Improving language models by retrieving from trillions of tokens

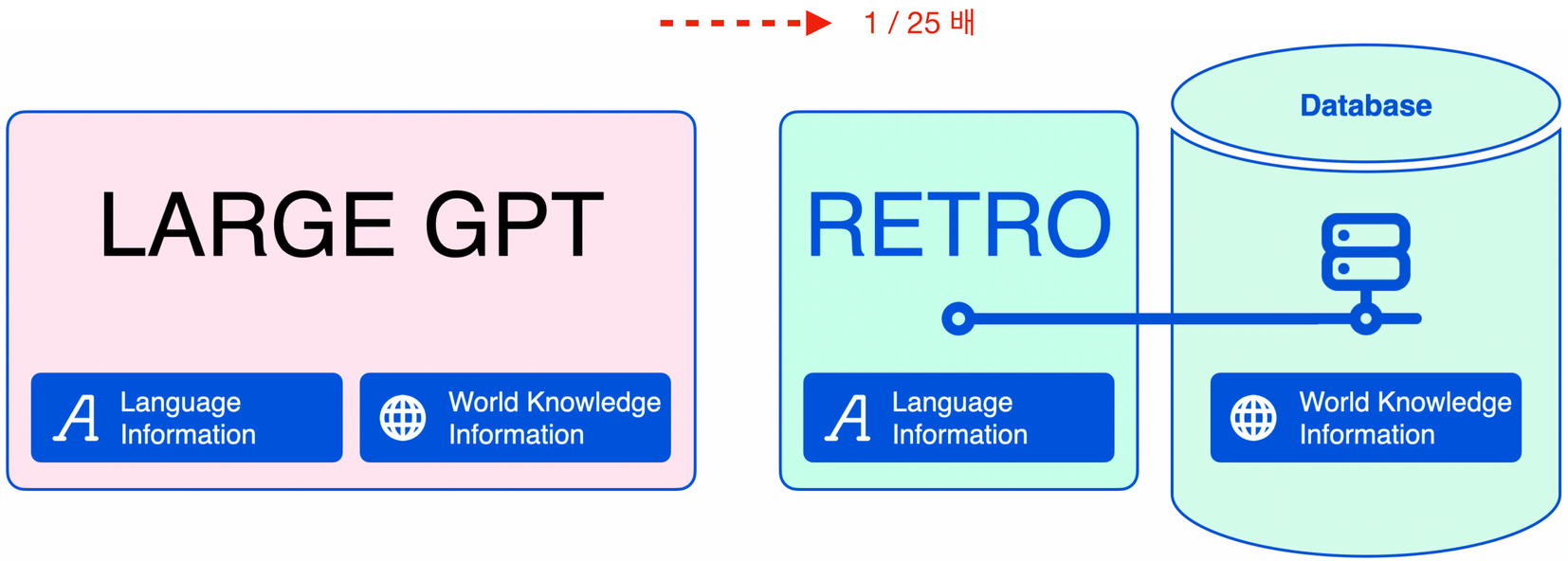
DeepMind, arXiv Feb 2022, 53회 인용

AI2-1 이현경 2022.12.20

Retrieval-Enhanced Transformer

검색 기능이 향상된 자동회귀 언어모델

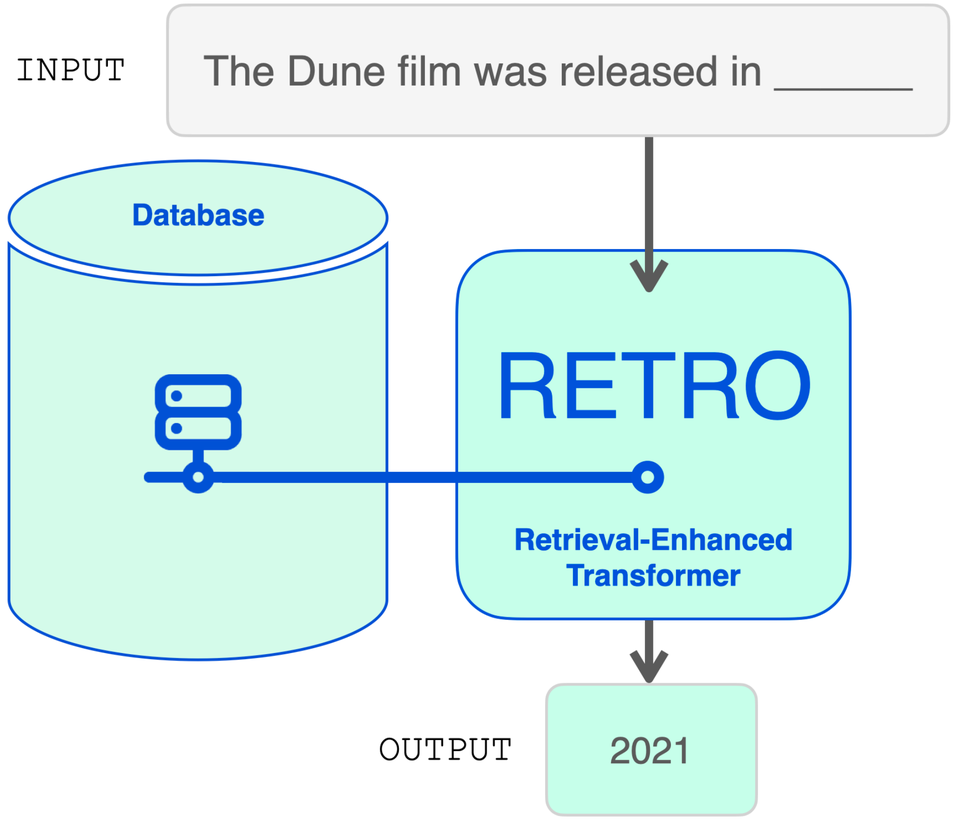
- 외부 KB (1.75T token)
- Transformer 기반 모델



Retrieval-Enhanced Transformer

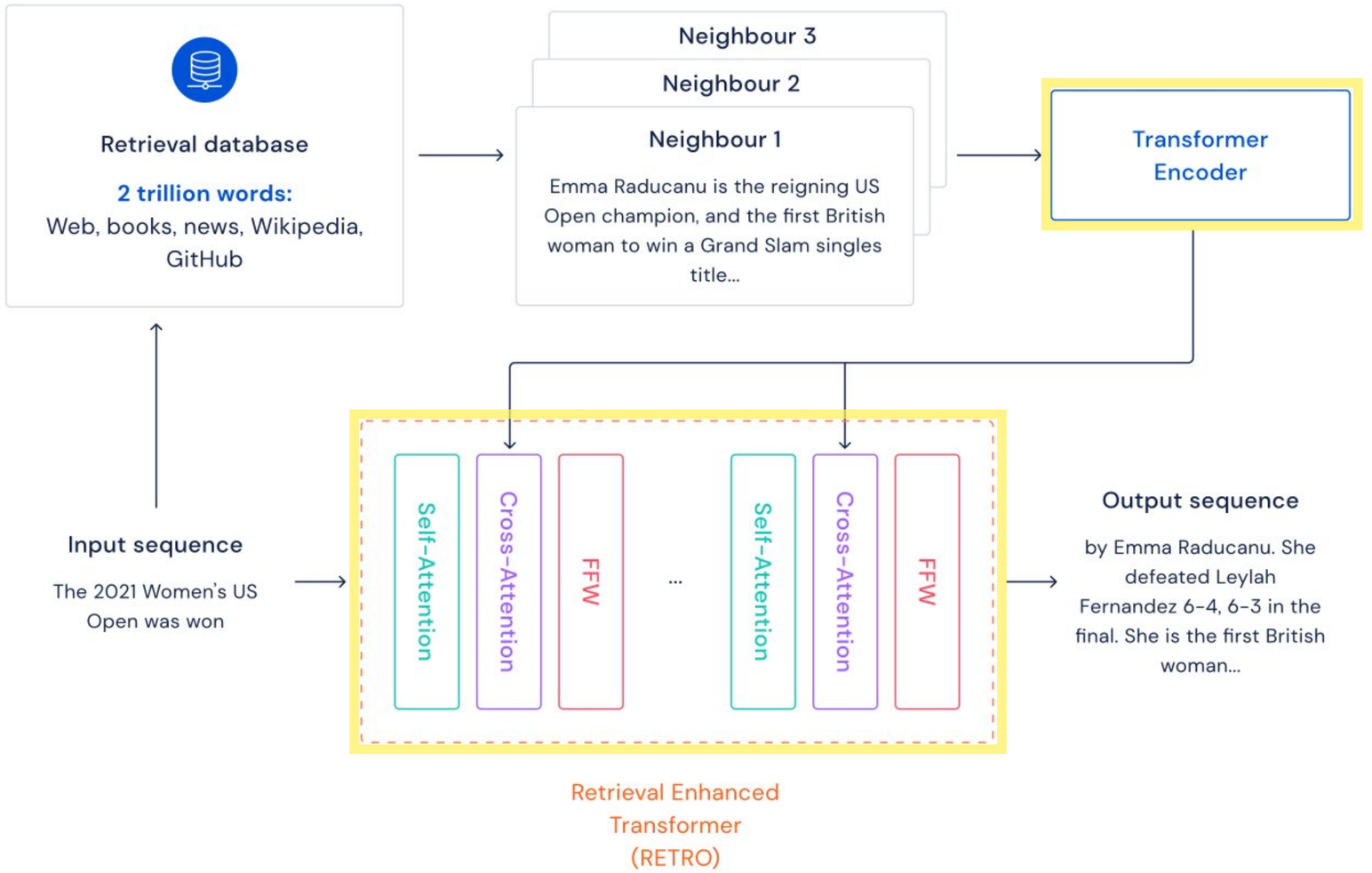
검색 기능이 향상된 자동회귀 언어모델

- Input sequence에 retrieval database 검색 결과를 보강
- Chunked cross-attention (CCA)



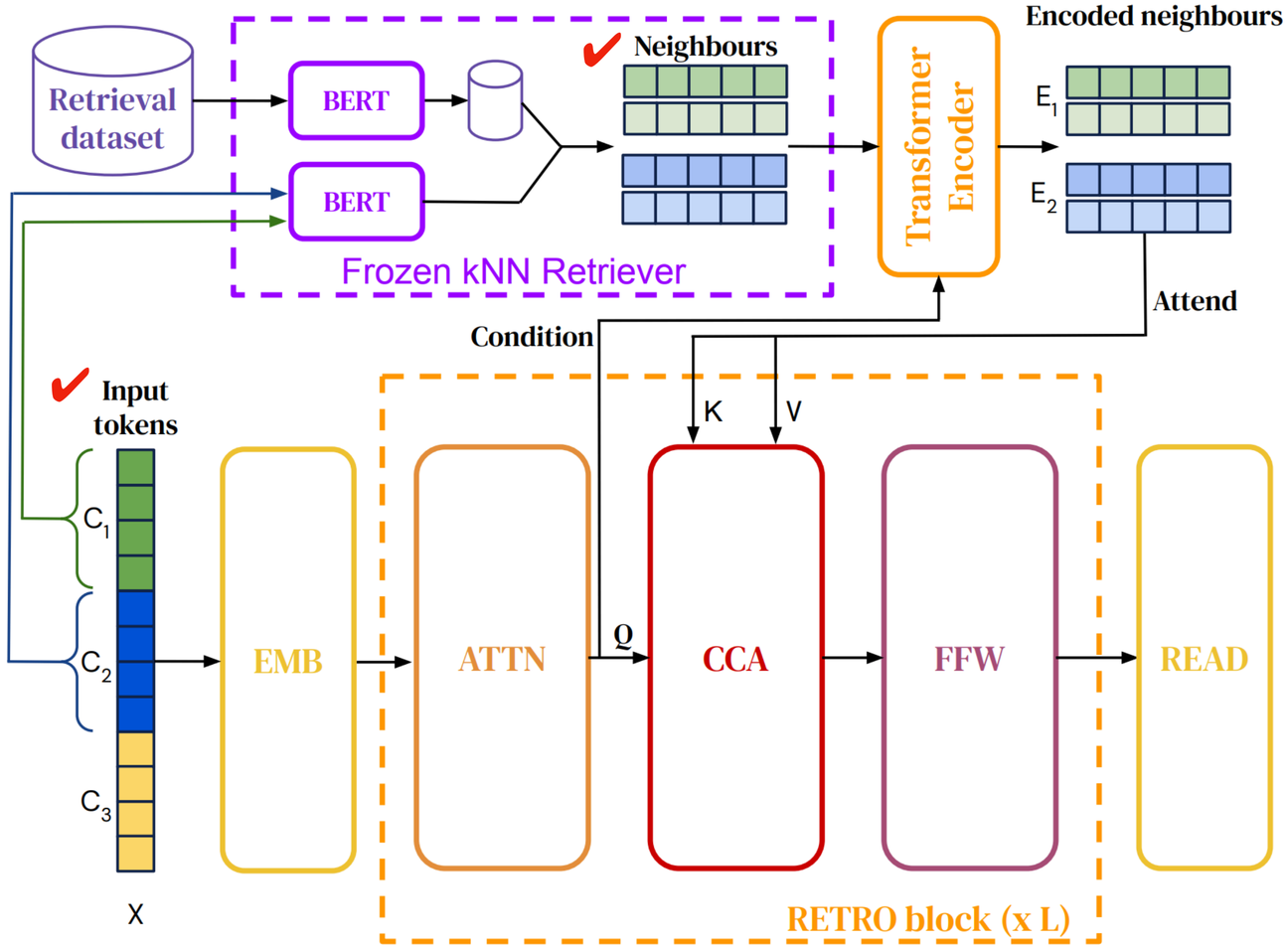
Retrieval-Enhanced Transformer

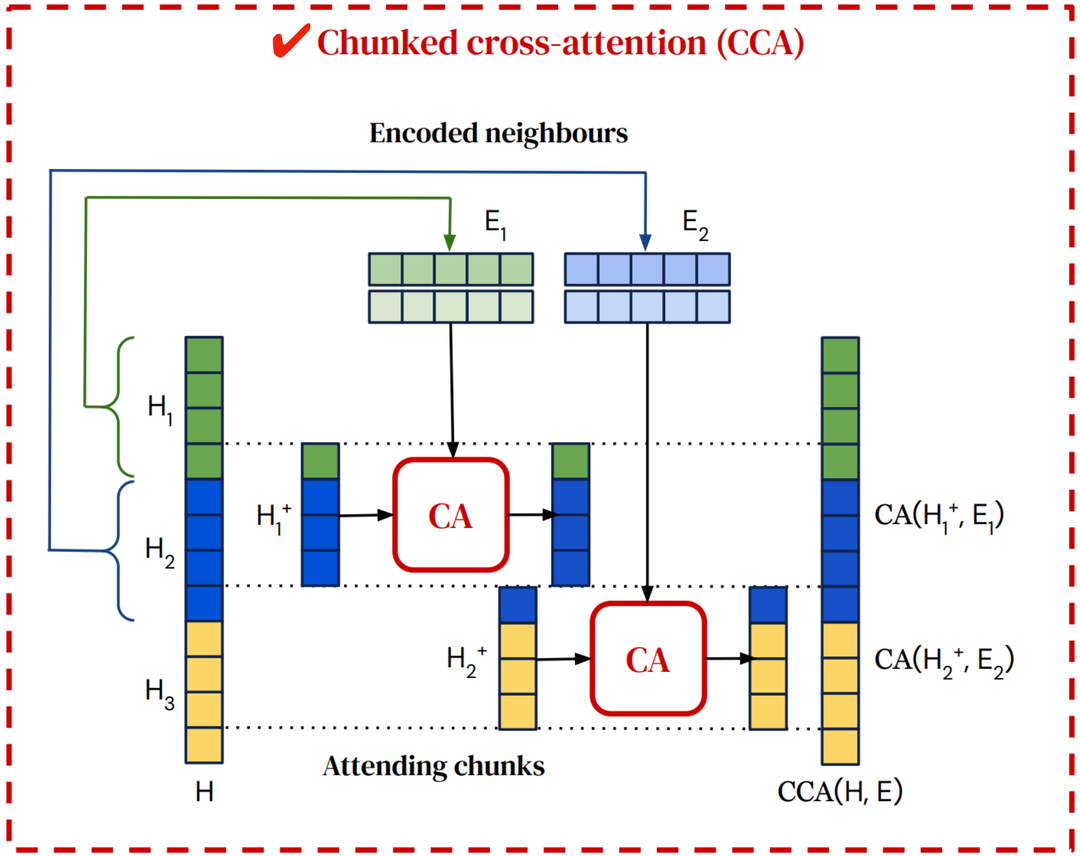
Encoder-decoder fully feed-foward



Retrieval-Enhanced Transformer

Encoder-decoder fully feed-foward





Retrieval-Enhanced Transformer

Autoregressive token models

- Retrieval-enhanced sequence log-likelihood

$$L\left(X|\theta,\mathcal{D}\right) \triangleq \sum_{u=1}^{l}\sum_{i=1}^{m} \ell_\theta\left(x_{(u-1)\,m+i}|(x_j)_{j<(u-1)\,m+i},\ (\mathrm{RET}_{\mathcal{D}}(C_{u'}))_{u'<u}\right).$$

  - 이전 토큰들
  - 이전 chunk들에서 검색해온 외부데이터

Retrieval-Enhanced Transformer

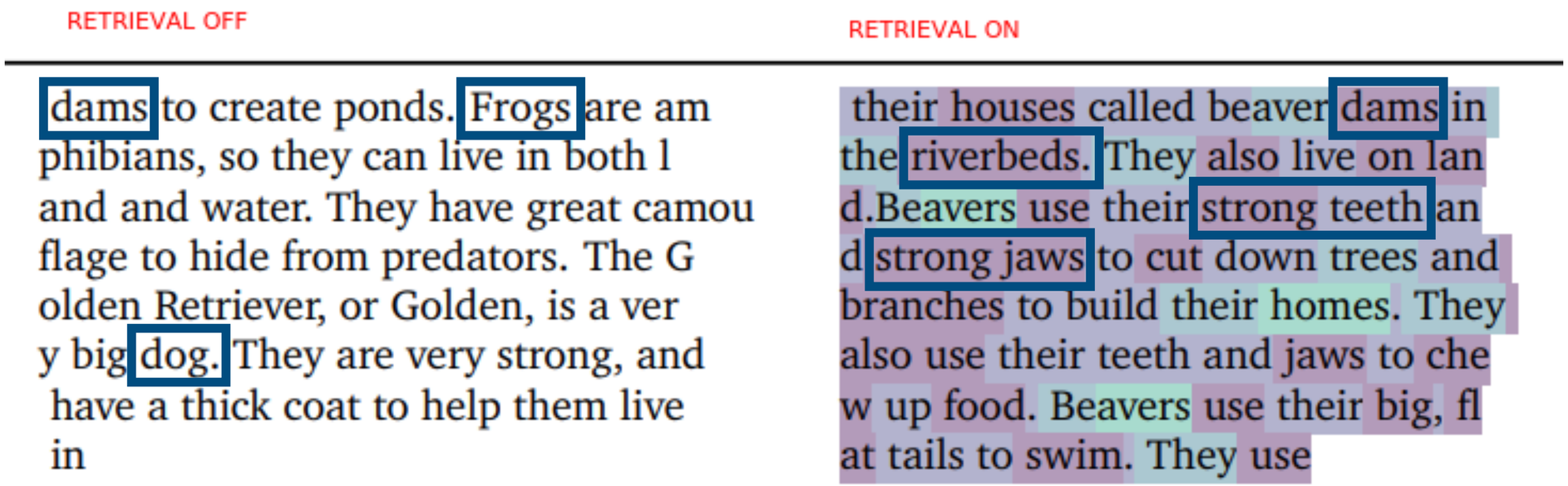
생성 예시

- Prompt

Beavers are interesting animals that live near rivers. They build

Retrieval-Enhanced Transformer
생성 예시

- Beaver -> Frogs -> Dog
- Beaver (dams, riverbed, strong teeth, jaw, …)



Retrieval-Enhanced Transformer

Contributions

- Pre-trained Frozen(BERT) 기반의 retrieving

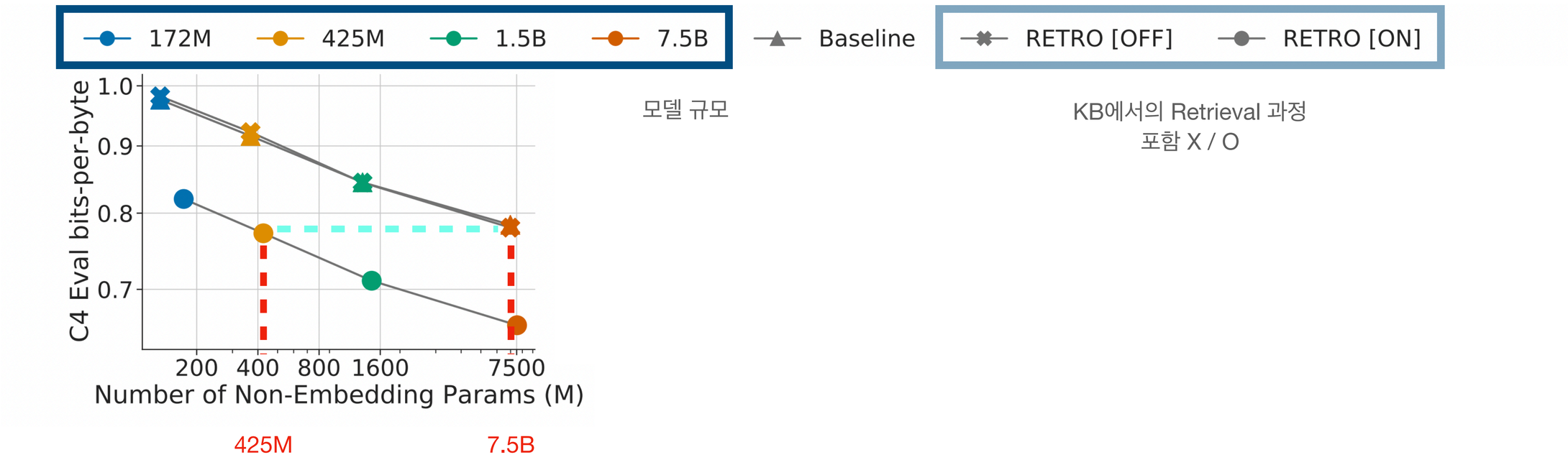
Table 3 | **Comparison of RETRO with existing retrieval approaches.**

	# Retrieval tokens	Granularity	Retriever training	Retrieval integration
Continuous Cache	$O(10^3)$	Token	Frozen (LSTM)	Add to probs
*k*NN-LM	$O(10^9)$	Token	Frozen (Transformer)	Add to probs
SPALM	$O(10^9)$	Token	Frozen (Transformer)	Gated logits
DPR	$O(10^9)$	Prompt	Contrastive proxy	Extractive QA
REALM	$O(10^9)$	Prompt	End-to-End	Prepend to prompt
RAG	$O(10^9)$	Prompt	Fine-tuned DPR	Cross-attention
FID	$O(10^9)$	Prompt	Frozen DPR	Cross-attention
EMDR^2	$O(10^9)$	Prompt	End-to-End (EM)	Cross-attention
RETRO (ours)	$O(\mathbf{10^{12}})$	**Chunk**	**Frozen (BERT)**	**Chunked cross-attention**

Retrieval-Enhanced Transformer

Contributions

- 적은 Model parameter로 동일한 성능



Retrieval-Enhanced Transformer

Contributions

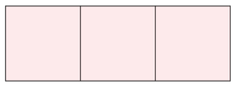
- SOTA 달성한 데이터셋
  - Wikitext103
  - The Pile

Method

RETRO's Retrieval Database

Key-Value 구조

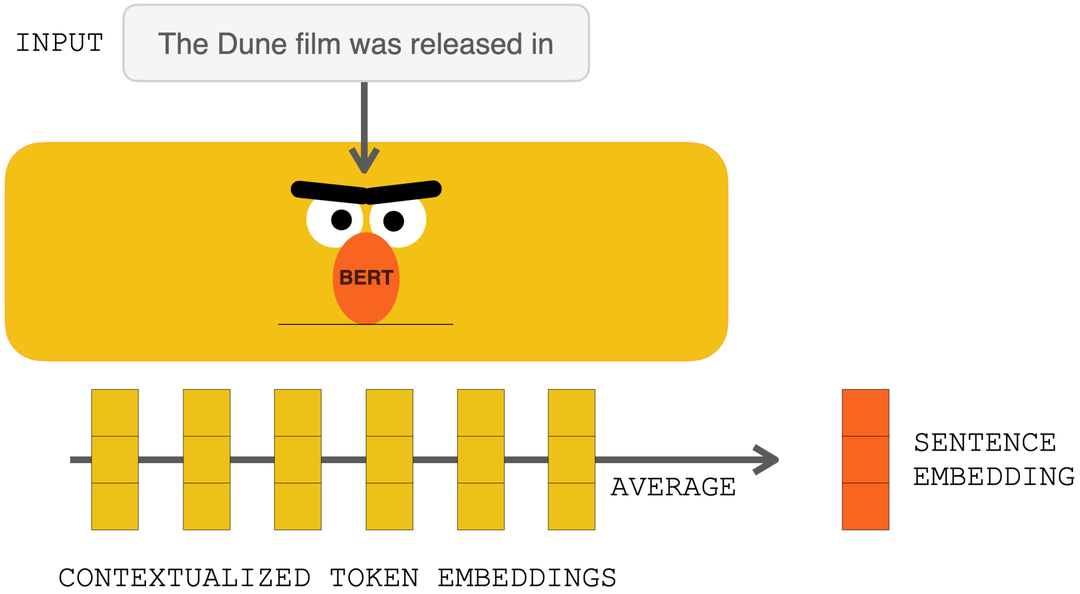
- Key: BERT sentence embedding
- Value: 2가지 text

Key (BERT sentence embedding)	Value (text. neighbor and completion chunks. Each up to 64 tokens in length)	
	Dune is a 2021 American epic science fiction film directed by Denis Villeneuve	NEIGHBOR
	It is the first of a planned two-part adaptation of the 1965 novel by Frank Herbert	COMPLETION
	Dune is a 1965 science fiction novel by American author Frank Herbert	NEIGHBOR
	originally published as two separate serials in Analog magazine	COMPLETION
...	...	

RETRO’s Retrieval Database

Key-Value 구조

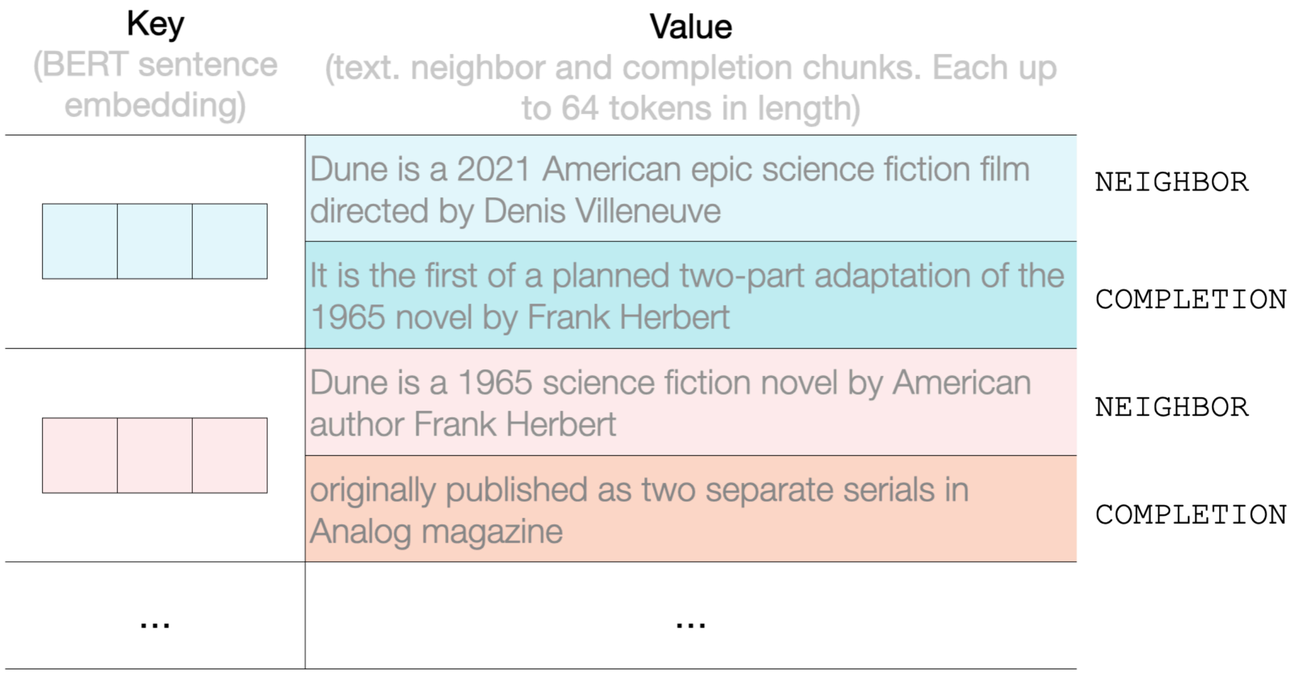
- Key: BERT sentence embedding
  - Mean pooling



RETRO's Retrieval Database

Key-Value 구조

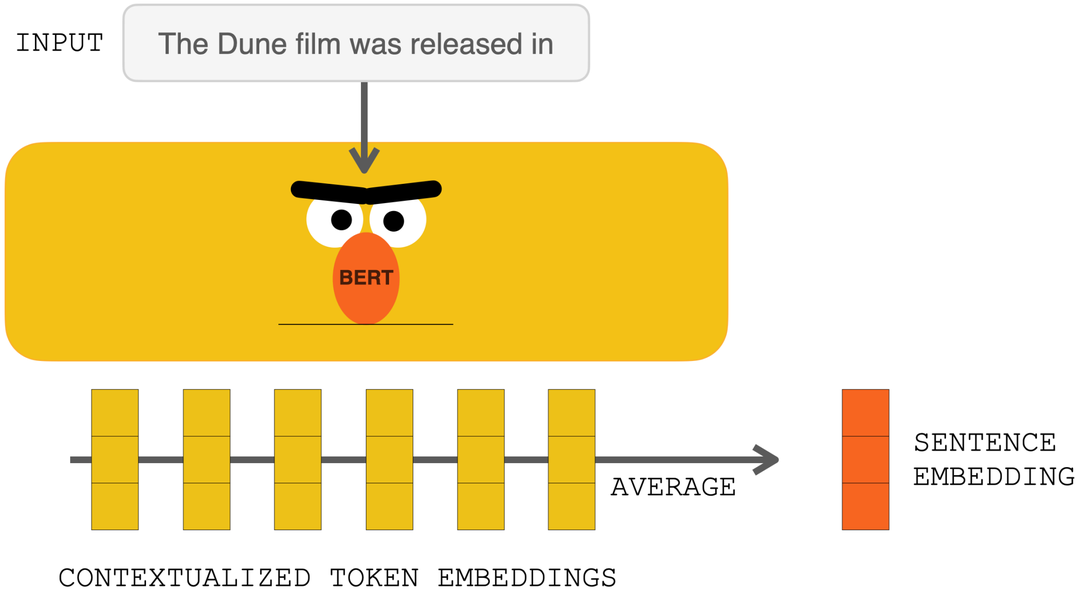
- Value: 2가지 text
  - 1. Neighbor (키 계산 시 사용됨)
  - 2. Completion (이어지는 다음 청크)



KNN Retrieval

Query

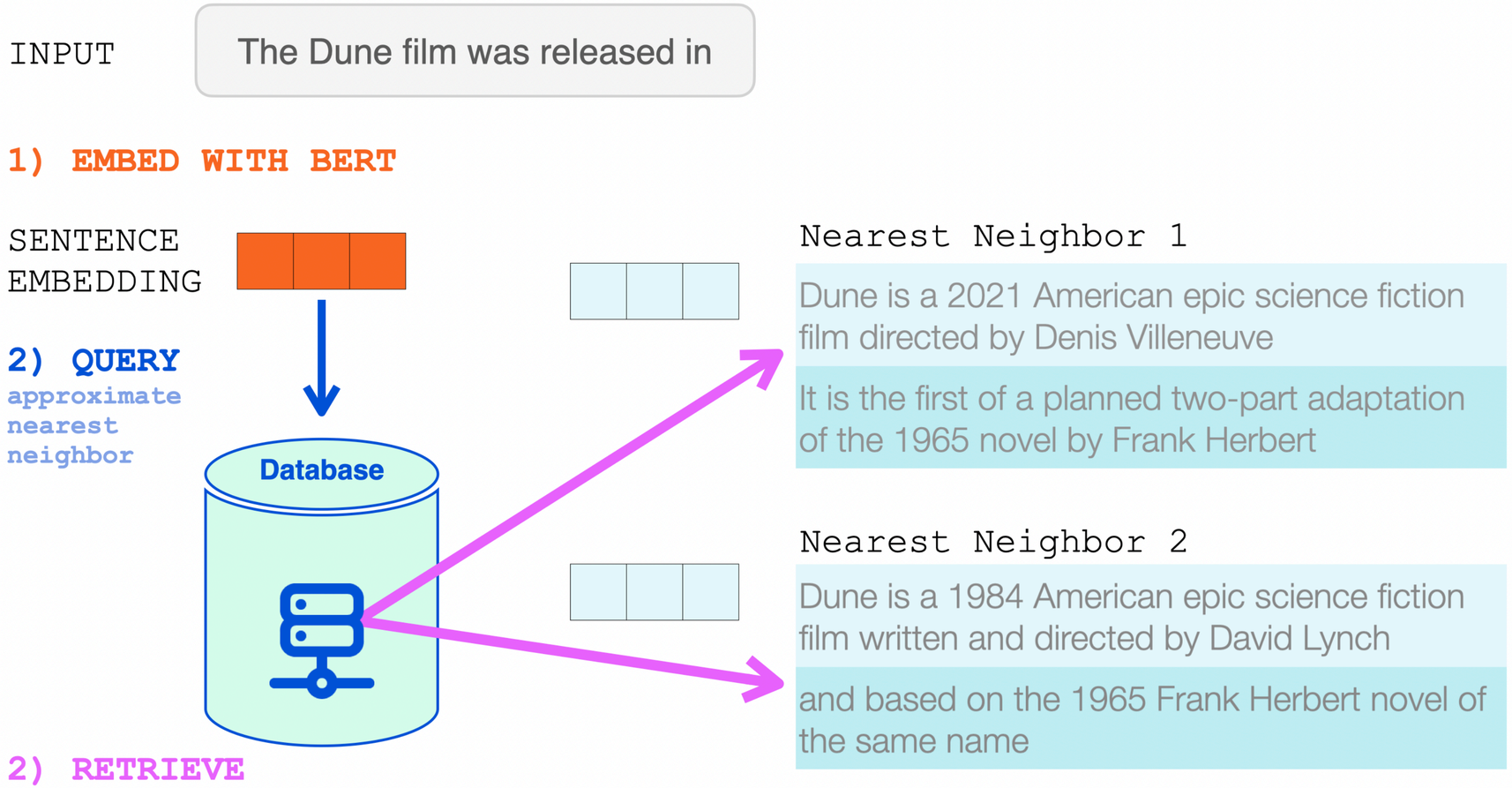
- Input tokens
  - Sentence embedding (Mean pooling)



KNN Retrieval

nearest neighbor search

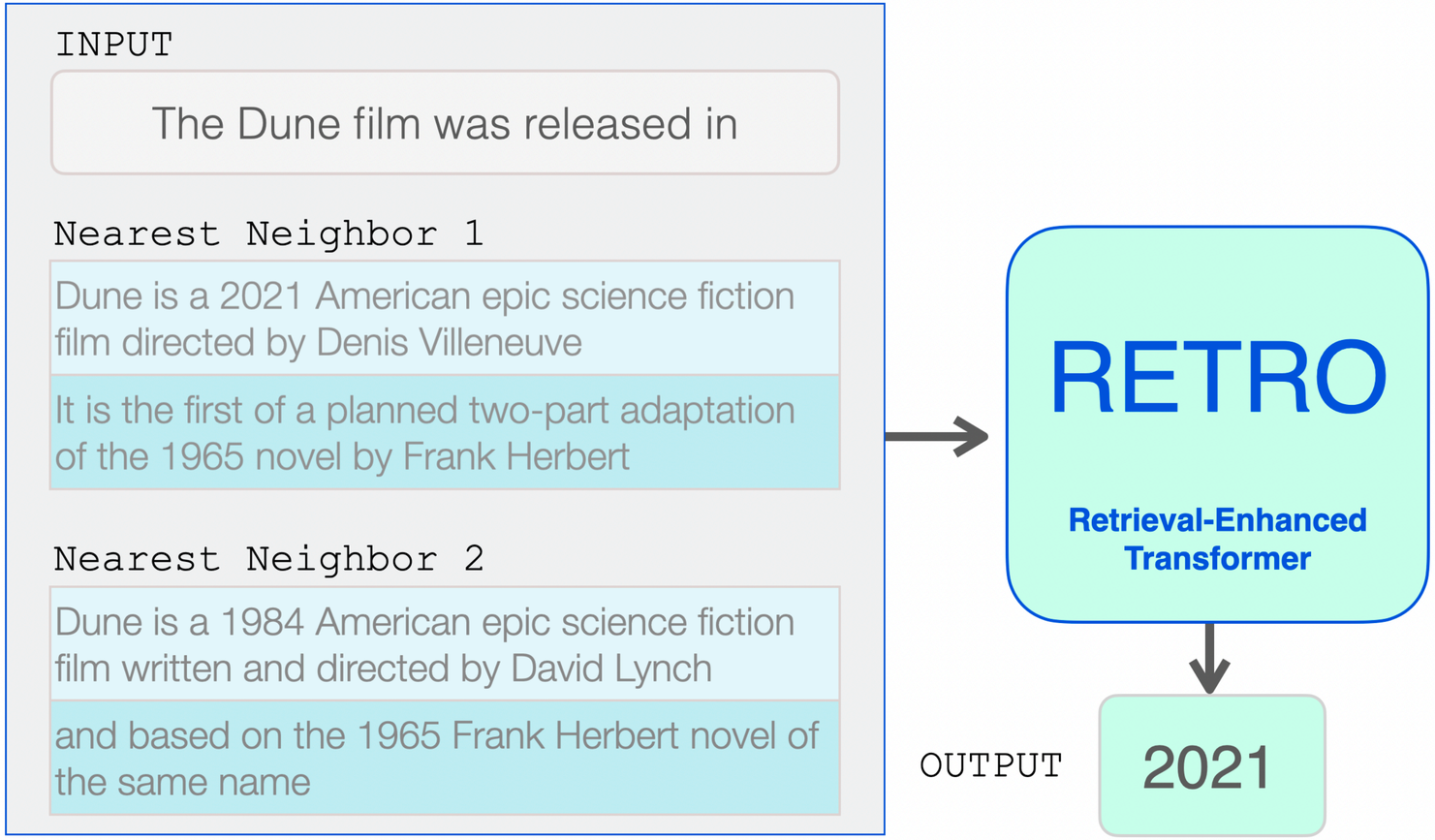
- ScaNN 라이브러리



Chunked cross-attention (CCA)

정보 통합

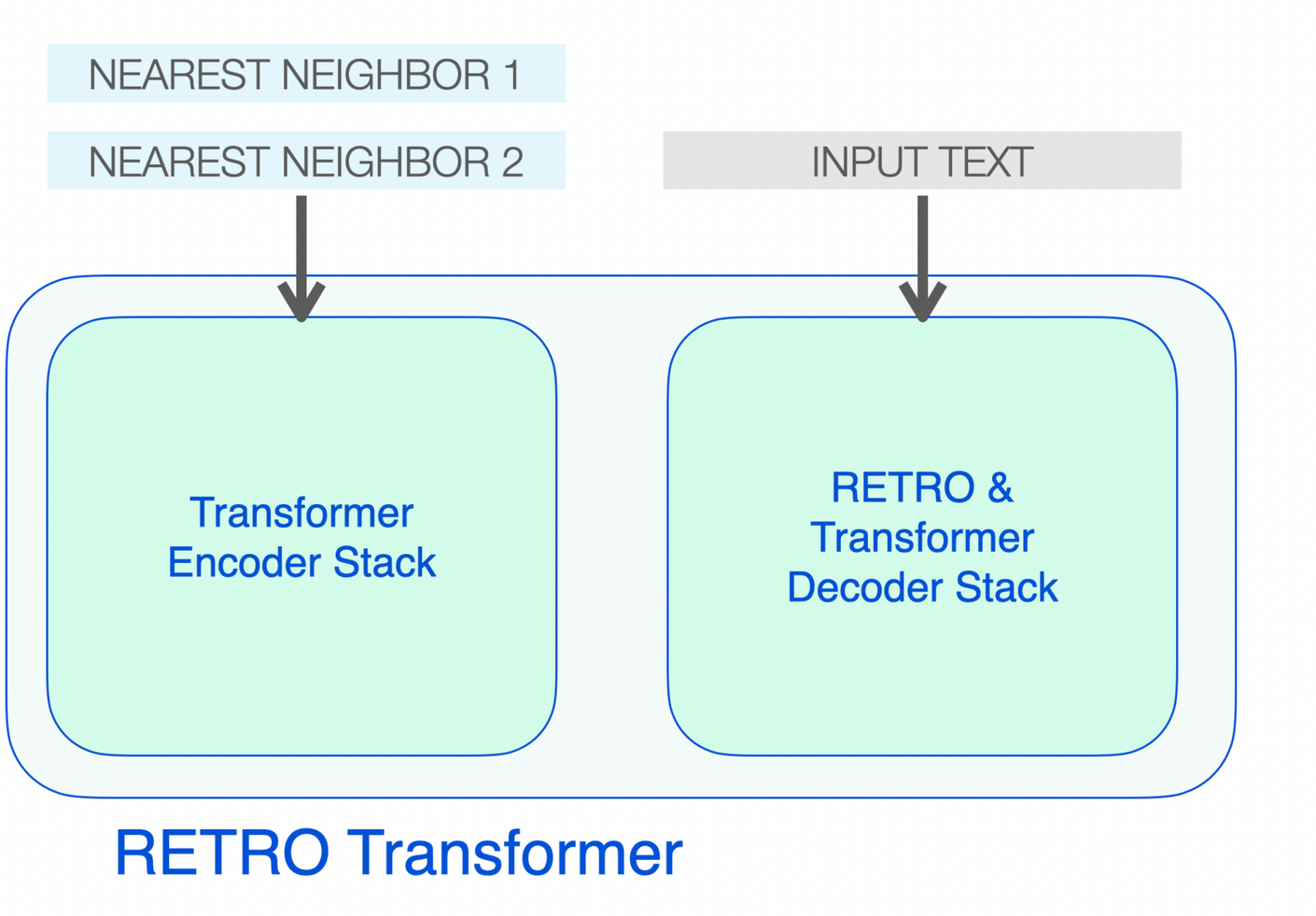
- Input tokens
- Retrieved neighbours



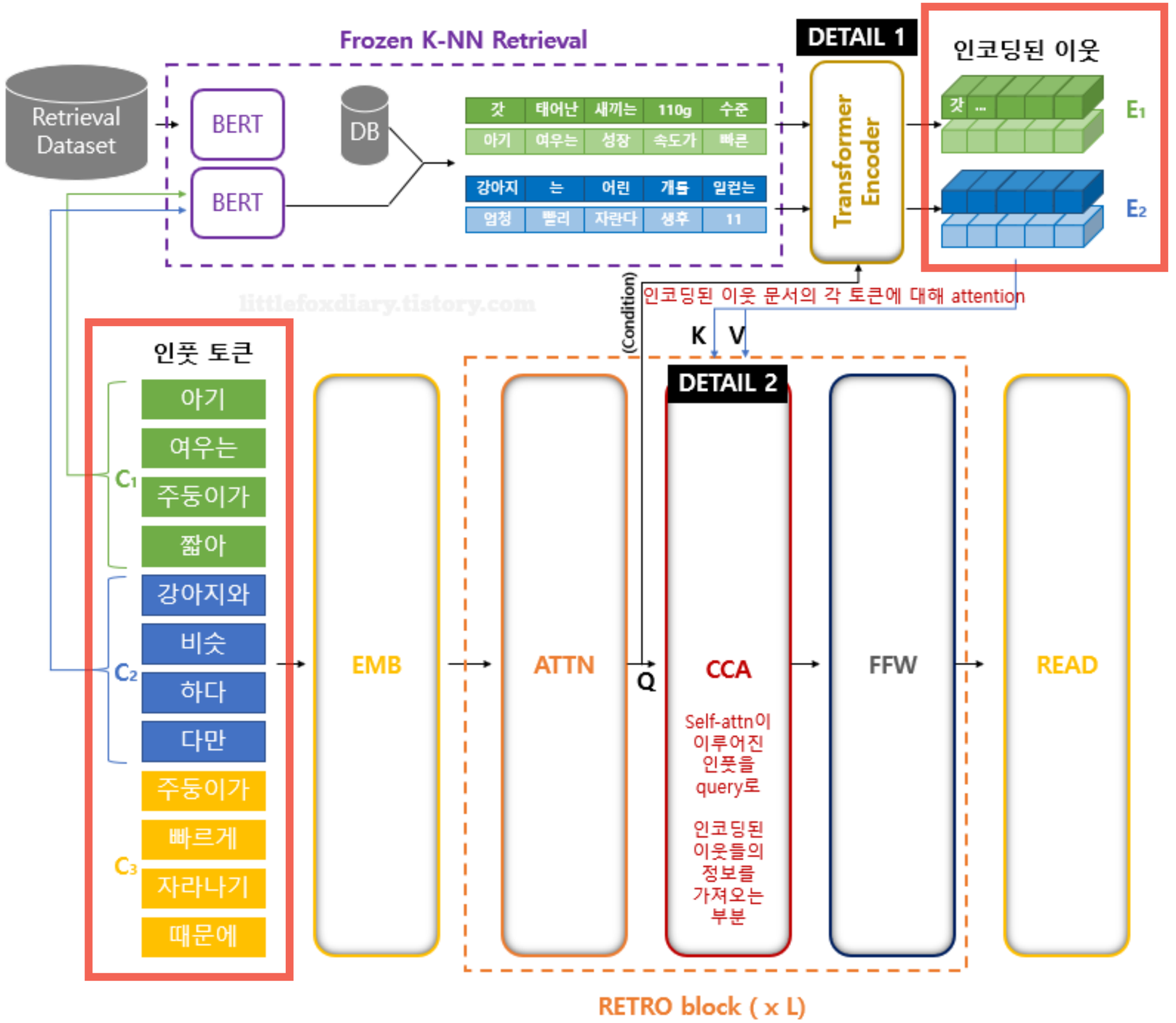
Chunked cross-attention (CCA)

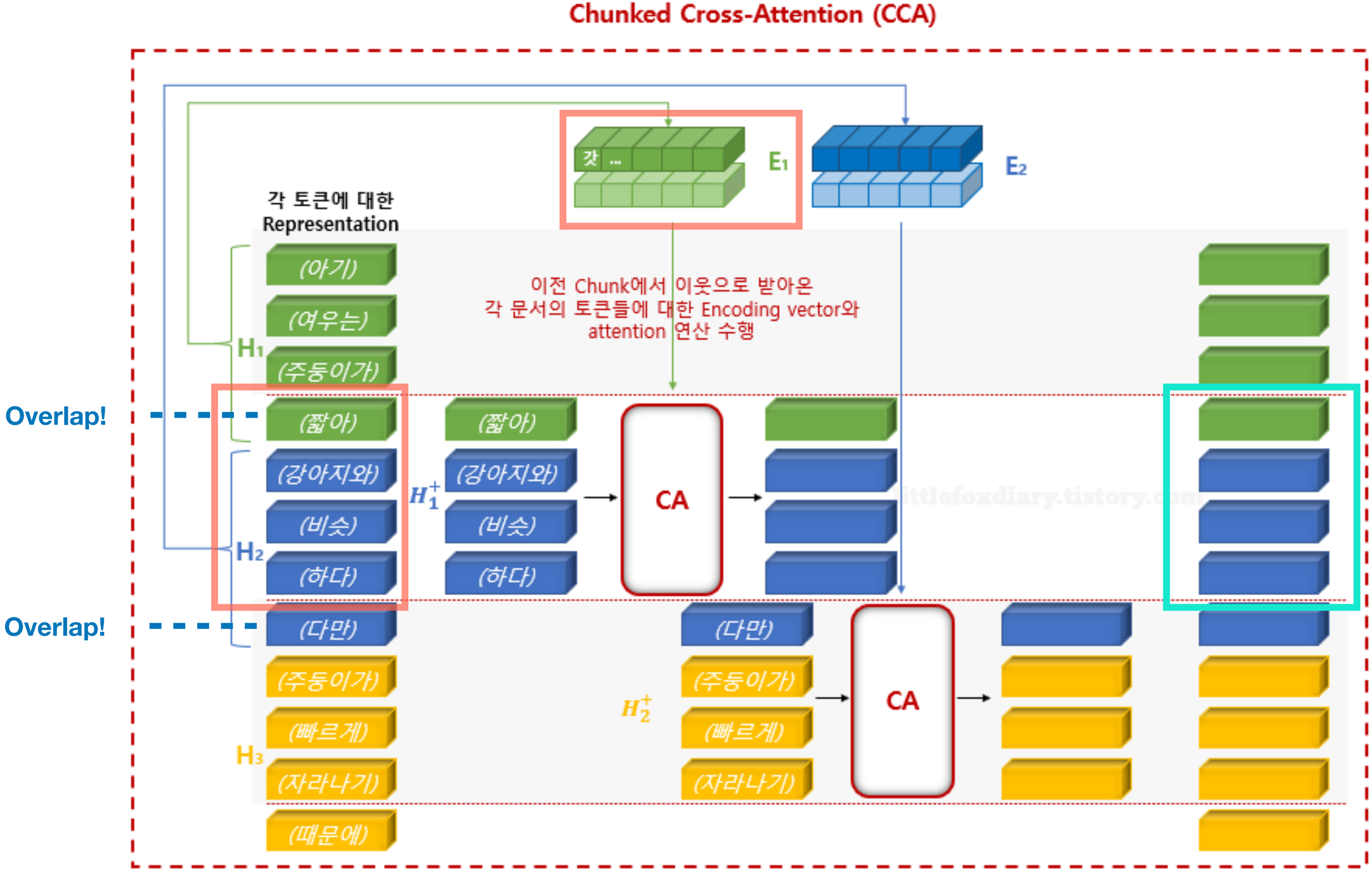
정보 통합

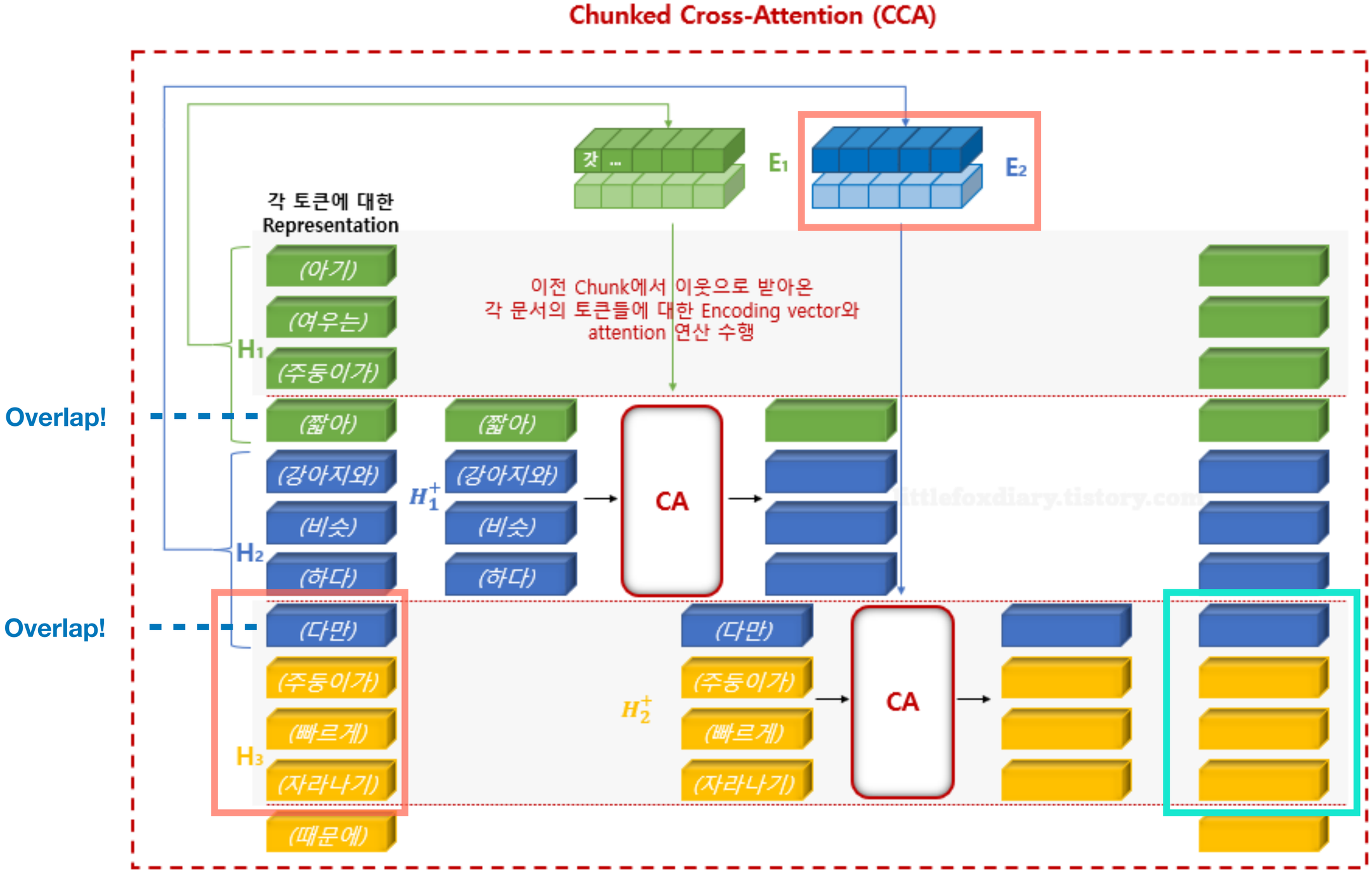
- Input tokens
- Retrieved neighbours

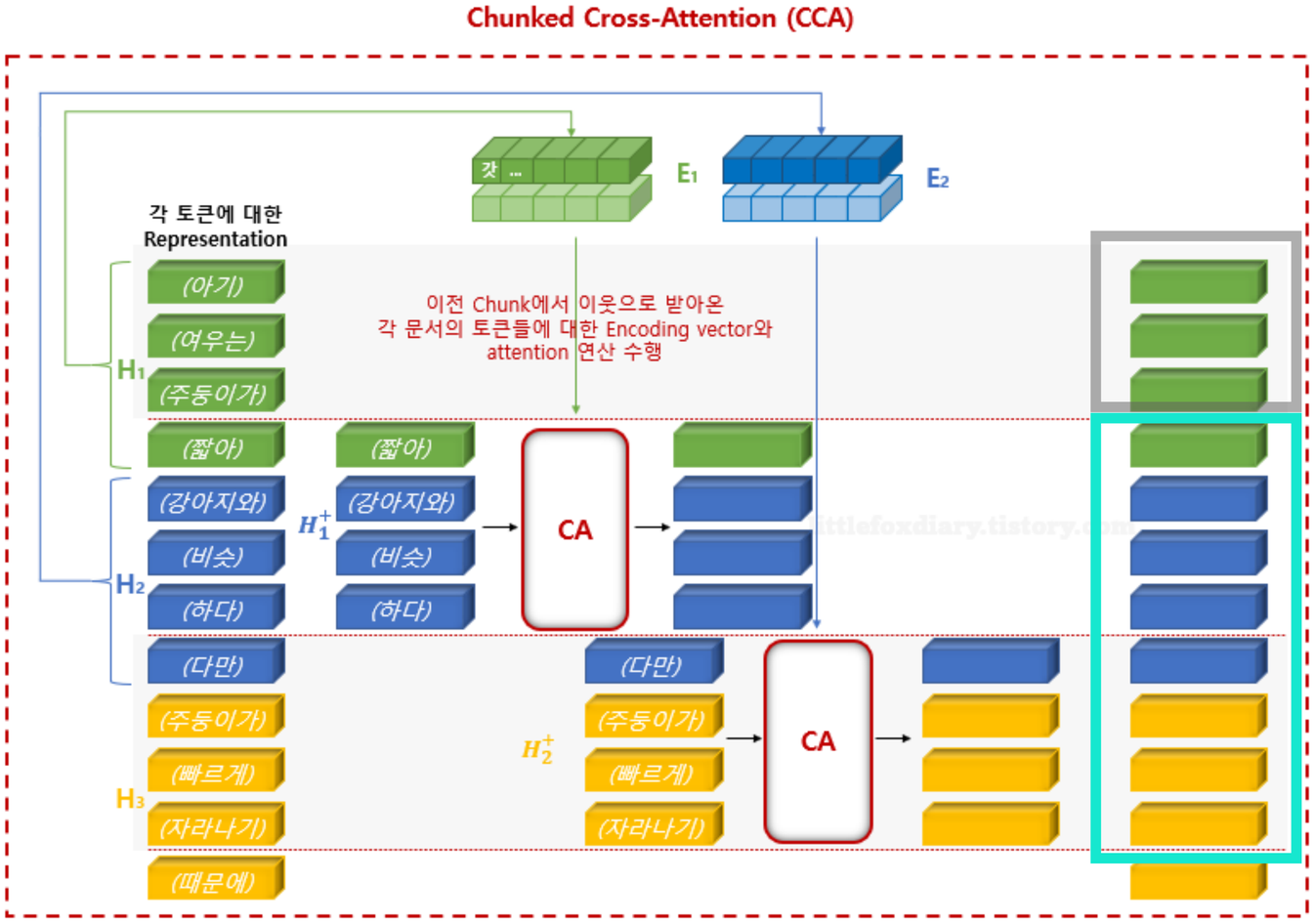


- Input tokens (multiple chunks)
- Retrieved neighbours (N & C chunks, multiple documents)




Chunked Cross-Attention (CCA)
갓 ...
E1
E2
각 토큰에 대한
Representation
(아기)
(여우는)
(주둥이가)
(짧아)
(강아지와)
(비슷)
(하다)
(다만)
(주둥이가)
(빠르게)
(자라나기)
(때문에)
H1
H2
H3
Overlap!
Overlap!
이전 Chunk에서 이웃으로 받아온
각 문서의 토큰들에 대한 Encoding vector와
attention 연산 수행
H_1^+
H_2^+
CA
CA

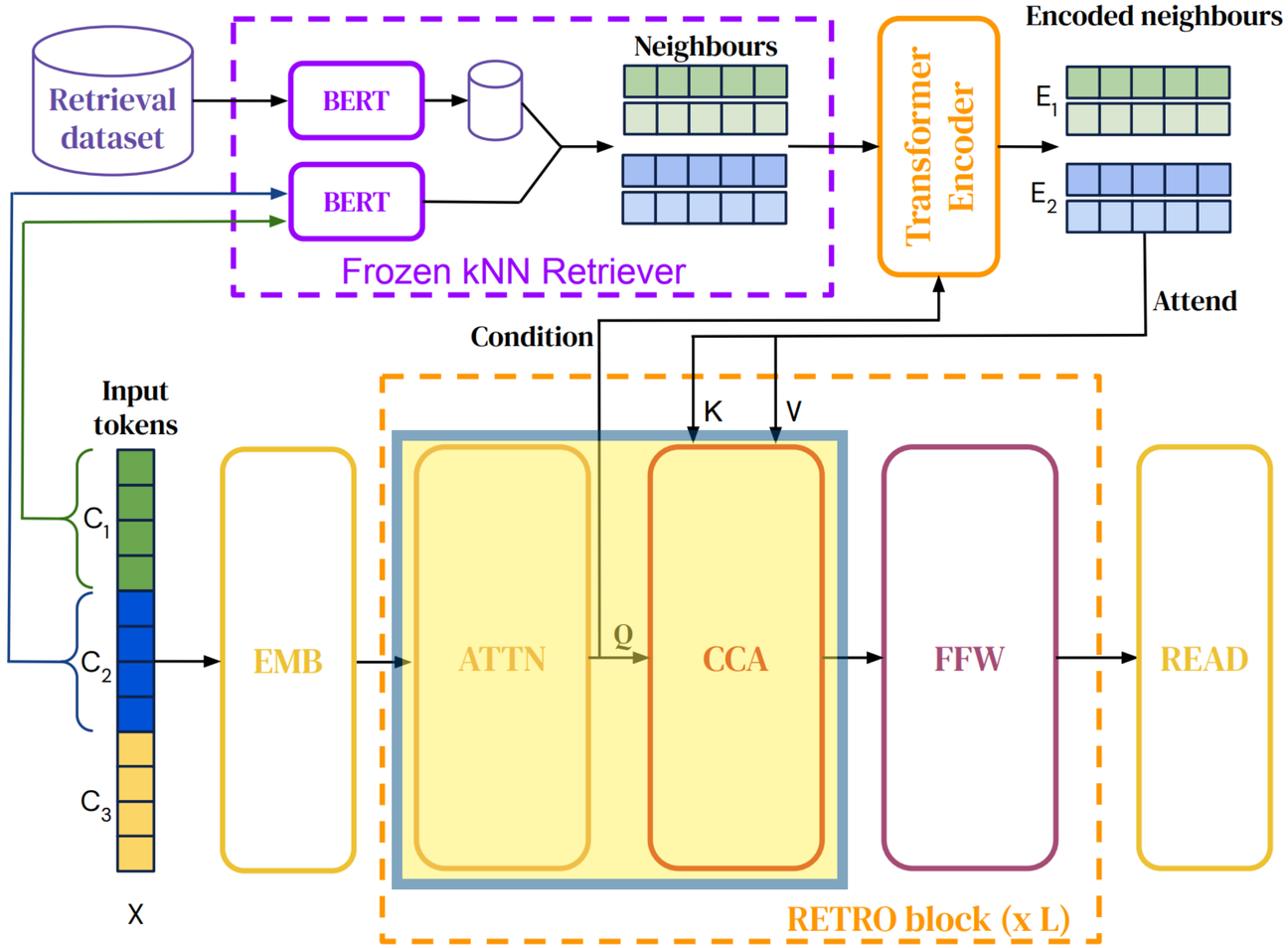

Chunked Cross-Attention (CCA)
각 토큰에 대한
Representation
갓 ...
E1
E2
(아기)
(여우는)
(주둥이가)
H1
이전 Chunk에서 이웃으로 받아온
각 문서의 토큰들에 대한 Encoding vector와
attention 연산 수행
Overlap!
(짧아)
(강아지와)
(비슷)
(하다)
H2
CA
Overlap!
(다만)
(주둥이가)
(빠르게)
(자라나기)
(때문에)
H3
CA

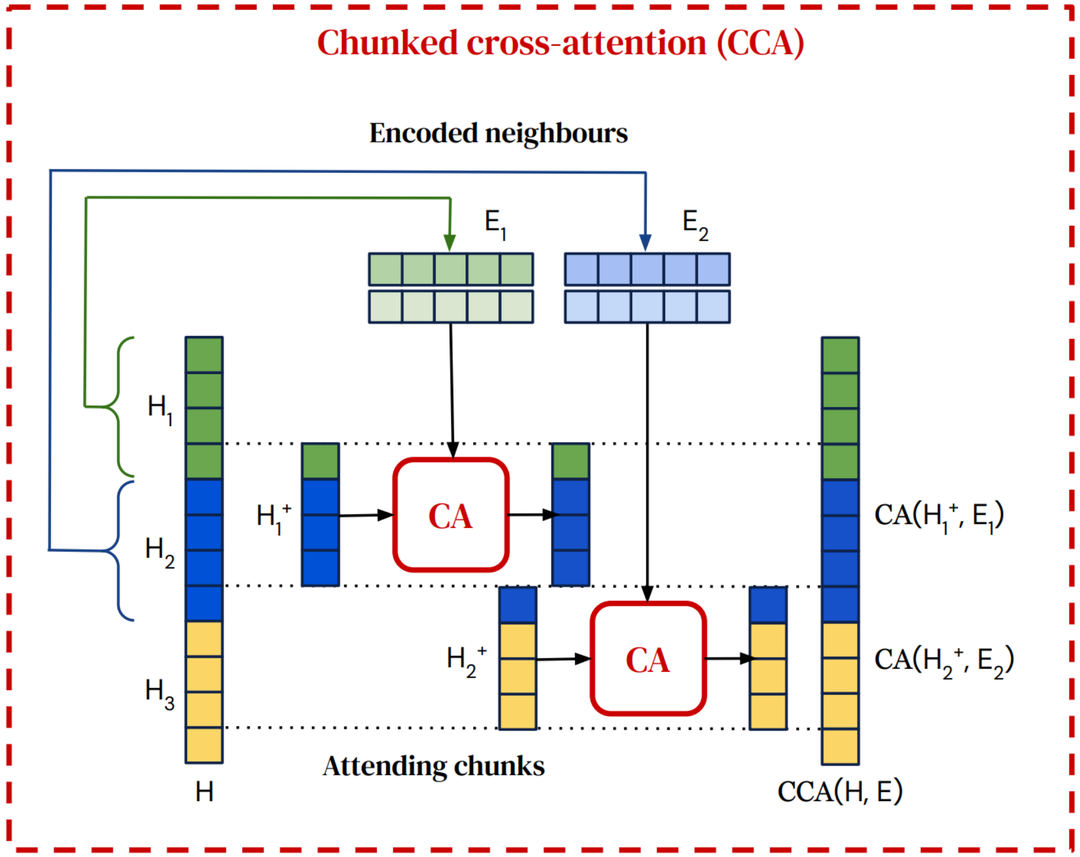


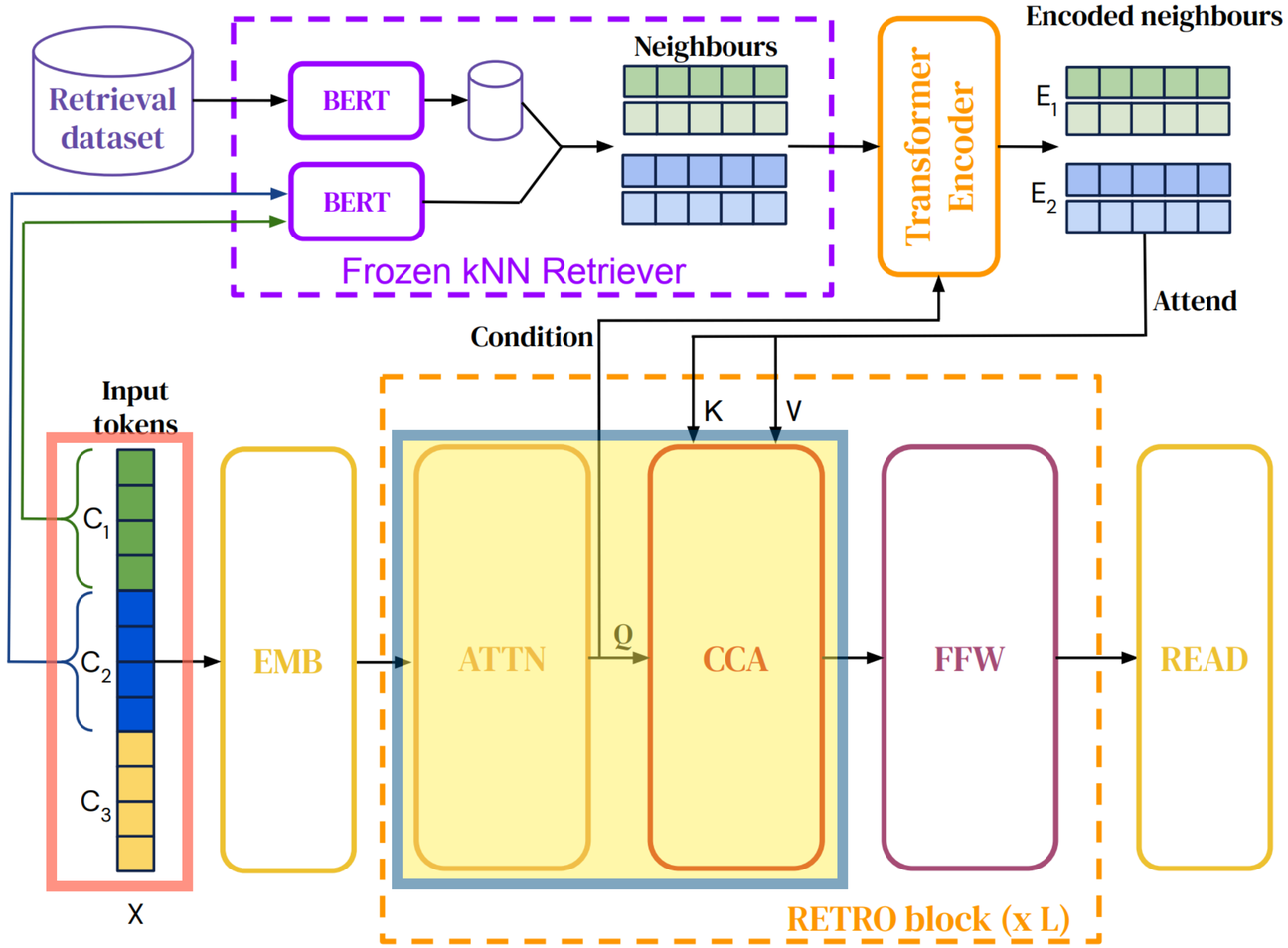

```
if n < self.chunk_size:
    return torch.zeros_like(x)
```

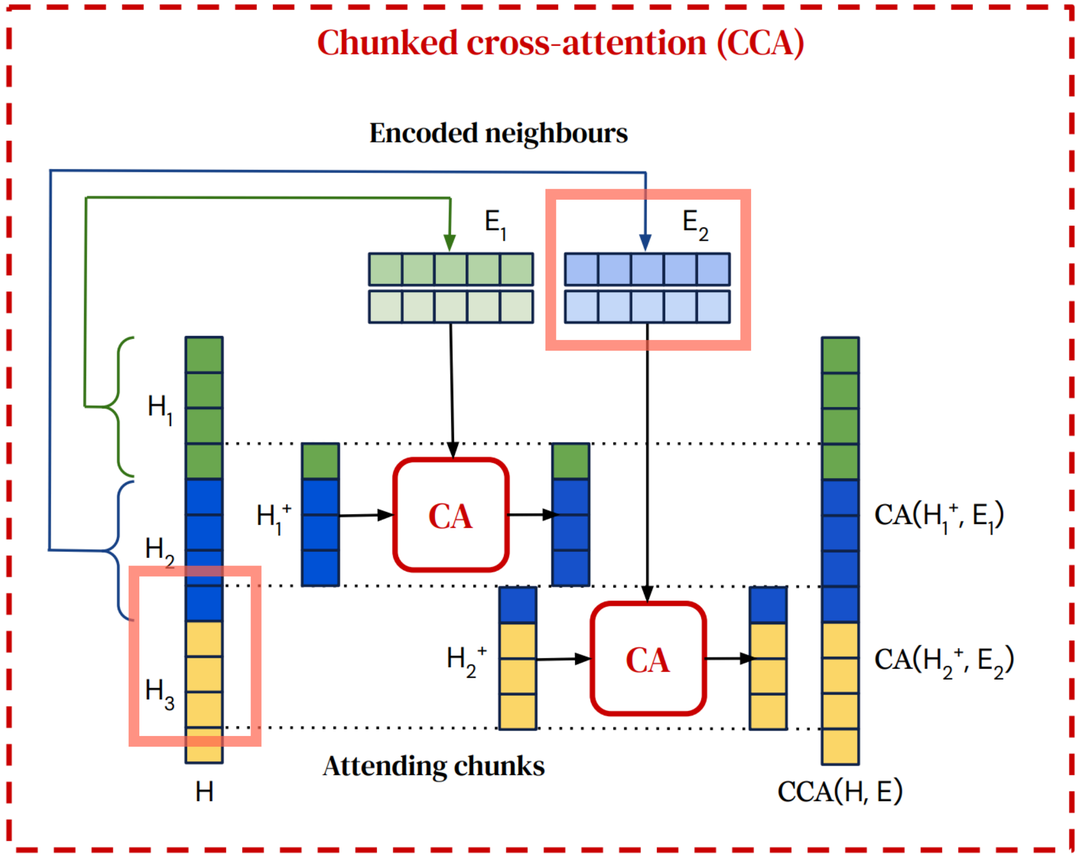
```
self.null_k = nn.Parameter(torch.randn(inner_dim)) if null_kv else None
self.null_v = nn.Parameter(torch.randn(inner_dim)) if null_kv else None
```

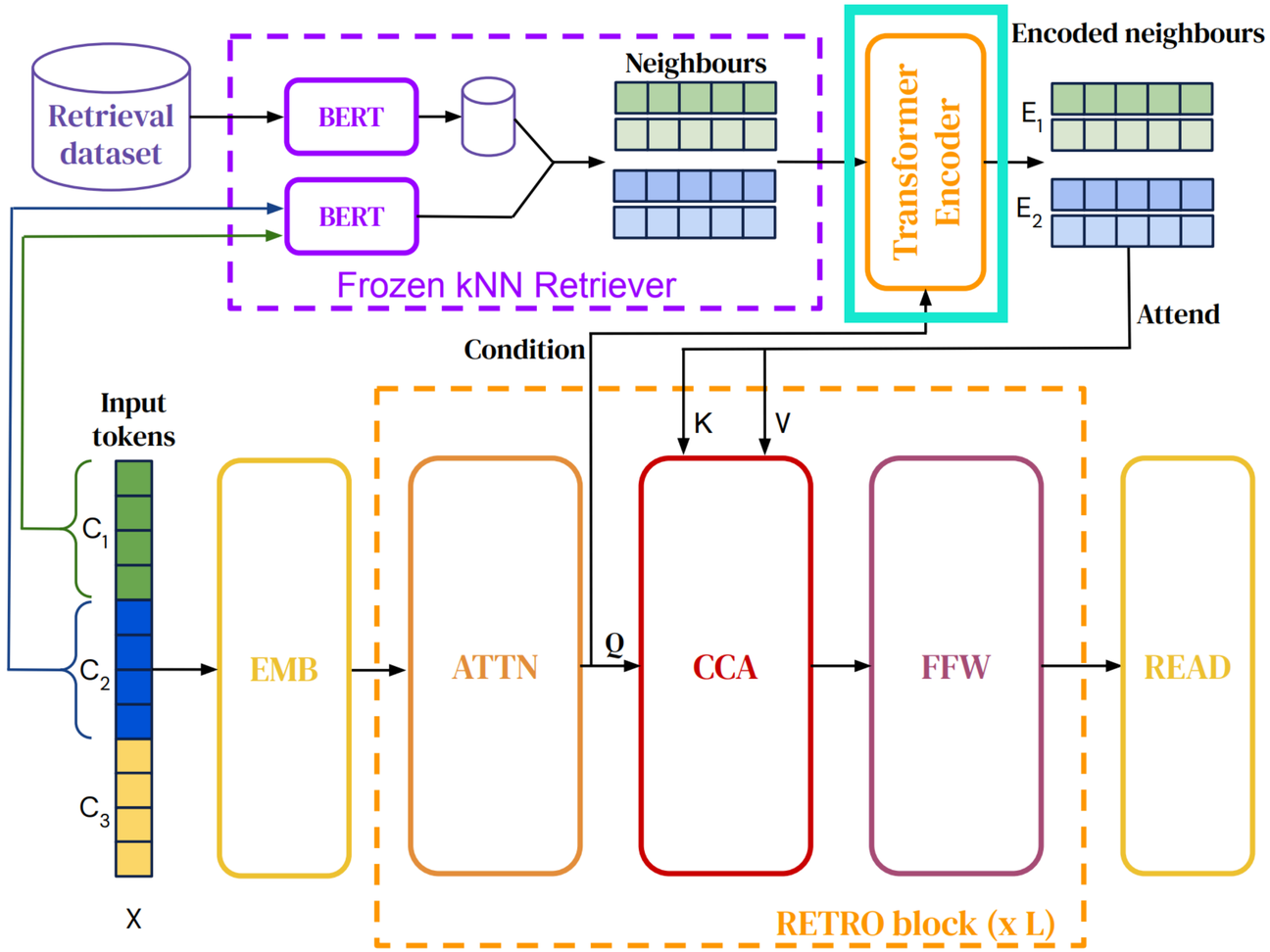
$$L\left(X|\theta,\mathcal{D}\right) \triangleq \sum_{u=1}^{l}\sum_{i=1}^{m} \ell_\theta\left(x_{(u-1)\,m+i}|(x_j)_{j<(u-1)\,m+i},\ \left(\mathrm{RET}_{\mathcal{D}}(C_{u'})\right)_{u'<u}\right).$$






Retrieval dataset
BERT
BERT
Frozen kNN Retriever
Neighbours
Transformer Encoder
Encoded neighbours
E1
E2
Attend
Condition
Input tokens
C1
C2
C3
X
EMB
ATTN
K
V
Q
CCA
FFW
READ
RETRO block (x L)


Chunked cross-attention (CCA)
Encoded neighbours
E1
E2
H1
H2
H3
H
H1+
CA
H2+
CA
CA(H1+, E1)
CA(H2+, E2)
Attending chunks
CCA(H, E)




```python
class RMSNorm(nn.Module):
    def __init__(
        self,
        dim,
        *,
        eps = 1e-8,
        gated = False
    ):
        super().__init__()
        self.eps = eps
        self.scale = dim ** -0.5
        self.gamma = nn.Parameter(torch.ones(dim))
        self.weight = nn.Parameter(torch.ones(dim)) if gated else None

    def forward(self, x):
        norm = x.norm(keepdim = True, dim = -1) * self.scale
        out = (x / norm.clamp(min = self.eps)) * self.gamma

        if not exists(self.weight):
            return out
```

Implement

Retrieval-Enhanced Transformer

구현 방법

- Train from scratch ✔
- RETRO-fiting on pre-trained transformers
  - Pre-trained weights (frozen)
  - Retrieval encoder (initialised)
  - Cross-attention weights (initialised)

RETRO's Retrieval Database

Training dataset와 retrieval data 둘다

- MassiveText (Rae et al., 2021)
  - 5 subset, Multi-lingual

Source	Token count (M)	Documents (M)	Multilingual	Sampling frequency
Web	977,563	1,208	Yes	55%
Books	3,423,740	20	No	25%
News	236,918	398	No	10%
Wikipedia	13,288	23	Yes	5%
GitHub	374,952	143	No	5%

- SentencePiece Tokenizer

Encoder-Decoder

RETRO Transformer 구조

- Encoder (2 layers)

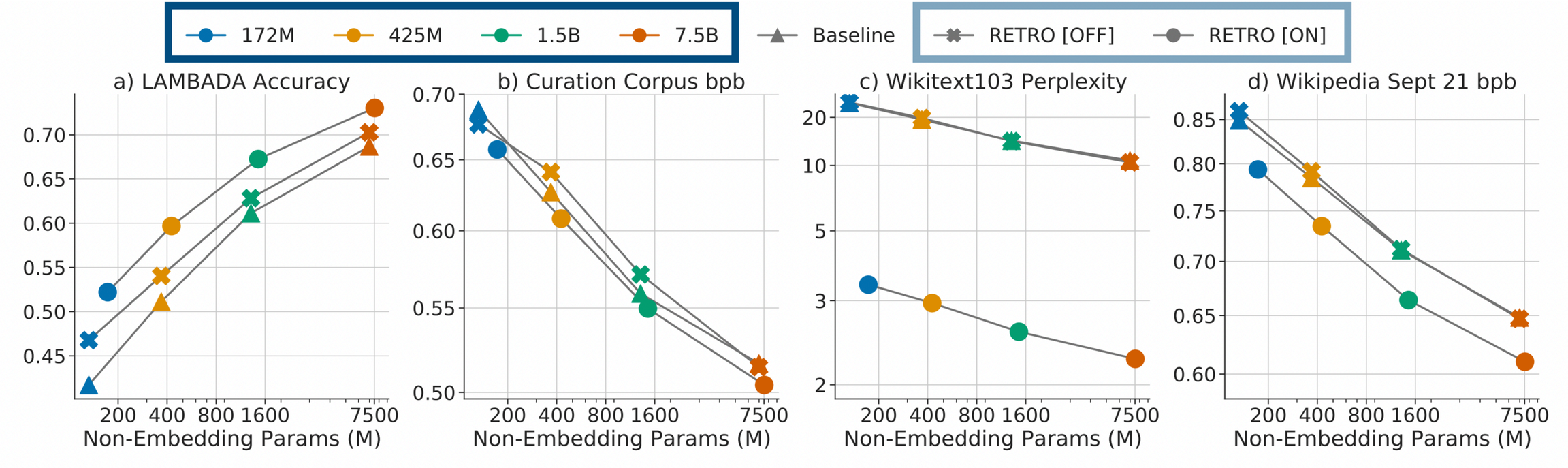
- Decoder (12 layers)
  - Standard Transformer Decoder Block (ATTN + FFNN)
  - RETRO Decoder Block (ATTN + **CCA** + FFNN) - 6, 9, 12 번째 layer

Results

Results

Language modeling

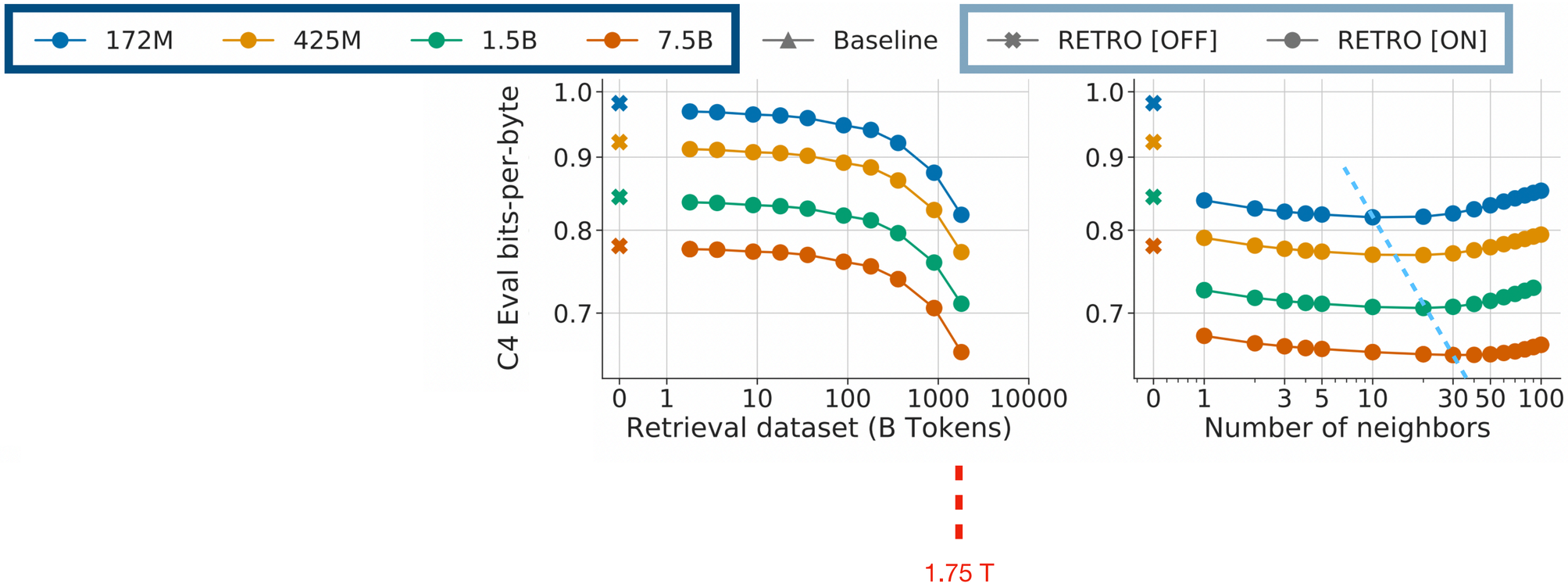
- Model scaling



Results

Language modeling

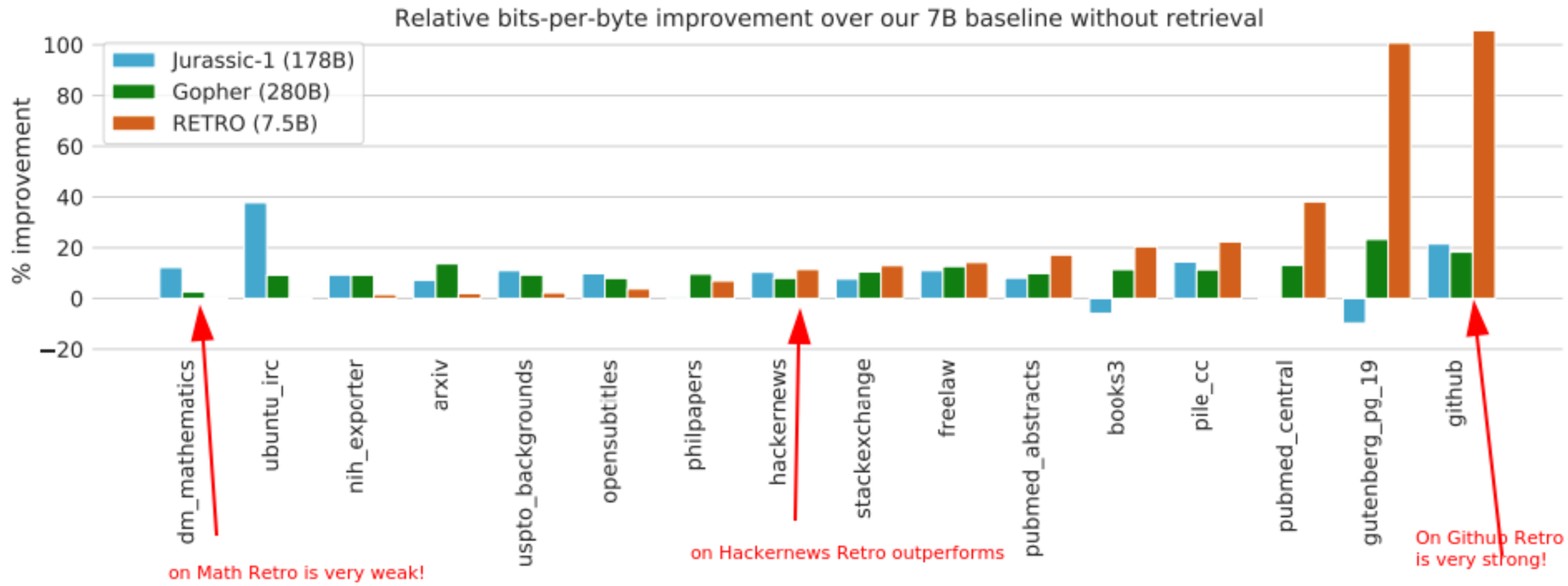
- Data scaling



Results

Language modeling

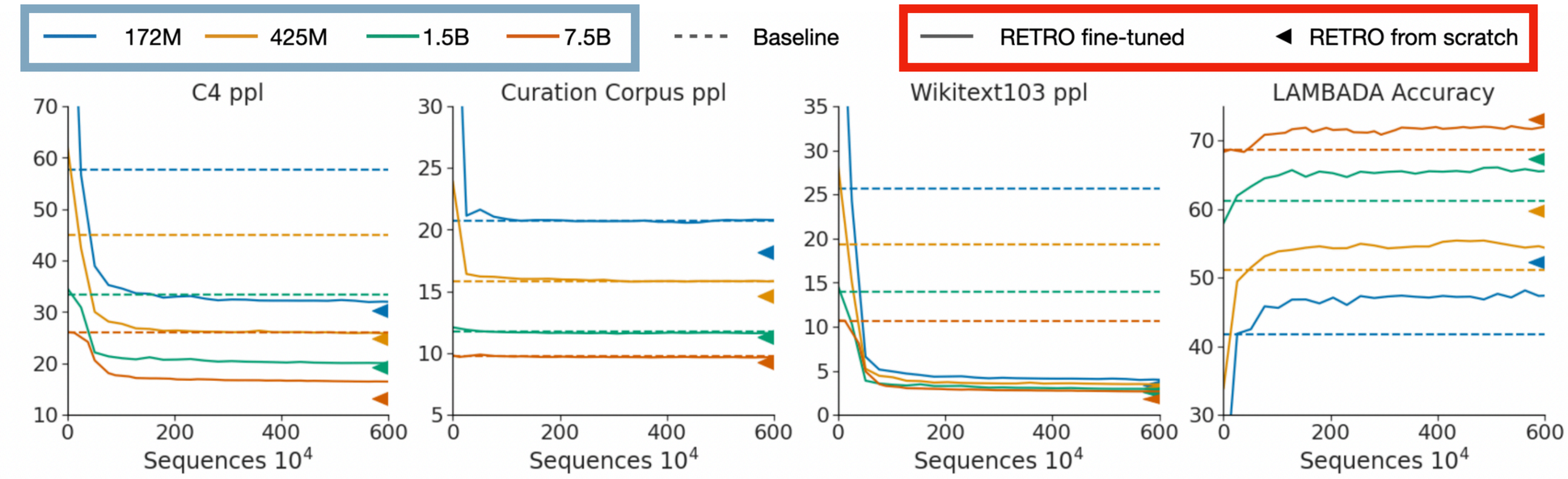
- The Pile



Results

RETRO-fitting baseline transformer

- Train only CCA+neighbour encoder parameter (10% ↓)
- Requiring only 6 million sequence (3%)



Results

Question answering

- Fine-tune (Natural Questions)
  - 7.5B pre-trained RETRO model

Table 5 | **Question answering results.** Exact match accuracy on Natural Questions.

Model	Test Accuracy
REALM (Guu et al., 2020)	40.4
DPR (Karpukhin et al., 2020)	41.5
RAG (Lewis et al., 2020)	44.5
EMDR^2 (Sachan et al., 2021)	52.5
FID (Izacard and Grave, 2021)	51.4
FID + Distill. (Izacard et al., 2020)	**54.7**
Baseline 7B (closed book)	30.4
RETRO 7.5B (DPR retrieval)	45.5

Conclusion

장점

- 모델
  - RETRO 모델은 가볍다
  - RETRO-fitting을 하면 더 가볍다
- 외부 KB
  - 지속적 업데이트 가능
  - 설명가능력
- Task
  - Long-sequence 생성 태스크에서 SOTA 달성
  - QA 태스크에 적용 가능성

Thank you