):
#     count = 0
#
#     if _match(toks, [(Token.IDENTIFIER, Token.STR, Token.NUMBER)]):
#         if toks[0]._type == Token.IDENTIFIER:
#             cls = Identifier
#         elif toks[0]._type == Token.NUMBER:
#             cls = Number
#
#         value = cls(toks[0]._value)
#         count += 1
#
#         alias, j = _parse_alias(toks[count:])
#         count += j
#         value.as_ = alias['as']
#         value.alias = alias['alias']
#
#         return value, count
#
#     raise InvalidIdentifier(toks)
#
# def parse(toks):
#     for statement, unused_count in _parse(toks):
#         yield statement
#
# Path: format_sql/tokenizer.py
# class Token:
#     AS = 'AS'
#     ASC = 'ASC'
#     ......
#       normalized = value.split()[-1].upper()
#       if normalized == 'JOIN':
#           return Token.JOIN
#
#       return None

# Path: tests/test_parser.py
import pytest
from format_sql.parser import _parse_identifier, parse
from format_sql.tokenizer import Token


......
@pytest.mark.parametrize(('tokens', 'expected_value', 'expected_count'), [
    ......
])
def test_parse_identifier(tokens, expected_value, expected_count):

    result, count = _parse_identifier(tokens)
```

format_sql/parser.py

```
def _parse_identifier(toks):
    count = 0

    if _match(toks, [(Token.IDENTIFIER, Token.STR
        if toks[0]._type == Token.IDENTIFIER:
            cls = Identifier
        elif toks[0]._type == Token.NUMBER:
            cls = Number

        value = cls(toks[0]._value)
        count += 1

        alias, j = _parse_alias(toks[count:])
        count += j
        value.as_ = alias['as']
        value.alias = alias['alias']

        return value, count

    raise InvalidIdentifier(toks)

def parse(toks):
    for statement, unused_count in _parse(toks):
        yield statement
```

format_sql/tokenizer.py

```
class Token:
    AS = 'AS'
    ASC = 'ASC'
    ......
    THEN = 'THEN'
    ELSE = 'ELSE'

    def __init__(self, token_type, token_value):
        ......

    @staticmethod
    def get_token(value):
        normalized = value.split()[0].upper()

        if normalized in TOKEN_RES.keys():
            return normalized

        normalized = value.split()[-1].upper()
        if normalized == 'JOIN':
            return Token.JOIN

        return None
```

RepoBench样例

Liu et al. RepoBench: Benchmarking Repository-Level Code Auto-Completion Systems. 2023.

上下文类别2：合成长文本

- **合成长文本**：并非自然存在的、人工构造的较长的模型输入
- 相比于真实长文本，合成长文本能以**更低的成本**构建大量样本
- 构造的方式主要分为两种
  - **片段拼接**
  - **规则生成**

上下文类别2：合成长文本

- **片段拼接：**通过将相对独立的文本片段拼接，构建较长的模型输入

- 拼接的内容经常包括
  - **来自不同文档的片段（多文档问答/摘要、多表格理解、计数、检索等）**
    - 例如：DuReader、MultiNews、SpaceDigest in ZeroSCROLLS
  - **上下文学习的示例**
    - 例如：Few-shot learning tasks in LongBench

上下文类别2：合成长文本

- **规则生成：**通过一定的规则，自动生成用于特定任务的长文本。一般仅用于测试模型能力，缺少实际应用价值。

- 例1：在长文章中随机位置插入一句无关的话，并询问模型关于这句话的问题

- 例2：自动生成大量key-value pair，要求模型返回给定key对应的value

```
line torpid-kid: REGISTER_CONTENT is <24169>
line moaning-conversation: REGISTER_CONTENT is <10310>
…
line tacit-colonial: REGISTER_CONTENT is <14564>
What is the <REGISTER_CONTENT> in line moaning-conversation?
```

来自LongEval的样本

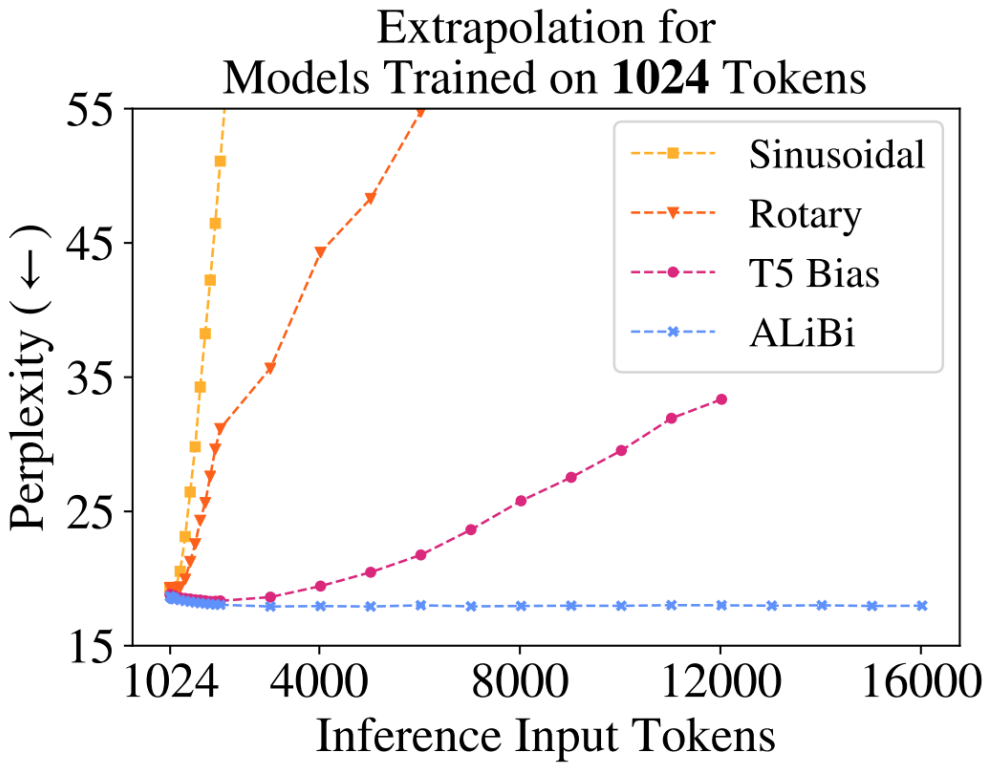
https://github.com/stunningpixels/lou-eval https://lmsys.org/blog/2023-06-29-longchat/

任务类别

- **语言建模**
- **局部定位**
- **全局理解**

任务类别1：语言建模

- 给定长文本语料库，计算其**困惑度**
- 最直接的衡量大模型长文本能力的方式
- 形式与下游任务差距较大



Press et al. Train Short, Test Long: Attention with Linear Biases Enables Input Length Extrapolation. ICLR 2022

任务类别2：细节定位

- 长文本输入中大部分信息是冗余的，需要模型**精准定位到与任务相关的局部信息**

- 重要的局部信息可能是单条，也可能是多条（多跳推理）

Question: Ralph Hefferline was a psychology professor at a university that is located in what city?
Evidence Chain: Ralph Hefferline -> Columbia University

P1: Ralph Hefferline	**P2: Columbia University**
Ralph Franklin Hefferline was a psychology professor at Columbia University.	Columbia University is a private Ivy League research university in Upper Manhattan, New York City.

来自HotpotQA的样本

任务类别2：细节定位

- 长文本输入中大部分信息是冗余的，需要模型**精准定位到与任务相关的局部信息**

- 常见任务包括：
  - 细节信息的问答、抽取（HotpotQA、DuReader）
  - 信息检索（LongChat-Lines、Passage Retrieval in LongBench）
  - 计数（Space Digest in ZeroSROLLS, Passage Count in LongBench）

样例：
1. 根据宁德时代2022年财报，公司现任董事长是谁？
2. 以上关于成都的段落中，第几段介绍了成都的旅游景点？
3. 给你50条对酒店的用户评论中，帮我统计一下里面有多少条是差评？

任务类别3：全局理解

- 需要模型对长文本进行**全局的分析和理解**
- 常见任务包括：
  - 概括性的问答、摘要、基于查询的摘要
    （GovReport，QMSum，VCSum）
  - 代码补全（LCC，RepoBench-P）
  - 上下文学习（Few-shot learning tasks in LongBench）
- 一些数据集同时包含细节定位和全局理解
  （NarrativeQA）

样例：
1. 请简单总结一下下面这篇关于Chain-of-thought的论文。
2. 根据老友记第一季的剧本，概括一下Ross对Rachel的态度。
3. 根据上面这篇关于查理一世的文章，将以下事件按时间线重排序。

长文本评测套件

- 如此多的长文本数据集，用哪些？怎么用？

长文本评测套件

- 研究者们将多类长文本任务进行整合成**评测套件**（evaluation suite）
- 希望**更方便、更全面、更公平**地评估模型能力

评测套件	研究单位	关注点
ZeroSCROLLS	Tel Aviv University、Meta	整合现有数据
L-Eval	复旦大学	关注评测公平性
LongBench	清华大学	引入中文任务
BAMBOO	中国人民大学	避免数据污染
S3Eval	中科院自动化所	灵活的人造任务
M4LE	香港中文大学、华为	不同粒度的关注点
LooGLE	北京通用人工智能研究院	多样的长依赖任务

ZeroSCROLLS（TAU, Meta）

ZeroScrolls

- 整合之前的问答、摘要任务
- 引入两个新的合成任务：计数和重排序

Dataset	Task	Domain	Metric	Avg #Words
GovReport (Huang et al., 2021)	Summarization	Government	ROUGE	7,273
SummScreenFD (Chen et al., 2022)	Summarization	TV	ROUGE	5,663
QMSum (Zhong et al., 2021)	QB-Summ	Meetings	ROUGE	10,839
SQuALITY (Wang et al., 2022)	QB-Summ	Literature	ROUGE	4,971
Qasper (Dasigi et al., 2021)	QA	Science	F1	3,531
NarrativeQA (Kočiský et al., 2018)	QA	Literature, Film	F1	49,384
QuALITY (Pang et al., 2022)	MC-QA	Literature, Misc	Accuracy	4,248
MuSiQue (Trivedi et al., 2022)	QA	Wikipedia	F1	1,749
SpaceDigest (New)	Aggregation	Reviews	ES	5,481
BookSumSort (New)	Aggregation	Literature	C_{idx}	6,840

Shaham et al. ZeroSCROLLS: A Zero-Shot Benchmark for Long Text Understanding. 2023.

L-Eval（复旦）

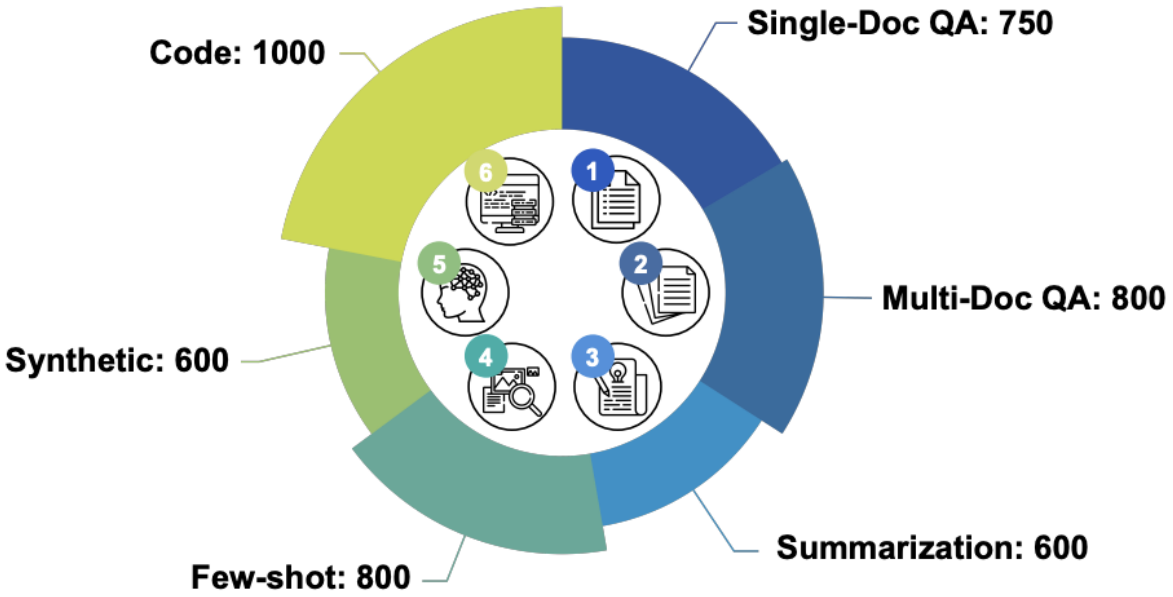
- 关注评测公平性
  - 引入更多closed-ended任务
  - 发现自动化指标与人类评价一致性较低
  - 在prompt中应明确期望的输出长度

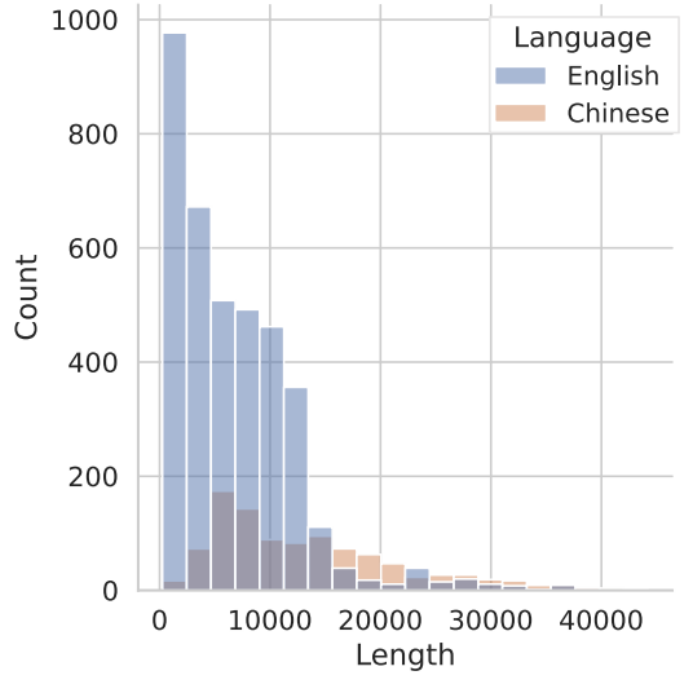
Dataset	Question-style	Domain	Avg len	Max len
		Closed - Ended Tasks		
TOEFL	Multiple choice	English test	3,907	4,171
GSM(16-shot)†	Solving math problems	In-context examples	5,557	5,638
QuALITY †	Multiple choice	Gutenberg	7,169	8,560
Coursera*	Multiple choice	Advanced courses	9,075	17,185
TopicRet†	Retriving topics	Conversation	12,506	15,916
SFcition*	True or False Questions	Scientific fictions	16,381	26,918
CodeU*	Deducing program outputs	Python Codebase	31,575	36,509
		Open - Ended Tasks		
MultiDoc2Dial	Goal-oriented dialogues	Grounded documents	3,905	7888
Qasper	QA on papers	NLP papers	5,019	6,547
LongFQA*	QA on earning call	Finance	6,032	7824
NQ	QA from Google Search	Wikipedia	23,698	47,726
CUAD	Extracting key information	Law	30,966	68,625
NarrativeQA	QA on narratives	Gutenberg	62,335	210,541
Multi-News	Multi-doc Summarization	Multiple News articles	7,320	19,278
GovReport	Single-doc Summarization	Government reports	7,495	27,128
BigPatent	Single-doc Summarization	Lengthy patents	7,718	12,867
SummScreen	Transcripts Summarization	TV series transcripts	10,688	14,544
Openreview†	Paper writing & reviewing	Papers from Openreview	11,170	33,303
QMSum	Query-based summarization	Meeting transcripts	16,692	33,310
SPACE†	Aspect-based summarization	Reviews on Hotels	19,978	22,158

An et al. L-Eval: Instituting Standardized Evaluation for Long Context Language Models. 2023.

LongBench（清华）

- 引入代码和上下文学习任务，包含多个中文任务
- LongBench-E子集：采用更均匀的长度分布





Bai et al. LongBench: A Bilingual, Multitask Benchmark for Long Context Understanding. 2023.

BAMBOO（人大）

BAMBOO

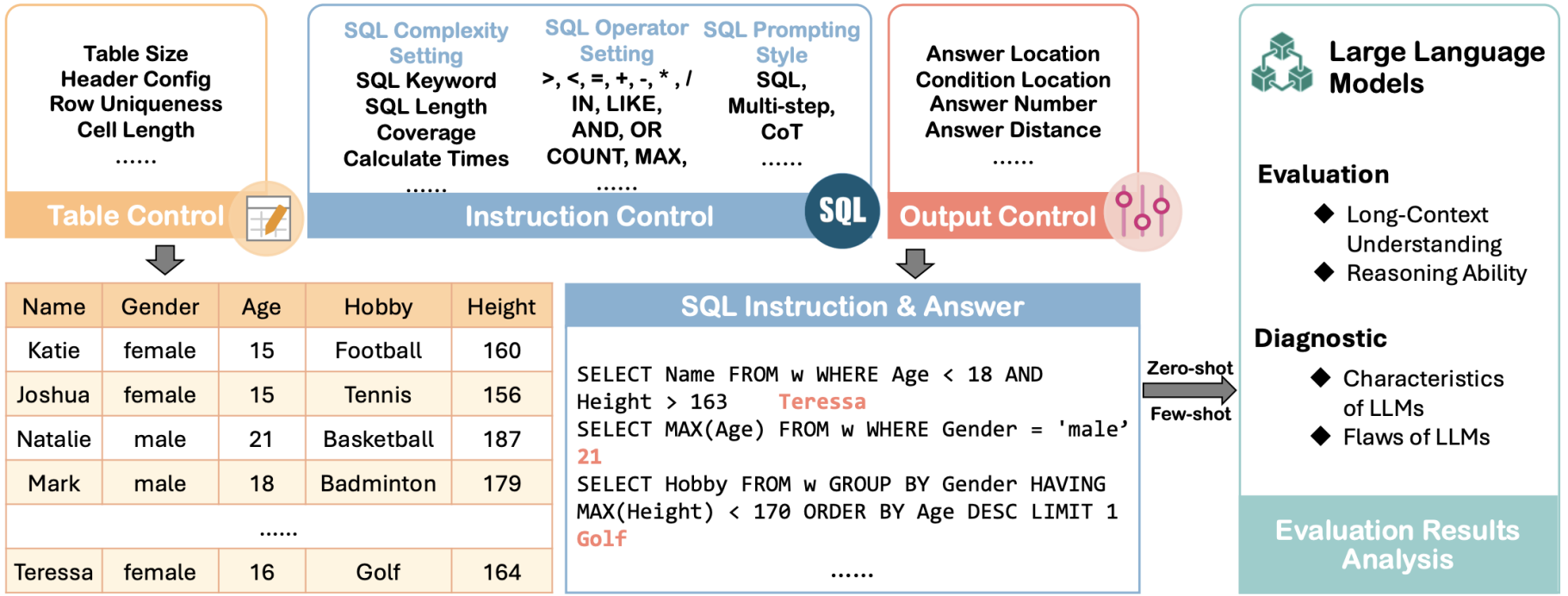
- 基于2023年的新文章构建数据，避免数据污染
- 发现了“扩展税”的存在

Dataset	#Len	#Example	Metric	Domain
AltQA	2297/8320	200/200	accuracy	Wikipedia
PaperQA	2330/4866	40/40	accuracy	Paper
MeetingQA	2195/7951	100/100	accuracy	Meeting
SenHallu	2297/4619	200/200	P/R/F1	Paper
AbsHallu	2415/4699	200/200	P/R/F1	Paper
ShowsSort	2046/4506	200/200	CI	TV Shows
ReportSumSort	3753/8309	150/150	CI	Reports
ShowsPred	1771/3641	100/100	accuracy	TV Shows
MeetingPred	2960/9313	100/100	accuracy	Meeting
PraviteEval	1701/3376	152/152	pass@1	Code

Dong et al. BAMBOO: A Comprehensive Benchmark for Evaluating Long Text Modeling Capacities of Large Language Models. 2023.

S3Eval（自动化所）

- 灵活的人造复杂任务
- 从表格、指令、输出多方面构造SQL查询任务

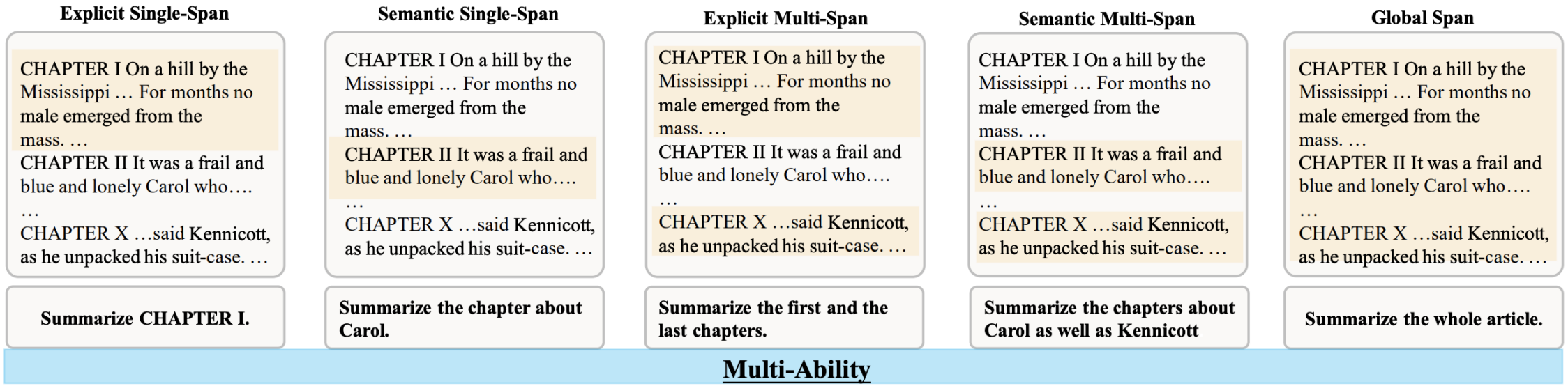


Lei et al. S3Eval: A Synthetic, Scalable, Systematic Evaluation Suite for Large Language Models. 2023.

M4LE（港中文，华为）

M^{4}LE

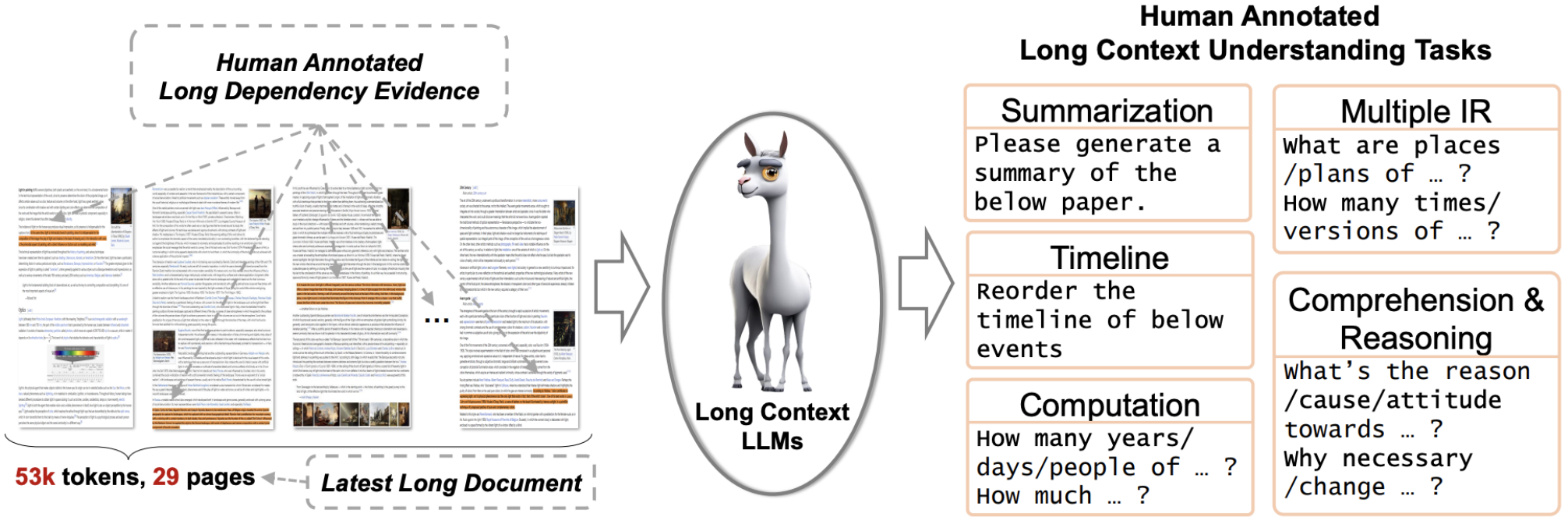
- 不同粒度的关注点：单片段、多片段、全局
- 包含中文任务



Kwan et al. M4LE: A Multi-Ability Multi-Range Multi-Task Multi-Domain Long-Context Evaluation Benchmark for Large Language Models. 2023.

LooGLE（通院）

- 多样的长距离依赖任务
- 目前最长的平均输入长度（19,367词）



Li et al. LooGLE: Can Long-Context Language Models Understand Long Contexts?. 2023.

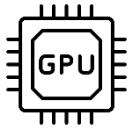
长文本评测套件的选取

- 用于评估中文能力（LongBench、M4LE）
- 更平衡的评估策略（L-Eval、LongBench）
- 避免数据泄露（BAMBOO、S3-Eval）
- 关注长距离依赖（LooGLE、M4LE）

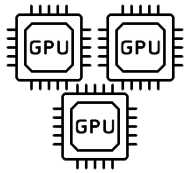
目录

内容总结

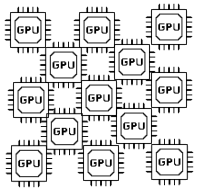
- 技术路线选择



  - 训练资源不支持更新模型参数 → Retrieval Augmentation
    - 即插即用，但长距离依赖的建模能力有限



  - 训练资源支持对已有模型小规模训练 → Interpolation
    - 以适中的代价充分利用现有模型



  - 训练资源支持从头训练模型 → Sparse Attention
    - 计算开销少，但加速框架需要额外适配

展望与挑战

- 模型支持的上下文长度是否有极限?
- 原生训练和插值扩展的长文本模型在能力上的区别?
- 超长文本（32k+）处理能力如何评价?

参考文献

- Xiao et al. Efficient Streaming Language Models with Attention Sinks. 2023.
- Chevalier et al. Adapting Language Models to Compress Contexts. 2023
- Xu et al. Retrieval meets Long Context Large Language Models. 2023.
- Chen et al. Walking Down the Memory Maze: Beyond Context Limit through Interactive Reading. 2023.
- Khattab et al. Demonstrate-Search-Predict. 2022
- Beltagy et al. Longformer: The Long-Document Transformer. 2020
- Kitaev et al. Reformer: The Efficient Transformer. ICLR 2020
- Ding et al. LONGNET: Scaling Transformers to 1,000,000,000 Tokens. 2023
- Chen et al. Extending Context Window of Large Language Models via Positional Interpolation. 2023.
- Xiong et al. Extending Context Window of Large Language Models via Positional Interpolation. 2023.
- Press et al. Train Short, Test Long: Attention with Linear Biases Enables Input Length Extrapolation. ICLR 2022.
- Kočiský et al. "The narrativeqa reading comprehension challenge." TACL 2018
- Dasigi et al. "A dataset of information-seeking questions and answers anchored in research papers." , NAACL 2021
- Liu et al. RepoBench: Benchmarking Repository-Level Code Auto-Completion Systems. 2023.
- Shaham et al. ZeroSCROLLS: A Zero-Shot Benchmark for Long Text Understanding. 2023.
- An et al. L-Eval: Instituting Standardized Evaluation for Long Context Language Models. 2023.
- Bai et al. LongBench: A Bilingual, Multitask Benchmark for Long Context Understanding. 2023.
- Dong et al. BAMBOO: A Comprehensive Benchmark for Evaluating Long Text Modeling Capacities of Large Language Models. 2023.
- Lei et al. S3Eval: A Synthetic, Scalable, Systematic Evaluation Suite for Large Language Models. 2023.
- Kwan et al. M4LE: A Multi-Ability Multi-Range Multi-Task Multi-Domain Long-Context Evaluation Benchmark for Large Language Models. 2023.
- Li et al. LooGLE: Can Long-Context Language Models Understand Long Contexts?. 2023.

北京大学
PEKING UNIVERSITY

谢谢！