ACL 2023 Tutorial:

Retrieval-based Language Models and Applications

Akari Asai, Sewon Min, Zexuan Zhong, Danqi Chen

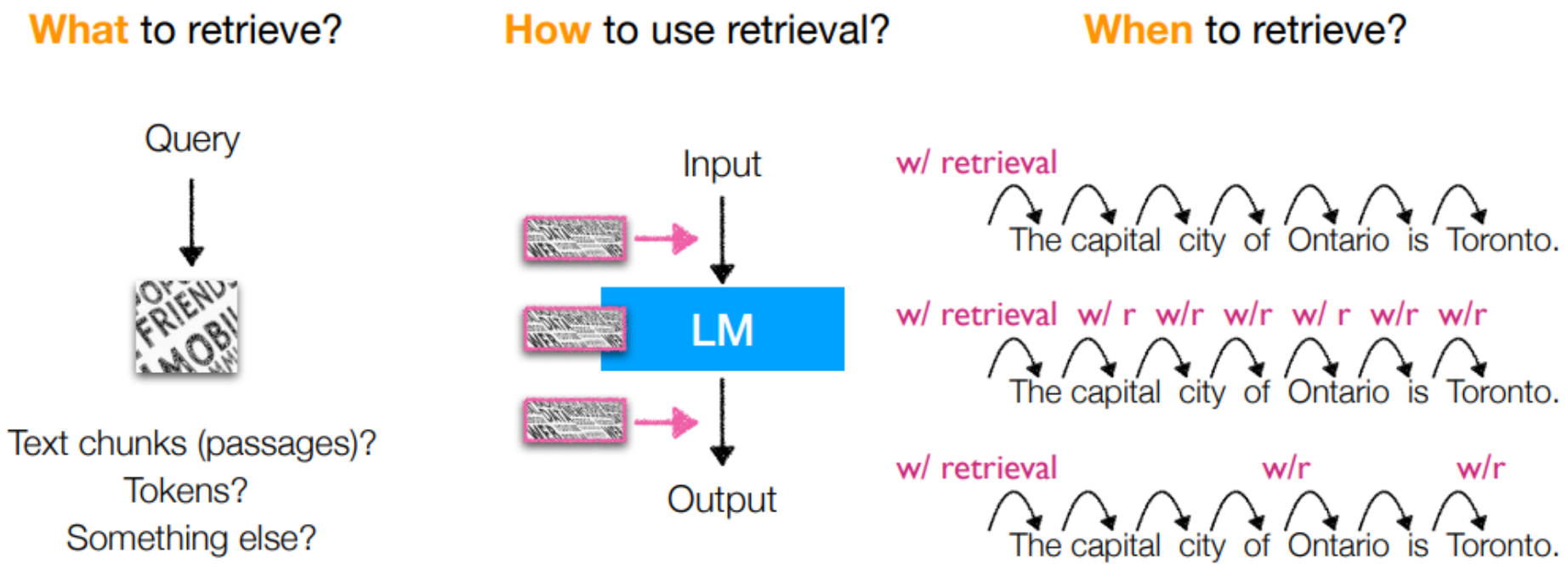
University of Washington, Princeton University

발표자: 송선영

2024/01/16

Retrieval-based language model

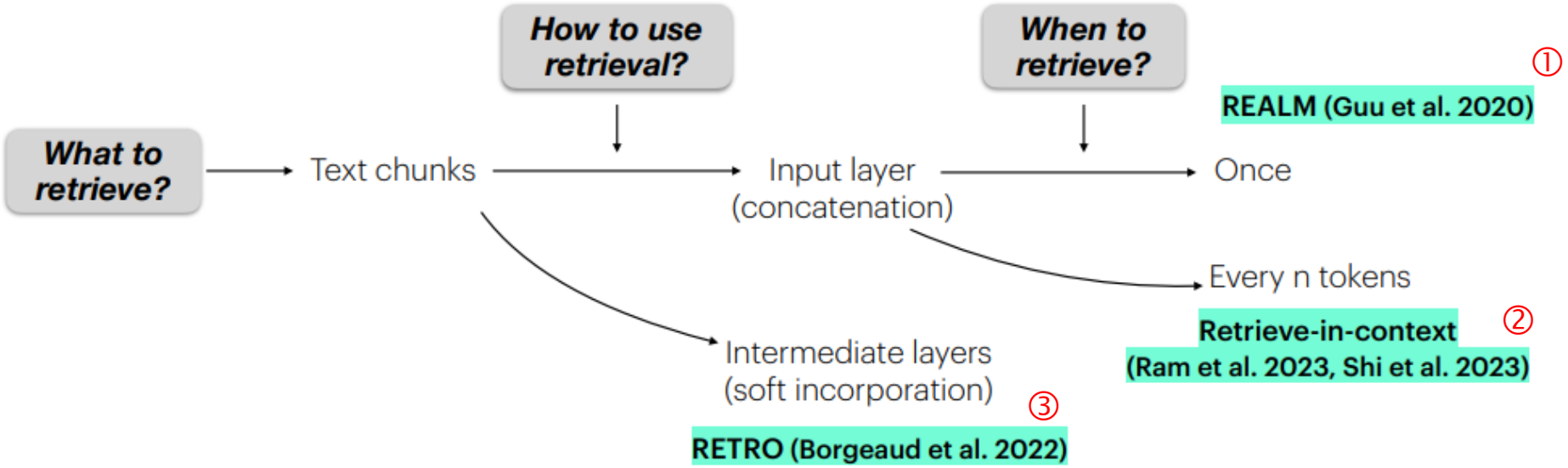
- We can categorize retrieval-based language models based on **three key questions**
  - What do we retrieve from a datastore?
  - How do we incorporate retriever result to the language model?
  - When do we use retrieval?



Retrieval-based language model

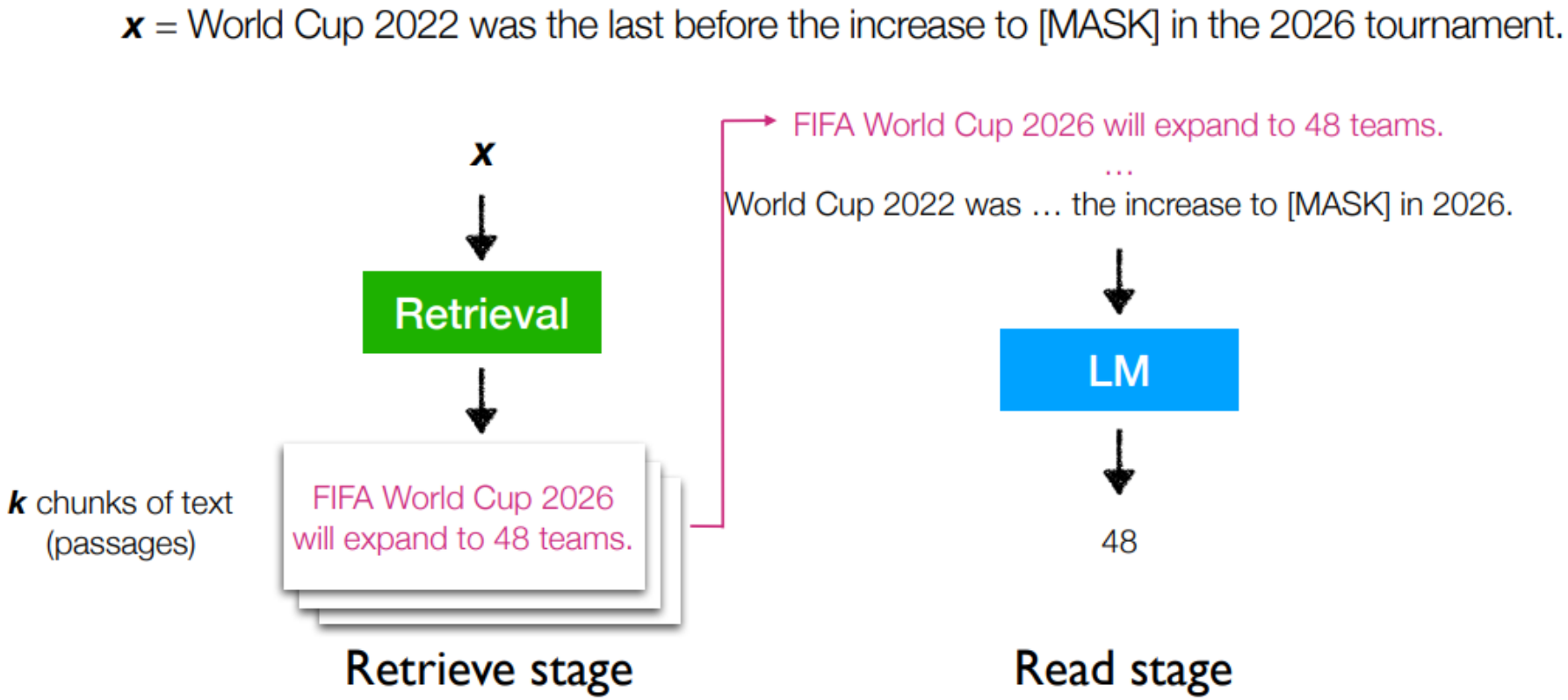
- **Road map of Key papers**
  - REALM
  - Retrieve-in-context
  - RETRO

	What do retrieve?	How to use retrieval?	When to retrieve?
REALM (Guu et al 2020)	Text chunks	Input layer	Once
Retrieve-in-context LM (Shi et al 2023, Ram et al 2023)	Text chunks	Input layer	Every n tokens
RETRO (Borgeaud et al. 2021)	Text chunks	Intermediate layers	Every n tokens



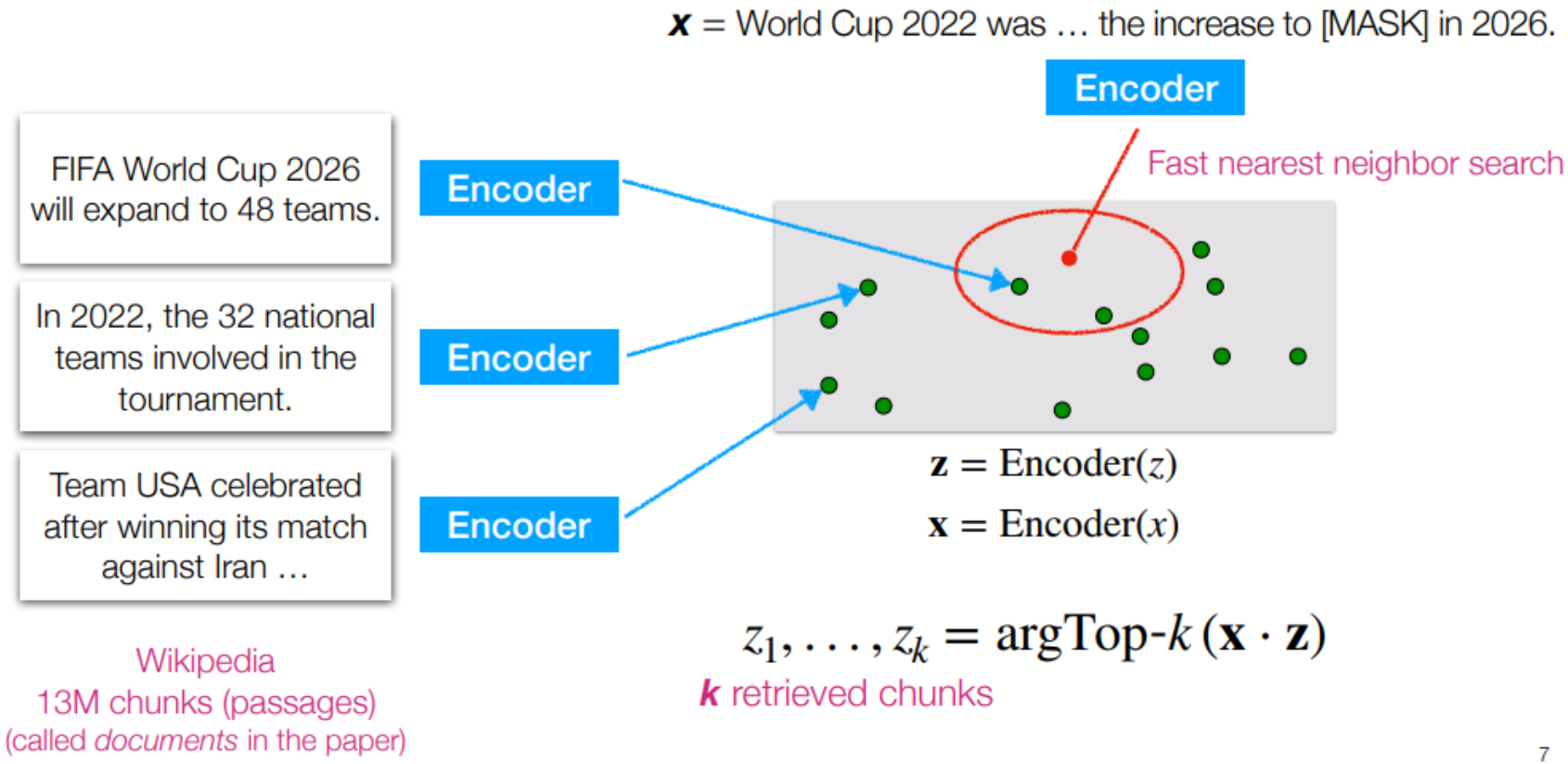
REALM (Guu et al. 2020)

- A masked language model



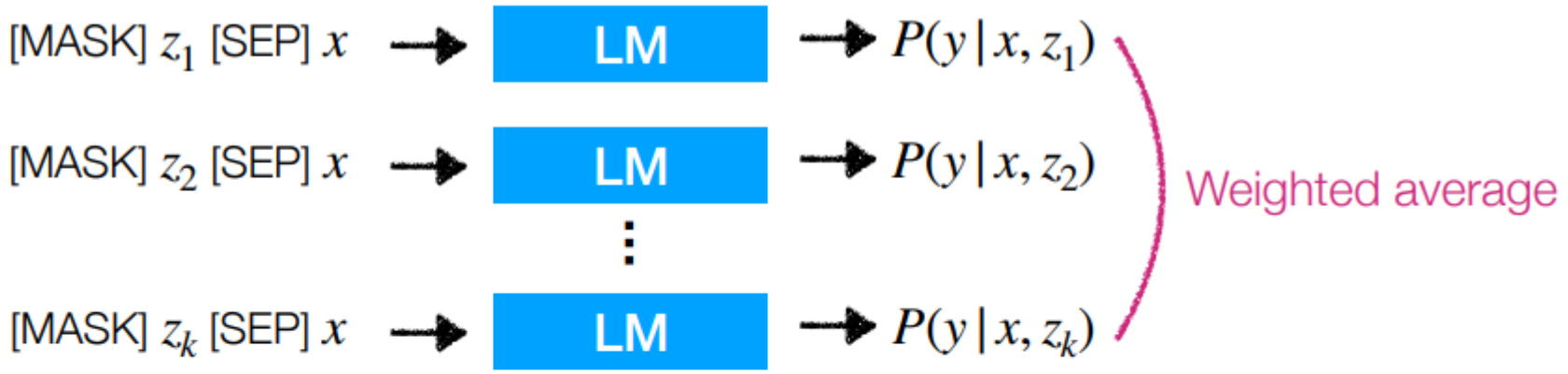
REALM – Retrieve stage

- Datastore: 13 million text chunks of Wikipedia



7

REALM – Read stage



0 if not one of top k

$$\sum_{z \in \mathcal{D}} P(z \mid x) P(y \mid x, z)$$

Need to approximate
→ Consider top k chunks only

from the retrieve stage

from the read stage

Retrieval-in-context LM (Ram et al. 2023, Shi et al. 2023)

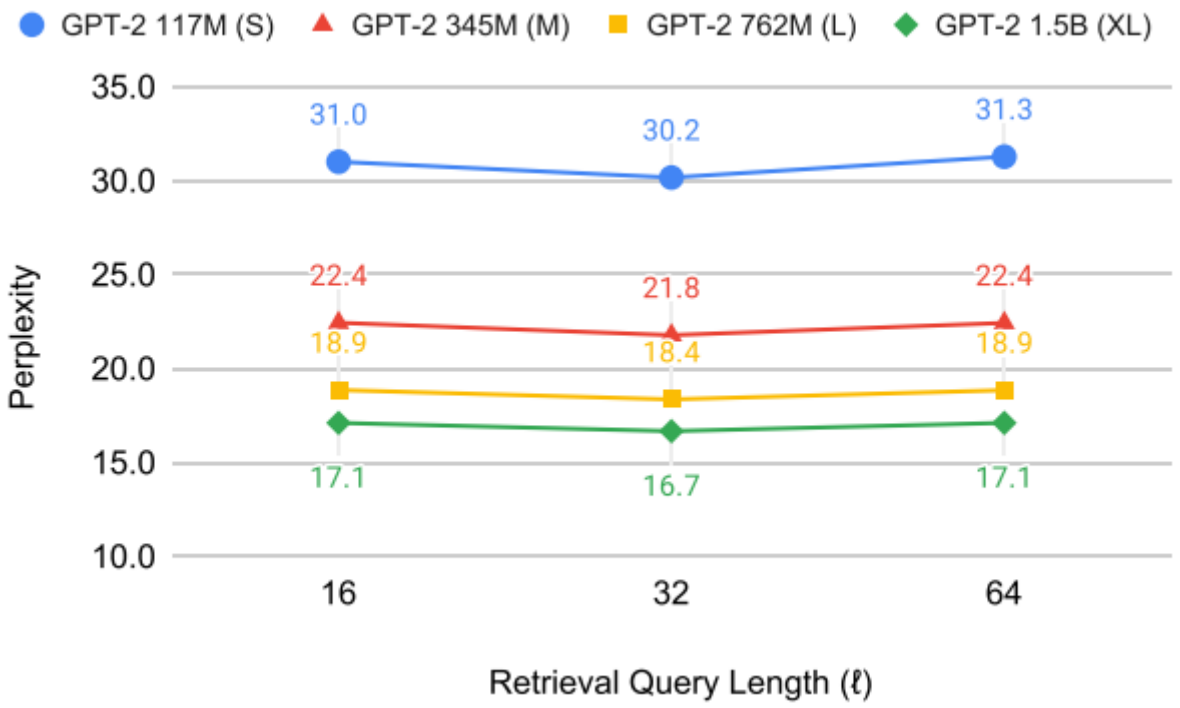
- An auto-regressive language model

Retrieval-in-context LM - experiments

1. Is it necessary to use the entire prefix as an input to the retrieval system?

Retrieval-in-context LM - experiments

1. Is it necessary to use the entire prefix as an input to the retrieval system?



→ Shorter prefix (more recent tokens) as a query helps

Retrieval-in-context LM - experiments

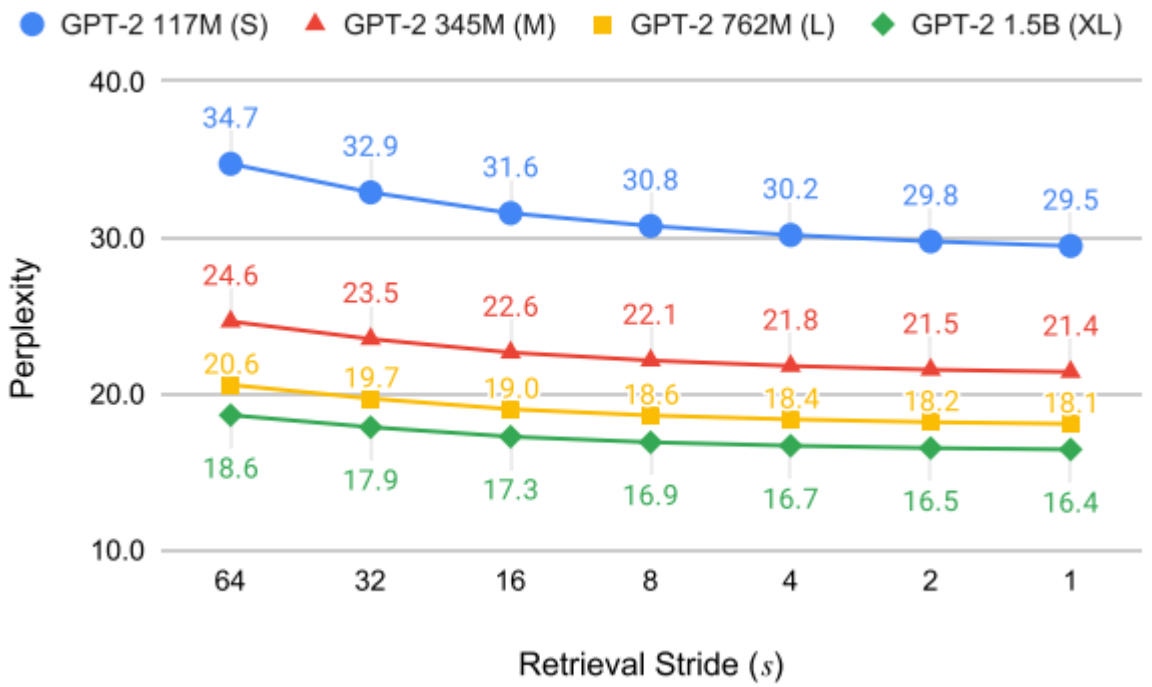
2. How frequent should we use retrieval?

Retrieval-in-context LM - experiments

2. How frequent should we use retrieval?

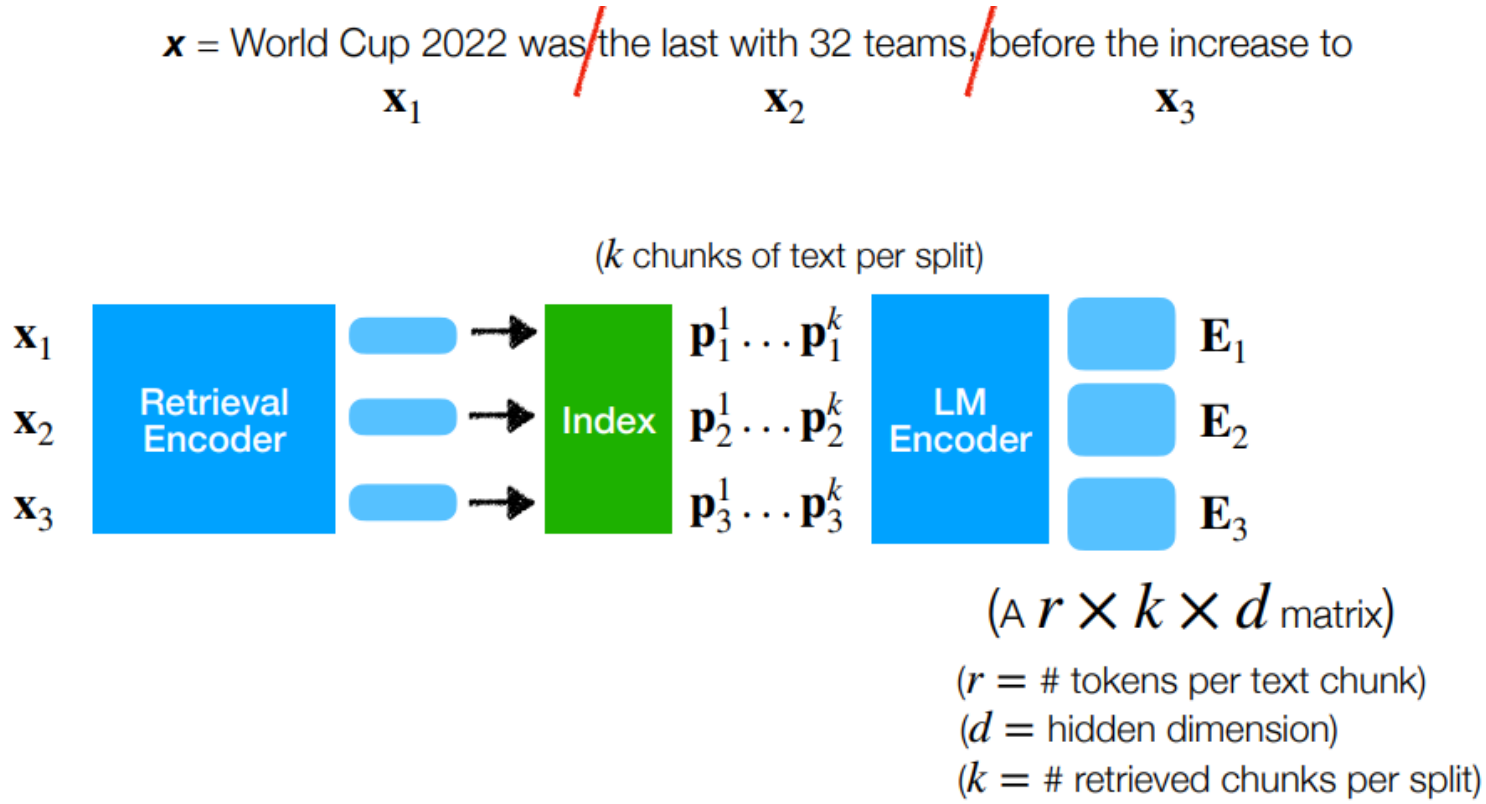
Retrieval-in-context LM - experiments

2. How frequent should we use retrieval?



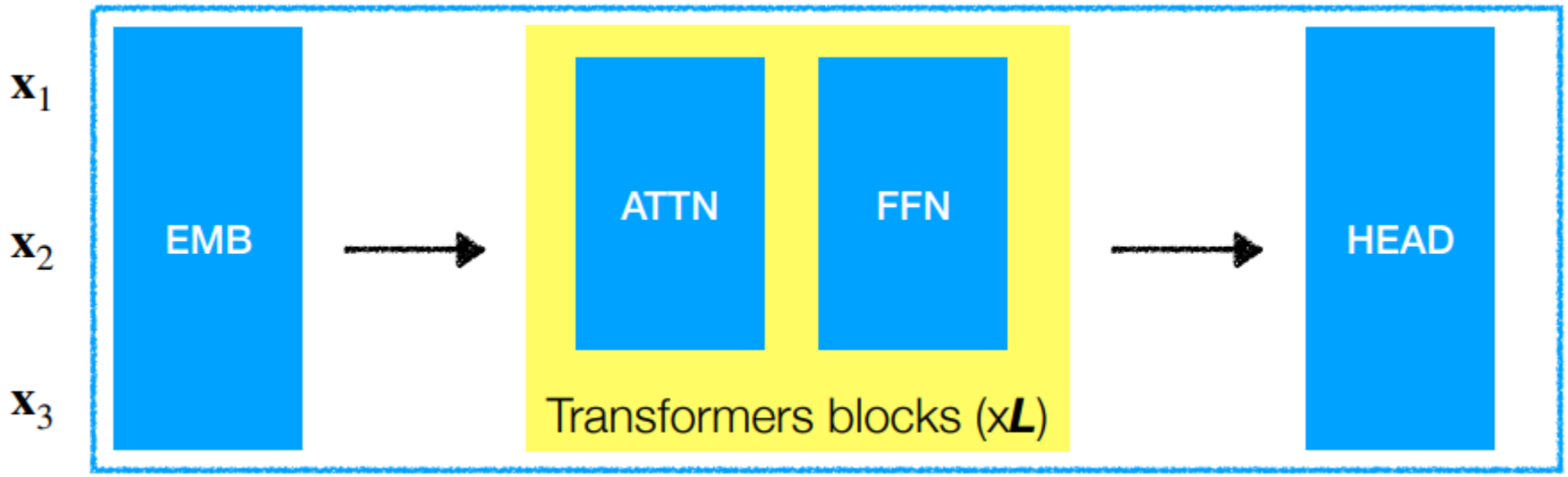
RETRO (Borgeaud et al. 2022)

- An auto-regressive language model
- Designed for many chunks, frequently, more efficiently
- Scale the datastore (1.8T tokens)



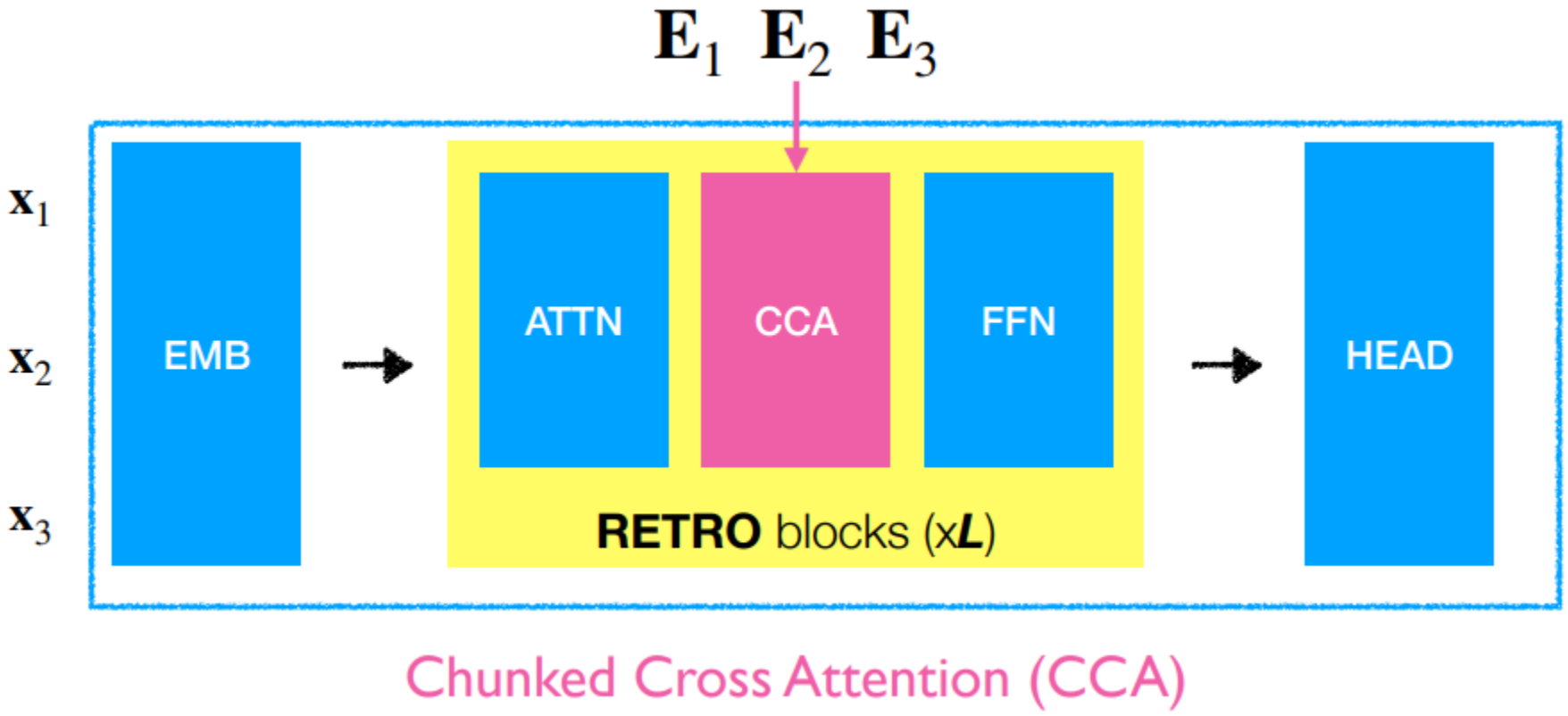
Decoder in RETRO

- Regular decoder



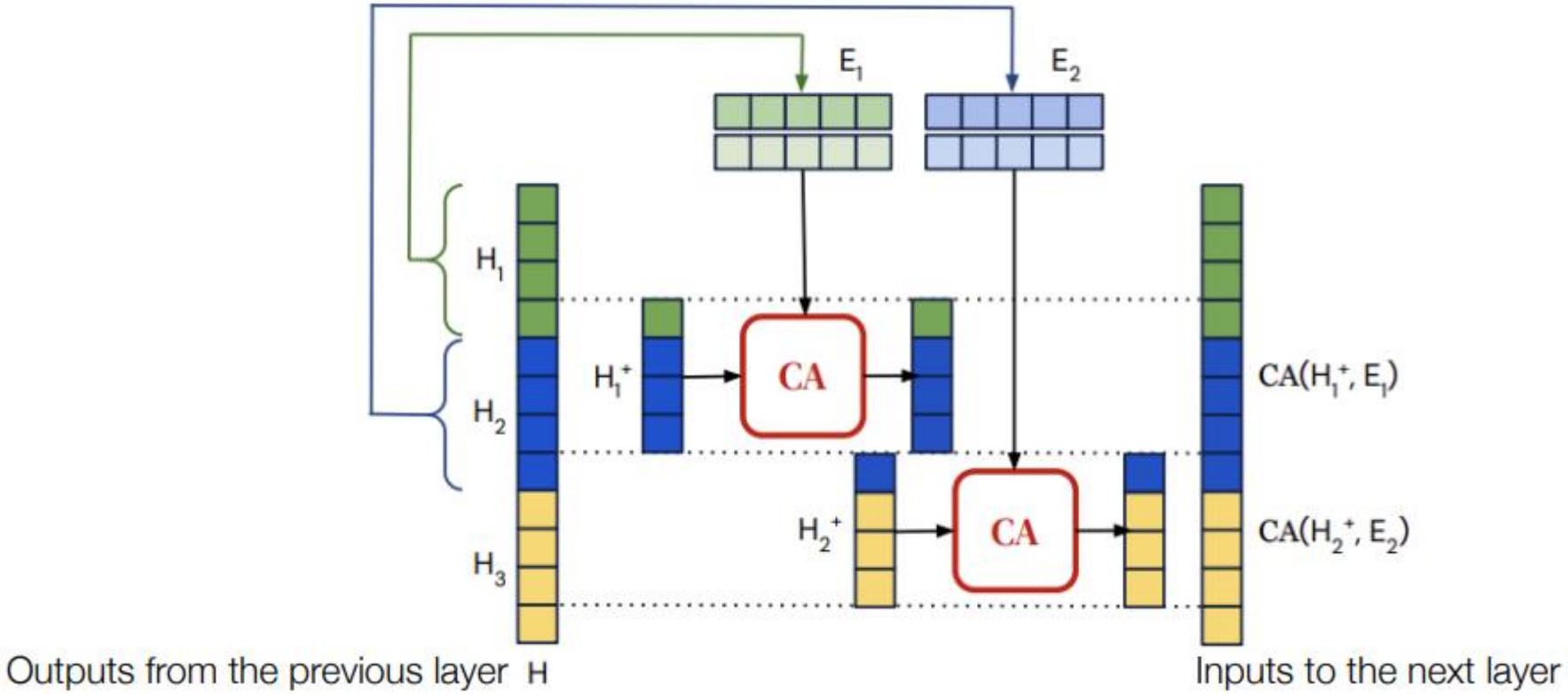
Decoder in RETRO

- RETRO decoder



Decoder in RETRO

- What is the CAA(Chunked Cross Attention) ?
  - To compute a cross-attention between the retriever result and the representation from the previous layer



Thank You

감사합니다.