# Retrieval is Accurate Generation

# Current Mainstream

- How it works?
  - left-to-right, word-by-word

$$\mathbf{X} = (X_1, \ldots, X_T)$$

$$P(\mathbf{X}) = \prod_t P(X_t \mid X_{<t}) = \prod_t P(X_t \mid C_t)$$

  - select a token from a vocabulary (*fixed, finite, and standalone*)

$$P_\theta(x|c) = \frac{\exp \mathbf{h}_c^\top \mathbf{w}_x}{\sum_{x'} \exp \mathbf{h}_c^\top \mathbf{w}_{x'}}$$

  - RNN, CNN and Transformer

$\mathbf{h}_c$ is a function of $c$, and $\mathbf{w}_x$ is a function of $x$

$\mathbf{h}_c^\top \mathbf{w}_x$ is called a *logit*

# Current Mainstream

- How it works?
  - left-to-right, word-by-word

$$\mathbf{X} = (X_1, \ldots, X_T)$$

$$P(\mathbf{X}) = \prod_t P(X_t \mid X_{<t}) = \prod_t P(X_t \mid C_t)$$

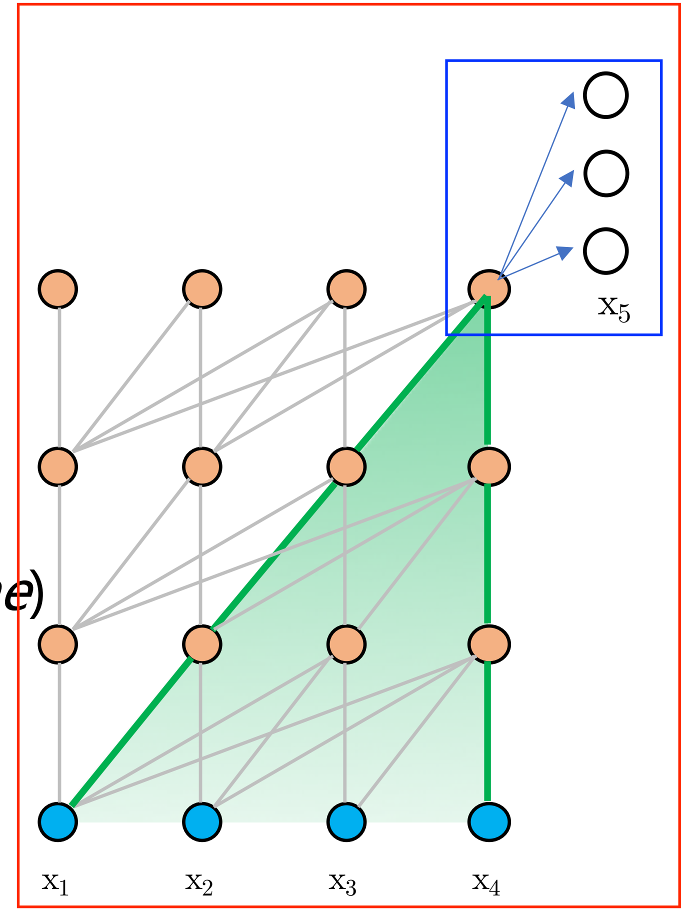
  - select a token from a vocabulary (*fixed, finite, and standalone*)

$$P_{\boxed{\theta}}(x|c) = \frac{\exp \mathbf{h}_c^\top \mathbf{w}_x}{\sum_{x'} \exp \mathbf{h}_c^\top \mathbf{w}_{x'}}$$

  - RNN, CNN and Transformer

$\boxed{\mathbf{h}_c \text{ is a function of } c}$, and $\mathbf{w}_x$ is a function of $x$

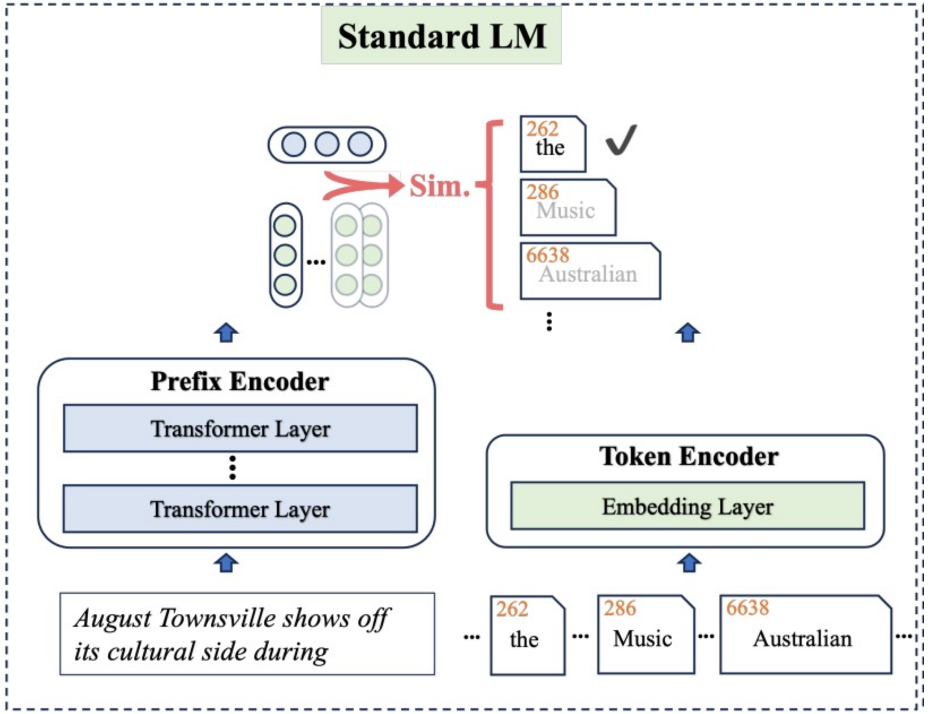
$\boxed{\mathbf{h}_c^\top \mathbf{w}_x \text{ is called a } \textit{logit}}$



# Current Mainstream

- How it works?
  - left-to-right, word-by-word
  - select a token from a vocabulary

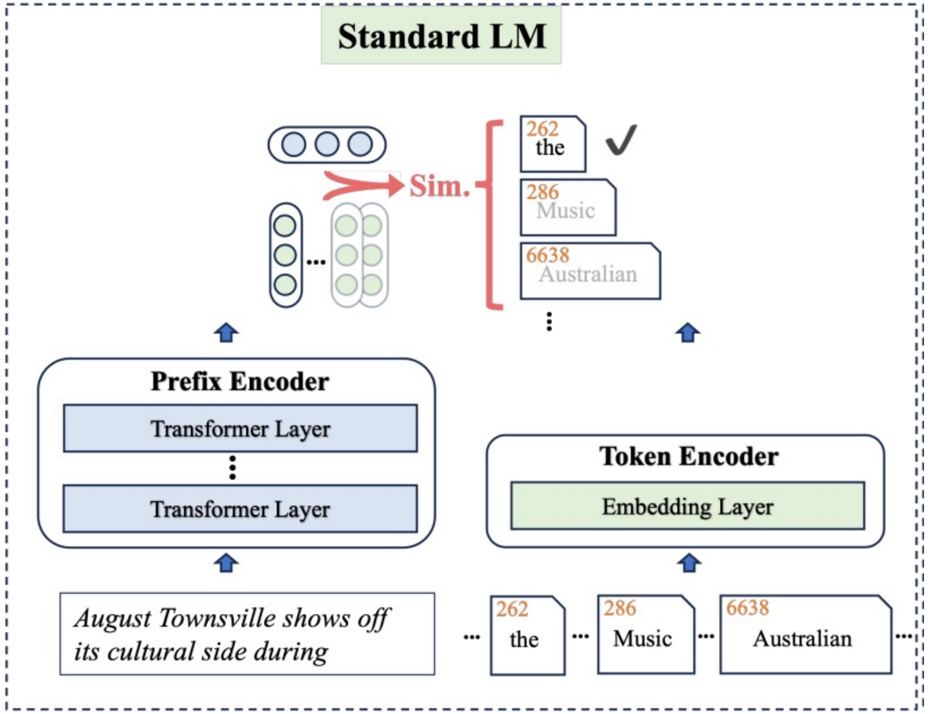
prefix ↔️ tokens in vocabulary
(retrieval)



# Current Mainstream vs Our Proposal (CoG)

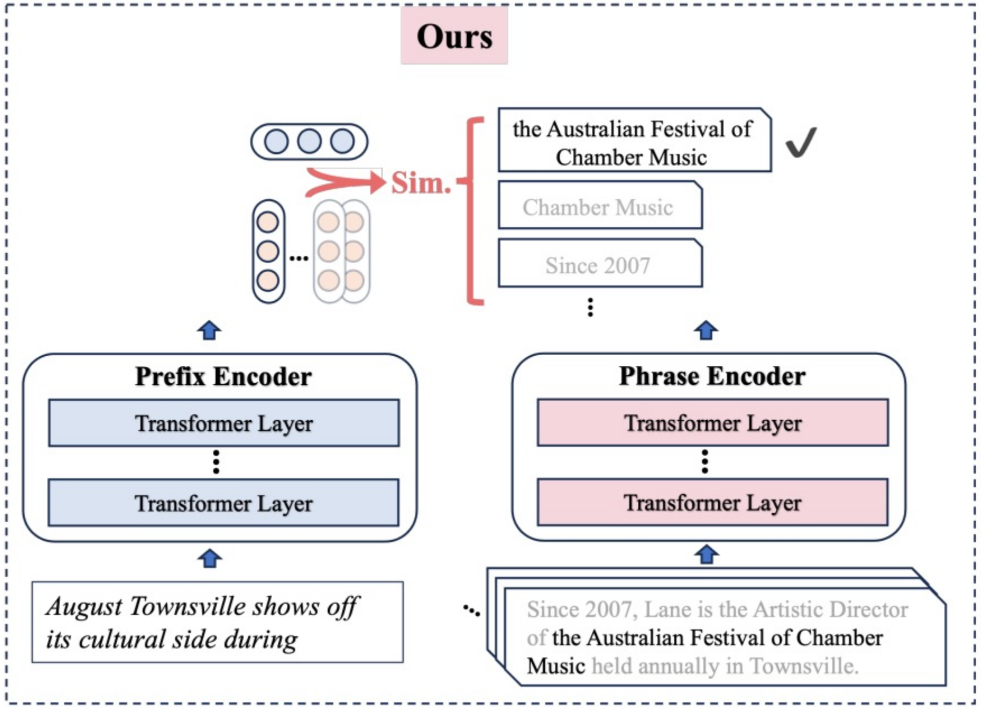
- How it works?
  - left-to-right, word-by-word
  - select a token from a vocabulary

prefix ↔ tokens in vocabulary
(retrieval)




- How it works?
  - left-to-right, phrase-by-phrase
  - select a phrase from memories

prefix ↔ phrases in memories
(retrieval, too)



# Our Proposal (CoG)

- How it works
  - left-to-right, phrase-by-phrase
  - select a phrase from memories

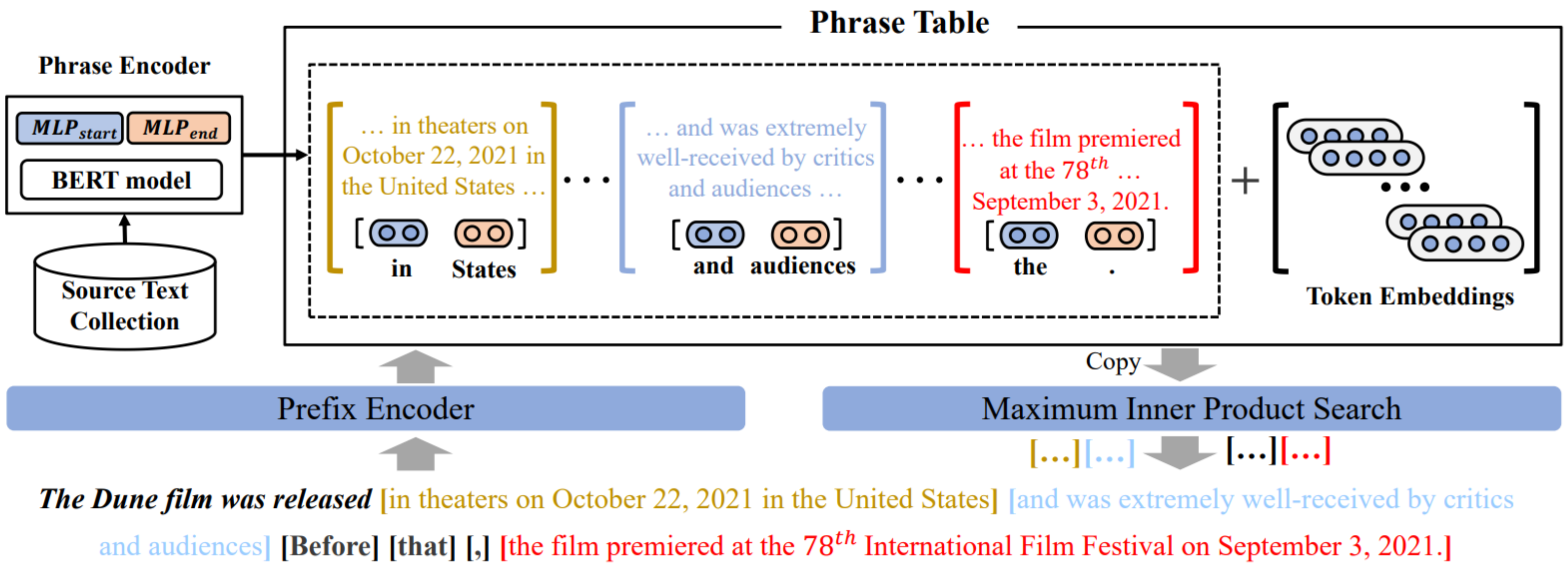
fixed, finite, and standalone vs. dynamic, extensible, and contextualized

<table>
<tr><td colspan="2">Flag burning ... "flag burning" refers only to burning a flag as an act of protest.</td><td>sends</td><td colspan="3">... Each song has a powerful message designed to stop and make you think about your life ...</td></tr>
<tr><td colspan="2">Flag burning is a propaganda tool, such as burning Effigies of world leaders.</td><td>sends</td><td colspan="3">the song ... sends a powerful message through its lyrics, telling listeners to 'keep going' and to fight for ...</td></tr>
<tr><td colspan="2">Flag burning ... situation escalated further after the parliamentary elections in ...</td><td colspan="4">... for its "very bold move making tonight plant-based. It really sends a powerful message." Soon after, Critics' Choice and SAG ...</td></tr>
<tr><td>Flag</td><td>burning</td><td>sends</td><td>a</td><td>powerful</td><td>message</td></tr>
</table>

**(Current mainstream is a specialized case of COG)**

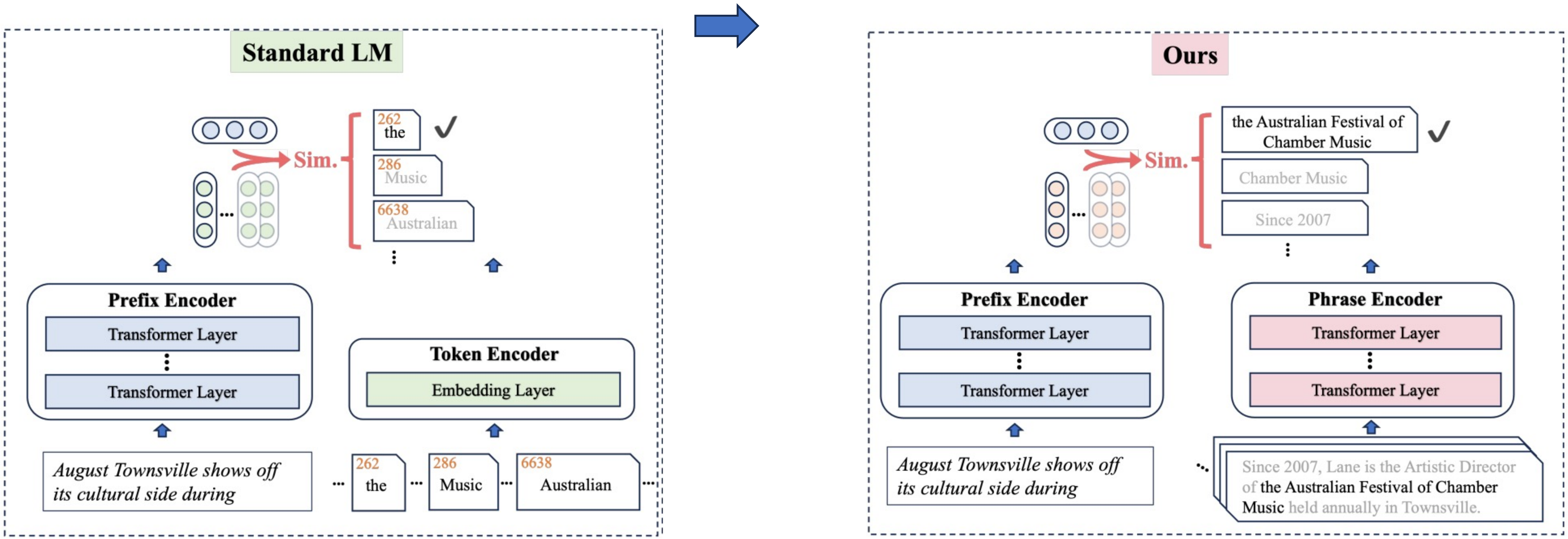
# Our Proposal (CoG)

- How it works
  - left-to-right, phrase-by-phrase
  - select a phrase from memories
- Advantages    fixed, finite, and standalone vs. dynamic, extensible, and contextualized
  - **Accuracy**: the semantics of phrases are enhanced by their surrounding contexts
  - **Interpretability**: each retrieved phrase can be traced back to its original source
  - **Extensibility**: memories can be edited and used in a *plug-and-play* fashion



# Our Proposal (CoG)

- **Efficiency?**
  - fewer generation steps, multiple tokens at one step
  - more parameters moved to the target side, phrase representations can be pre-computed



# Our Proposal (CoG)

- Key challenges:
  - The construction of training data.
    - How a string of text is segmented?
    - What is the best source for each phrase?
  - Scaling up to millions of millions of documents and phrases.
    - filtering, clustering, and deduplication
    - dimensionality reduction, quantization
    - fast vector search algorithm
    - hierarchical search

# Our Proposal (CoG)

- The construction of training data.
  - How a string of text is segmented?
  - What is the best source for each phrase?
- Our method:
  - Linguistics-motivated heuristics
    - Syntactic structure
    - Distributional sparsity
    - Semantic similarity
  - Iterative self-reinforcement
    - To adjust the generation paths based on its own preferences

# Our Proposal (CoG)

- Training objective
  - InfoNCE + negative sampling N(p)

$$\mathcal{L}_p = \frac{\exp(E_p(p) \cdot E_c(s))}{\exp(E_p(p) \cdot E_c(s)) + \sum_{t \in N(p)} \exp(E_p(p) \cdot E_c(t))}$$

- Negative set construction
  - In-batch negative
  - Hard negative
    - top phrases that are not chosen in iterative self-reinforcement (top-k approximation!)

# Our Proposal (CoG)

- Inference
  - We employ FAISS, a library for vector similarity search, for efficient retrieval
  - Continuation Generation:
    - top-k recall -> softmax (next phrase prob. distribution) -> top-p sampling or others
  - Likelihood Estimation:
    - Summing all possible generation paths
    - The generation prob. of each step is calculated as above
    - Compute efficiently using dynamic programming

*The Moon rises:*
*1. → The→moon→rises*
*2. → The moon→rises*
*3. → The moon rises*

# Experiments

- Baselines

We compare the proposed method with standard LM in the zero-shot setting, also drawing the following state-of-the-art retrieval-augmented methods as baselines:

**Base LM** is the standard token-level language model using the Transformer (Vaswani et al., 2017) architecture. We fine-tune the pre-trained GPT-2[5] (Radford et al., 2019).

$k$**NN-LM** (Khandelwal et al., 2020) is a retrieval-augmented LM that interpolates the next-token distribution of the base LM with a $k$-nearest neighbors ($k$NN) model.

**RETRO** (Borgeaud et al., 2022)[6] is a retrieval-augmented LM incorporated with a pre-trained document retriever, a document encoder and a cross-attention mechanism.

**CoG** (Lan et al., 2023)[7] is another retrieval-augmented LM that adopts a two-stage search pipeline. It first retrieves semantically-relevant documents, and then considers all $n$-grams within them as candidate phrases.

- Tasks
  - Knowledge-intensive tasks
    - OpenbookQA, ARC-Chanllenge, TruthfulQA, MedMCQA, MedUSIMLE
  - Open-ended text generation
- Phrase Index: Wikipedia, 137, 101, 097

# Experiments

- Knowledge-intensive tasks

| | TruthfulQA | OpenbookQA | ARC-Challenge | MedMCQA | Med-USMILE |
|---|---|---|---|---|---|
| Base LM (w/o FT) | 30.27 | 22.67 | 24.52 | 27.96 | 24.89 |
| Base LM | 29.73 | 23.47 | 23.92 | 28.33 | 24.19 |
| $k$NN-LM | 30.27 | 22.93 | 24.82 | 27.96 | 24.72 |
| RETRO | 27.53 | 26.13 | 22.21 | 25.68 | 25.33 |
| CoG | 34.11 | 35.47 | 27.24 | 29.07 | 25.07 |
| Ours | **34.27** | **36.27** | **28.27** | **29.44** | **25.69** |

- Switch to large index (3x) **without training**

| | TruthfulQA | OpenbookQA | ARC-Challenge | MedMCQA | Med-USMILE |
|---|---|---|---|---|---|
| Ours | 34.27 | 36.27 | **28.27** | 29.44 | 25.69 |
| w/ enlarged index | **39.59** | **37.07** | 27.14 | **31.63** | **27.87** |

- Switch to specialized index **without training**

| | MedMCQA | Med-USMILE |
|---|---|---|
| Base LM (FT) | 28.79 | 25.15 |
| General index | 29.44 | 25.69 |
| Medical index | **29.50** | **26.38** |

# Experiments

- Open-ended text generation

| | **MAUVE↑** | **Coherence↓** | **Diversity↑** | **Latency↓** |
|---|---|---|---|---|
| Base LM (w/o FT) | 69.68 | 3.64 | 83.14 | 1.00x |
| Base LM | 42.61 | 3.56 | 78.72 | 1.00x |
| $k$NN-LM | 13.07 | 5.63 | **88.10** | 6.29x |
| RETRO | 62.39 | 4.82 | 80.96 | 1.51x |
| CoG | 52.27 | **2.08** | 55.04 | 4.40x |
| Ours | **81.58** | 3.25 | 76.26 | **1.29x** |

Table 4: Results for open-ended text generation.

| **Model** | **Fluency** | **Coherence** | **Informativeness** | **Grammar** |
|---|---|---|---|---|
| Base LM (w/o FT) | 2.91 | 2.33 | 2.35 | 3.00 |
| Base LM | 2.81 | 2.37 | 2.40 | 2.79 |
| Ours | **2.95** | **2.70** | **2.67** | **3.02** |

Table 5: Human evaluation results.