ACL 2023 Tutorial:

Retrieval-based Language Models and Applications

Akari Asai, Sewon Min, Zexuan Zhong, Danqi Chen

University of Washington, Princeton University

발표자: 송선영

2024/01/23

Retrieval-based language model

	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
kNN-LM (Khandelwal et al. 2020)	Tokens	Output layer	Every token
FLARE (Jiang et al. 2023)	Text chunks	Input layer	Every n tokens *(adaptive)*
Adaptive kNN-LM (He et al 2021, Alon et al 2022, etc)	Tokens	Output layer	Every n tokens *(adaptive)*
Entities as Experts (Fevry et al. 2020), Mention Memory (de Jong et al. 2022)	Entities or entity mentions	Intermediate layers	Every entity mentions
Wu et al. 2022, Bertsch et al. 2023, Rubin & Berant. 2023	Text chunks ***from the input***	Intermediate layers	Once or every n tokens

Retrieval-based language model

	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
kNN-LM (Khandelwal et al. 2020)	Tokens	Output layer	Every token
FLARE (Jiang et al. 2023)	Text chunks	Input layer	Every n tokens *(adaptive)*
Adaptive kNN-LM (He et al 2021, Alon et al 2022, etc)	Tokens	Output layer	Every n tokens *(adaptive)*
Entities as Experts (Fevry et al. 2020), Mention Memory (de Jong et al. 2022)	Entities or entity mentions	Intermediate layers	Every entity mentions
Wu et al. 2022, Bertsch et al. 2023, Rubin & Berant. 2023	Text chunks ***from the input***	Intermediate layers	Once or every n tokens

REALM (Guu et al. 2020)

- A masked language model

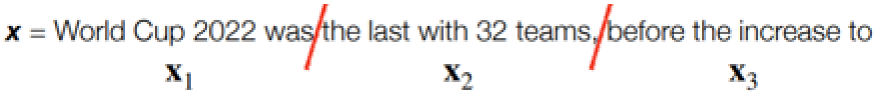
Retrieval-in-context LM (Ram et al. 2023, Shi et al. 2023)

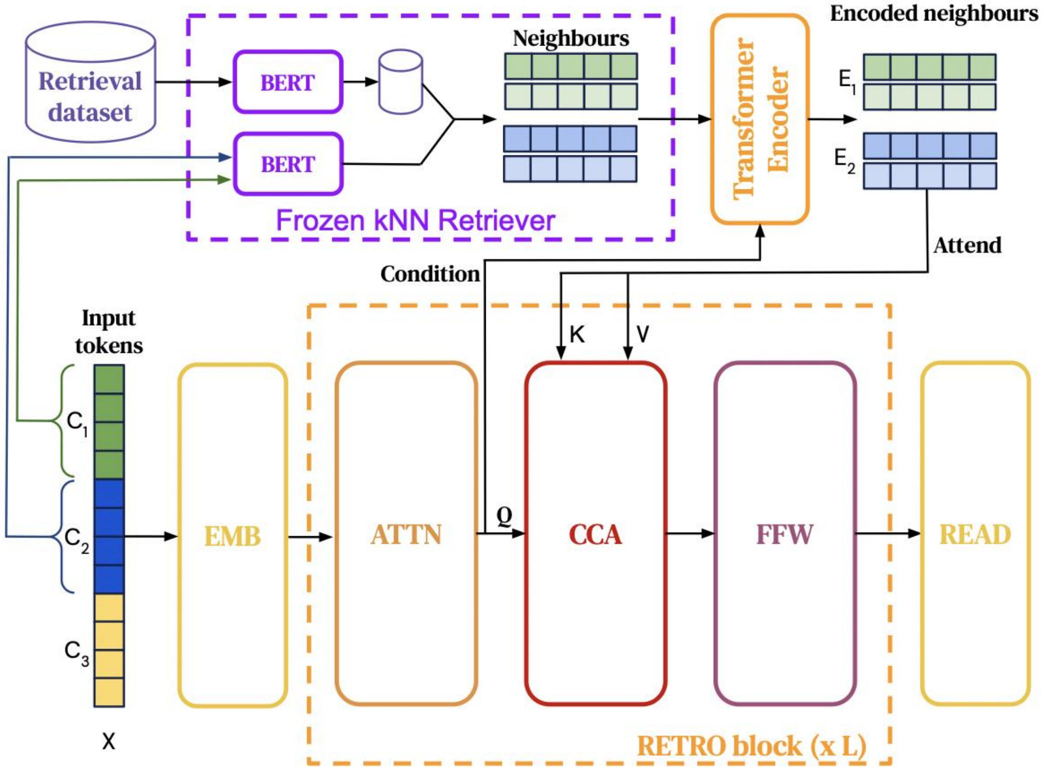
- An auto-regressive language model

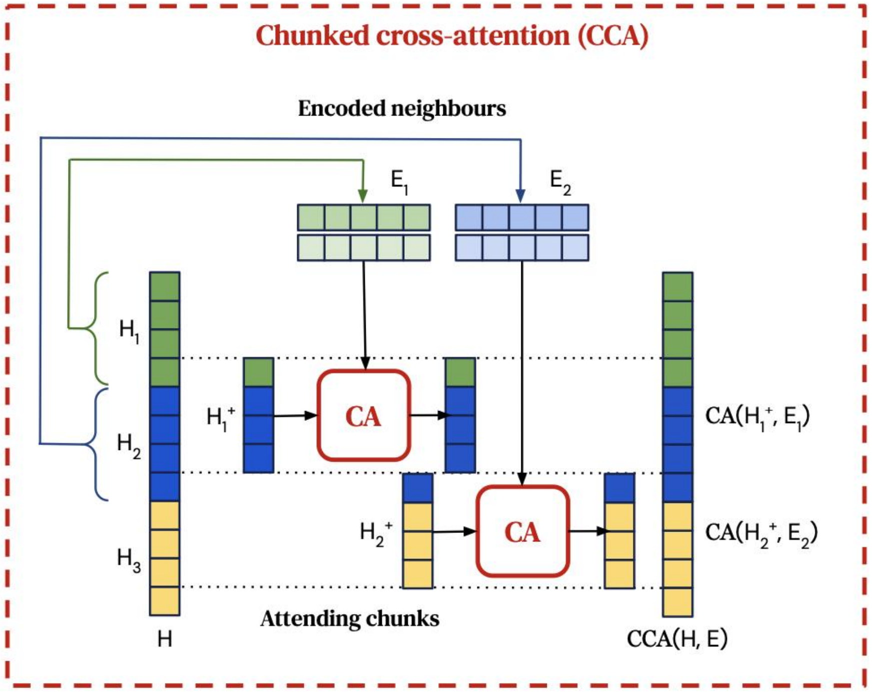
$\boldsymbol{x}$ = World Cup 2022 was the last with 32 teams, before the increase to

RETRO (Borgeaud et al. 2022)

- Using Chunked cross-attention (CCA)

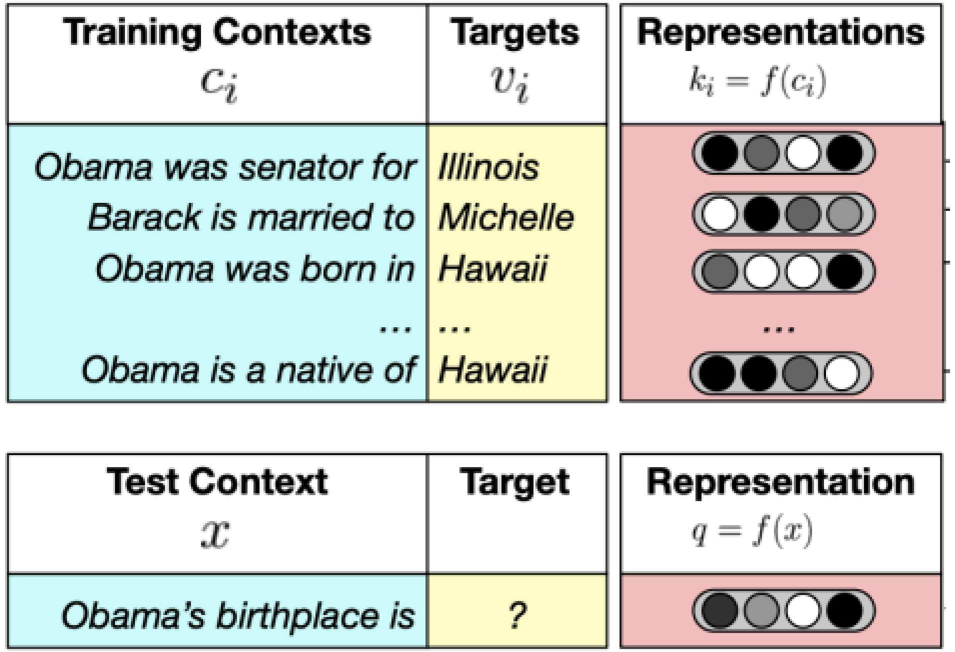






kNN-LM (Khandelwal et al. 2020)

- The size of the datastore = # of tokens in the corpus (not vocab size)



Which tokens <u>in a datastore</u> are close to the next token?

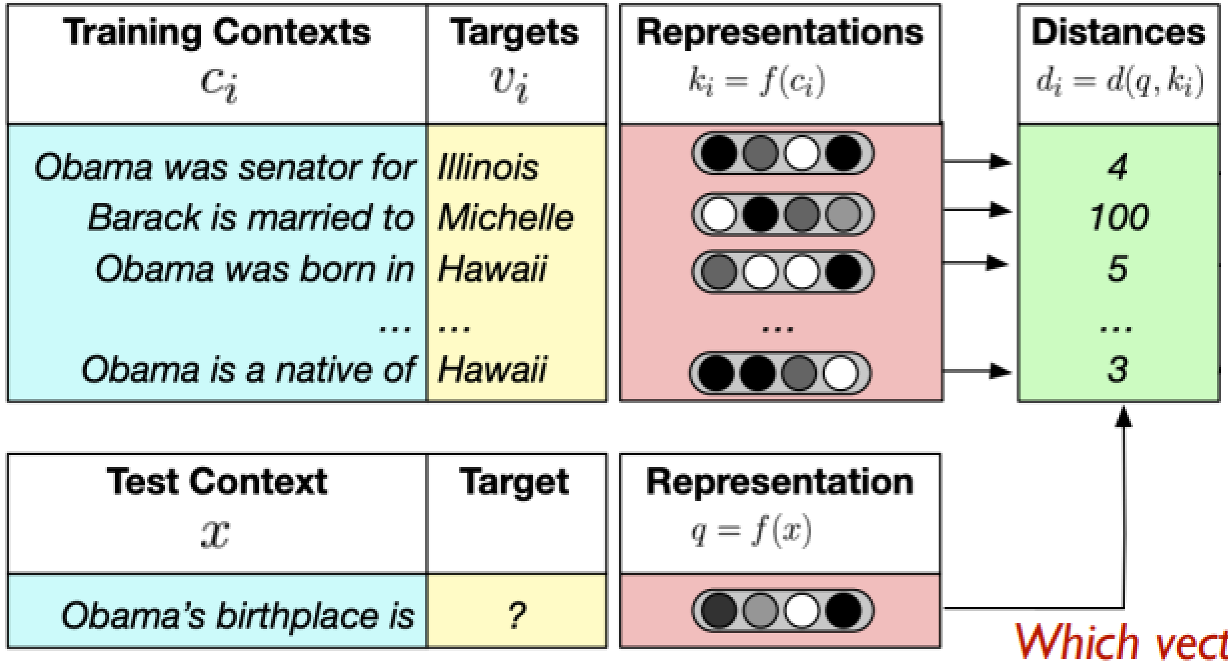
=

Which prefixes in a datastore are close to the prefix we have?

=

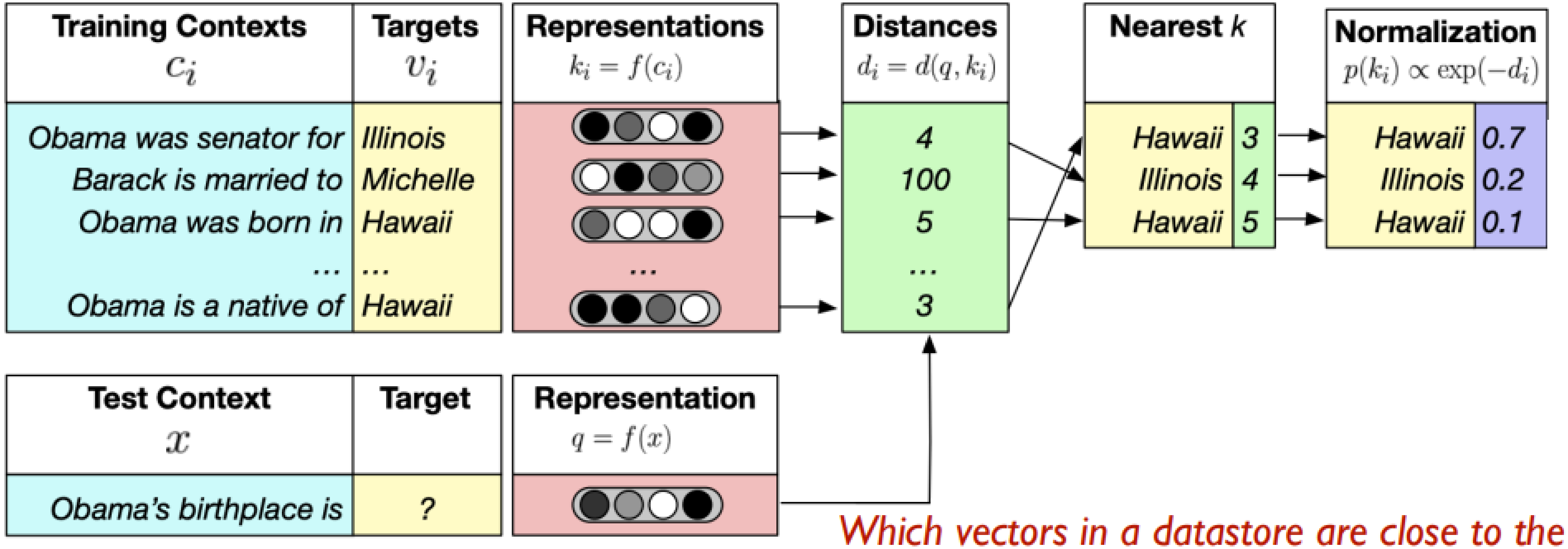
Which vectors in a datastore are close to the vector we have?

kNN-LM (Khandelwal et al. 2020)



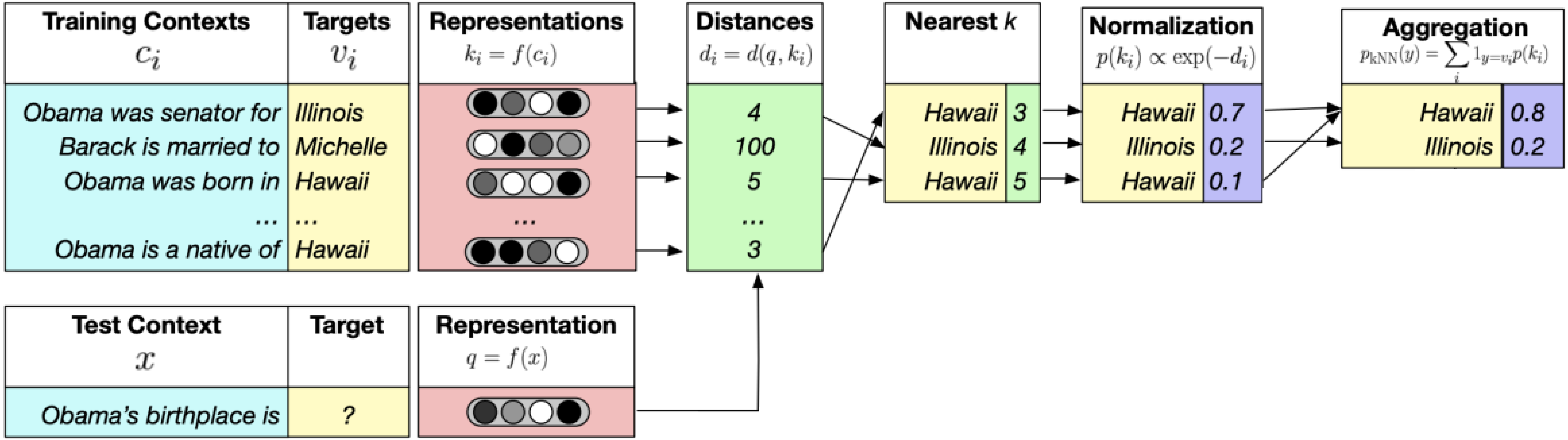
Which vectors in a datastore are close to the vector we have?

kNN-LM (Khandelwal et al. 2020)

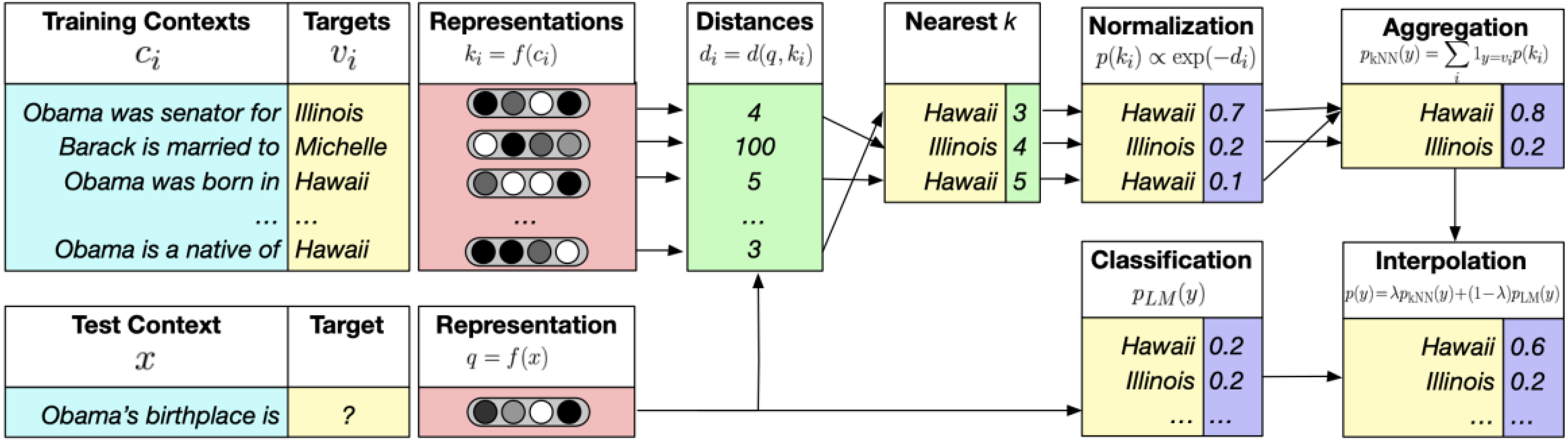


Which vectors in a datastore are close to the vector we have?

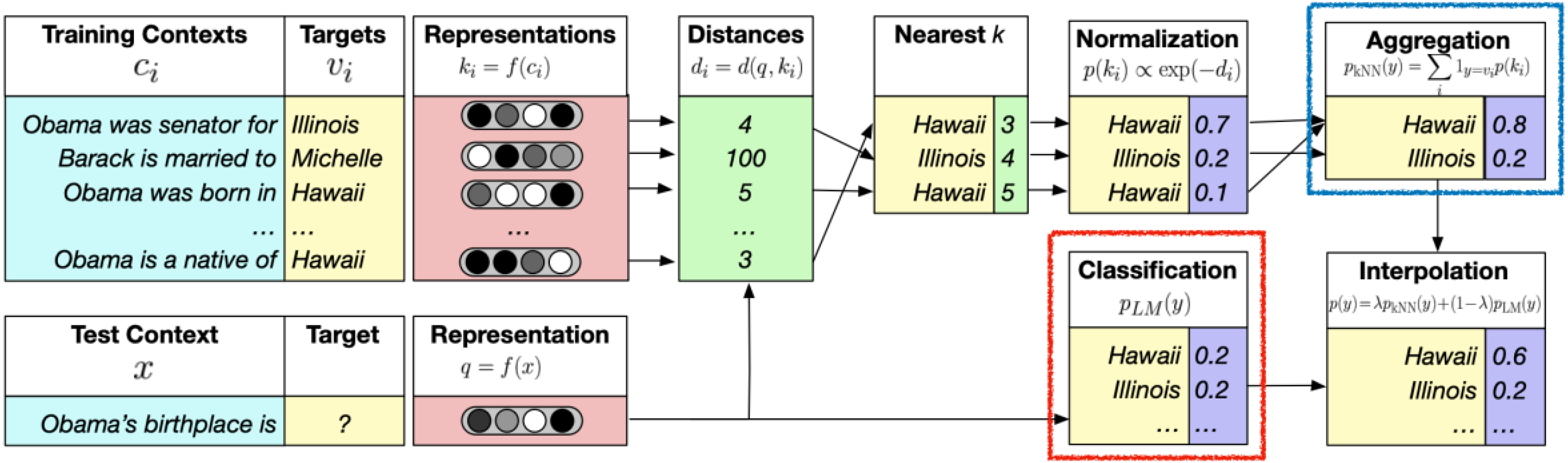
kNN-LM (Khandelwal et al. 2020)



kNN-LM (Khandelwal et al. 2020)



kNN-LM (Khandelwal et al. 2020)



λ: hyperparameter

$$P_{k\text{NN-LM}}(y \mid x) = (1-\lambda) P_{\text{LM}}(y \mid x) + \lambda P_{k\text{NN}}(y \mid x)$$

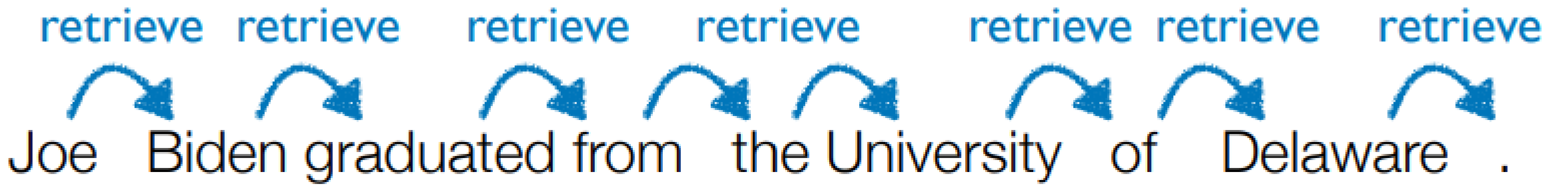
Adaptive kNN-LM (He et al. 2021, Along et al. 2022)

- Adaptive retrieval of **tokens**

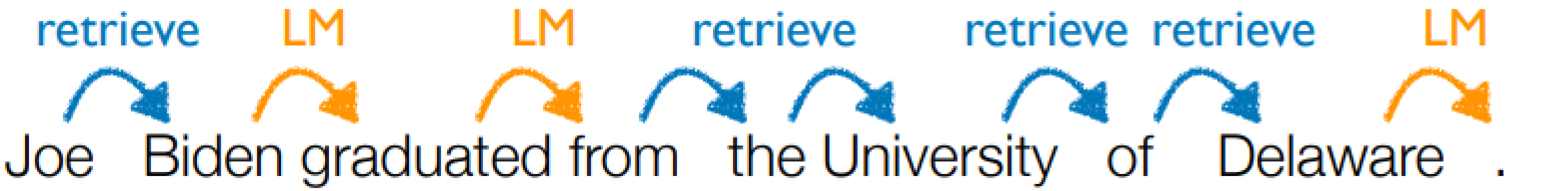
kNN-LM (Khandelwal et al. 2020)	Tokens	Output layer	Every token
Adaptive kNN-LM (He et al 2021, Alon et al 2022, etc)	Tokens	Output layer	Every n tokens *(adaptive)*

Adaptive kNN-LM (He et al. 2021, Along et al. 2022)

kNN-LM



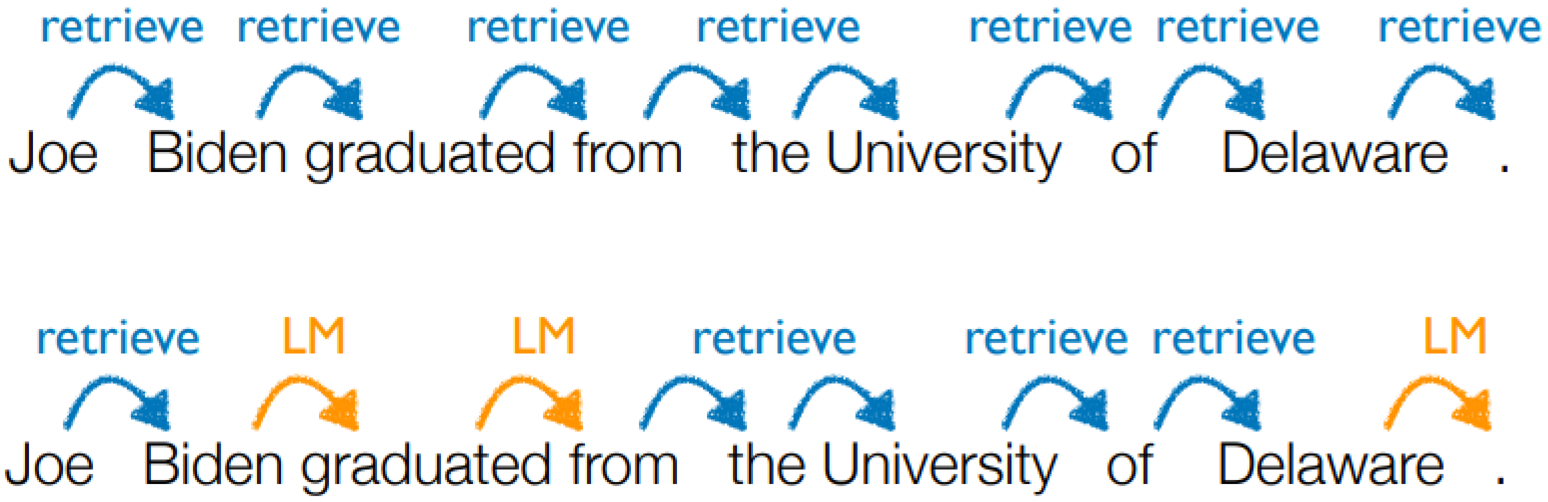
Adaptive kNN-LM



Adaptive kNN-LM (He et al. 2021, Along et al. 2022)

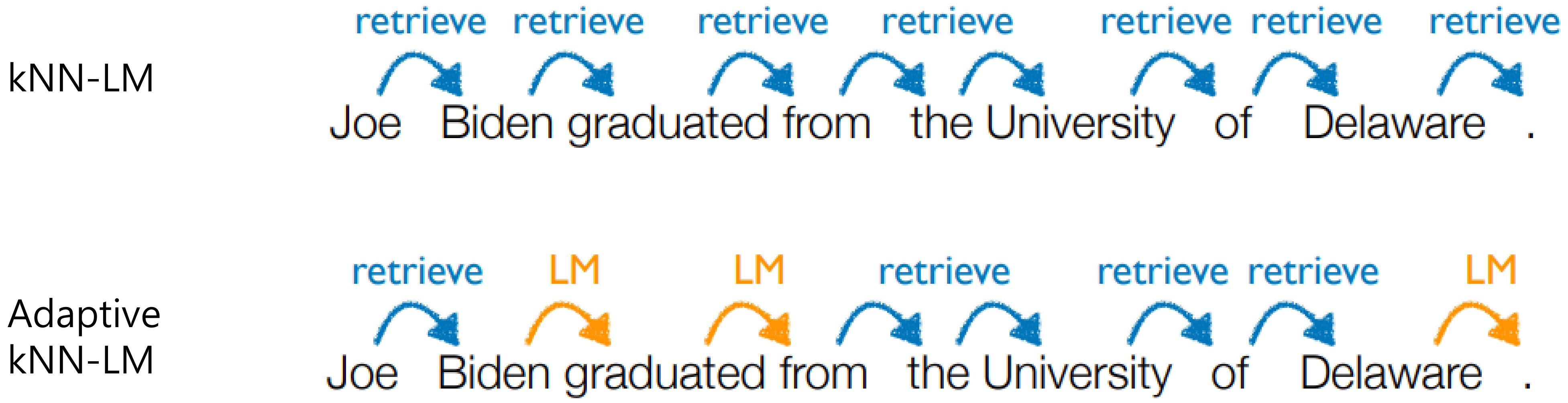
kNN-LM

Adaptive kNN-LM



$$P_{k\mathrm{NN-LM}}(y\,|\,x) = (1 - \lambda(x))P_{\mathrm{LM}}(y\,|\,x) + \lambda(x)P_{k\mathrm{NN}}(y\,|\,x)$$

Adaptive kNN-LM (He et al. 2021, Along et al. 2022)



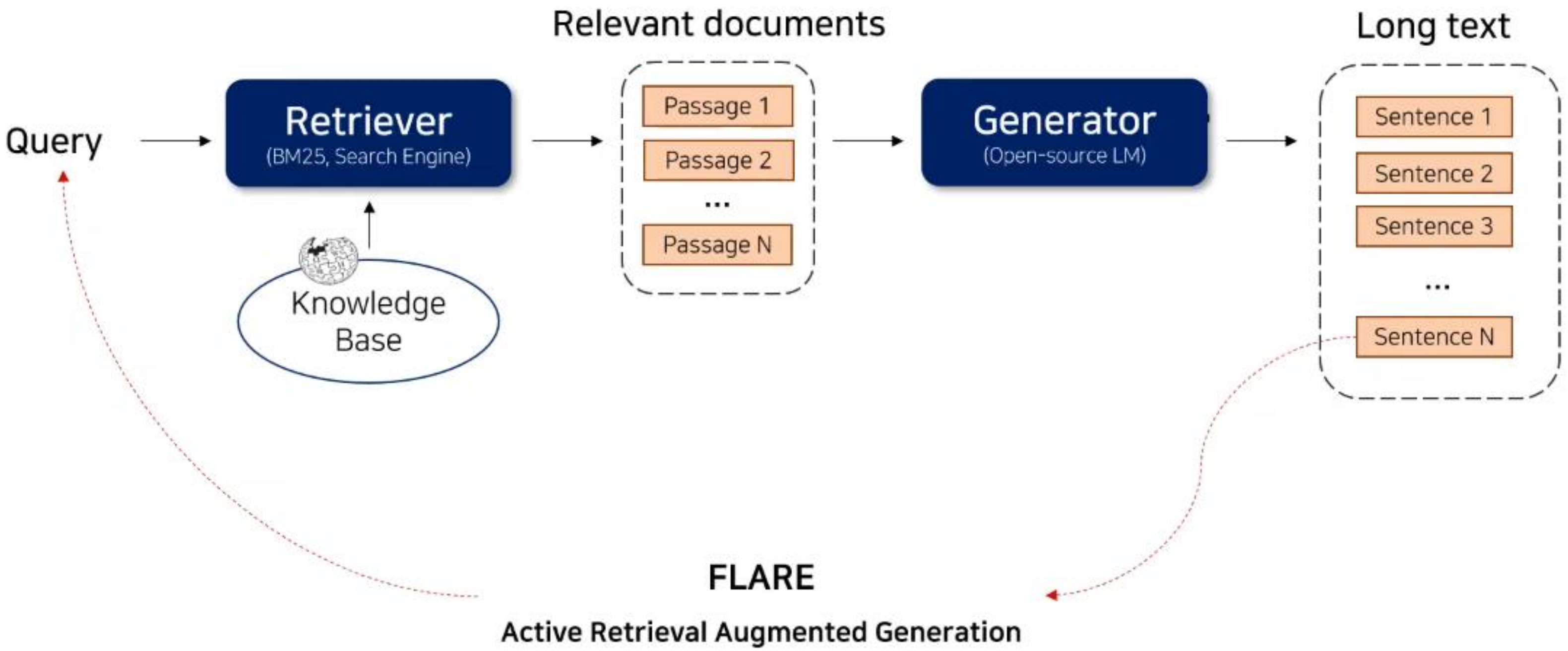
$$P_{k\mathrm{NN-LM}}(y\,|\,x) = (1 - \underline{\lambda(x)})P_{\mathrm{LM}}(y\,|\,x) + \underline{\lambda(x)}P_{k\mathrm{NN}}(y\,|\,x)$$

$$\mathcal{L} = \frac{1}{T}\sum_{t}\left[\log p(w_t|c_t;\lambda_\theta(c_t)) - a \cdot \lambda_\theta(c_t)\right]$$

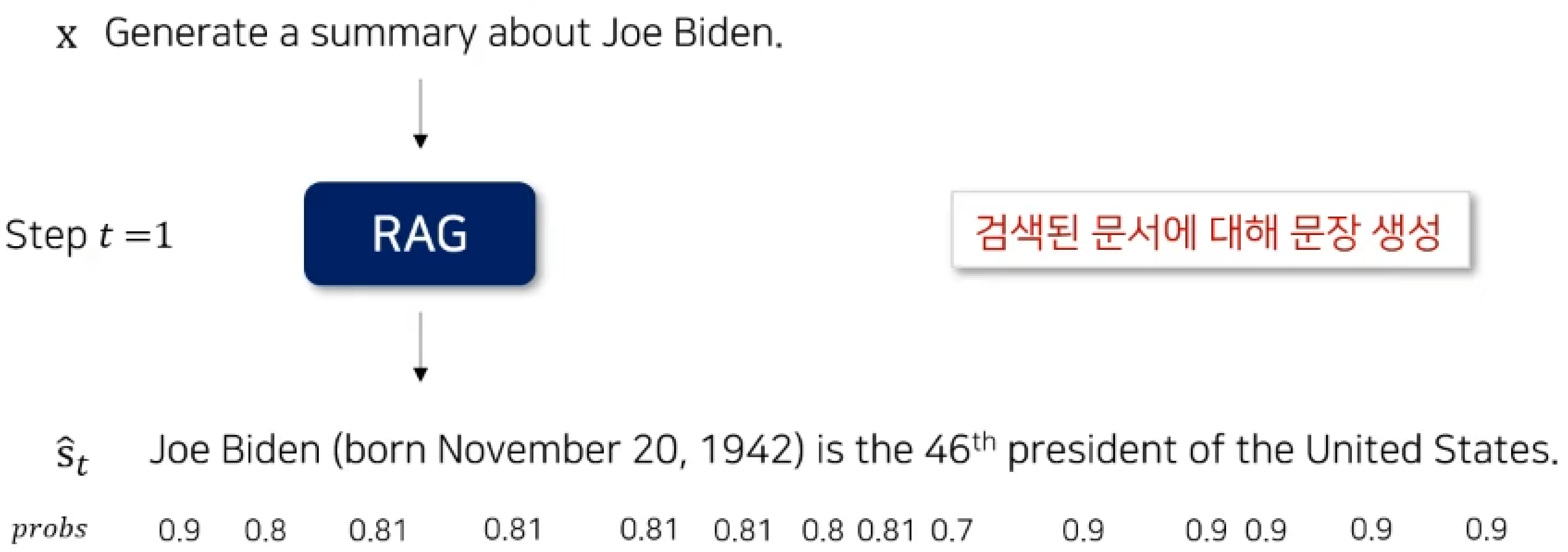
FLARE (Jiang et al. 2023)

- Adaptive retrieval of **chunks**

Retrieve-in-context LM (Shi et al 2023, Ram et al 2023)	Text chunks	Input layer	Every n tokens
FLARE (Jiang et al. 2023)	Text chunks	Input layer	Every n tokens *(adaptive)*



FLARE (Jiang et al. 2023)



출처: https://www.youtube.com/watch?v=lEqeXo-IbKw

FLARE (Jiang et al. 2023)

- Query formulation

x Generate a summary about Joe Biden.

$\hat{s}_t$를 기반으로, 검색을 trigger 할 지 query formulation 진행

Step t =1

RAG

$$q_t = \begin{cases} \emptyset & \text{if all tokens of } \hat{s}_t \text{ have probs} \geq \theta \\ \text{mask}(\hat{s}_t) \text{ or } \text{qgen}(\hat{s}_t) & \text{otherwise} \end{cases}$$

$\hat{s}_t$ Joe Biden (born November 20, 1942) is the 46th president of the United States.

probs 0.9 0.8 0.81 0.81 0.81 0.81 0.8 0.81 0.7 0.9 0.9 0.9 0.9 0.9

$\theta \in [0, 1]$

0.7

FLARE (Jiang et al. 2023)

- Query formulation
  1. Mask (implicit)
  2. Qgen (explicit)

x Generate a summary about Joe Biden.

y_1 Joe Biden (born November 20, 1942) is the 46th president of the United States.

$\hat{s}_t$를 기반으로, 검색을 trigger 할 지 query formulation 진행

Step t =2

RAG

$$q_t = \begin{cases} \emptyset & \text{if all tokens of } \hat{s}_t \text{ have probs} \geq \theta \\ \text{mask}(\hat{s}_t) \text{ or qgen}(\hat{s}_t) & \text{otherwise} \end{cases}$$

$\hat{s}_t$ Joe Biden attended the University of Pennsylvania, where he earned a law degree.

probs 0.9 0.8 0.81 0.74 0.62 0.7 0.58 0.9 0.9 0.9 0.7 0.5 0.73

θ 0.7

FLARE (Jiang et al. 2023)

1. Implicit query by masking

$\hat{s}_t$ Joe Biden attended the University of Pennsylvania, where he earned a law degree. β

$probs$ 0.9 0.8 0.81 0.74 0.62 0.7 0.58 0.9 0.9 0.9 0.7 0.5 0.73 0.8

Joe Biden attended , where he earned .

RAG

출처: https://www.youtube.com/watch?v=lEqeXo-IbKw

FLARE (Jiang et al. 2023)

2. Explicit query by question generation (qgen)

$\hat{s}_t$ Joe Biden attended the University of Pennsylvania, where he earned a law degree. β

probs 0.9 0.8 0.81 0.74 0.62 0.7 0.58 0.9 0.9 0.9 0.7 0.5 0.73 0.8

Ask a question to which the answer is "the University of Pennsylvania"

Ask a question to which the answer is "a law degree"

$\hat{s}_t$ 에서 probs가 낮은 span z 를 대상으로 explicit query 생성

LM
(chatGPT)

What university did Joe Biden attend?

What degree did Joe Biden earn?

Prompt 3.2: zero-shot question generation

User input $\boldsymbol{x}$.
Generated output so far $\boldsymbol{y}_{\leq t}$.

Given the above passage, ask a question to which the answer is the term/entity/phrase "$\boldsymbol{z}$".

출처: https://www.youtube.com/watch?v=lEqeXo-IbKw

FLARE (Jiang et al. 2023)

2. Explicit query by question generation

What university did Joe Biden attend?

What degree did Joe Biden earn?

RAG

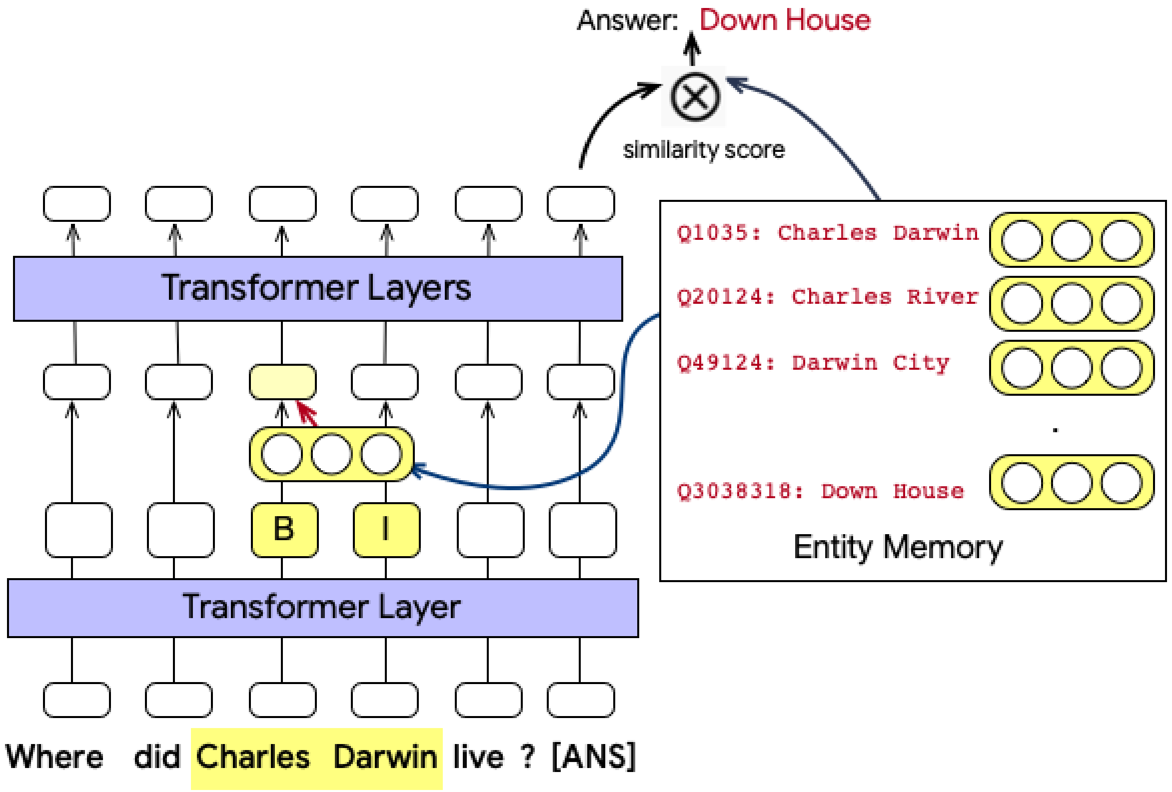
s_2 He graduated from the University of Delaware in 1965 with a Bachelor of Arts in history and political science

출처: https://www.youtube.com/watch?v=lEqeXo-IbKw

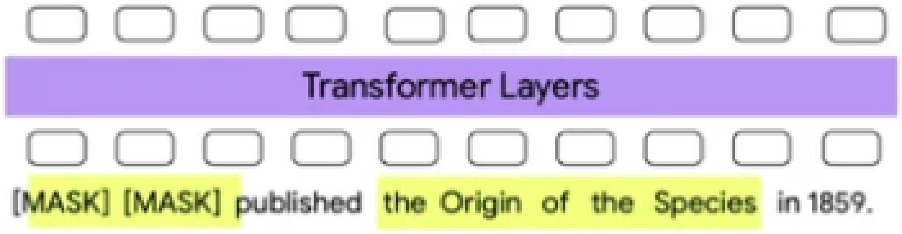
Entities as Experts (Fevry et al. 2020)

- Entities as Experts (EAE)

Entities as Experts (Fevry et al. 2020), Mention Memory (de Jong et al. 2022)	Entities or entity mentions	Intermediate layers	Every entity mentions

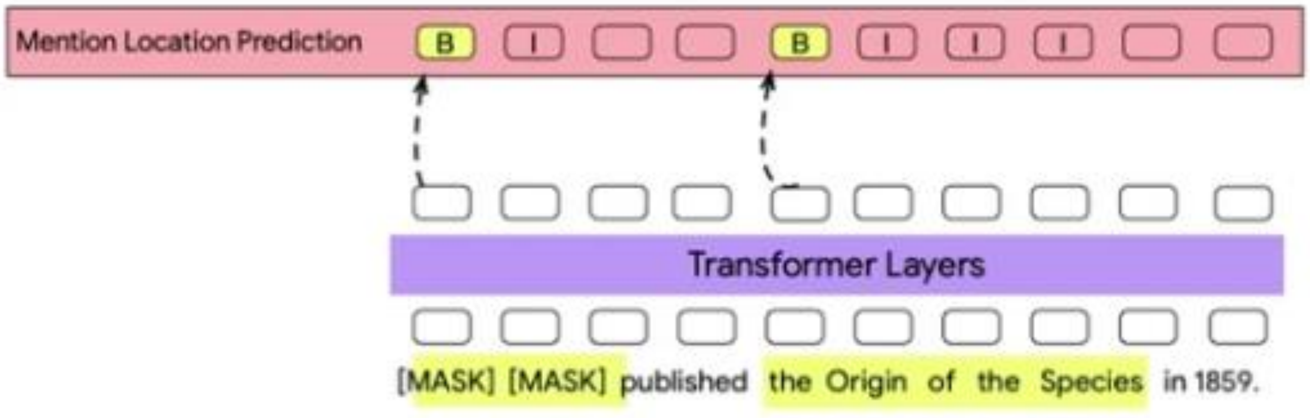


Entities as Experts (Fevry et al. 2020)



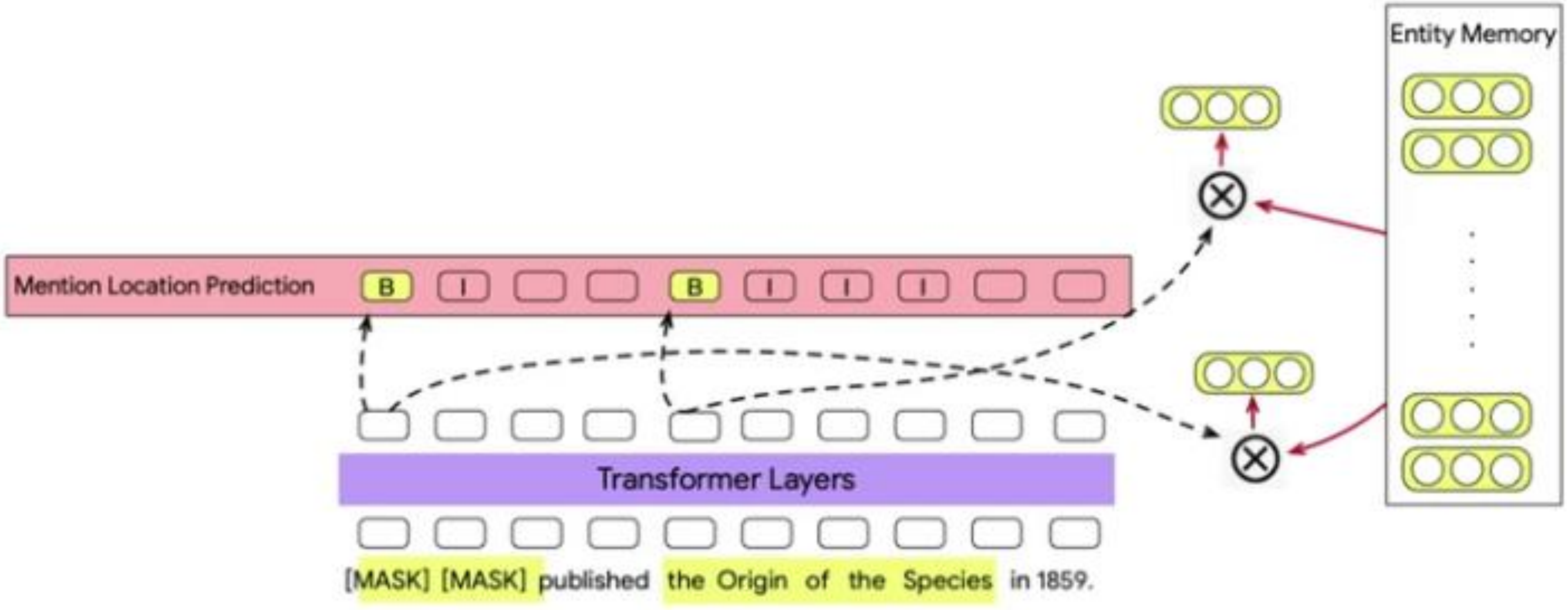
"Charles Darwin published the Origin of the Species in 1859."

Entities as Experts (Fevry et al. 2020)



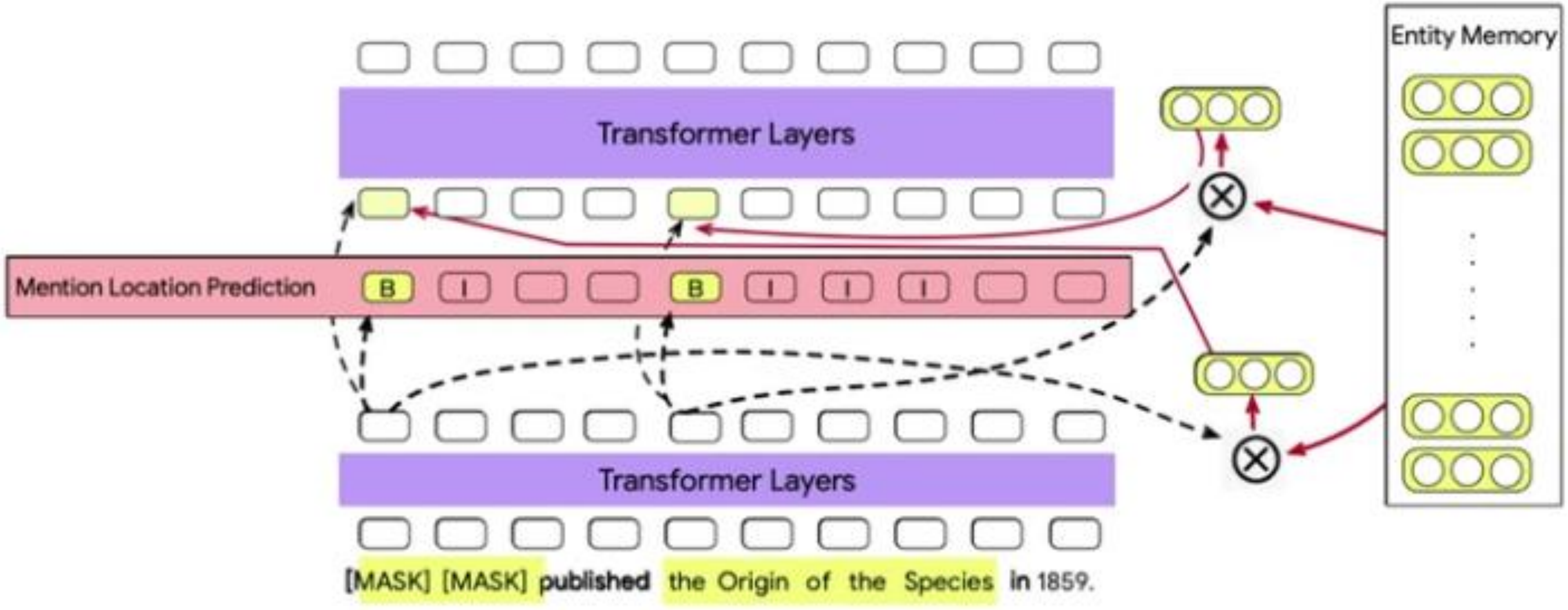
"Charles Darwin published the Origin of the Species in 1859."

Entities as Experts (Fevry et al. 2020)



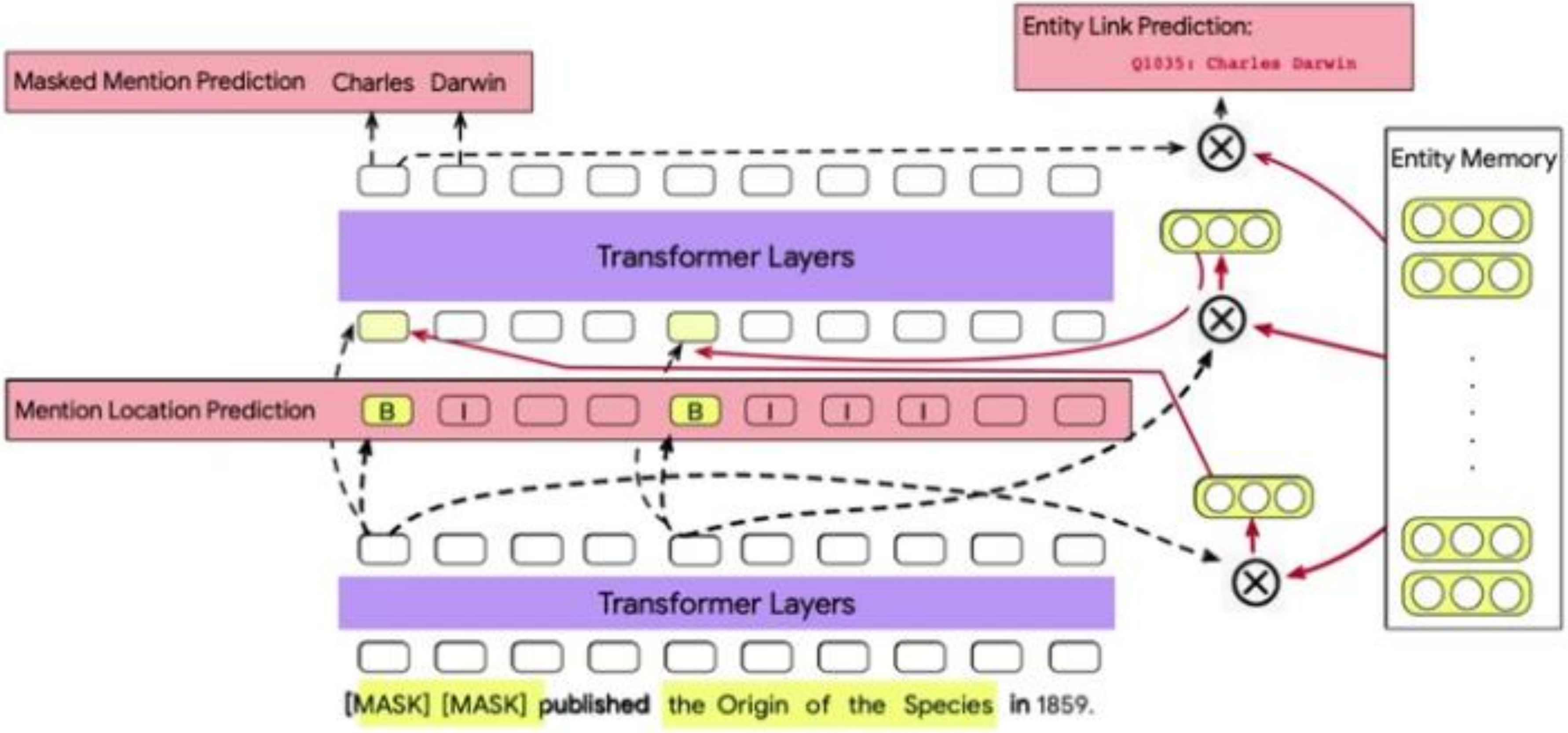
“Charles Darwin published the Origin of the Species in 1859.”

Entities as Experts (Fevry et al. 2020)



"Charles Darwin published the Origin of the Species in 1859."

Entities as Experts (Fevry et al. 2020)

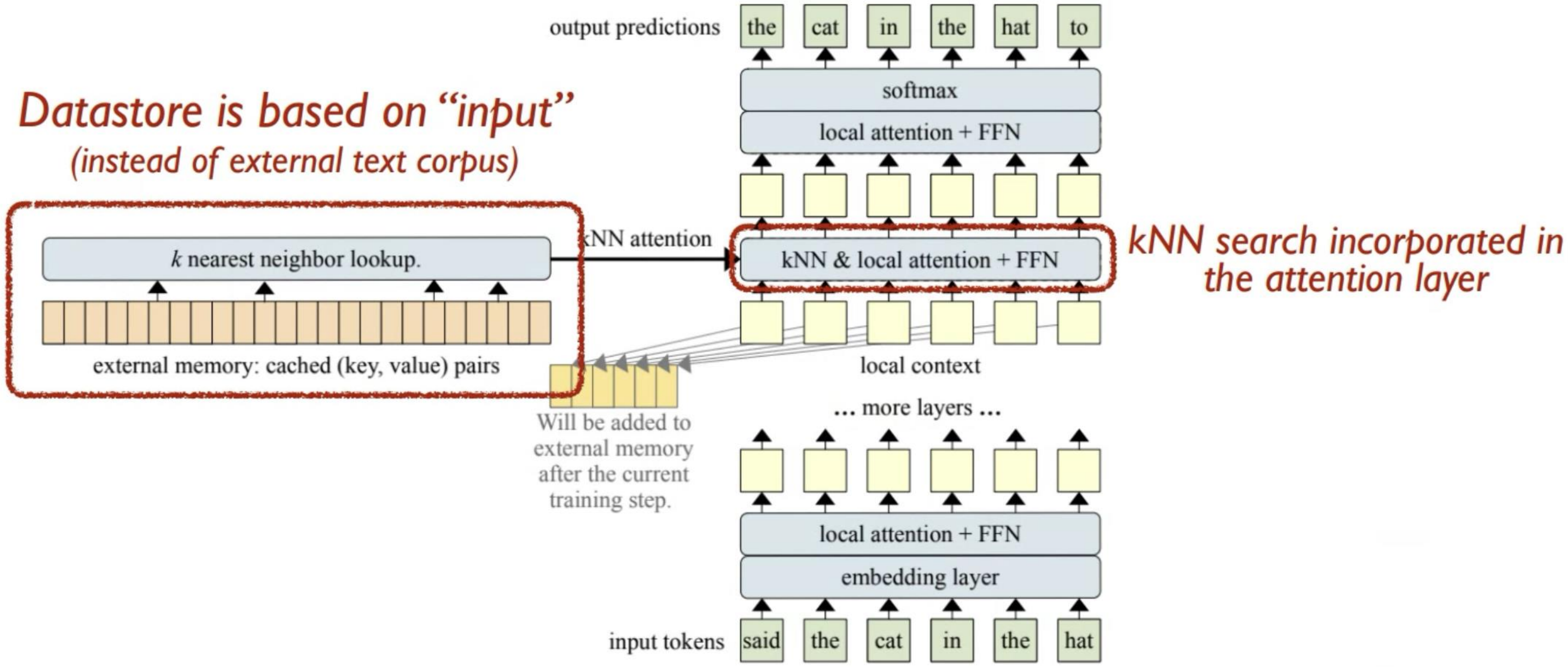


“Charles Darwin published the Origin of the Species in 1859.”

Retrieval for long-range LM (Wu et al. 2022)

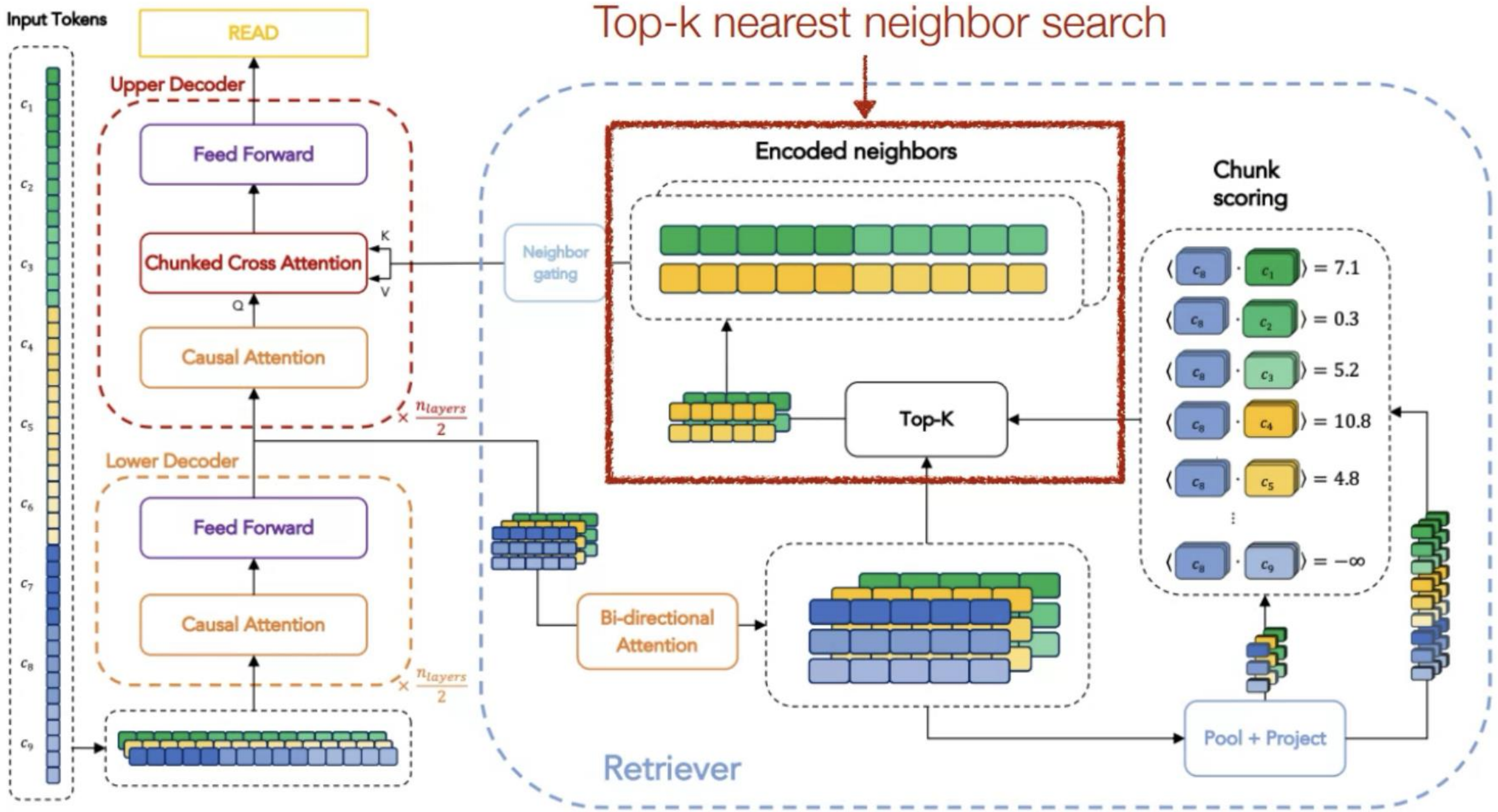
- It is hard to process long input (entire book) because transformer have a sequence length limit

Wu et al. 2022, Bertsch et al. 2023, Rubin & Berant. 2023	Text chunks ***from the input***	Intermediate layers	Once or every n tokens



Retrieval for long-range LM (Rubin & Brent et al. 2023)

- This is from the RETRO



Thank You

감사합니다.