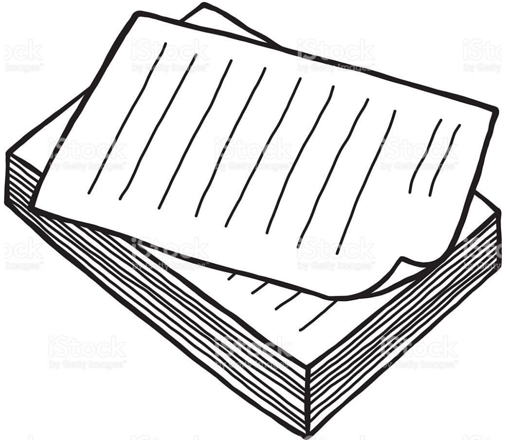
Unlimiformer: Long-Range Transformers with Unlimited Length Input

Amanda Bertsch Uri Alon Graham Neubig Matt Gormley

The inputs we'd like to work with keep getting bigger…

1,000 tokens



10,000 tokens

100,000 tokens

…and our models don't scale that well

100,000 tokens

- Sparse attention
  - Pretraining is hugely expensive
  - Fixed maximum length
- Hierarchical summarization
  - Cascading errors
  - Can't see the big picture
- ???

The ***length*** of the context window is fixed… what about the ***content***?

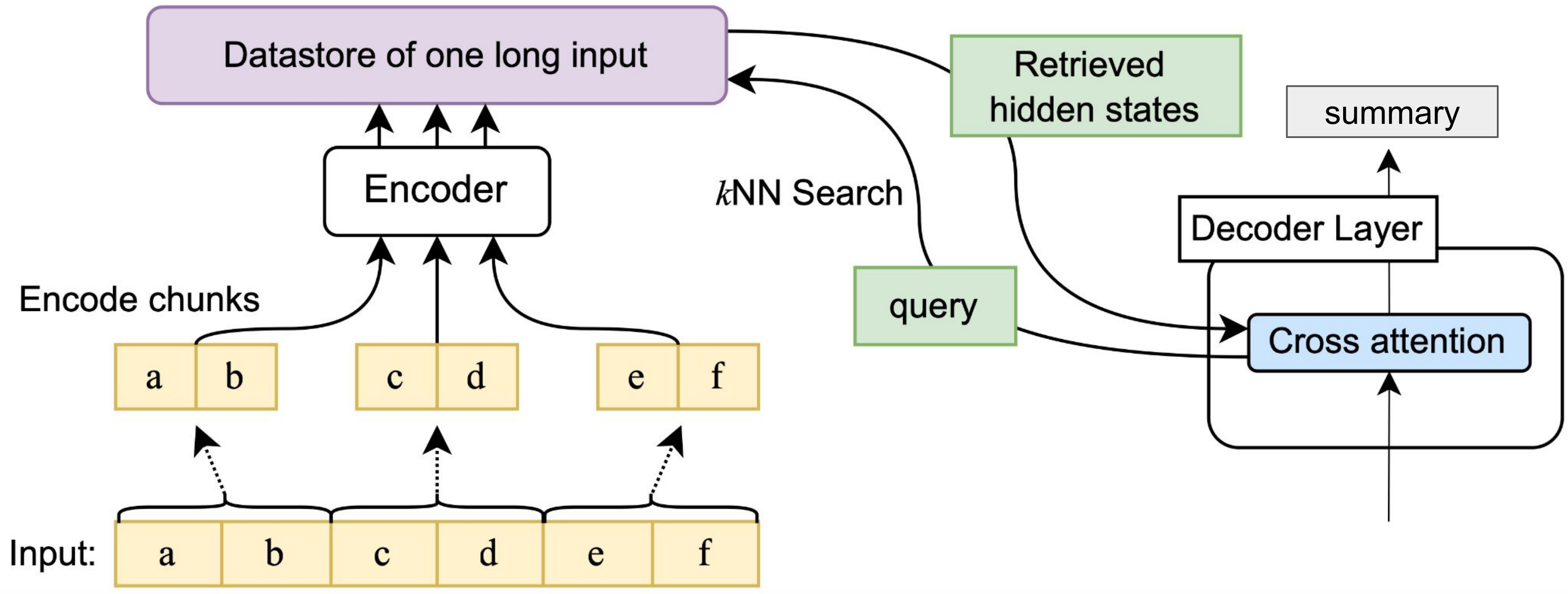
Retrieval-augmented generation

100,000 tokens

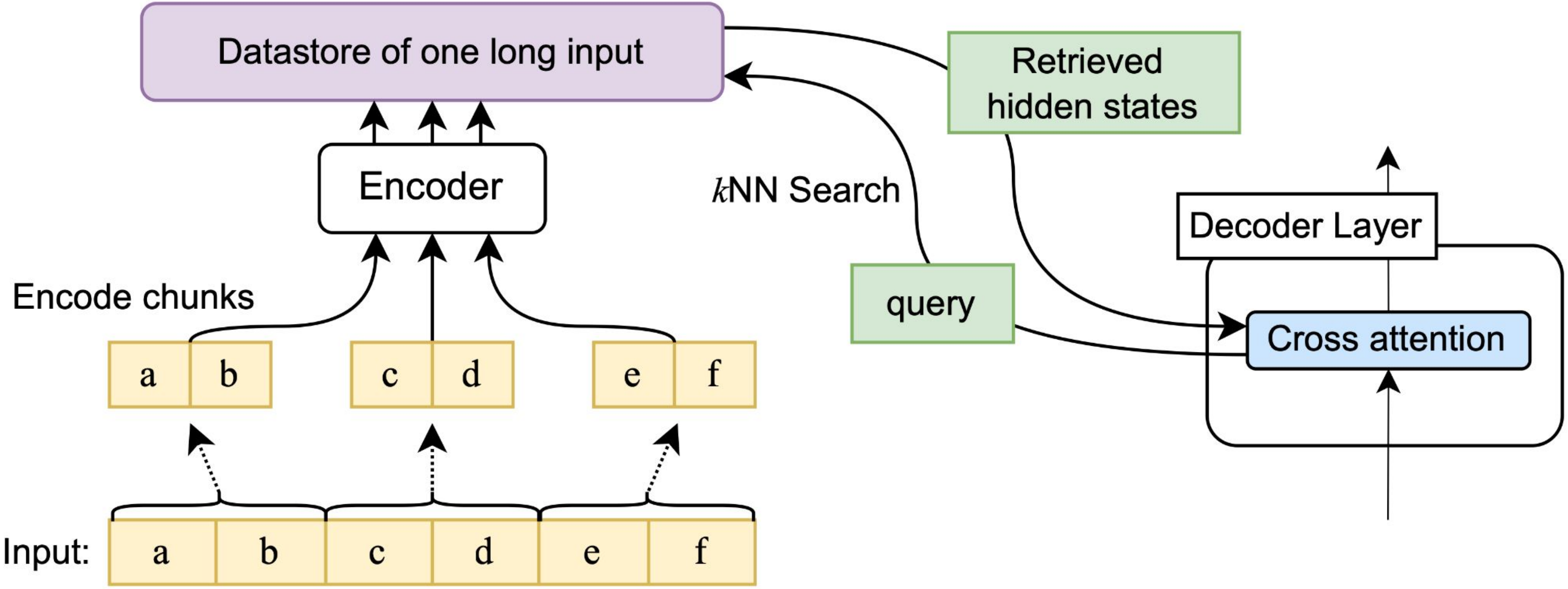
RETRO, Memorizing Transformers, etc:

- maintain a “base context” and augment with retrieved text
  - Unlimiformer has no “base context”
- add a layer (or a few layers) that cross attend to both external memory and the context
  - Unlimiformer cross attends only to external memory at every layer
- retrieve from set of relevant documents for QA or full pretraining corpus/recent examples for LM
  - Unlimiformer retrieves from the same long sequence
  - The datastore is static and unique for a single example

Unlimiformer



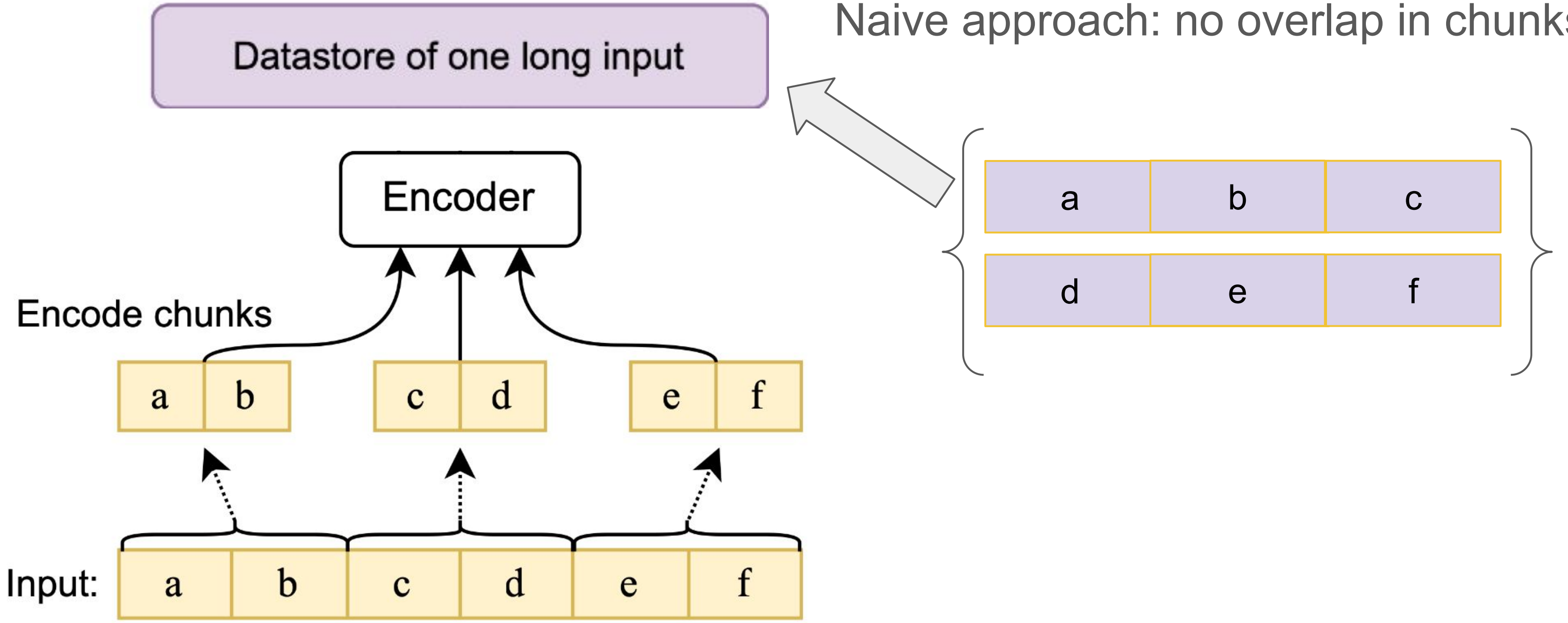
How do we do encoding?



Number of encoder passes: ⌈input len / encoder max len⌉

How do we do encoding?

Naive approach: no overlap in chunks



Number of encoder passes: ⌈input len / encoder max len⌉

What about context?

embeddings with no left context: a d

embeddings with left+right context: b e

embeddings with no right context: c f

What about positional embeddings?

encoding:

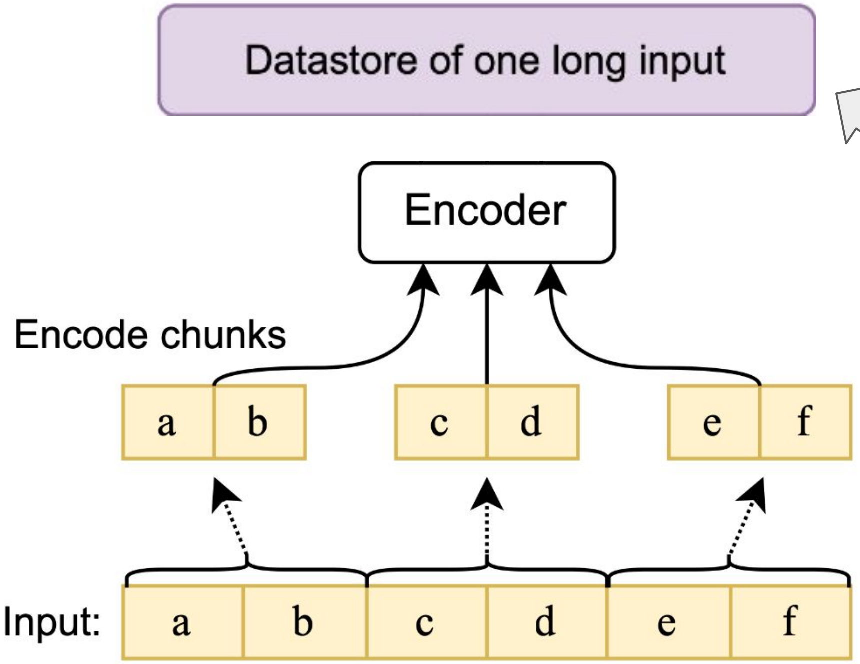
a	b	c
d	e	f

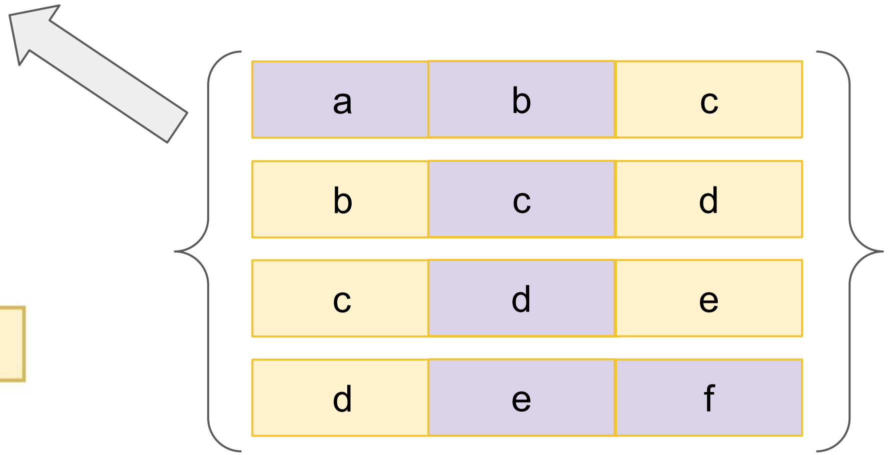
positional embeddings:

a	b	c	d	e	f
1	2	3	1	2	3

How do we do encoding?

Overlapping chunks: all tokens in the middle of the input have left and right context!



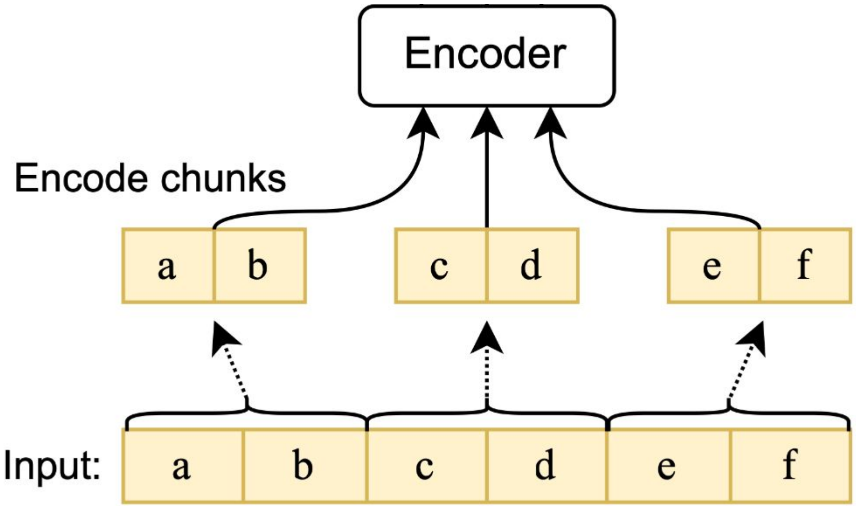


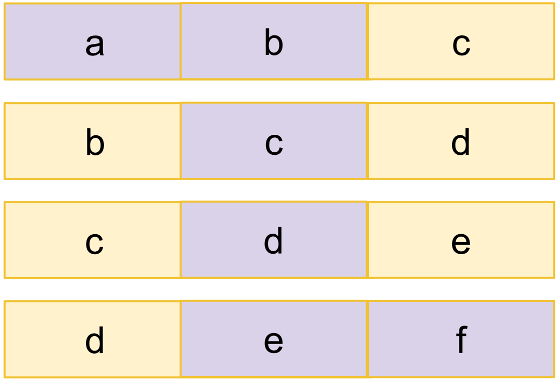
in practice: use embeddings from middle half of window

Number of encoder passes: ⌈input len / (0.5 * encoder max len)⌉ - 1

How do we do encoding?

Overlapping chunks: all tokens in the middle of the input have left and right context!





in practice: use embeddings from middle half of window

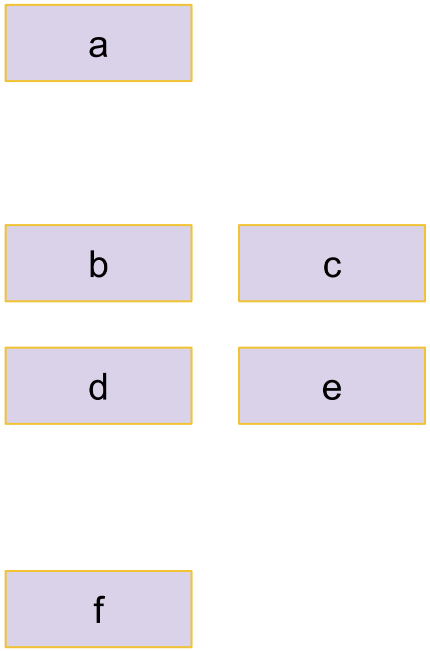
Number of encoder passes: ⌈input len / (0.5 * encoder max len)⌉ - 1

What about context?

embeddings with no left context:

embeddings with left+right context:

embeddings with no right context:



What about positional embeddings?

encoding:

a	b	c
b	c	d
c	d	e
d	e	f

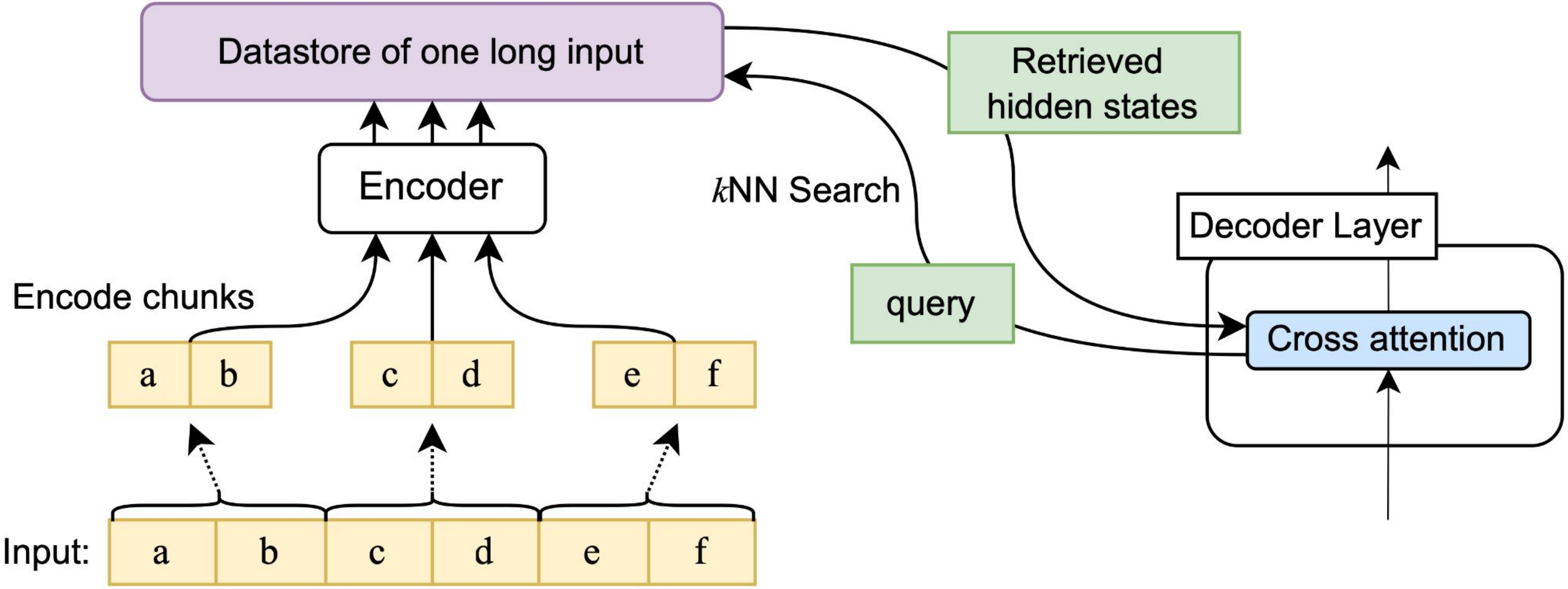
positional embeddings:

a	b	c	d	e	f
1	2	2	2	2	3

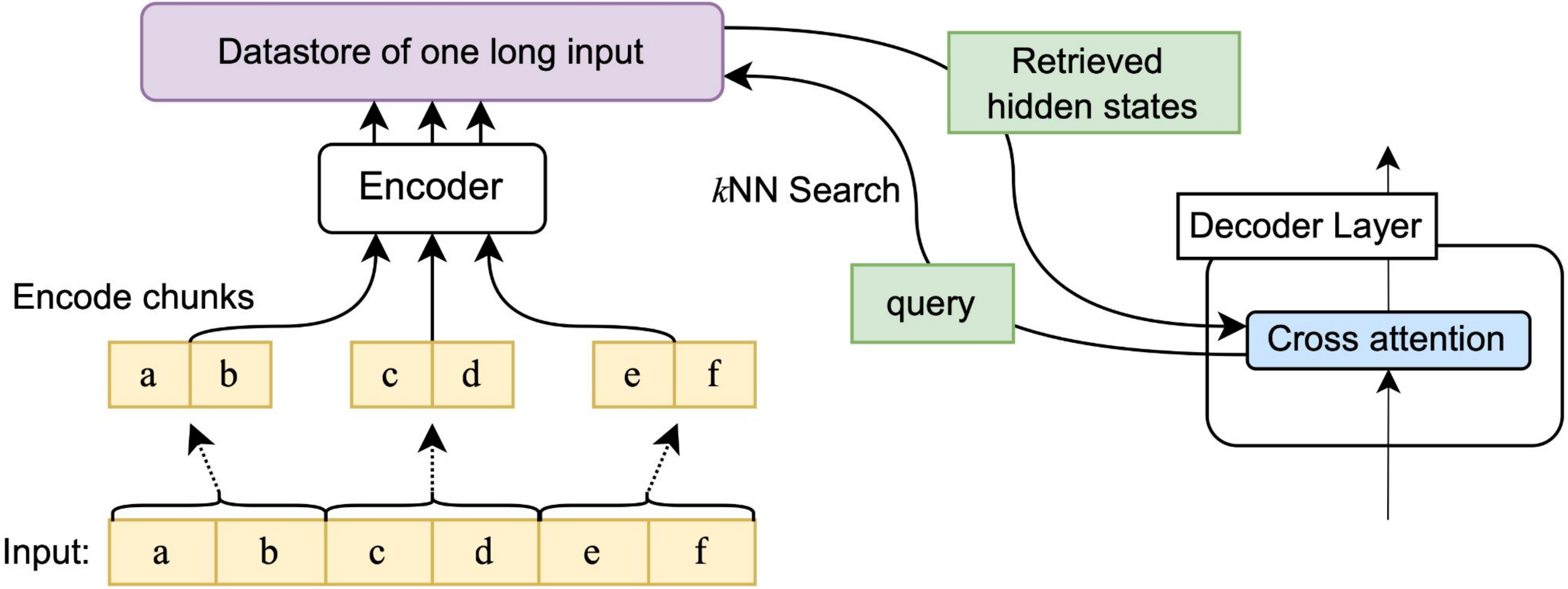
also…

the decoder positional embeddings are unaffected!

What is the datastore?



How do we choose the context window?



How do we choose the context window? cross-attention

decoder hidden state

encoder hidden state

layer specific

head specific

$$QK^T = (\boldsymbol{h}_d \boxed{W_q})\,(\boldsymbol{h}_e \boxed{W_k})^\top$$

Memorizing Transformers (Wu et al. ICLR'2022)

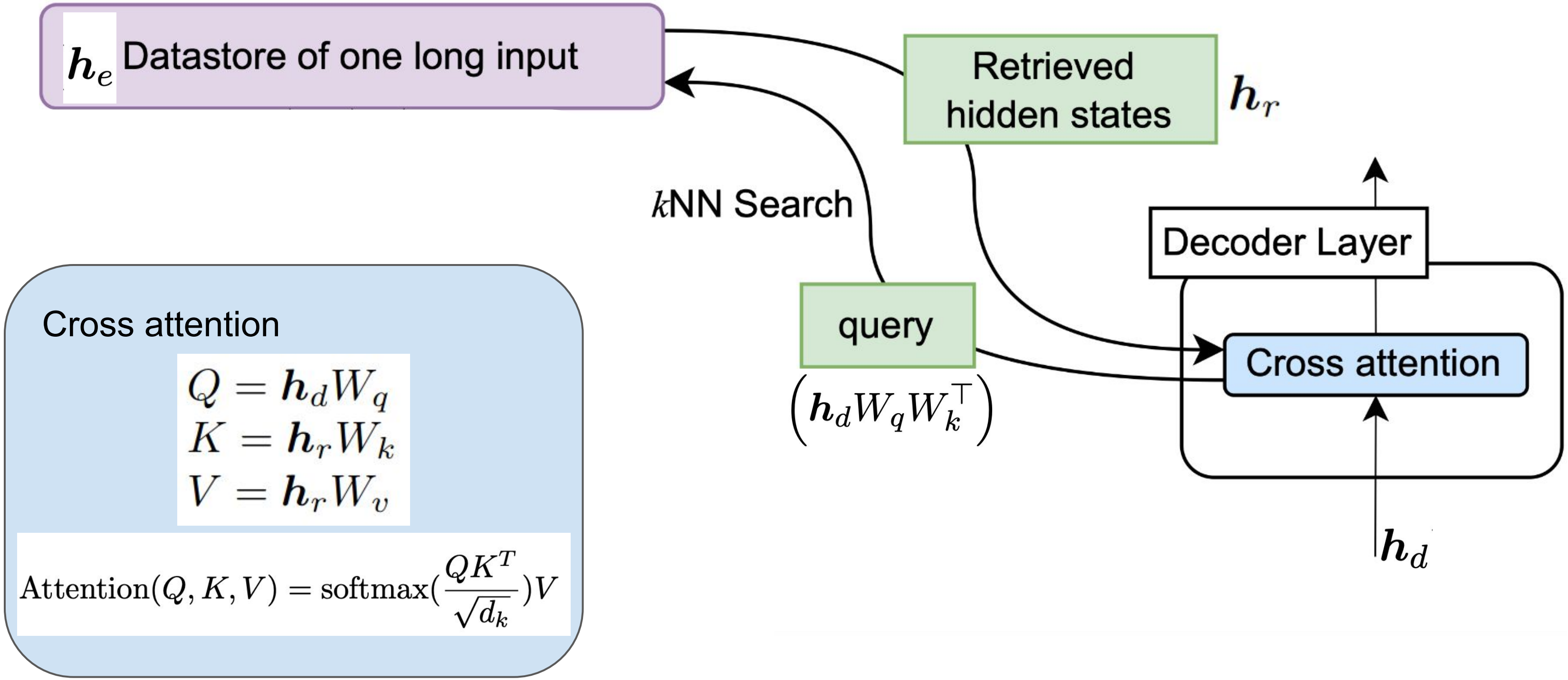
kept two datastores for each <layer,head> pair

Overall datastores: 2 X layers X heads

Project the query differently for every layer/head

We can keep a **single** datastore of the encoded hidden states

How do we choose the context window?



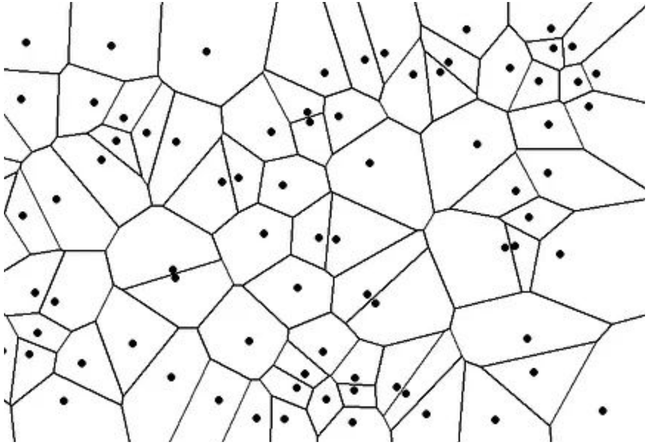
How do we do efficient search?

Datastore of one long input

faiss Public

FAISS search:

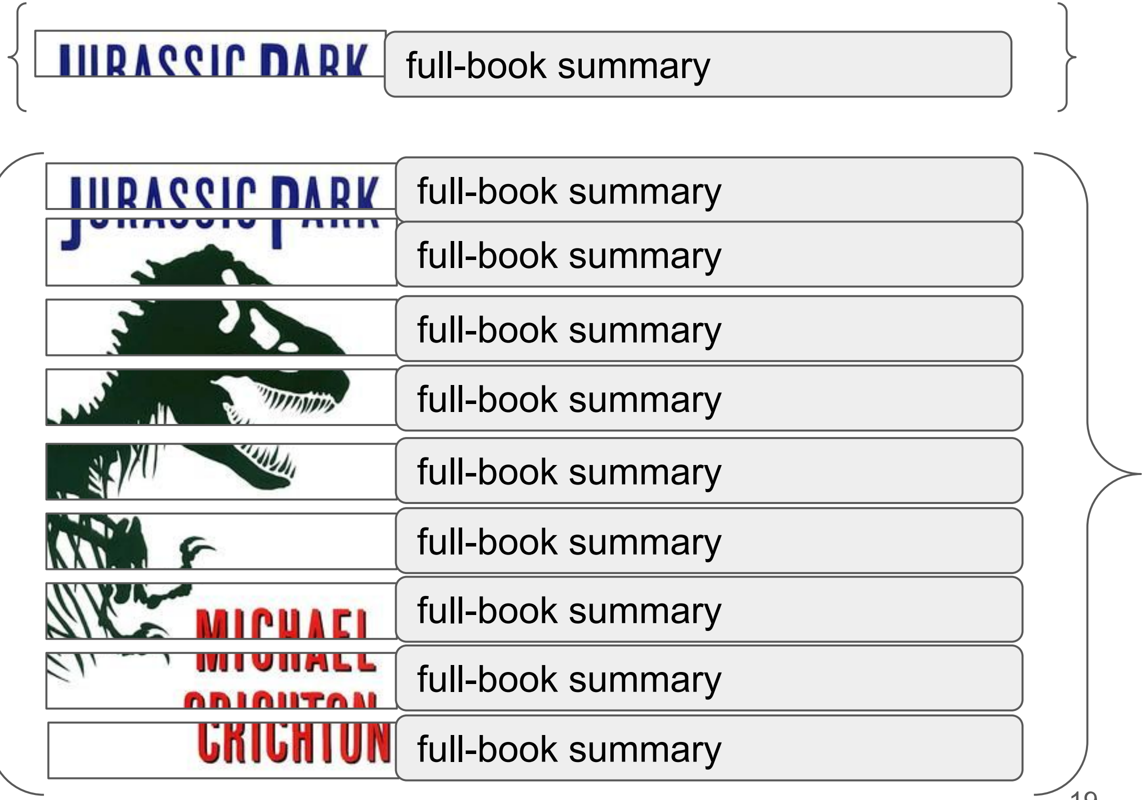
- *Supports datastores on GPU, CPU, or disk*
- *Approximate*
- *Sublinear*



Data augmentation (not Unlimiformer-specific!)

standard finetuning

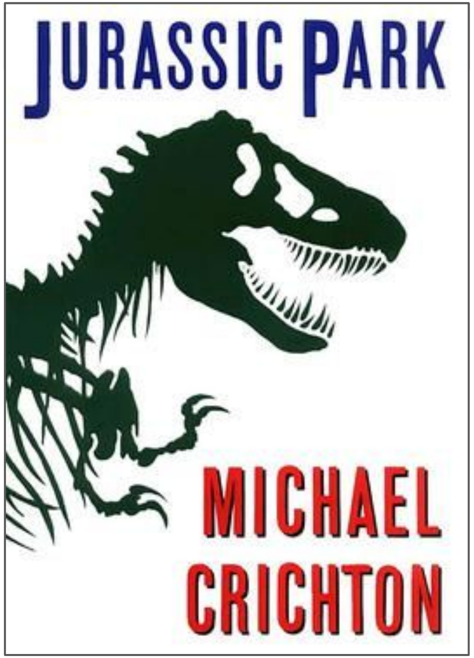
chunked finetuning



How do we train Unlimiformer?

Summarize:

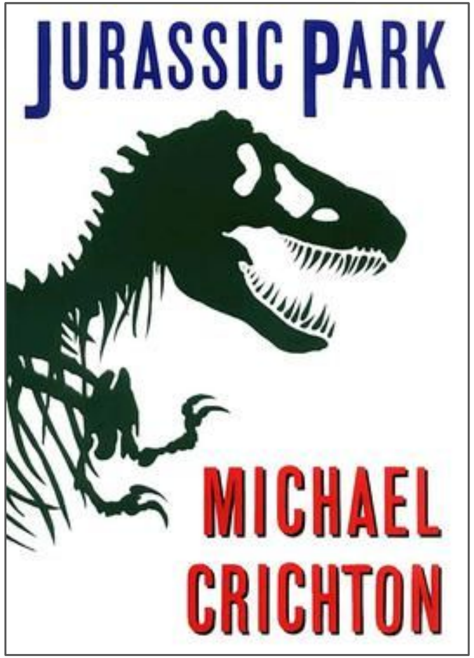
Running example:
book summarization



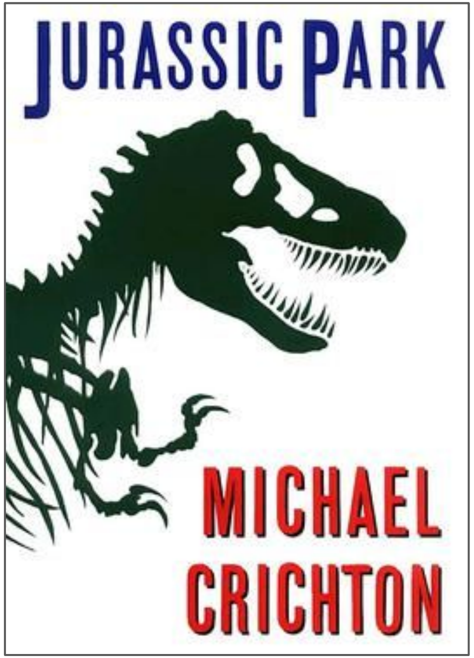
117,645
words

Normal training: truncating all inputs

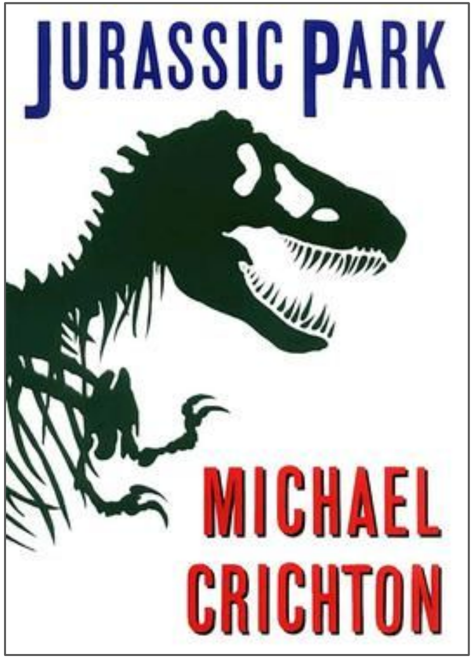
During training:



During early stopping:



During test-time:

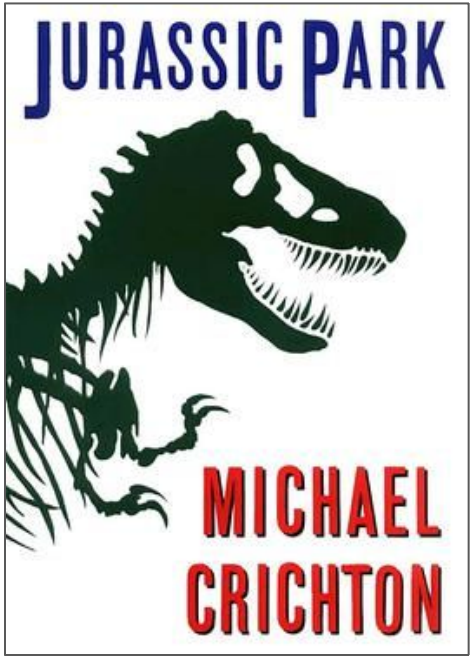


Adding Unlimiformer after training

During training:

During early stopping:

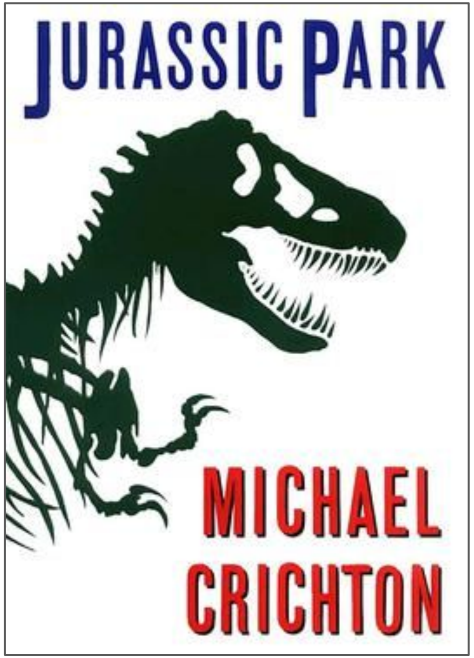
During test-time:



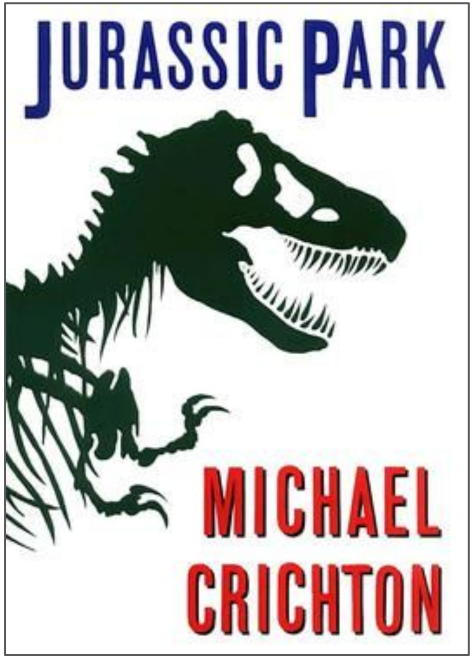
Low cost training: Unlimiformer-aware early stopping

During training:

During early stopping:



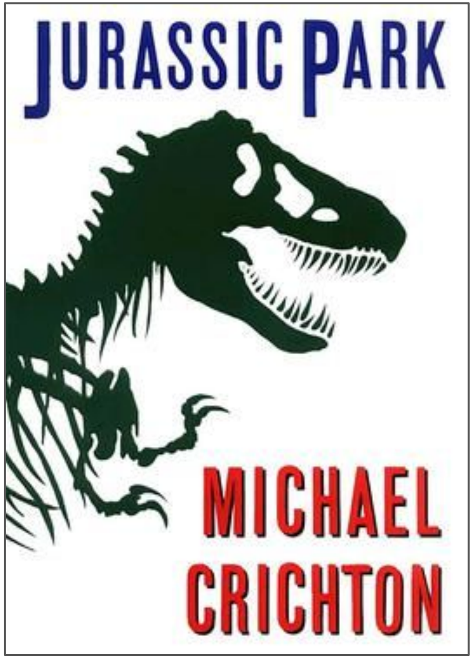
During test-time:



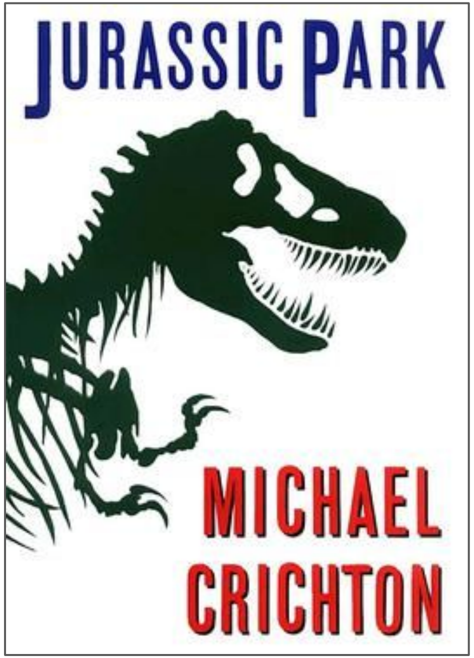
Higher cost training methods

During training:

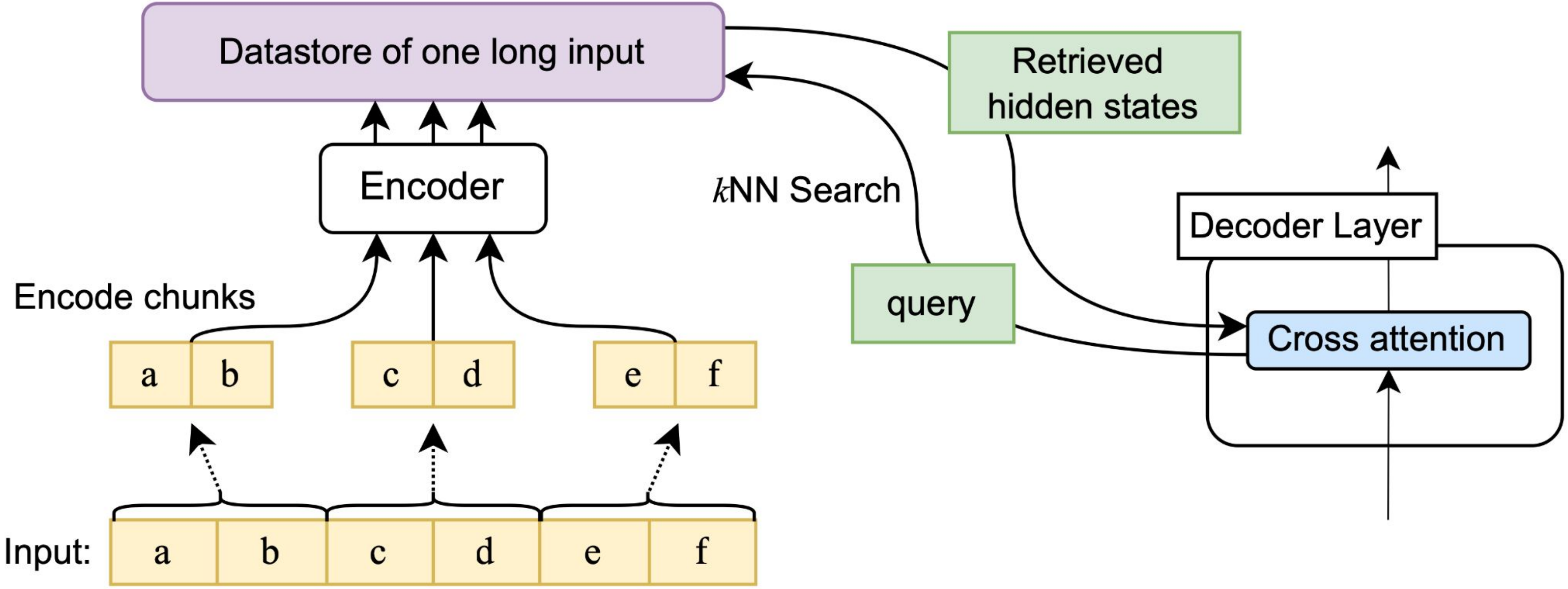
During early stopping:



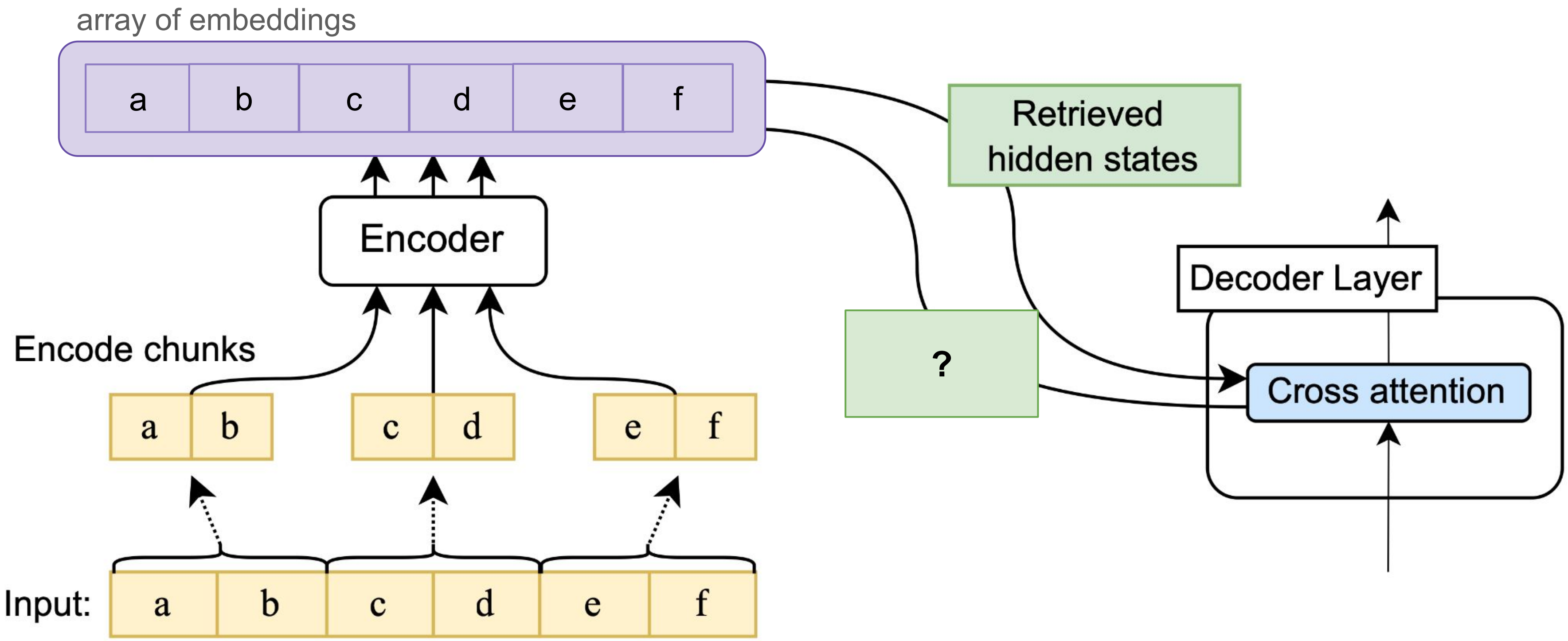
During test-time:



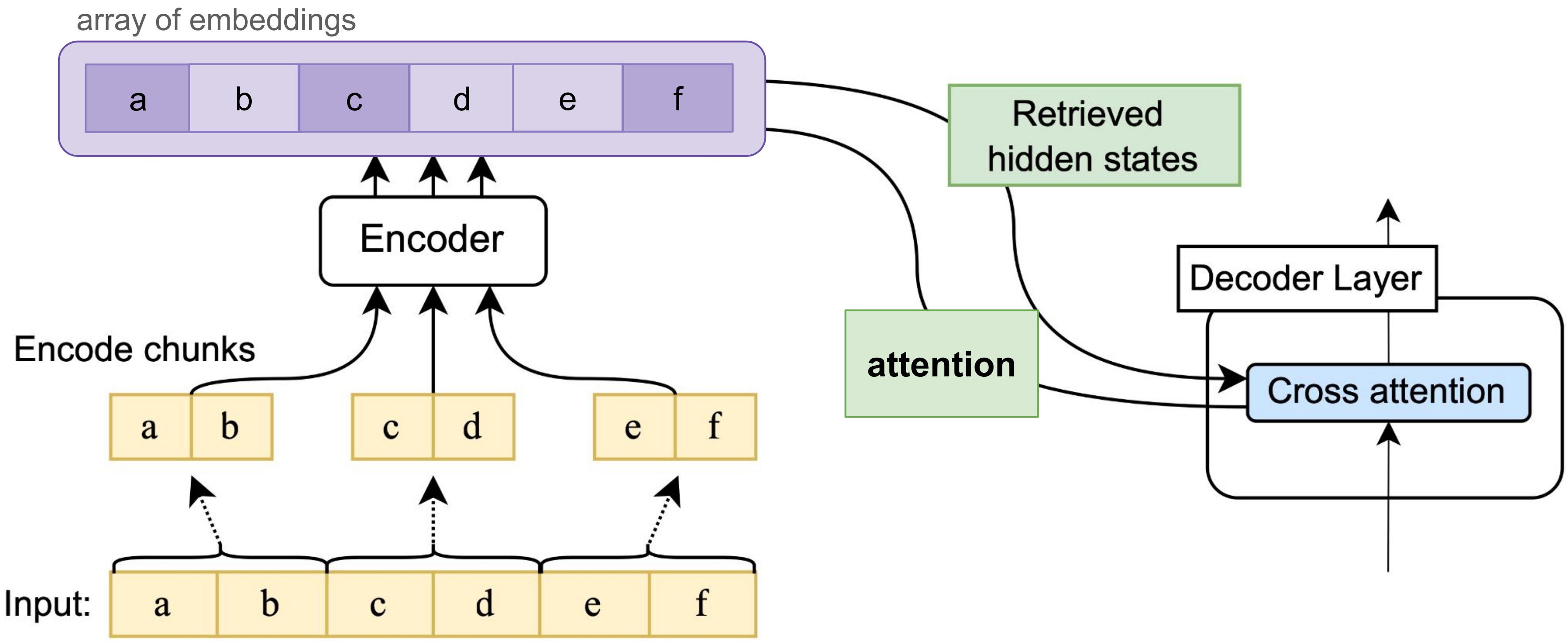
Higher cost training: which embeddings to backprop through?



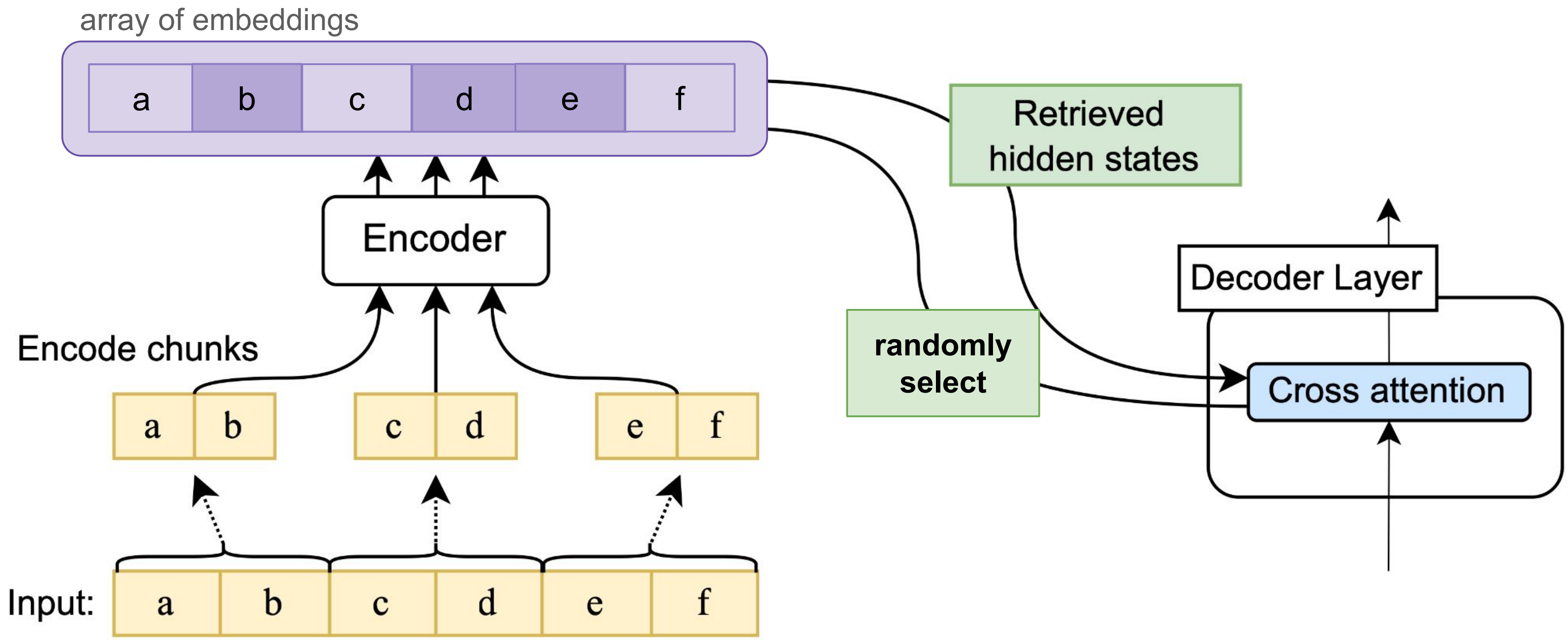
Higher cost training: which embeddings to backprop through?



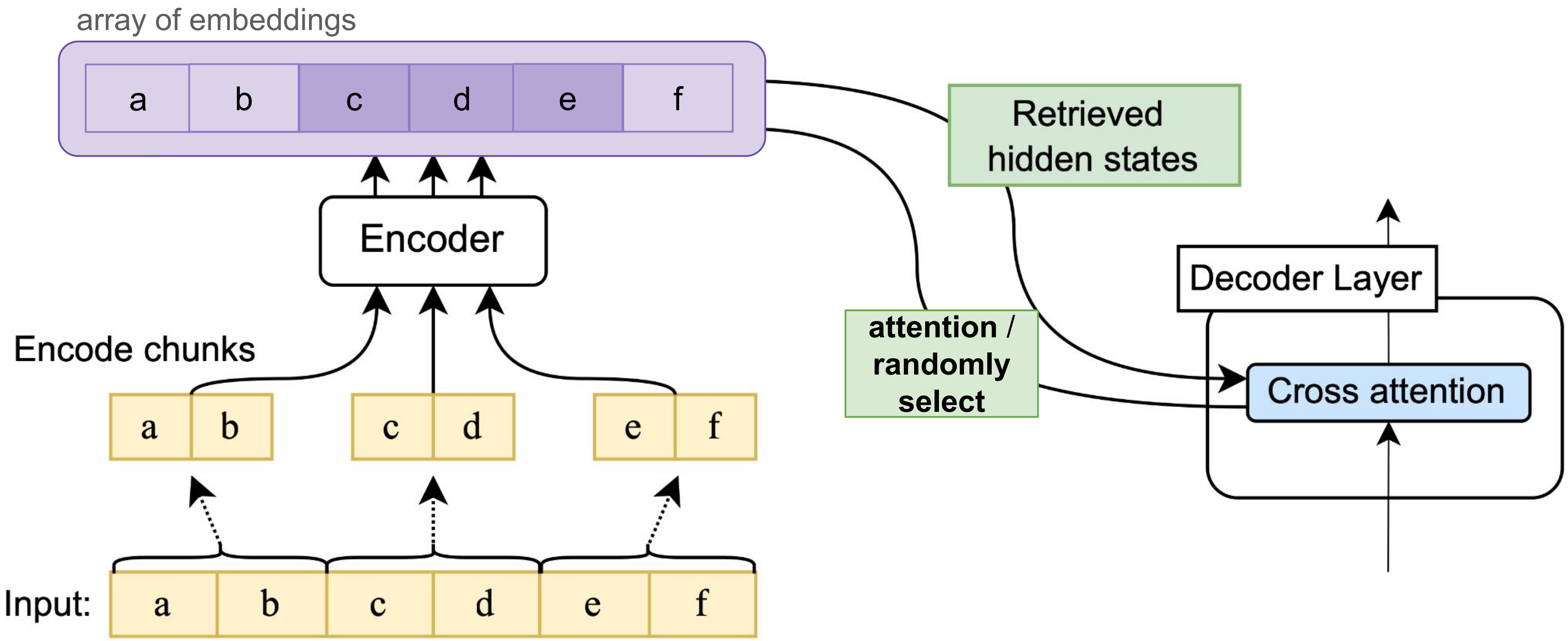
Higher cost training: retrieval training



Higher cost training: random-encoded



Higher cost training: alternating

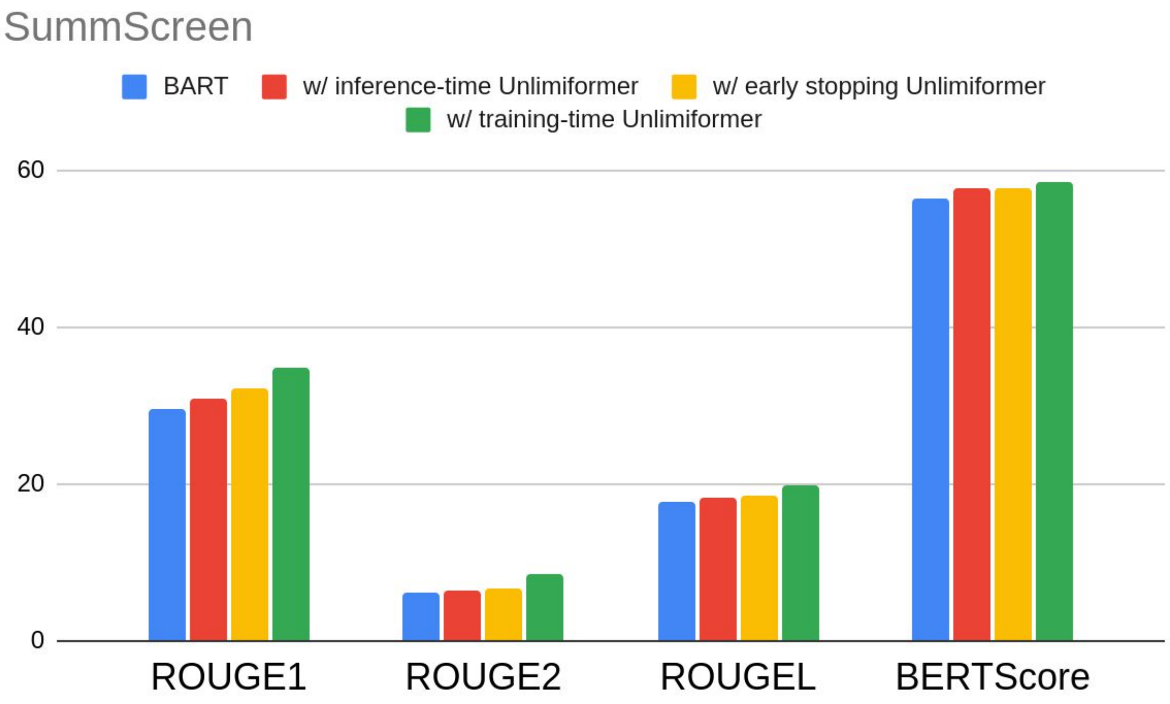


Results on SummScreen

Domain: TV screenplays

Avg input length: 8,987

Avg output length: 137

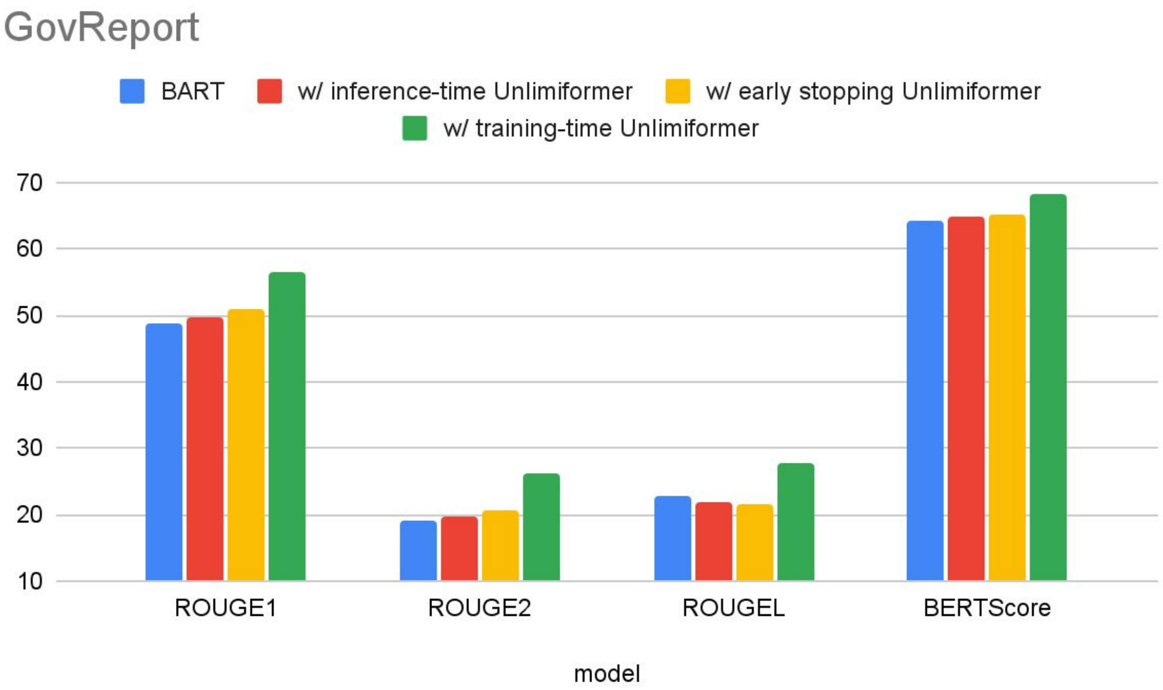


Results on GovReport

Domain: government reports

Avg input length: 9,616

Avg output length: 597

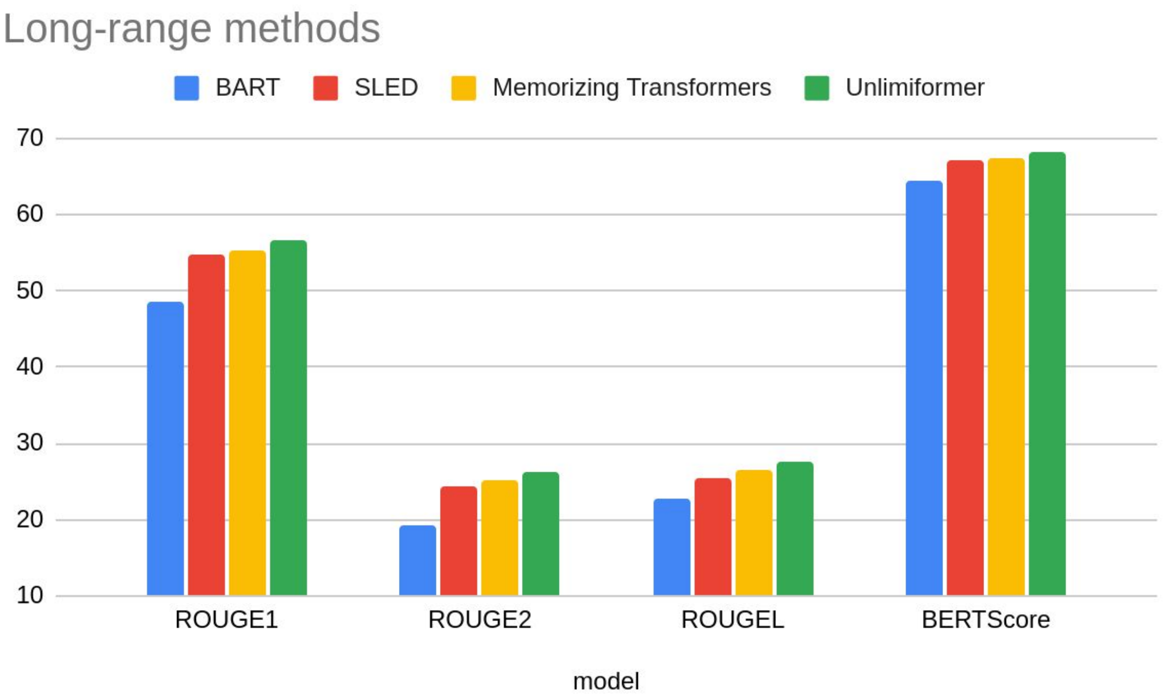


Comparison to other long-range methods [GovReport]

Domain: government reports

Avg input length: 9,616

Avg output length: 597

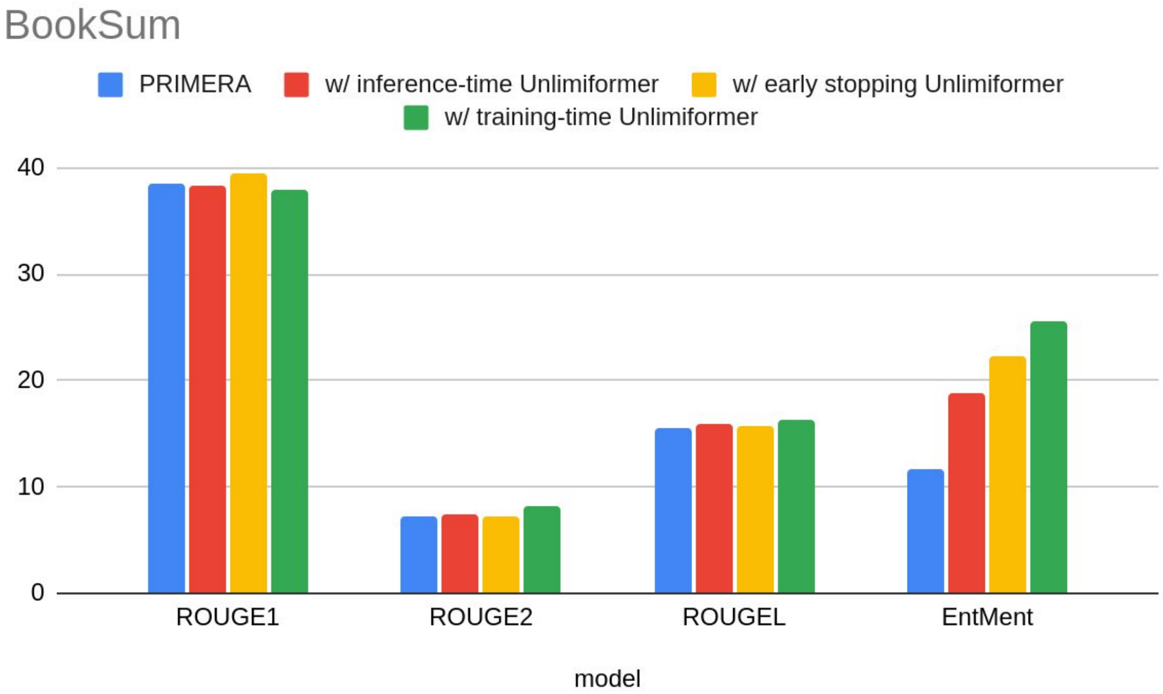


Results on BookSum

Domain: public-domain novels

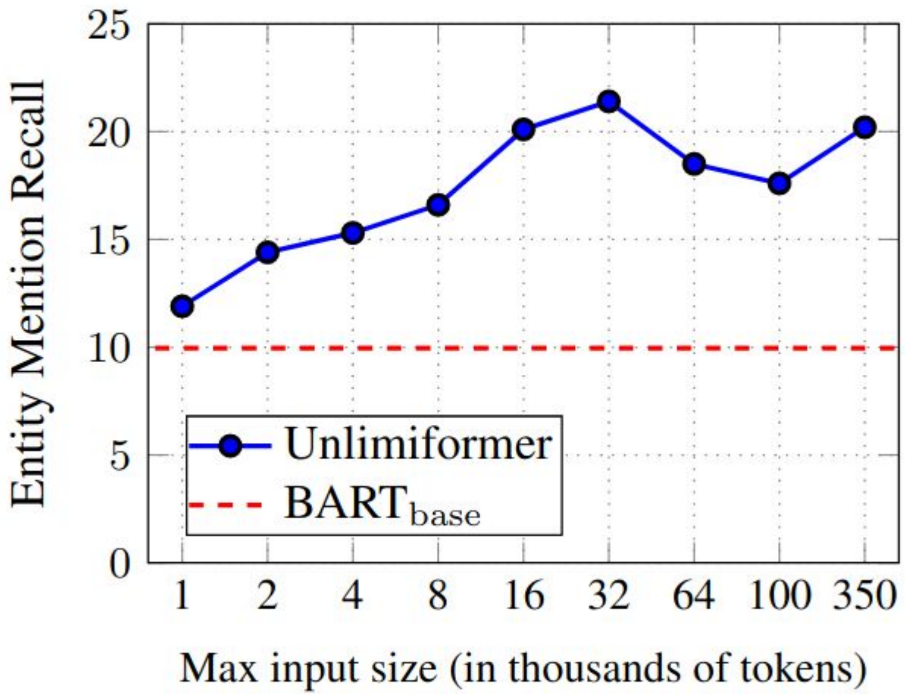
Avg input length: 143,301

Avg output length: 1,294



EntMent

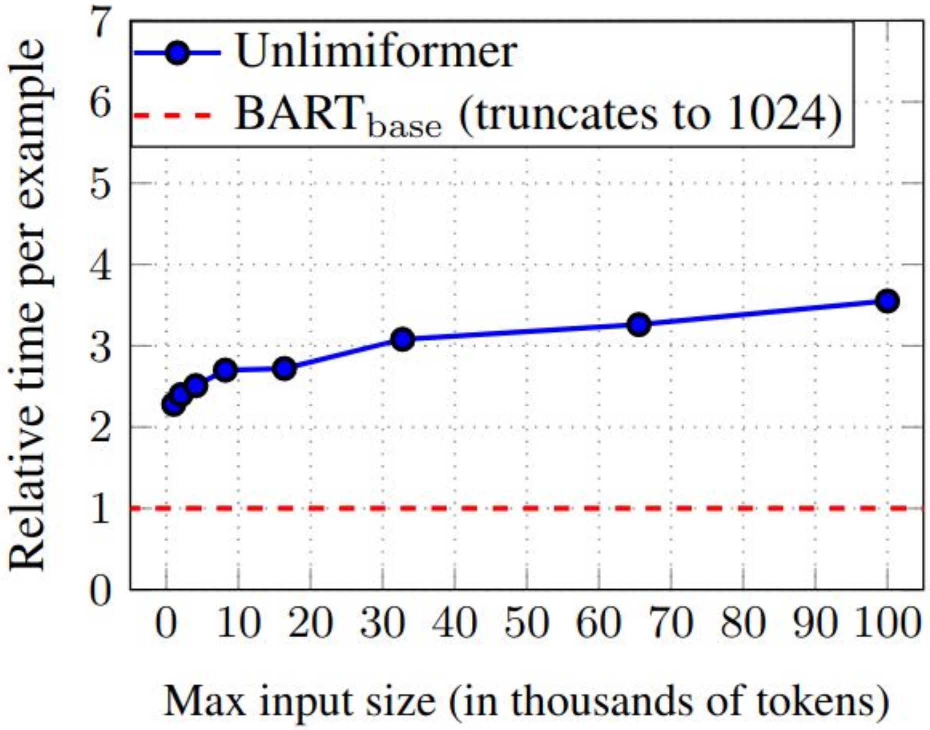
Idea: **important to include entities from the gold summary**



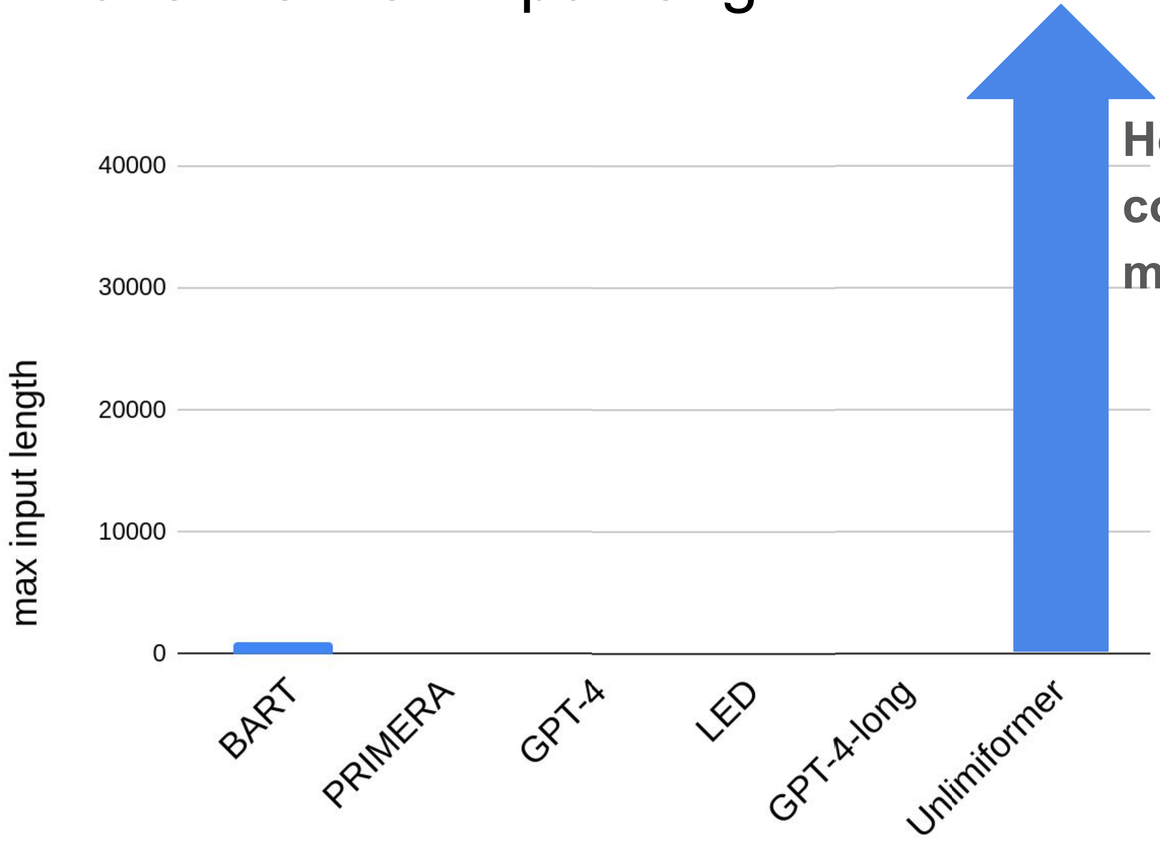
Computational cost

Additional cost from:

- Encoding additional input
- Datastore construction
- Datastore search



What’s the max input length?



How big is your computer’s memory?

What (could be) next?

- Decoder-only models with Unlimiformer: LLaMA and Falcon
- Multi-doc summarization with Unlimiformer

- Better evaluation for long text
- Generation of long text
- Training to include *all* input

questions?

Amanda Bertsch

Uri Alon

Graham Neubig

Matt Gormley