基于检索的语言模型

Retrieval-Based LM

杜威 52265901025@stu.ecnu.edu.cn

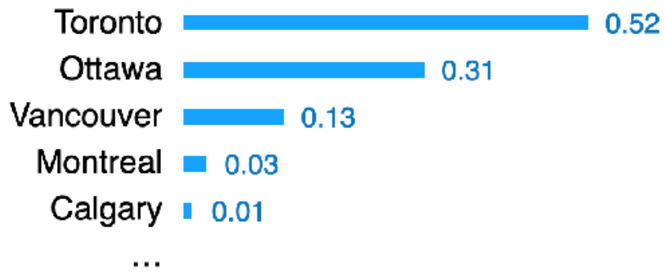
Retrieval-based language models (LMs)

Retrieval-based LMs = Retrieval + LMs

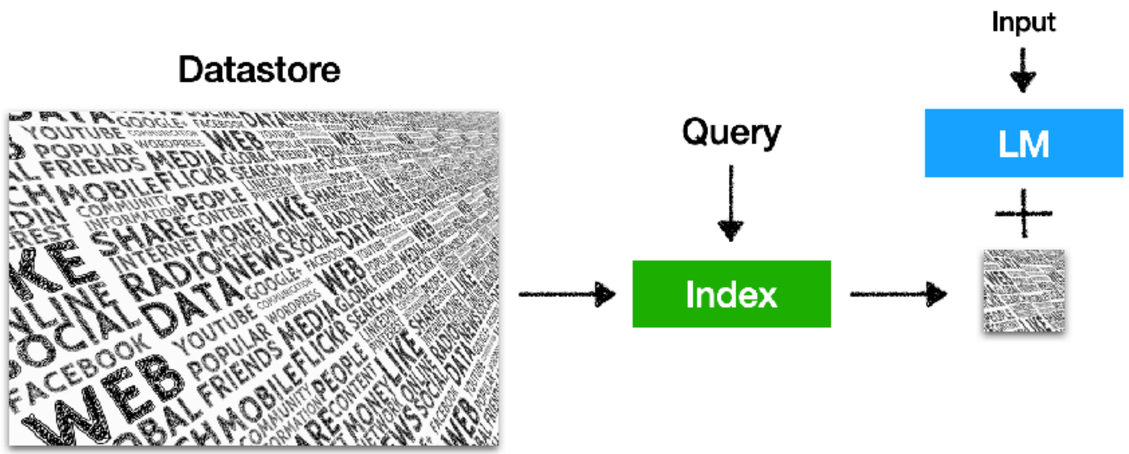
- It is a **language model** $P(x_n \mid x_1, x_2, \cdots, x_{n-1})$

The capital city of Ontario is ___

(can be broadly extended to masked language models or encoder-decoder models)



- It retrieves from an **external datastore** (at least during inference time)



(Also referred to semiparametric and non-parametric models)

From :https://acl2023-retrieval-lm.github.io/slides/1-intro.pdf

Why retrieval-based LMs?

- LLMs can’t memorize all (long-tail) knowledge in their parameters
- LLMs’ knowledge is easily outdated and hard to update
- LLMs’ output is challenging to interpret and verify
- LLMs are shown to easily leak private training data
- LLMs are *large* and expensive to train and run

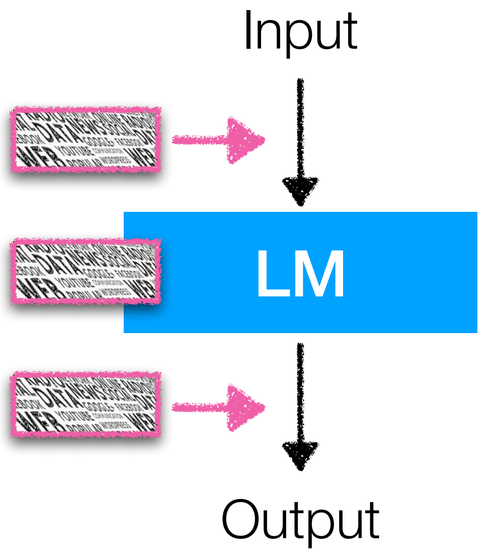
Categorization of retrieval-based LMs

What to retrieve?

Query

Text chunks (passages)?
Tokens?
Something else?

How to use retrieval?



When to retrieve?

From: https://acl2023-retrieval-lm.github.io/slides/3-architecture.pdf

Long-range Language Modeling with Self-retrieval

Ohad Rubin **Jonathan Berant**
The Blavatnik School of Computer Science, Tel Aviv University
{ohad.rubin,joberant}@cs.tau.ac.il

Introduction

Long-range Language Modeling with Self-retrieval

What to retrieval

- **Chunk**
- Tokens
- Others

How to use retrieval

- Input layer
- **Intermediate layers**
- Output layer

When to retrieval

- Once
- **Every n tokens (n>1)**
- Every token

Motivation

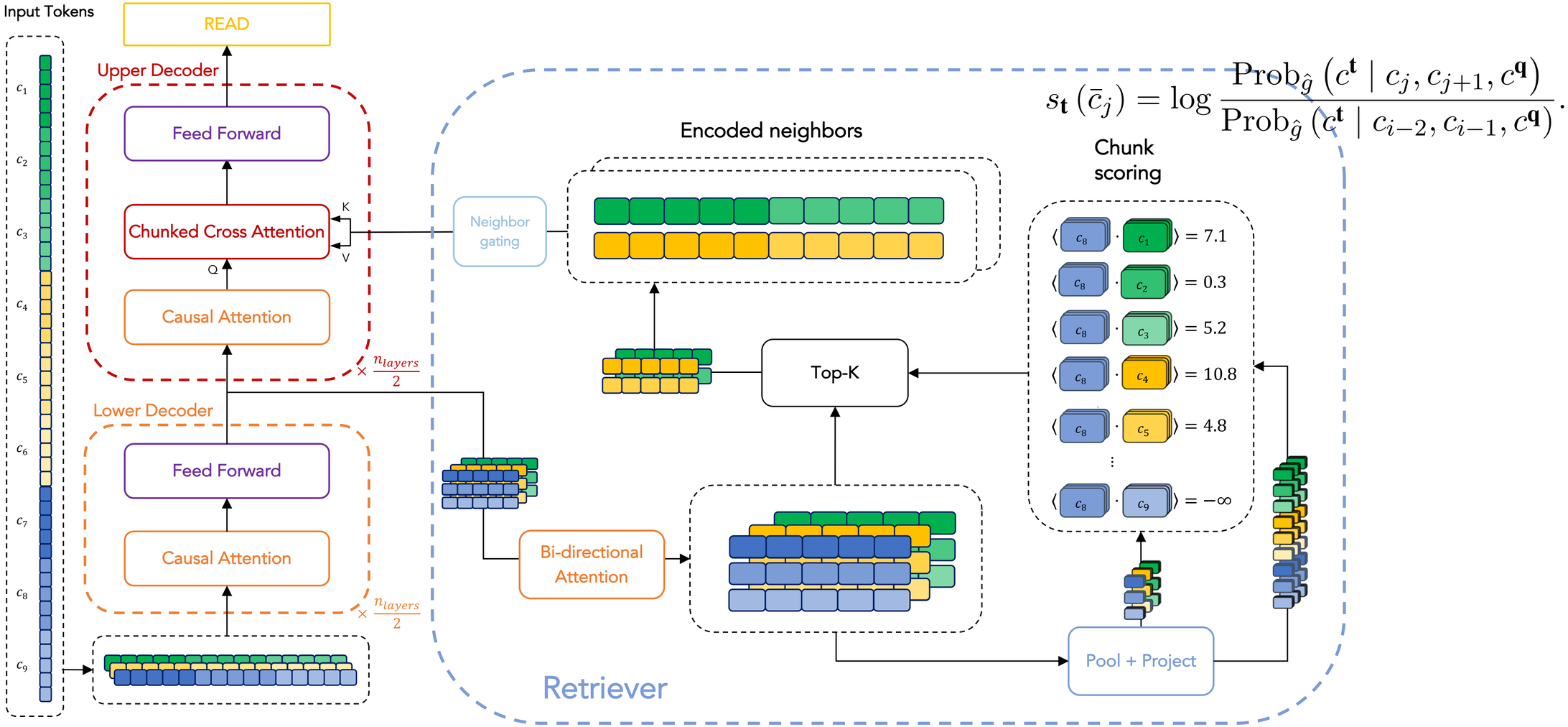
Long-range Language Modeling with Self-retrieval

- Retriever is not trained jointly as a native component of the LM, which limits the ability of the LM and the retriever to adapt to one another.

- Propose the *Retrieval- Pretrained Transformer* (RPT), an architecture and training procedure for jointly training a retrieval-augmented LM from scratch for the task of modeling long texts.(books, documents)

Architecture

Long-range Language Modeling with Self-retrieval



Training

Long-range Language Modeling with Self-retrieval

- LambdaRank Loss

$$L_{\text{ret}}(c^{\mathbf{q}}) = \sum_{\{j,l:\bar{c}_l \in \mathcal{R}^{\mathbf{q}}_{\text{pos}}, s_{\mathbf{t}}(\bar{c}_l) > s_{\mathbf{t}}(\bar{c}_j)\}} \lambda_{jl} \max\left(0, \tau - (s_{\mathbf{q}}(c_l) - s_{\mathbf{q}}(c_j))\right)$$

- Training Loss

$$L_{\text{LM}} + \alpha_{\text{ret}} L_{\text{ret}}$$

Results

Long-range Language Modeling with Self-retrieval

Model	ArXiv	Code	PG19	Books3	Params
TRANSFORMER-XL (OURS)	3.11	2.30	11.48	15.00	202M
RETRO W. BM25 (OURS)	2.94	2.17	11.44	14.60	236M
RPT-LEX	2.92	2.23	11.59	14.32	242M
RPT-SEM	2.77	2.17	10.96	13.91	242M
W. 3 NEIGHBOURS	2.75	2.16	**10.92**	**13.87**	242M
W. 4 NEIGHBOURS	**2.74**	**2.15**	10.93	13.91	242M
MEMORIZING TRANSFORMER	2.92	2.18	10.97	14.40	212M
BLOCK-RECURRENT TRANSFORMER	2.89	2.73	10.95	14.64	212M
RPT-LEX W. ORACLE	2.80	2.12	10.88	13.30	242M
RPT-SEM W. ORACLE	2.69	2.10	10.26	12.74	242M

Table 2: Test set perplexity for all datasets. Unless specified, we use 2 neighbours during inference.

COPY IS ALL YOU NEED

Tian Lan[♢,♡,*] **Deng Cai**[♢,*,†] **Yan Wang**[♢,†] **Heyan Huang**[♡] **Xian-Ling Mao**[♡]
[♢]Tencent AI Lab
[♡]School of Computer Science and Technology, Beijing Institute of Technology
{lantiangmftby,thisisjcykcd,yanwang.branden}@gmail.com
{hhy63,maoxl}@bit.edu.cn

Introduction

COPY IS ALL YOU NEED

What to retrieval

- Chunk
- Tokens
- **Phrase**

How to use retrieval

- Input layer
- Intermediate layers
- **Output layer**

When to retrieval

- Once
- Every n tokens (n>1)
- Every token
- **Adaptive**

Motivation

COPY IS ALL YOU NEED

- Retrieval 能不能在C位?

- LM辅助，Retrieval作为主要输出。

Motivation

COPY IS ALL YOU NEED

- Retrieval 能不能在C位?

- LM辅助，Retrieval作为主要输出。

Method

COPY IS ALL YOU NEED

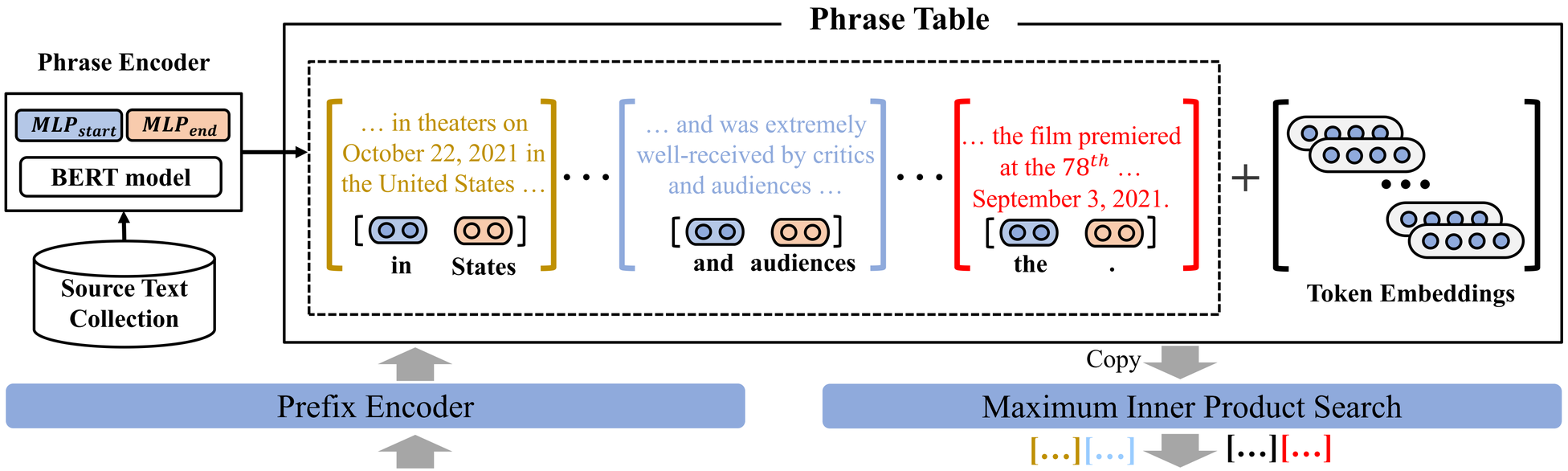


Figure 1: The overview of our proposed CoG. Given the prefix *The Dune film was released*, CoG retrieve 3 phrases (in different colors) from the documents and generates 3 tokens (*Before*, *that*, and the comma ,) from the fixed vocabulary to form the whole generation.

Method

COPY IS ALL YOU NEED

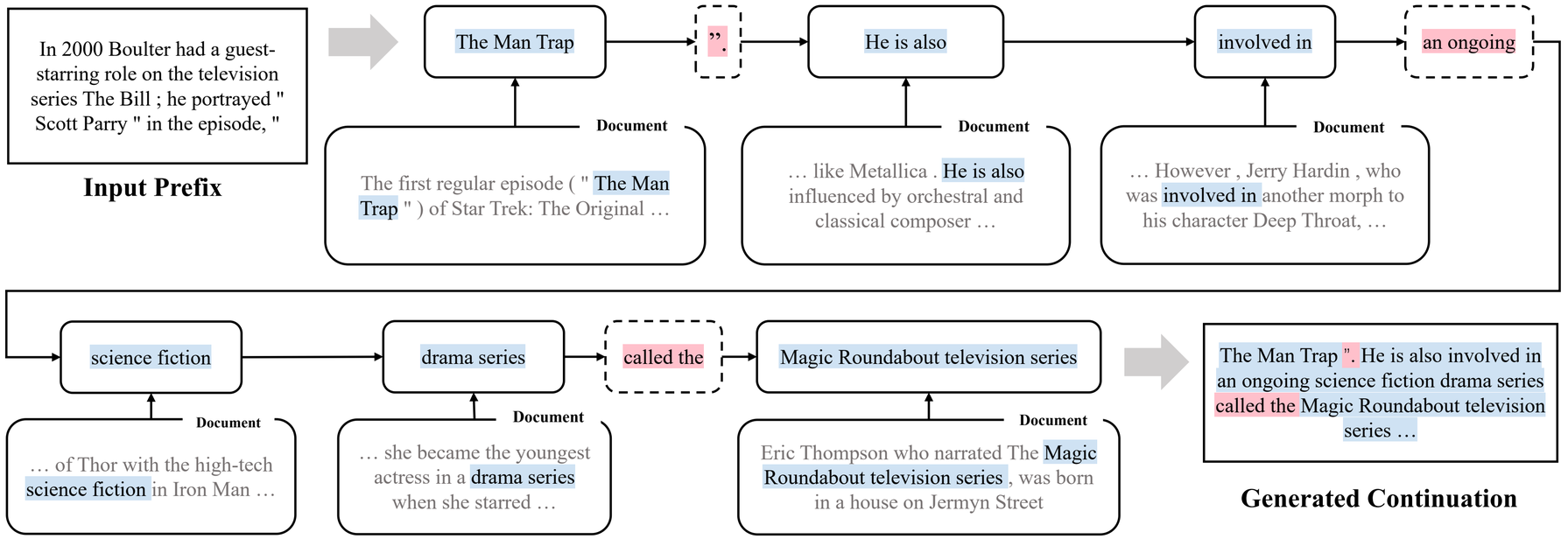


Figure 2: An example generated by CoG on the test set of WikiText-103. The dotted squares denote that the content (highlighted in red)is copied from the token vocabulary, and the solid squares denote that the content (highlighted in blue) is copied from other documents.

Method

COPY IS ALL YOU NEED

假设我们的语料包含n个文档 $\{D^1, \ldots, D^n\}$，对于每个文档 D^i，我们可以提取一个长度为 $e - s + 1$ 的片段 $k = D^i_{s:e}$（下文称为phrase，与原论文对齐），其中e和s代表了这个短语在文档中的起始位置和结束位置。我们将源文本集合中的所有短语表示为 $\mathcal{P}$，对于任意给定的句子前缀，我们的目标都是选择一个合适的短语来续写这个前缀。我们的模型会使用一个短语编码器 $\text{PhraseEncoder}\left(s, e, D^i\right)$ 为每个短语计算一个上下文化表示，那么所有可行的短语会形成一个Phrase Table: $\{(k,\ p_k) \mid k\ \in P\}$。在推理时，我们利用前缀的表示与短语表示的点积来计算前缀与短语的合适程度：

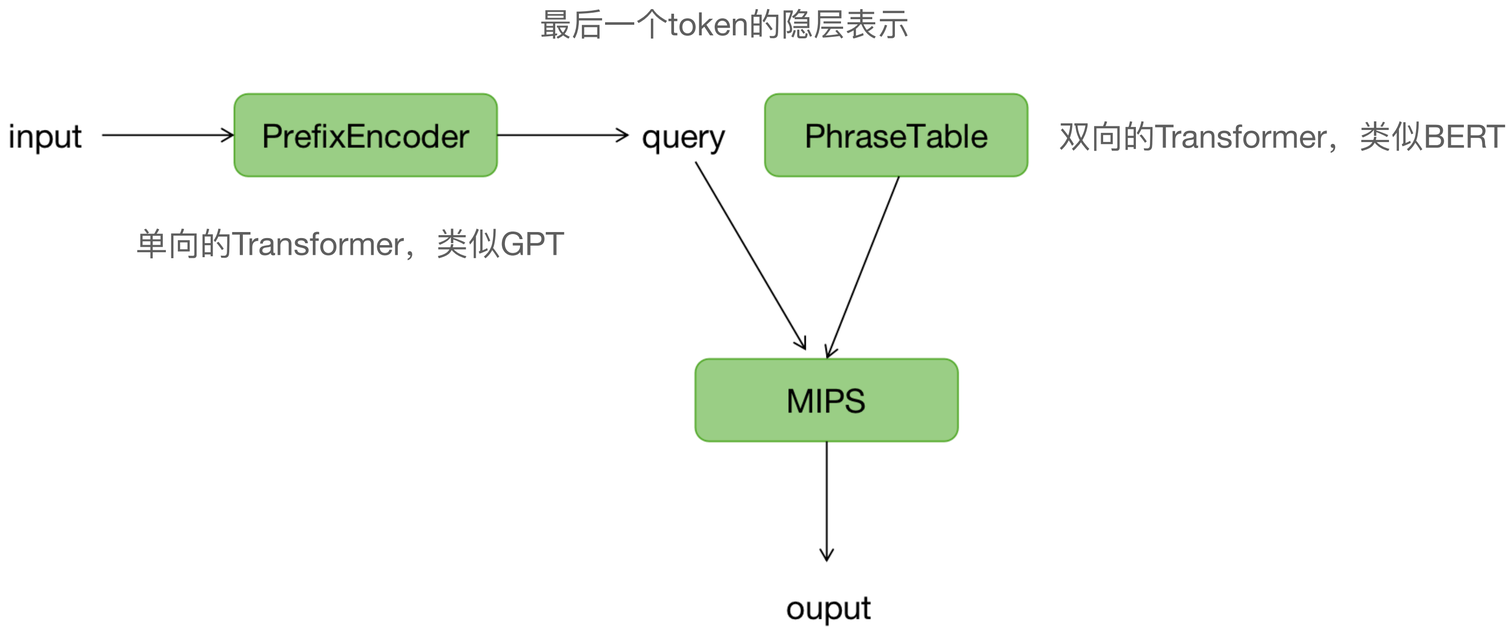
$$\mathcal{D}_{\text{start}} = \text{MLP}_{\text{start}}(\mathcal{D}), \mathcal{D}_{\text{end}} = \text{MLP}_{\text{end}}(\mathcal{D}).$$

$$\text{PhraseEncoder}(s, e, D) = [\mathcal{D}_{\text{start}}[s]; \mathcal{D}_{\text{end}}[e]] \in \mathbb{R}^d$$

$$\mathcal{D}_{\text{start}}, \mathcal{D}_{\text{end}} \in \mathbb{R}^{m \times \frac{d}{2}}$$

Method

COPY IS ALL YOU NEED



Training

COPY IS ALL YOU NEED

$$\mathcal{L}_p = -\frac{1}{n}\sum_{k=1}^{n}\log\frac{\exp(q_k \cdot p_k)}{\sum_{p\in\mathcal{P}_k}\exp(q_k \cdot p_p) + \sum_{w\in V}\exp(q_k \cdot v_w)}$$

其中 $\mathcal{P}_k$ 是可选phrase集合，V是token embedding集合。

另外，为了保留模型的token级别生成能力，我们还有一个标准的自回归模型loss：

$$\mathcal{L}_t = -\frac{1}{m}\sum_{i=1}^{m}\log\frac{\exp(q_i, v_{D_i})}{\sum_{w\in V}\exp(q_i, v_w)}$$

最终模型的loss是前面两者之和：

$$\mathcal{L} = \mathcal{L}_p + \mathcal{L}_t$$

Results

COPY IS ALL YOU NEED

Model	Decoding	MAUVE↑	Rep-2↓	Rep-3↓	Rep-4 ↓	Diversity↑	Latency (s)↓
Transformer	greedy	19.87	43.56	38.55	35.5	22.37	1.32
	nucleus	23.43	5.10	1.33	0.50	93.22	1.48
kNN-LM	greedy	19.92	43.79	38.76	35.69	22.13	10.36
	nucleus	22.50	**3.33**	**0.69**	**0.21**	**95.8**	10.42
RETRO	greedy	21.19	44.65	39.63	36.6	21.19	4.39
	nucleus	22.86	6.21	1.93	0.86	91.19	4.51
COG	greedy	26.01	28.14	23.80	21.40	43.03	**1.29**
	nucleus	**26.14**	7.31	2.66	1.28	89.07	1.54

Table 1: The automatic evaluation on the test set of WikiText-103. As for each model with nucleus sampling, we run 10 times and recorded the average MAUVE and Diversity scores.

Results

COPY IS ALL YOU NEED

Model	Decoding	MAUVE ↑	Diversity ↑
Transformer w/o FT	greedy	20.32	70.66
	nucleus	25.21	93.88
Transformer w/ FT	greedy	23.00	80.52
	nucleus	26.85	90.14
kNN-LM	greedy	23.31	19.85
	nucleus	24.75	**94.60**
RETRO	greedy	18.70	71.14
	nucleus	20.35	94.81
CoG	greedy	21.31	84.32
	nucleus	**28.14**	92.56

Table 4: The automatic evaluation on Law-MT.

Result

Improving language models by retrieving from trillions of tokens

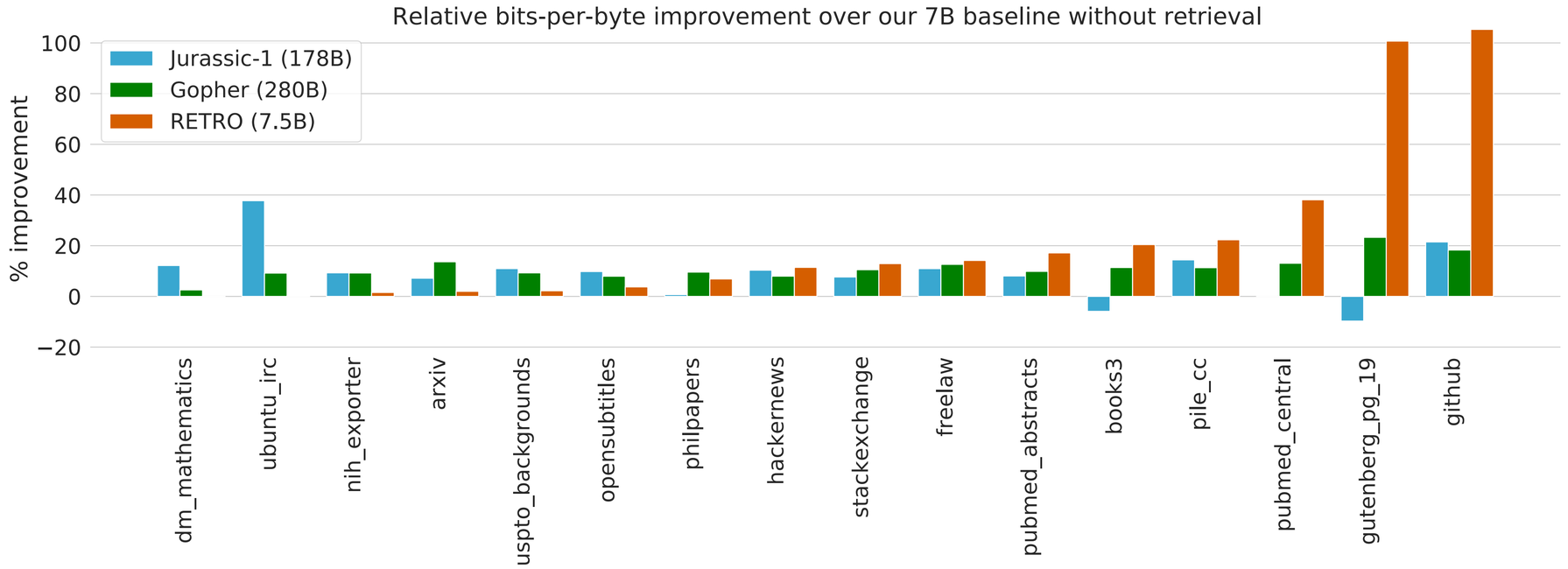


Figure 4 | **The Pile: Comparison of our 7B baseline against Jurassic-1, Gopher, and RETRO.** We observe that the retrieval model outperforms the baseline on all test sets and outperforms Jurassic-1 on a majority of them, despite being over an order of magnitude smaller.

END