



Feeding Lots Data to Language Models

CSCI 601-471/671 (NLP: Self-Supervised Models)

<https://self-supervised.cs.jhu.edu/sp2024/>

Logistics update

- By popular demand, project proposal deadline is pushed to Monday, Apr 1.
- Addressing a question:
 - *"I want to build an application X via prompt-engineering"*
 - Depending on what you mean by "Prompt-engineering" it can be something that high-school kids can do as well.
 - So, you need to argue either of these:
 - (1) this is a non-trivial "engineering"; or (2) it's a compelling application.
- Clarification on "default" ideas:
 - There are ways to go about these without massive compute.
 - For example, for the ideas involving positional encodings we have provided links to prior implementations that you can use.
 - If they seem complex or unclear, feel free to ask me or TA.

More on Proposals

- Anonymous? :)
- No abstract? :)

Expert Knowledge Generation on Surgical Procedures

Anonymous Author(s)

Affiliation
Address
email

Abstract

1
2
3
4

The abstract paragraph should be indented $\frac{1}{2}$ inch (3 picas) on both the left- and right-hand margins. Use 10 point type, with a vertical spacing (leading) of 11 points. The word **Abstract** must be centered, bold, and in point size 12. Two line spaces precede the abstract. The abstract must be limited to one paragraph.

More on Proposals

- What is the input to your system?
- “our curated surgical procedures”: → show an example
- Dataset: have you thought about using existing datasets?

38 3 Project Plan

39 The project has two parts. First, we will implement a PDF-to-text converter to obtain datasets on
40 the textbooks of surgical procedures available on the internet. The books and documents will be
41 manually curated and fed into the converter to serve as the training data. We expect the converter to
42 be able to accurately copy the text from the PDF pages. We expect the fine-tuned model to be able to
43 provide answers regarding how to perform a prompted surgery. The produced outputs will be used in
44 the downstream tasks described in Liu and Armand [2024].

45 3.1 Experiments

- 46 • *Models*: Llama 2 fine-tuned with surgical knowledge using LoRA
- 47 • *Benchmark*: We will evaluate the model based on human expert (of which we expect about
48 10) ratings and a GPT4 evaluation agents’ rating, with the human ratings weighted higher.
49 These can be numerical score ratings or comparison ratings between procedures generated
50 by the base Llama model vs the fine-tuned model.
- 51 • *Ablation Studies*: (1) Type of knowledge fine-tuned with: fine-tune with only the surgical
52 procedure dataset, and with both the surgical procedure dataset and the engineering knowl-
53 edge dataset. (2) An adapter for parameter-efficient training: fine-tune the vanilla model,
54 fine-tune the adapter, and fine-tune the model with adapter.

55 3.2 Success Metrics

56 We will ask human experts to rate the answers produced by the model. The experts will grade the
57 answers using: methodological accuracy (do all the instructions described recommend the correct
58 methods?), comprehensiveness (will the steps given encompass the entire procedure?), clarity (are
59 the instructions clear and the steps well-defined?), safety management, patient-centeredness, up-to-
60 date information, and practicality. The fine-tuned model is successful if the answers can achieve
61 satisfactory scores.

62 3.3 Datasets

63 We will use our curated surgical procedure textbooks as the dataset. The size of the dataset is to be
64 determined by the availability of the books and their relevance. Additional materials can be academic
65 articles and Wikipedia pages.

This is NOT Just an Academic Exercise

- Imagine you're in a tech company.
- You need to write memos, reports and proposals for before or after any project.
- Jeff Bezos: "There is no way to write a six-page narratively structured memo and not have clear thinking."

<https://medium.com/@nathan.baugh/welcome-to-the-jungle-38fdde285b6f>

<https://writingcooperative.com/the-anatomy-of-an-amazon-6-pager-fc79f31a41c9>



Back to our topic



Feeding Lots of Things to LM

- Books, scientific articles, government reports, videos, your daily experience, etc. they all are much longer than 2k tokens!!
- How do you enable language models process massive amounts of data?
- One approach: just scale up your model—train it on a much longer context window size.
 - The bottleneck, memory usage and number of operations in Self-Attention increases quadratically.

Plan: Feeding Lots of Things to LM

1. Understanding Positional encodings
2. Length generalization of Transformer LMs
3. Retrieval-augmented generation

Chapter goal: Get familiar with techniques that are related and important for length generalization of LMs.

Revisiting Encoding Positional Information

Recap: Self-Attention

- Given input \mathbf{x} :

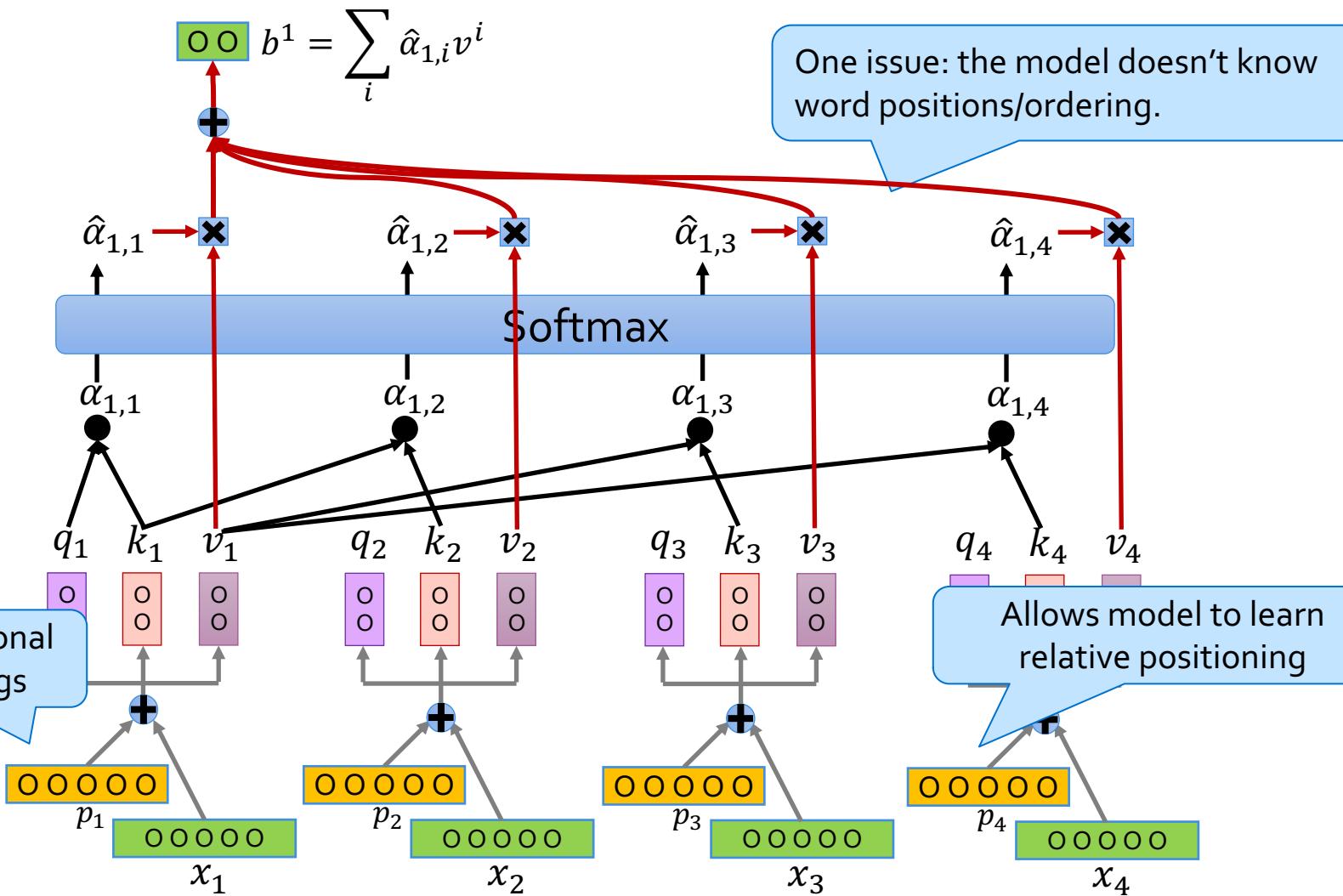
$$Q = \mathbf{W}^q \mathbf{x}$$

$$K = \mathbf{W}^k \mathbf{x}$$

$$V = \mathbf{W}^v \mathbf{x}$$

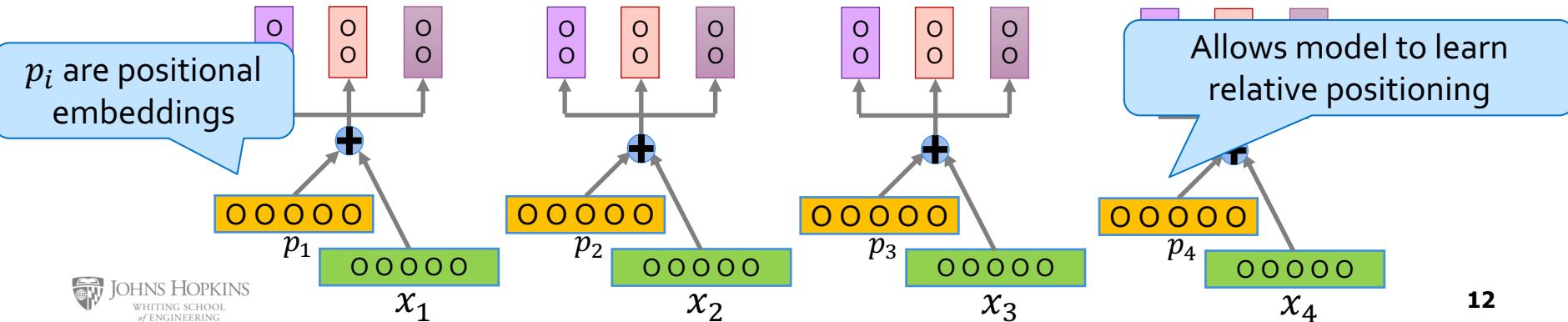
$$\text{Attention}(\mathbf{x}) = \text{softmax}\left(\frac{QK^T}{\alpha}\right)V$$

- Remember that, without positional encoding the input is just bag of words for self-attention.



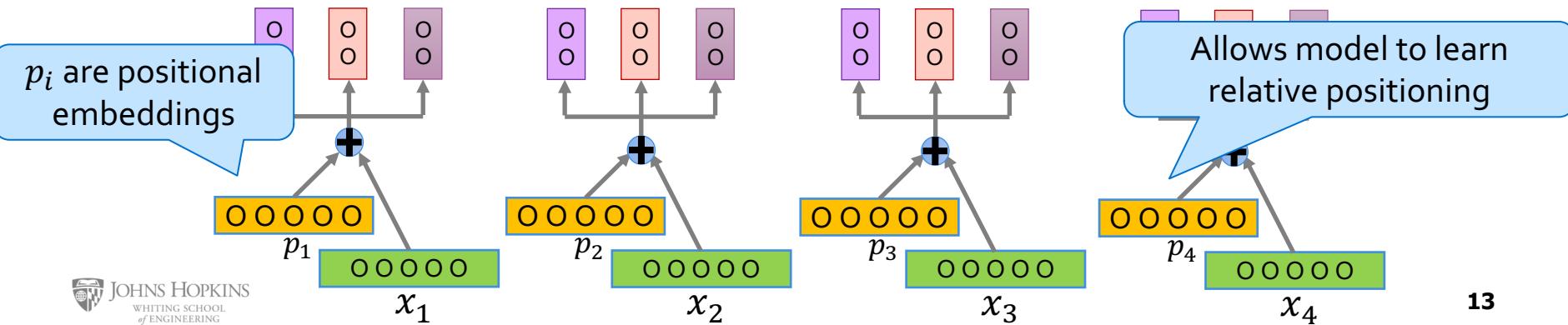
Positional Embeddings: The Flavors

- There are many different choices of positional encodings:
 - Some are learned (BERT, Devlin et al. 2018) and some are **fixed**.
 - Learn the position-specific embeddings through Backprop.
 - Some encode **absolute** position. Will see more on this.
 - Some encode **relative** position. Will see more on this.



Positional Embeddings: The Flavors

- Why “add”? Why not, say, “concatenate and then project”?
 - “concatenate and then project” would be a more general approach with more trainable parameters.
 - In practice, “sum” works fine that
 - The intuition here is that “summing” forms point clouds of word embedding information around position embeddings unique to each position.



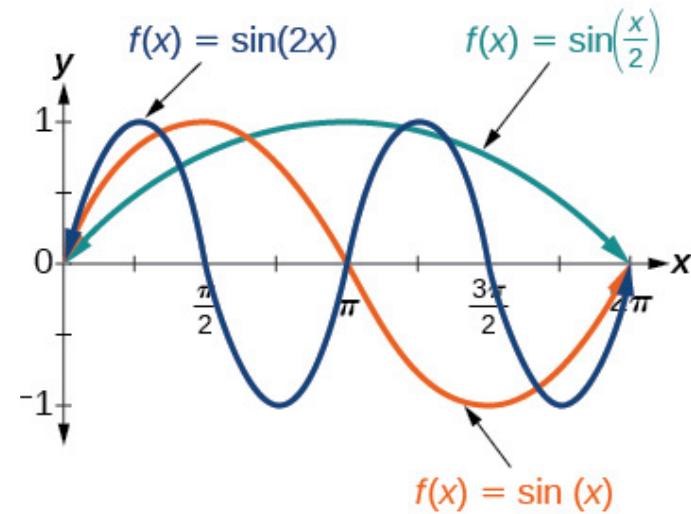
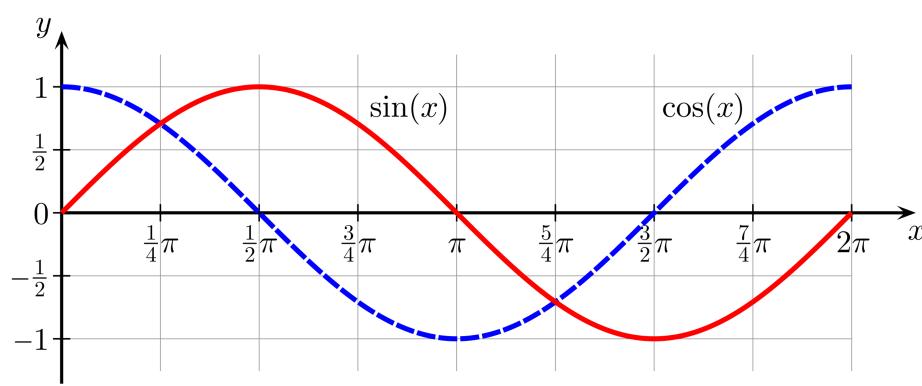
Absolute Positional Embeddings

- The idea is to create vectors that uniquely encoder each position.
- For example, consider vectors of binary values.
 - Example below shows 4-dimensional position encodings for 16 positions.

0 :	0	0	0	0	8 :	1	0	0	0
1 :	0	0	0	1	9 :	1	0	0	1
2 :	0	0	1	0	10 :	1	0	1	0
3 :	0	0	1	1	11 :	1	0	1	1
4 :	0	1	0	0	12 :	1	1	0	0
5 :	0	1	0	1	13 :	1	1	0	1
6 :	0	1	1	0	14 :	1	1	1	0
7 :	0	1	1	1	15 :	1	1	1	1

The issue with binary encoding is that the positional information is localized around a few bits.

Math Recap: Sine and Cosine Functions



Absolute Positional Embeddings

- Let t be a desired position. Then the i -th element of the positional vector is:

$$\vec{p}_t^{(i)} = f(t)^{(i)} := \begin{cases} \sin(\omega_k \cdot t), & \text{if } i = 2k \\ \cos(\omega_k \cdot t), & \text{if } i = 2k + 1 \end{cases} \quad \omega_k = \frac{1}{10000^{2k/d}}$$

- Here d is the maximum dimension.
- This provides unique vectors for each position.

Quiz

- Let t be a desired position:

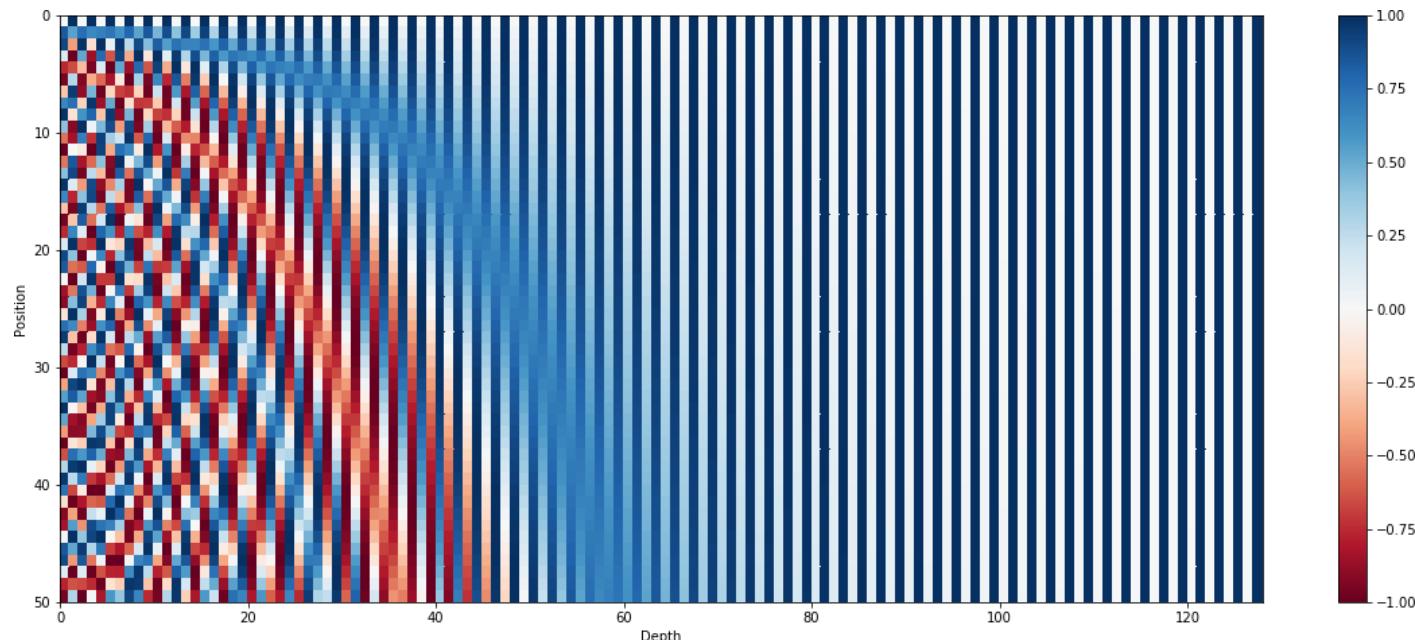
$$\vec{p}_t^{(i)} = f(t)^{(i)} := \begin{cases} \sin(\omega_k \cdot t), & \text{if } i = 2k \\ \cos(\omega_k \cdot t), & \text{if } i = 2k + 1 \end{cases}$$

$$\omega_k = \frac{1}{10000^{2k/d}}$$

- Q:** Are the frequencies increasing with dimension i ?
- Answer:** The frequencies are decreasing along the vector dimension.

Visualizing Absolute Positional Embeddings

- Here positions range from 0-50, for an embedding dimension of 130.



Limits of Absolute Positional Encoding

- We can have fixed positional embeddings for each index training position (e.g., 1, 2, 3, ... 1000).
 - What happens if we get a sequence with 5000 words at test time?
- We want something that can generalize to arbitrary sequence lengths.
 - Approach: encoding the relative positions, for example based on the distance of the tokens in a local window to the current token.

A Unified Perspective on Positional Encoding

- We are input sequence x_0, x_1, \dots and
 - Then the unnormalized attention value between position i , and j is:

$$QK_{ij} = (\mathbf{W}_q \mathbf{x}_i)^T (\mathbf{W}_k \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{W}_q^T \mathbf{W}_k \mathbf{x}_j$$

- Now also assume that positional embeddings are added to \mathbf{x}_i , i.e., they're $\mathbf{x}_i + \mathbf{p}_i$

$$QK_{ij} = (\mathbf{W}_q [\mathbf{x}_i + \mathbf{p}_i])^T (\mathbf{W}_k [\mathbf{x}_j + \mathbf{p}_j]) = \underbrace{\mathbf{x}_i^T \mathbf{W}_q^T \mathbf{W}_k \mathbf{x}_j}_{\text{The original attention term: how much attention should we pay to word } x_j \text{ given word } x_i} + \underbrace{\mathbf{x}_i^T \mathbf{W}_q^T \mathbf{W}_k \mathbf{p}_j}_{\text{How much attention should we pay to word } x \text{ given the position } p} + \underbrace{\mathbf{p}_i^T \mathbf{W}_q^T \mathbf{W}_k \mathbf{x}_j}_{\text{How much attention should position } p_i \text{ attend to position } p_j} + \underbrace{\mathbf{p}_i^T \mathbf{W}_q^T \mathbf{W}_k \mathbf{p}_j}_{}$$

The original attention term:
how much attention should we
pay to word x_j given word x_i

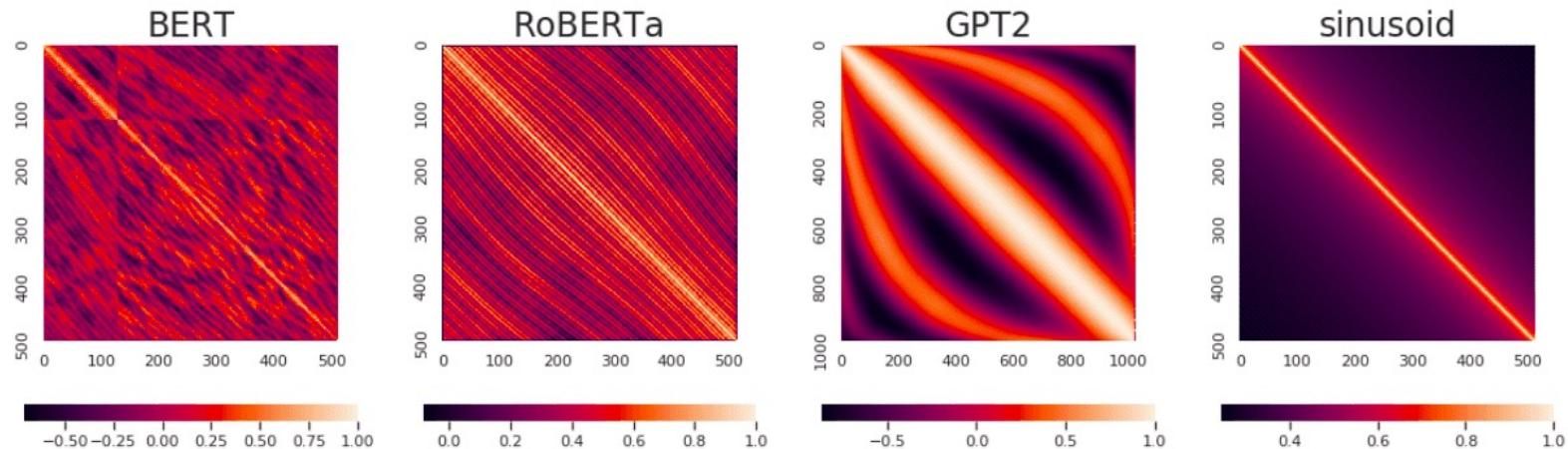
How much attention
should we pay to word x
given the position p

How much attention
should position p_i should
attend to position p_j

A Unified Perspective on Positional Encoding

- You can rewrite the statement from the previous slide in the following form:

$$QK_{ij} = (W_q[x_i + p_i])^T (W_k[x_j + p_j]) = \mathbf{x}_i^T W_q^T W_k \mathbf{x}_j + P_{ij}$$



Relative Positional Encoding

- You can rewrite the statement from the previous slide in the following form:

$$QK_{ij} = (W_q[x_i + p_i])^T (W_k[x_j + p_j]) = x_i^T W_q^T W_k x_j + P_{ij}$$

- Note, the values of P_{ij} encode the relative of i and j .
- How should we construct P_{ij} ?

How much attention
should position i should
attend to position j

Relative Positional Encoding

- There have been various choices:
 - T5 models simplify this into learnable relative embeddings \mathbf{P}_{ij} such that:
$$QK_{ij} = \mathbf{x}_i^T \mathbf{W}_q^T \mathbf{W}_k \mathbf{x}_j + \mathbf{P}_{ij}$$
 - DeBERTa learns relative positional embeddings $\tilde{\mathbf{p}}_{i-j}$ such that:
$$QK_{ij} = \mathbf{x}_i^T \mathbf{W}_q^T \mathbf{W}_k \mathbf{x}_j + \mathbf{x}_i^T \mathbf{W}_q^T \mathbf{W}_k \tilde{\mathbf{p}}_{i-j} + \tilde{\mathbf{p}}_{i-j}^T \mathbf{W}_q^T \mathbf{W}_k \mathbf{x}_j$$
 - Transformer-XL learns relative positional embeddings $\tilde{\mathbf{p}}_{i-j}$ and trainable vectors \mathbf{u}, \mathbf{v} s.t.:
$$QK_{ij} = \mathbf{x}_i^T \mathbf{W}_q^T \mathbf{W}_k \mathbf{x}_j + \mathbf{x}_i^T \mathbf{W}_q^T \mathbf{W}_k \tilde{\mathbf{p}}_{i-j} + \mathbf{u}^T \mathbf{W}_q^T \mathbf{W}_k \mathbf{x}_j + \mathbf{v}^T \mathbf{W}_q^T \mathbf{W}_k \tilde{\mathbf{p}}_{i-j}$$
 - ALiBi learns a scalar m such that:
$$QK_{ij} = \mathbf{x}_i^T \mathbf{W}_q^T \mathbf{W}_k \mathbf{x}_j - m |i - j|$$

Attention with Linear Biases (ALiBi)

- Add a constant value to the attention logits
- Apply a head specific scalar m

$$\begin{matrix} q_1 \cdot k_1 \\ q_2 \cdot k_1 & q_2 \cdot k_2 \\ q_3 \cdot k_1 & q_3 \cdot k_2 & q_3 \cdot k_3 \\ q_4 \cdot k_1 & q_4 \cdot k_2 & q_4 \cdot k_3 & q_4 \cdot k_4 \\ q_5 \cdot k_1 & q_5 \cdot k_2 & q_5 \cdot k_3 & q_5 \cdot k_4 & q_5 \cdot k_5 \end{matrix} + \begin{matrix} 0 \\ -1 & 0 \\ -2 & -1 & 0 \\ -3 & -2 & -1 & 0 \\ -4 & -3 & -2 & -1 & 0 \end{matrix} \cdot m$$

Length Extrapolation

- Note there are two interpretations of “length extrapolation”:

- Modest interpretation: LM should not crash when given long context, i.e., PPL should **not get worse** with more context.
- Strict interpretation: LM should exploit the added information, i.e., PPL should **get smaller** with more context.

This result shows that ALiBi is length generalizable according to the modest definition.
It does **not** tell us whether it is length generalization according to the strict interpretation.

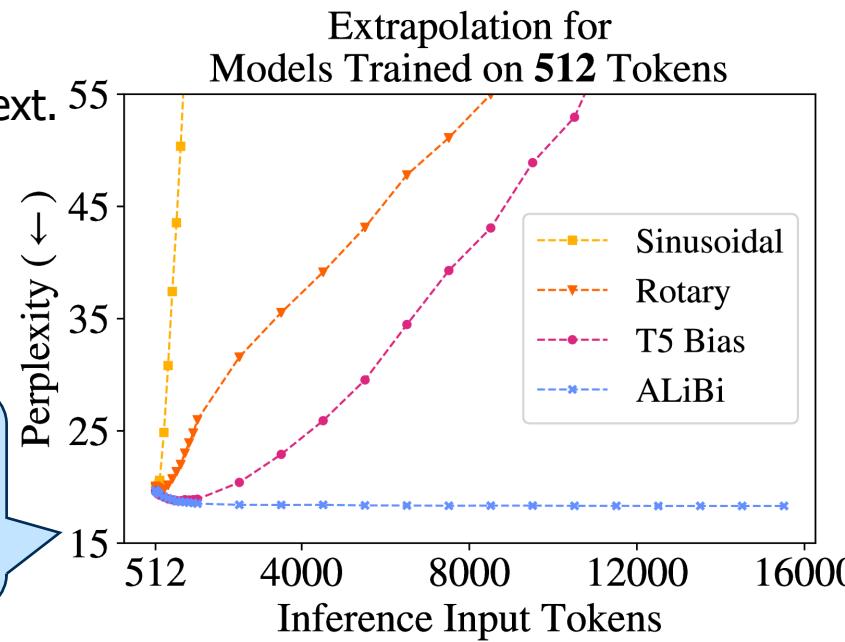
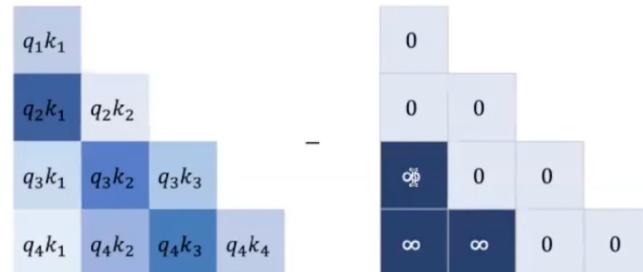
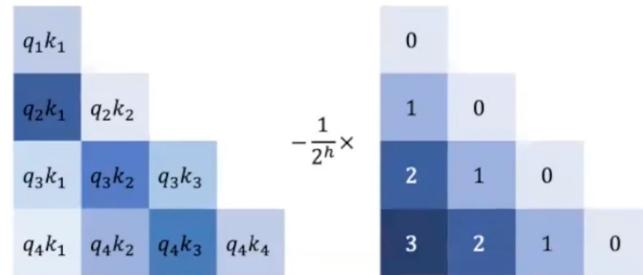


fig from Ofir Press (2023)

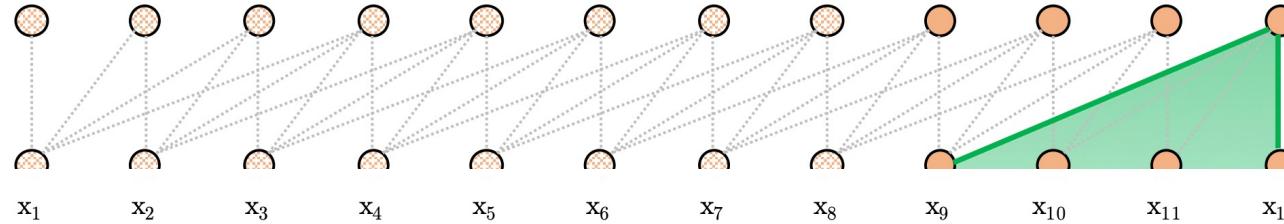
ALiBi vs. Fixed Window

- Hypothesis: ALiBi might be acting like a fixed-window.
- ALiBi scales the attention values based on their distance, effectively ignoring long-range context and acting like a moving window.
- This sounds like a bad idea, since we want language models to attend to long-range dependencies.
- But the good news is that, stacking self-attention layers mitigates such narrow attention window.



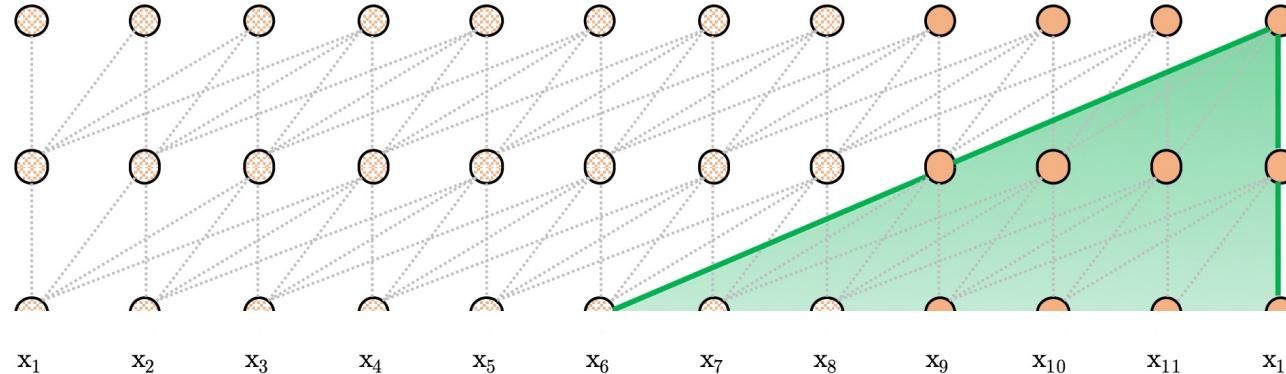
Fixed Window Attending to Long-Context

- Imagine you have a Self-Attention limited to a window of size 4.



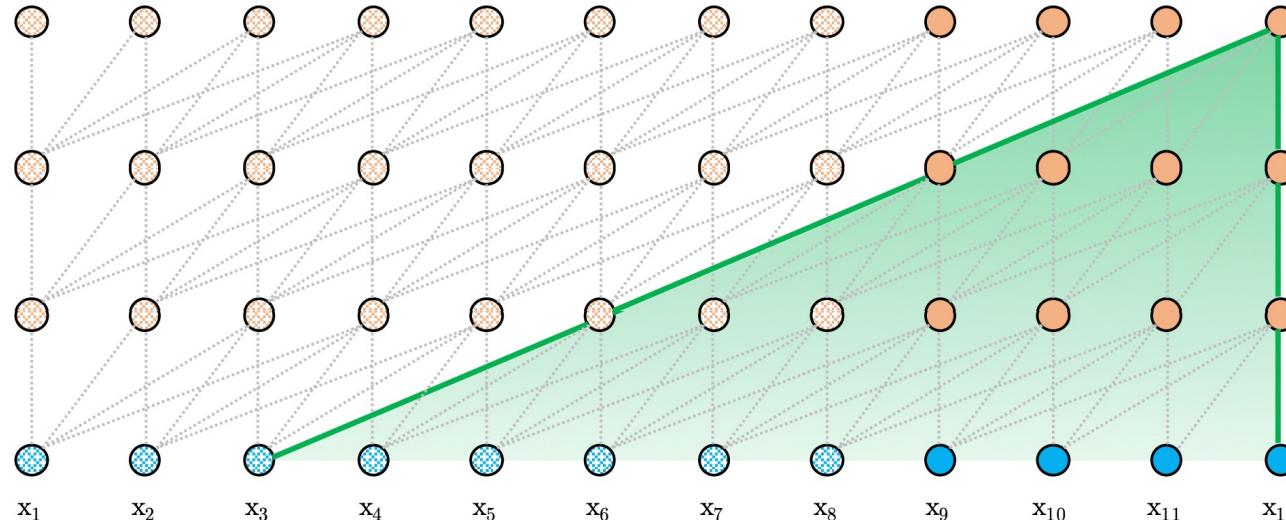
Fixed Window Attending to Long-Context

- Imagine you have a Self-Attention limited to a window of size 4.



Fixed Window Attending to Long-Context

- Imagine you have a Self-Attention limited to a window of size 4.

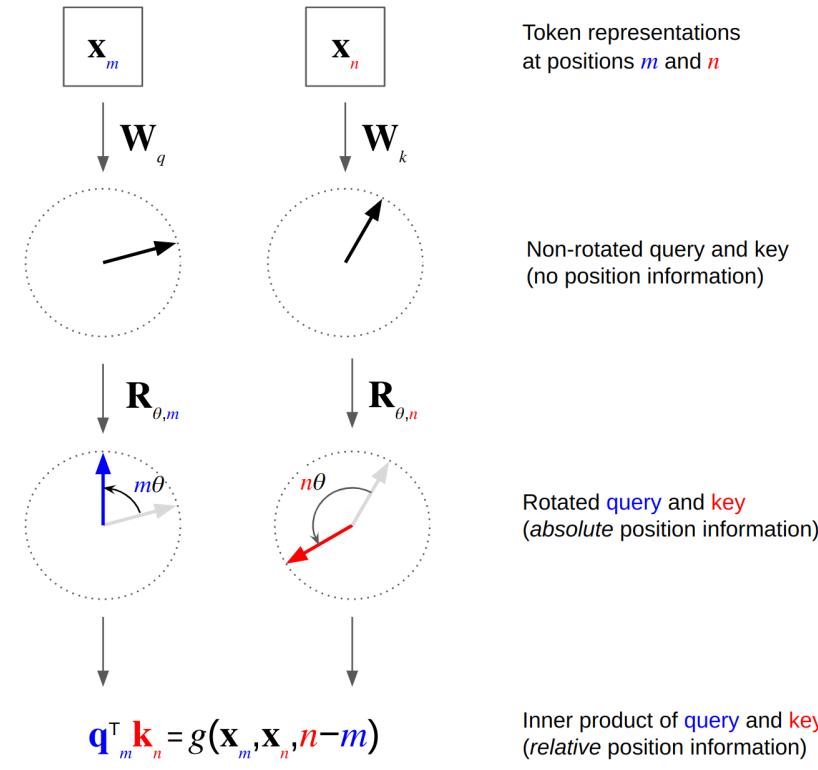


Rotary Positional Encoding (RoPE)

- Drop the additive positional encoding and make it multiplicative.

$$\begin{aligned} qk_{mn} &= (R_{\theta,m}W_qx_m)^T(R_{\theta,n}W_kx_n) \\ &= x_m^TW_q^TR_{\theta,m}^TR_{\theta,n}W_kx_n \end{aligned}$$

- θ : the size of rotation
- $R_{\theta,m}$: rotation matrix, rotates a vector it gets multiplied to proportional to θ and the position index m .
- Intuition: **nearby** words have **smaller relative rotation**.

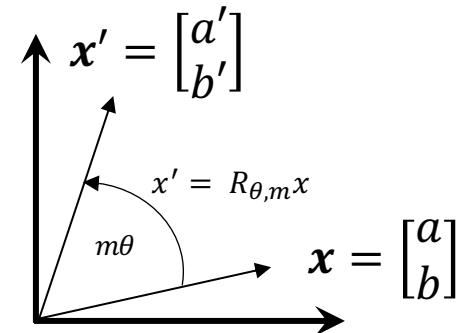


Thinking About Rotation Matrix

- In 2D, a rotation matrix can be defined in the following form:

$$R_{\theta,m} = \begin{pmatrix} \cos m\theta & -\sin m\theta \\ \sin m\theta & \cos m\theta \end{pmatrix}$$

- The rotation increases with increasing θ and m .



Thinking About Rotation Matrix

- In practice, we are rotating d dimensional embedding matrices.
- Idea: rotate different dimensions with different angles:
 - $\Theta = \{\theta_0, \theta_1, \theta_2, \theta_3, \dots, \theta_{d/2}\}$

$$\mathbf{R}_{\Theta,t}^d = \begin{pmatrix} \cos t\theta_1 & -\sin t\theta_1 & 0 & 0 & \cdots & 0 & 0 \\ \sin t\theta_1 & \cos t\theta_1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cos t\theta_2 & -\sin t\theta_2 & \cdots & 0 & 0 \\ 0 & 0 & \sin t\theta_2 & \cos t\theta_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \cos t\theta_{d/2} & -\sin t\theta_{d/2} \\ 0 & 0 & 0 & 0 & \cdots & \sin t\theta_{d/2} & \cos t\theta_{d/2} \end{pmatrix}$$

RoPE in its General Form

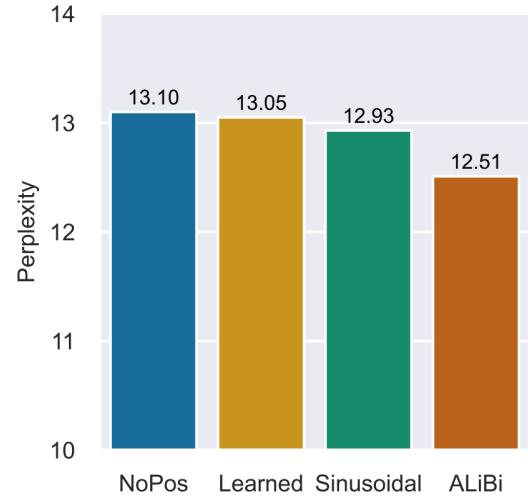
$$qk_{mn} = (R_{\Theta,m}^d W_q x_m)^T (R_{\Theta,m}^d W_k x_n),$$

- where $R_{\Theta,m}^d$ is a d -dimensional rotation matrix.
- Since $R_{\Theta,m}^d$ is a sparse matrix, its multiplication is implemented via dense operations:

$$\mathbf{R}_{\Theta,t}^d \mathbf{u} = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ \vdots \\ u_{d-1} \\ u_d \end{pmatrix} \otimes \begin{pmatrix} \cos m\theta_1 \\ \cos t\theta_1 \\ \cos t\theta_2 \\ \cos t\theta_2 \\ \vdots \\ \cos t\theta_{d/2} \\ \cos t\theta_{d/2} \end{pmatrix} + \begin{pmatrix} -u_2 \\ u_1 \\ -u_4 \\ u_3 \\ \vdots \\ -u_d \\ u_{d-1} \end{pmatrix} \otimes \begin{pmatrix} \sin t\theta_1 \\ \sin t\theta_1 \\ \sin t\theta_2 \\ \sin t\theta_2 \\ \vdots \\ \sin t\theta_{d/2} \\ \sin t\theta_{d/2} \end{pmatrix}$$

Positional Encodings May Not be Necessary!!

- Some results suggest that, you might be able to build low-PPL LMs even if you drop the positional embeddings.
- Notice that these results are for causal (decoder) Transformer architectures.
- Now the questions are:
 - How can Transformers do language modeling without positional embeddings?
 - Does low PPL for language modeling entail being good for any language ability? (e.g., ICL)



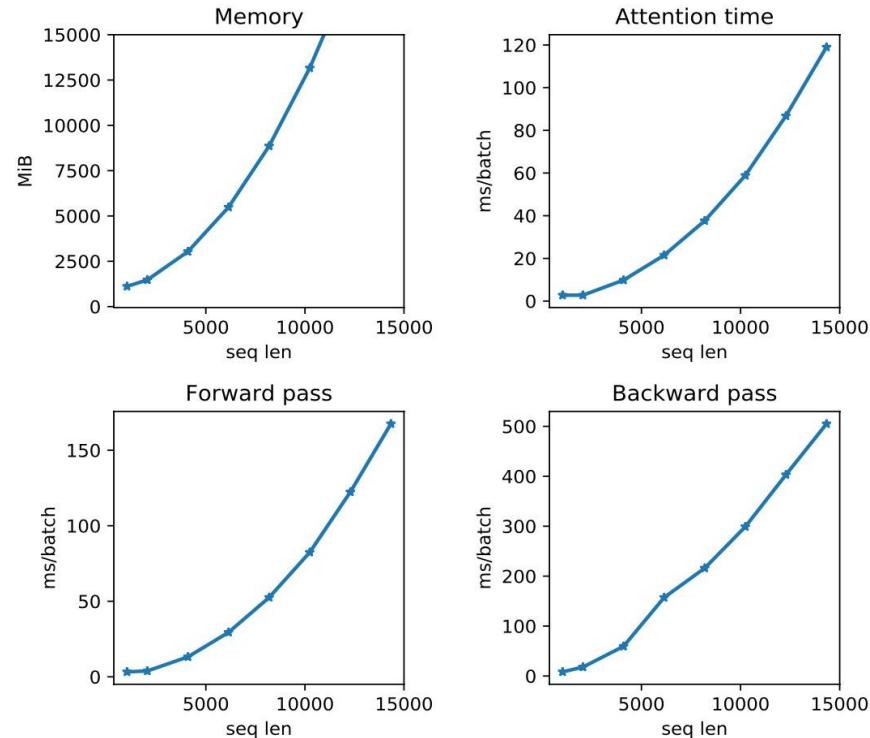
Summary

- Encoding positional information in language models is a non-trivial problem.
 - We discussed various proposal: learned, absolute, relative encoding, NoPos, etc.
- This is an important literature related to the length generalization of Transformers.
- This is an active research area and likely to change in the coming years.

Long Context: Efficiency and Generalization

Transformer LMs and Long Inputs

- **Length generalization:** Do Transformers work accurately on long inputs?
- **Efficiency considerations:** How efficient are LMs on long inputs?



Transformer LMs and Long Inputs

- A wide variety of topics.
- We will cover few only.

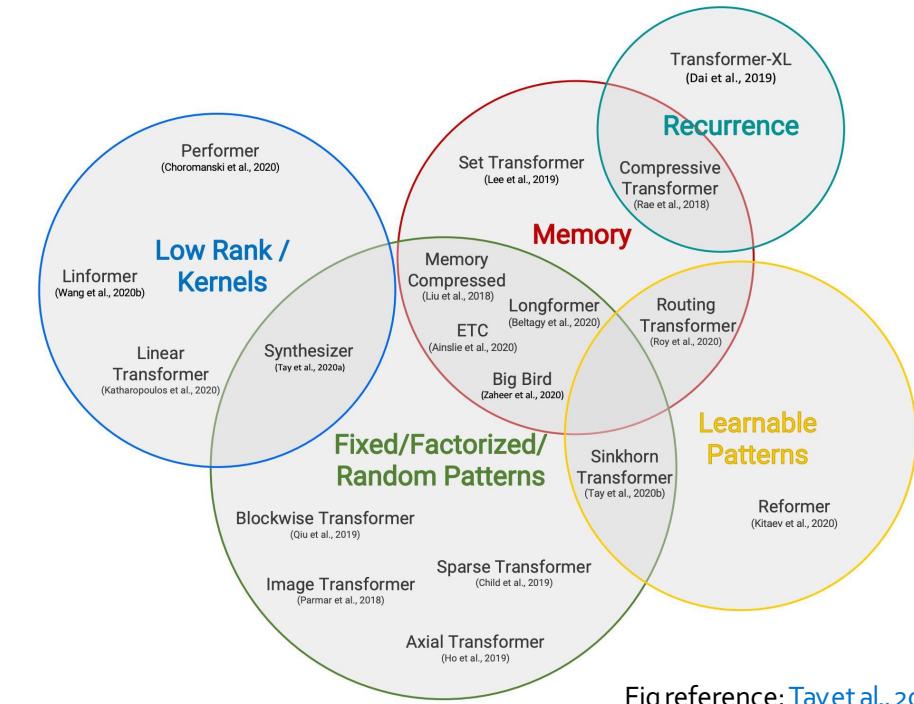


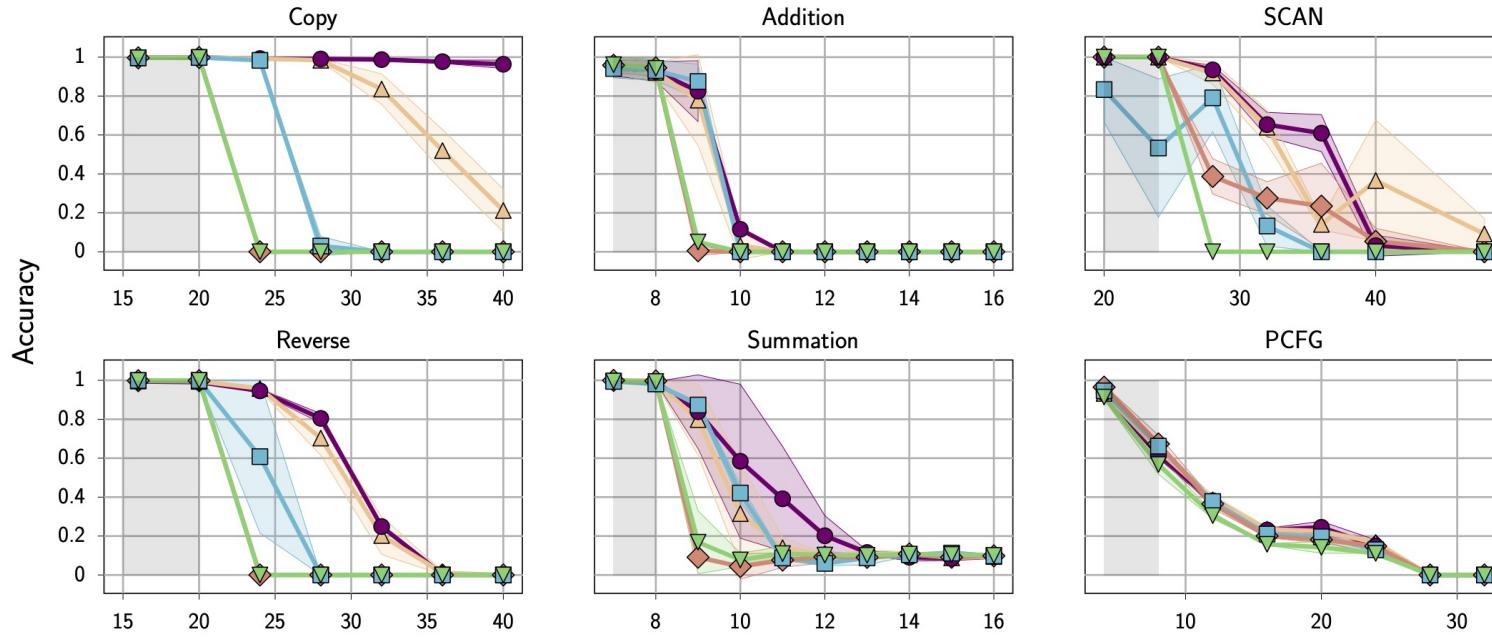
Fig reference: [Tay et al., 2020](#)

Length Generalization

Table 1: Examples of the input and output of the tasks.

Task	Input Example	Output Example
Primitive Tasks		
Copy	Copy the following words: <w1> <w2> <w3> <w4> <w5>	<w1> <w2> <w3> <w4> <w5>
Reverse	Reverse the following words: <w1> <w2> <w3> <w4> <w5>	<w5> <w4> <w3> <w2> <w1>
Mathematical and Algorithmic Tasks		
Addition	Compute: 5 3 7 2 6 + 1 9 1 7 ?	The answer is 5 5 6 4 3.
Polynomial Eval.	Evaluate x = 3 in (3 x ** 0 + 1 x ** 1 + 1 x ** 2) % 10 ?	The answer is 5.
Sorting	Sort the following numbers: 3 1 4 1 5 ?	The answer is 1 1 3 4 5.
Summation	Compute: (1 + 2 + 3 + 4 + 7) % 10 ?	The answer is 7.
Parity	Is the number of 1's even in [1 0 0 1 1] ?	The answer is No.
LEGO	If a = -1; b = -a; c = +b; d = +c. Then what is c?	The answer is +1.
Classical Length Generalization Datasets		
SCAN	jump twice and run left	JUMP JUMP TURN_LEFT RUN
PCFG	shift prepend K10 R1 K12 , E12 F16	F16 K10 R1 K12 E12

Length Generalization



Positional Encoding

■ NoPE ▲ T5's Relative PE ■ ALiBi □ Rotary ▼ Absolute Position Embedding

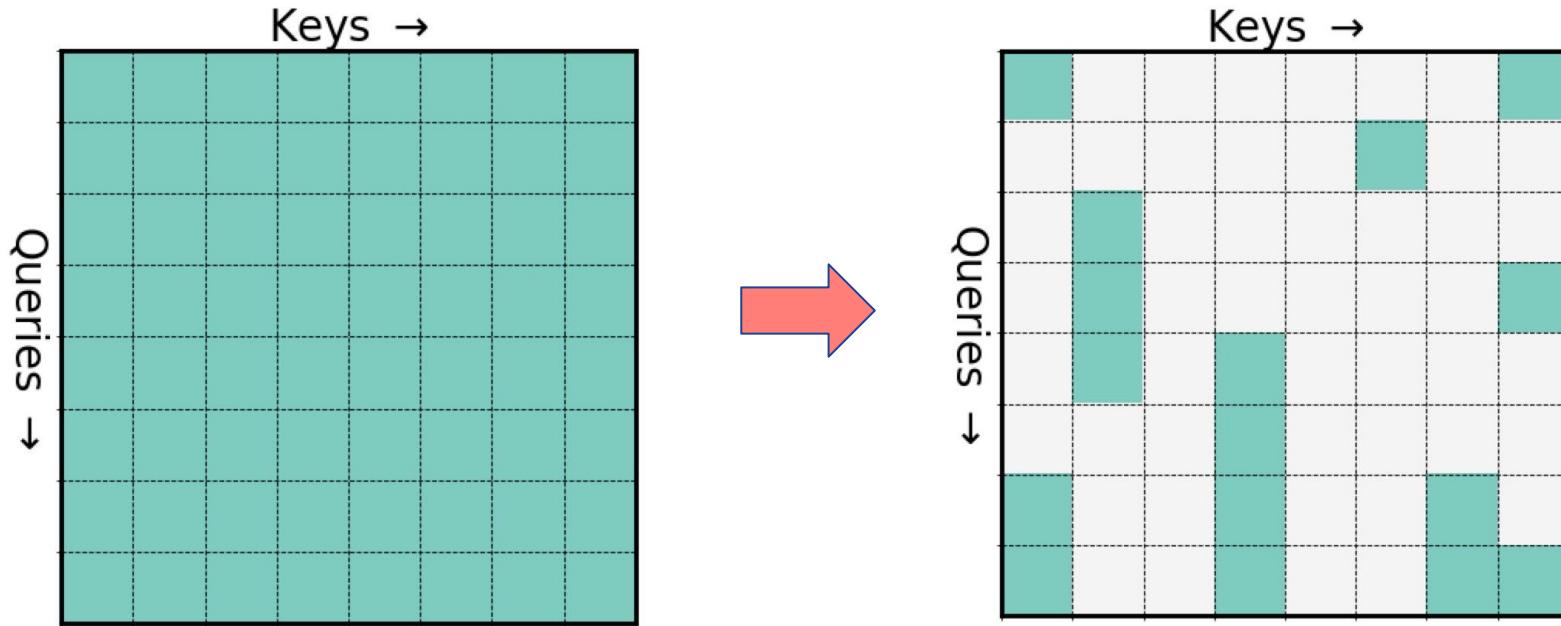


Efficiency considerations



Sparse Attention Patterns

- The idea is to make the attention operation sparse

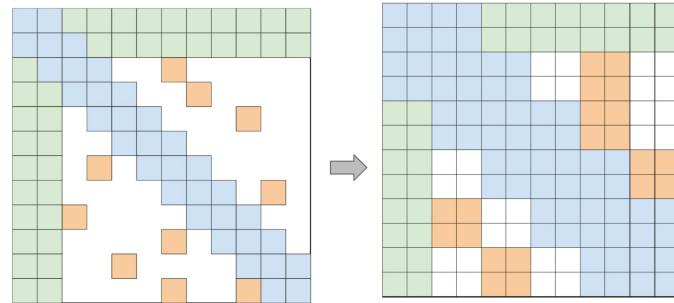


Sparse Attention Patterns: Challenge

- Ok sparsity is great, but how to efficiently implement this?
- **Challenge:** Arbitrary sparse matrix multiplication is not supported in DL libraries
- **A solution:** Perform computations in blocks

Pre-specified Sparsity Patterns: Computations

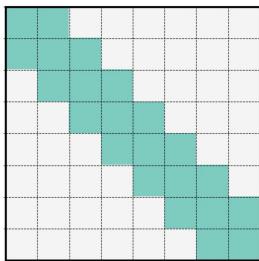
- Efficient blockified implementation
- There are libraries for implementing blockified sparse matrix multiplication.
 - Can be hardware specific
 - Block Sparse ([Gray et al., 2017](#))
 - TVM toolkit ([Chen et al., 2018](#))
 - cuSPARSE



Pre-specified Sparsity Patterns

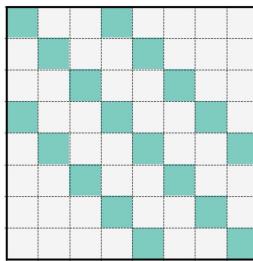
- A variety of patterns has been explored in the past work
 - Longformer ([Beltagy et al., 2020](#)), Sparse Transformer ([Child et al., 2019](#)), ...

Sliding window



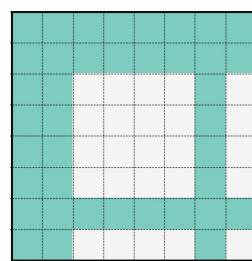
Sparse Transformer
Longformer

Dilated



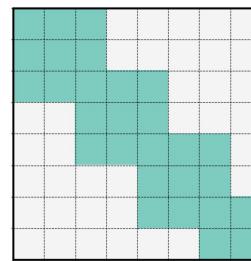
Longformer

Global



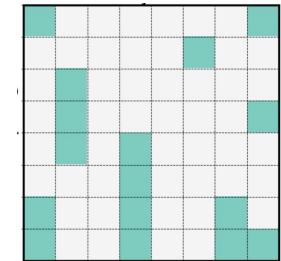
Big Bird

Blocked



Big Bird
Sinkhorn

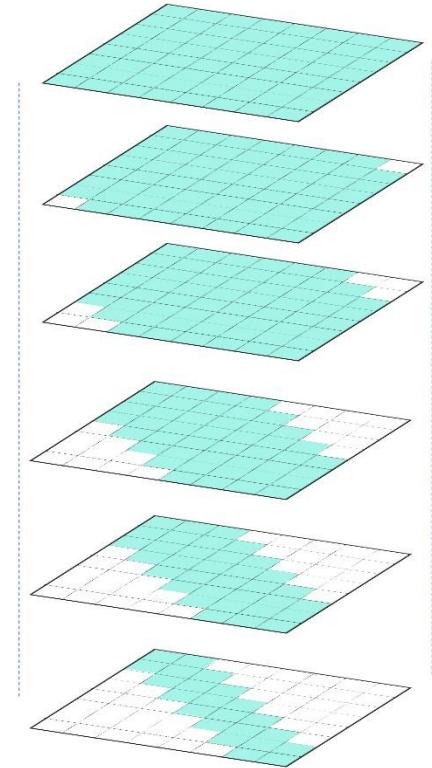
Random



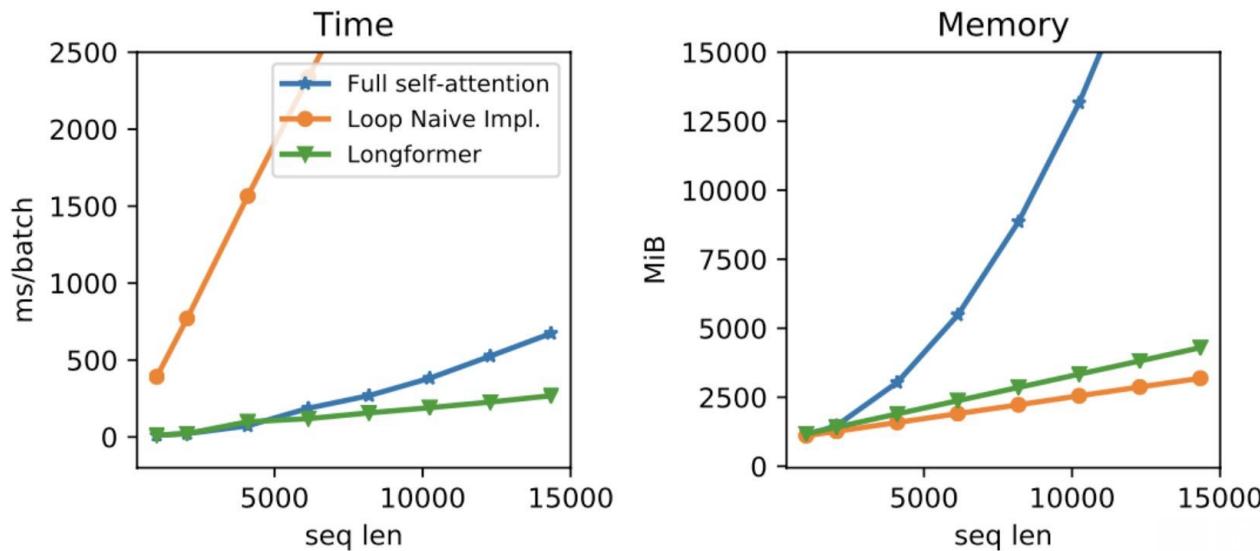
Big Bird

Pre-specified Sparsity Patterns

- Different layers and attention heads can follow different patterns
- A common setup is to have earlier layers with sparser attention pattern.
 - Longformer ([Beltagy et al., 2020](#))



Pre-specified Sparsity Patterns: Computations



A Notable Adoption: GPT-3

- Sparse patterns also used in GPT-3 ([Brown et al., 2020](#))

2.1 Model and Architectures

We use the same model and architecture as GPT-2 [RWC⁺19], including the modified initialization, pre-normalization, and reversible tokenization described therein, with the exception that we use alternating dense and locally banded sparse attention patterns in the layers of the transformer, similar to the Sparse Transformer [CGRS19]. To study the dependence of ML performance on model size, we train 8 different sizes of model, ranging over three orders of magnitude from 125 million parameters to 175 billion parameters, with the last being the model we call GPT-3. Previous work [KMH⁺20]

Summary

- How well do Transformers work on long sequences? Not so well.
- How can we make them more efficient? Induce sparsity.
- Many open questions we did not get into:
 - Limitations of sportifying Transformers
 - Alternatives to Transformers (e.g., state-space models)
 - ...

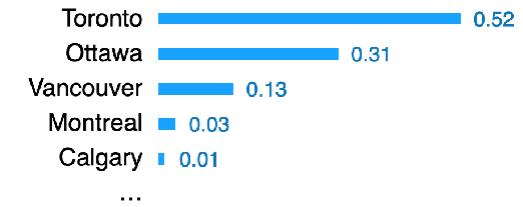
Retrieval-Augmented LMs

Retrieval-based Language Models

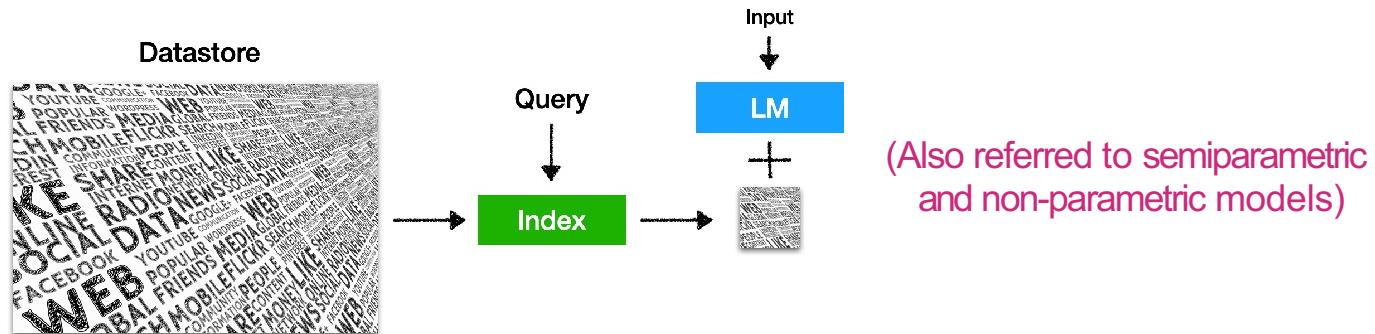
- It is a **language model** $P(x_n | x_1, x_2, \dots, x_{n-1})$

The capital city of Ontario is __

(can be broadly extended to masked language models or encoder-decoder models)



- It retrieves from an **external datastore** (at least during inference time)



Why retrieval-based LMs?

LLMs can't memorize all (long-tail) knowledge in their parameters



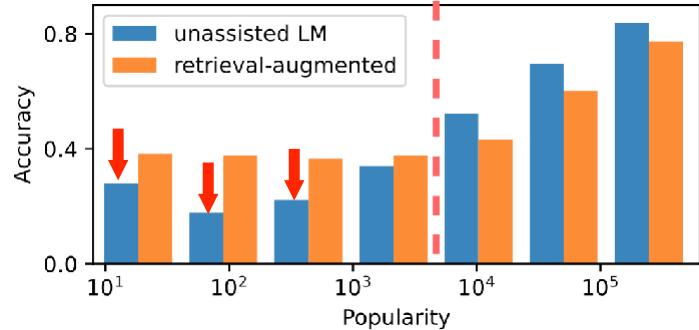
List 5 important papers authored by Geoffrey Hinton



Geoffrey Hinton is a renowned computer scientist ...
Here are five important papers authored by him:

1. **"Learning Internal Representations by Error Propagation"** (with D. E. Rumelhart and R. J. Williams) - This paper, published in **1986**, ..
2. **"Deep Boltzmann Machines"** (with R. Salakhutdinov) - Published in **2009**, ..
3. **"Deep Learning"** (with Y. Bengio and A. Courville) - Published as a book in **2016**,...
4. **"Attention Is All You Need"** (with V. Vaswani, N. Shazeer, et al.) - Published in **2017**, this paper introduced the Transformer model,...

What is Kathy Saltzman's occupation?



(Mallen et al., 2023)

GPT-3 davinci-003: 20%-30% accuracy

Why retrieval-based LMs?

LLMs' knowledge is easily outdated and hard to update



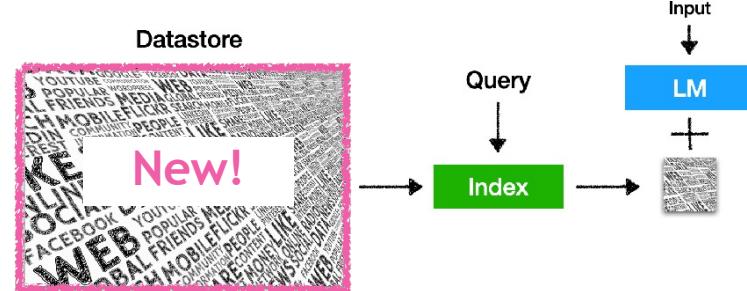
Who is the CEO of Twitter?



As of my **knowledge cutoff in September 2021**, the CEO of Twitter is **Jack Dorsey**....

- Existing **knowledge editing** methods are still NOT scalable (**active research!**)
- The datastore can be easily **updated** and **expanded** - even without retraining!

The screenshot shows a Google search results page. The search bar contains the query "Who is the CEO of Twitter?". Below the search bar are navigation links for All, News, Images, Shopping, Videos, and More, along with a Tools button. The main content area displays the search results, starting with a snippet about Twitter's CEO. To the right of the snippet is a portrait photo of Linda Yaccarino.



Why retrieval-based LMs?

LLMs' output is challenging to interpret and verify

The screenshot shows a search interface with a query bar: "Create an itinerary for exploring Toronto over a span of two days." Below the query, there are two main sections. The left section, starting with a blue icon, contains a general introduction and a bulleted list for Day 1. The right section, also starting with a blue icon, contains a bulleted list for Day 2. Red arrows from the text in the Day 2 list point to specific numbered links at the bottom of the page.

Create an itinerary for exploring Toronto over a span of two days.

Toronto is a vibrant and diverse city with many attractions and activities to suit different tastes and preferences. Here are some suggestions for creating an itinerary for exploring Toronto over two days:

- **Day 1:** Start your day with a visit to the **CN Tower**, the iconic landmark that offers panoramic views of the city and beyond ¹. You can also experience the thrill of walking on the glass floor or the EdgeWalk, the world's highest hands-free walk ¹. Next, head to the nearby **St. Lawrence Market**, one of the world's best food markets, where you can sample a variety of cuisines and local specialties ². After lunch, take a stroll along **Queen West**, a trendy neighborhood with eclectic shops, galleries, cafes and street art ³. In the evening, enjoy a show at one of the many theaters or comedy clubs in the **Entertainment District**, or catch a game at the **Scotiabank Arena** if you're a sports fan.

• **Day 2:** Explore the history and culture of Toronto at the **Royal Ontario Museum**, one of the largest and most comprehensive museums in North America, featuring exhibits on art, natural history, world cultures and more ⁴. Then, hop on a ferry to the **Toronto Islands**, a group of islands that offer a relaxing escape from the city, with beaches, parks, trails and amusement rides ³ ⁵. You can also rent a bike or kayak to explore the islands at your own pace. For dinner, head to **Chinatown**, one of the largest and most vibrant in North America, where you can find a variety of Asian cuisines and shops ³.

I hope this helps you plan your trip to Toronto. Have fun! 😊

Learn more:

- 1. cntower.ca
- 2. travel.usnews.com
- 3. bing.com
- 4. rom.on.ca
- 5. tripadvisor.com

Why retrieval-based LMs?

LLMs are **large** and expensive to train and run



Long-term goal: can we possibly reduce the **training and inference costs**, and scale down the size of LLMs?

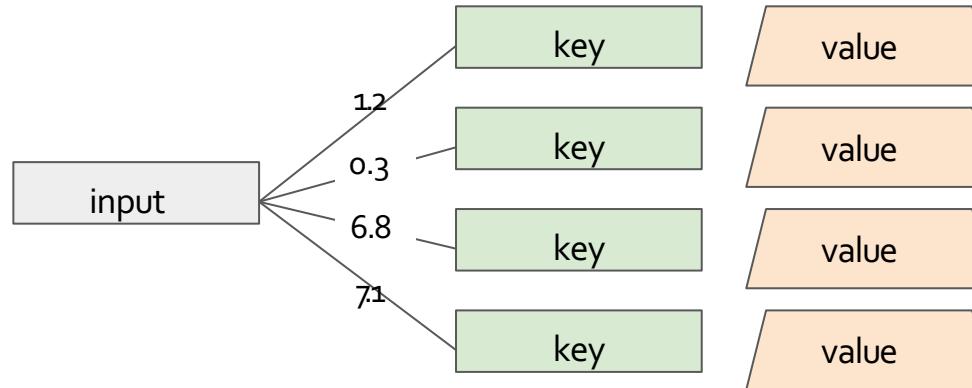
e.g., RETRO (Borgeaud et al., 2021): “obtains comparable performance to GPT-3 on the Pile, despite using **25x fewer parameters**”

What are the Key Design Questions?

- What are your memories?
 - Documents, database records, training examples, etc.
- How to retrieve memories?
 - Use an off-the-shelf search engine (e.g. Google, StackOverflow).
 - How to train your own memory retriever.
- How to use retrieved memories?
 - "Text fusion"
 - Common failure modes:
 - Underutilization: model ignores retrieved memories.
 - Overreliance: model depends too much on memories!

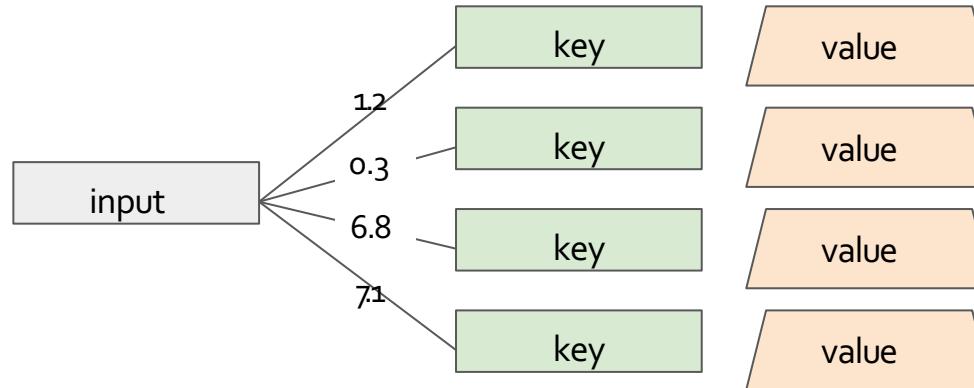
Anatomy of a Neural Retriever

1. Score the input against each key.
2. Return the value for the highest scoring key.



Anatomy of a Neural Retriever

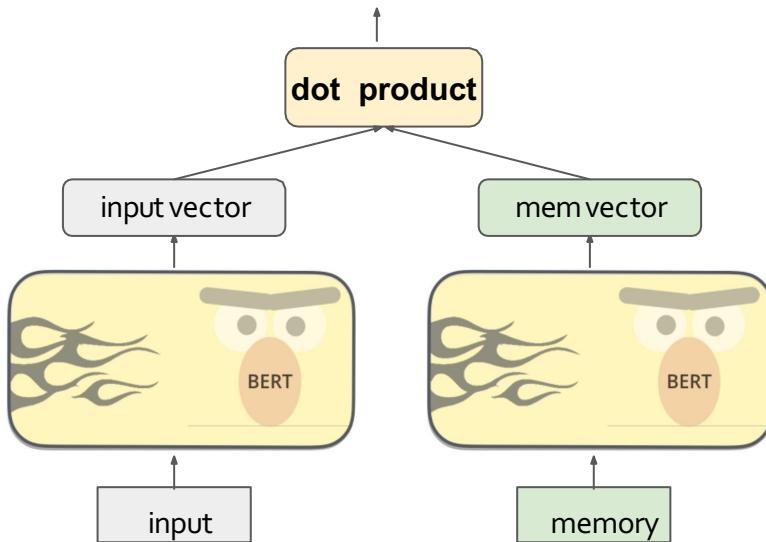
1. Score the input against each key.
2. Return the value for the highest scoring key.



A similarity function: $\text{sim}(\text{input}, \text{key}) \rightarrow \text{score}$

Defining Similarity Metrics

$$\text{sim}(I, M) = \text{Encoder}(I) \times \text{Encoder}(M)$$



- **Advantages:**

- Differentiable -- can optimize with gradient descent.

- **Disadvantages:**

- Works well for data on which your LM is pre-trained on.

Defining Similarity Metrics: Approach 2

$$\text{sim}(I, M) = \text{tf}(I, M) \times \log N/d_I$$

- $\text{tf}(I, M)$: # of occurrences of I in M .
- N : # of documents
- d_I : # of documents that contain I .

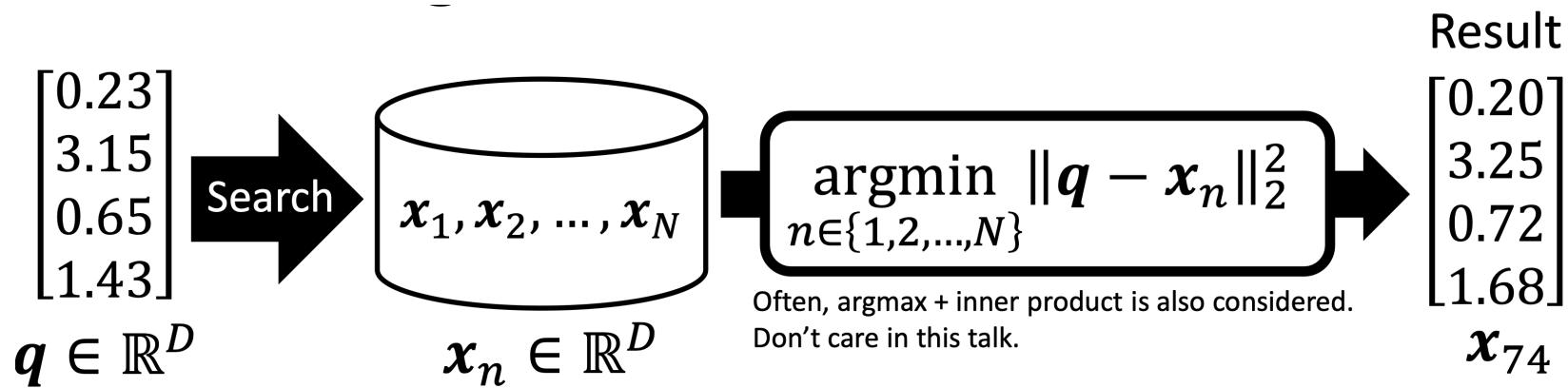
- **Advantages:**

- Differentiable -- can optimize with gradient descent.

- **Disadvantages:**

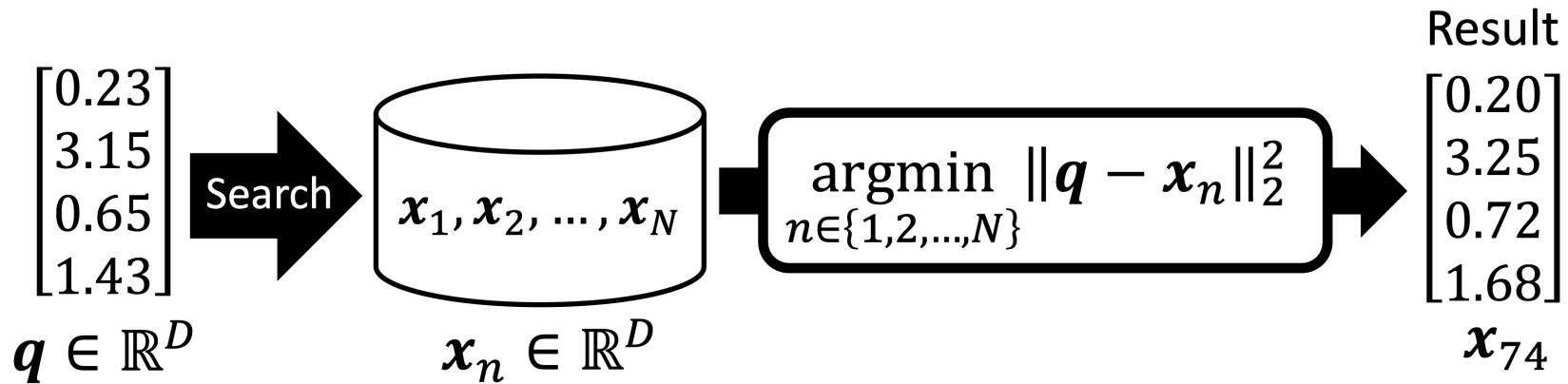
- Works well for data on which your LM is pre-trained on.

Finding Nearest Neighbors



- N D -dim database vectors: $\{x_n\}_{n=1}^N$
- Given a query q , find the closest vector from the database
- One of the fundamental problems in computer science
- Solution: linear scan, $O(ND)$, slow ☹

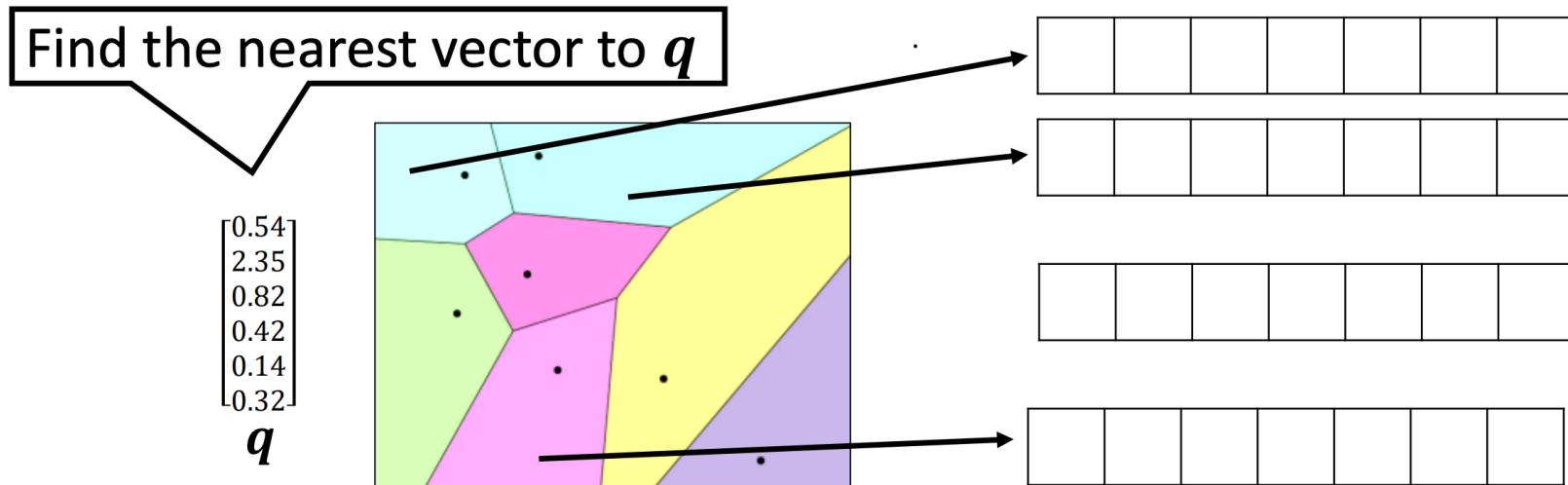
Approximate Finding Nearest Neighbors



- Faster search
- Don't necessarily have to be exact neighbors
- Trade off: runtime, accuracy, and memory-consumption

Approximate NNs: Algorithms, Libraries, Services

- Space partitioning + data compression



- More information: <https://github.com/facebookresearch/faiss/wiki>

Approximate NNs: Algorithms, Libraries, Services

Algorithm

- Scientific paper
- Math
- Often, by researchers

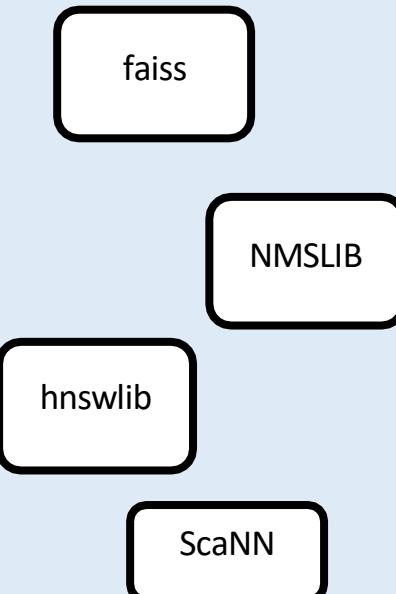
Product Quantization +
Inverted Index (PQ, IVFPQ)
[Jégou+, TPAMI 2011]

Hierarchical Navigable
Small World (HNSW)
[Malkov+, TPAMI 2019]

ScaNN (4-bit PQ)
[Guo+, ICML 2020]

Library

- Implementations of algorithms
- Usually, a search function only
- By researchers, developers, etc



Service (e.g., vector DB)

- Library + (handling metadata, serving, scaling, IO, CRUD, etc)
- Usually, by companies

Pinecone

Qdrant

Milvus

jina

Vald

Vertex AI
Matching Engine

Weaviate

Three levels of technology

Product Quantization +
Inverted Index (PQ, IVFPQ)
[Jégou+, TPAMI 2011]

Hierarchical Navigable
Small World (HNSW)
[Malkov+, TPAMI 2019]

faiss

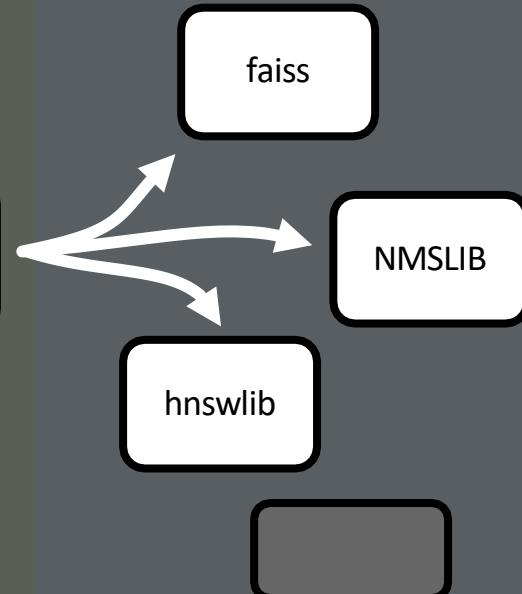
One library may implement
multiple algorithms

- “I benchmarked faiss”
- “I benchmarked PQ in faiss”

Three levels of technology

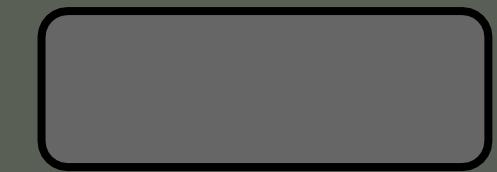
One algorithm may be implemented in multiple libraries

Hierarchical Navigable
Small World (HNSW)
[Malkov+, TPAMI 2019]



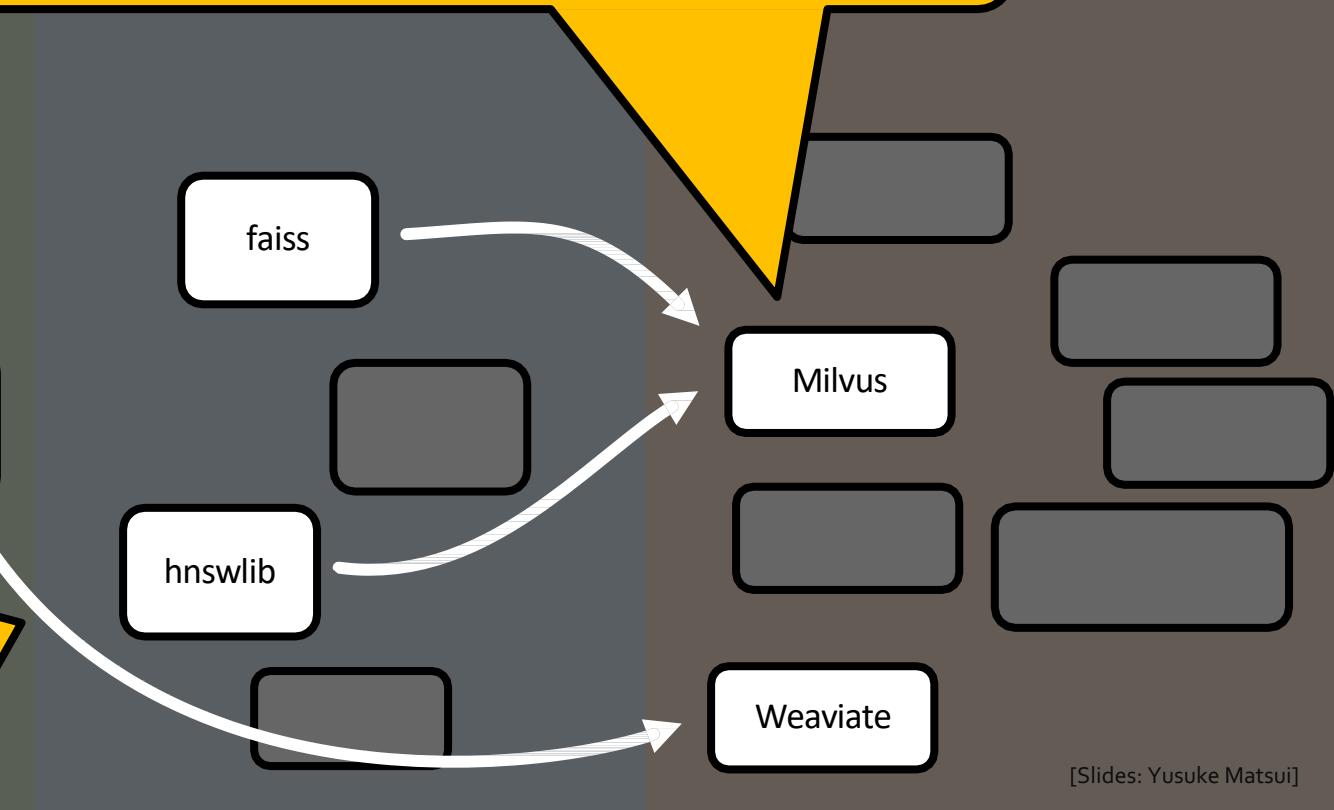
Three levels of technology

One service may use some libraries



Hierarchical Navigable
Small World (HNSW)
[Malkov+, TPAMI 2019]

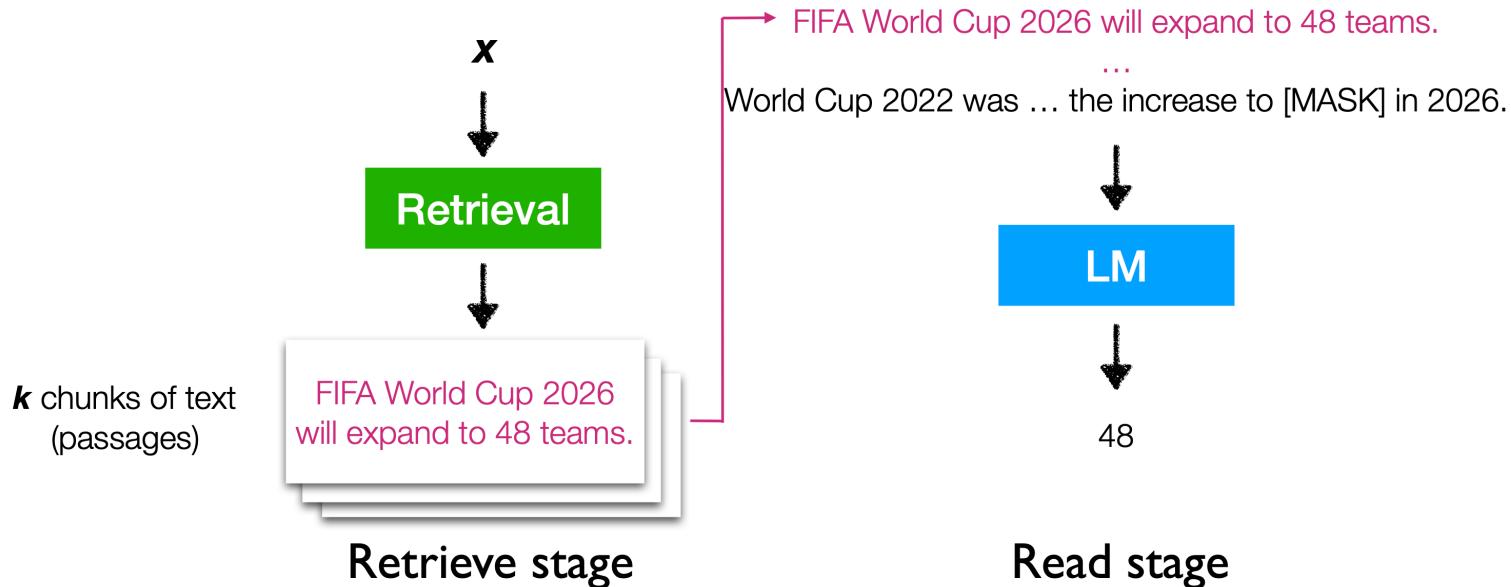
... or re-implement
algorithms from
scratch (e.g., by Go)



Let's assume that we have our retrieval engine and data ready

Retrieval-Augmented LM

- x = World Cup 2022 was the last before the increase to [MASK] in the 2026 tournament.

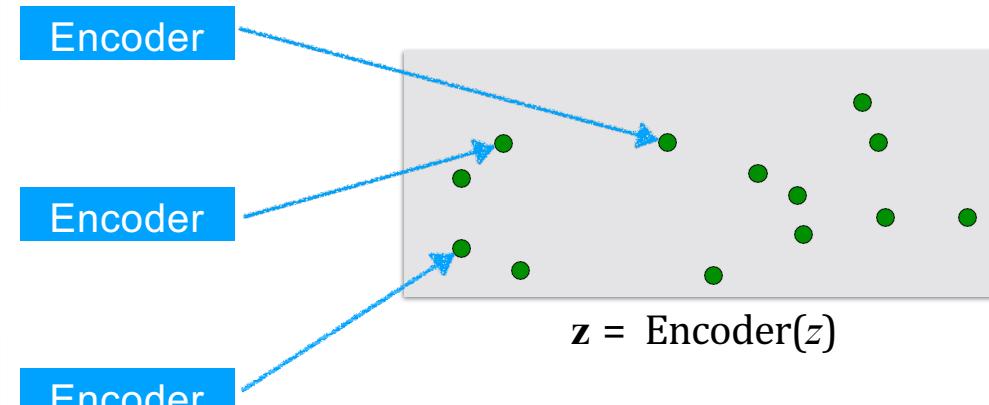


Retrieval-Augmented LM

FIFA World Cup 2026
will expand to 48 teams.

In 2022, the 32
national teams involved
in the tournament.

Team USA celebrated
after winning its match
against Iran ...



Wikipedia

13M chunks (passages)
(called *documents* in the paper)

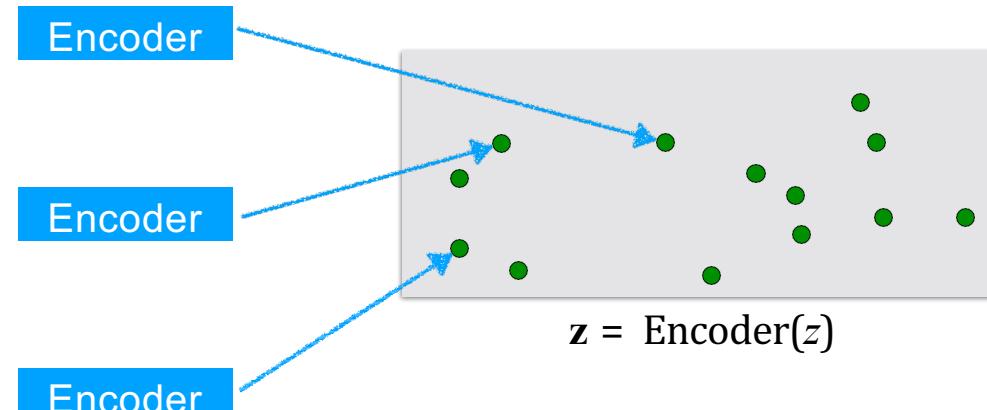
Retrieval-Augmented LM

x = World Cup 2022 was ... the increase to [MASK] in 2026.

FIFA World Cup 2026
will expand to 48 teams.

In 2022, the 32
national teams involved
in the tournament.

Team USA celebrated
after winning its match
against Iran ...

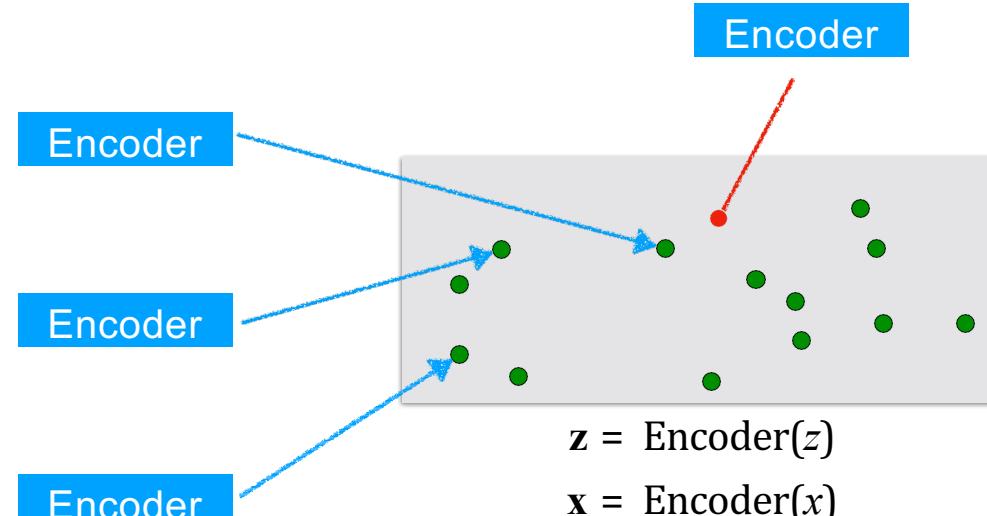


Wikipedia
13M chunks (passages)
(called *documents* in the paper)

Retrieval-Augmented LM

x = World Cup 2022 was ... the increase to [MASK] in 2026.

- FIFA World Cup 2026 will expand to 48 teams.
- In 2022, the 32 national teams involved in the tournament.
- Team USA celebrated after winning its match against Iran ...

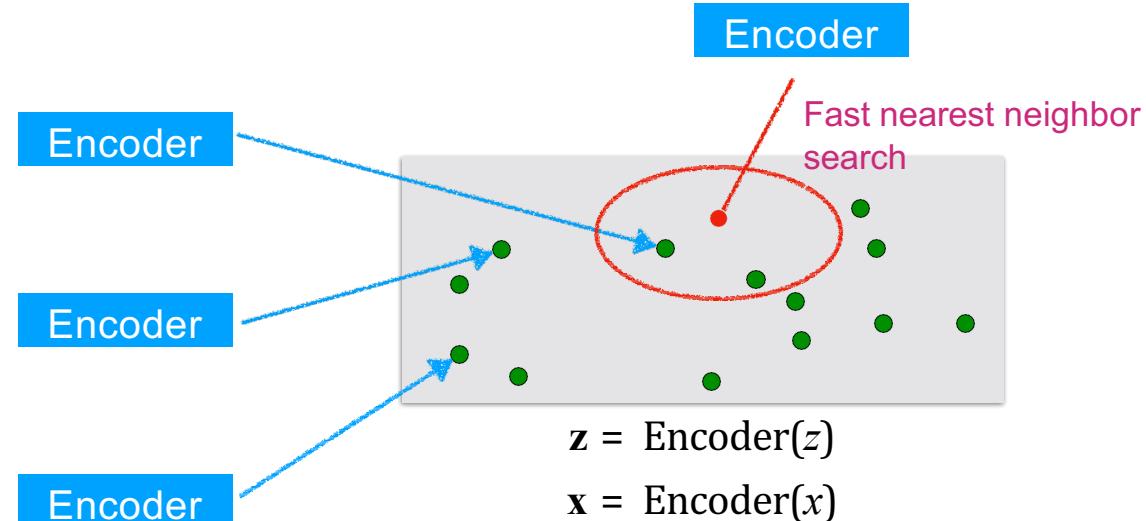


Wikipedia
13M chunks (passages)
(called *documents* in the paper)

Retrieval-Augmented LM

x = World Cup 2022 was ... the increase to [MASK] in 2026.

- FIFA World Cup 2026 will expand to 48 teams.
- In 2022, the 32 national teams involved in the tournament.
- Team USA celebrated after winning its match against Iran ...



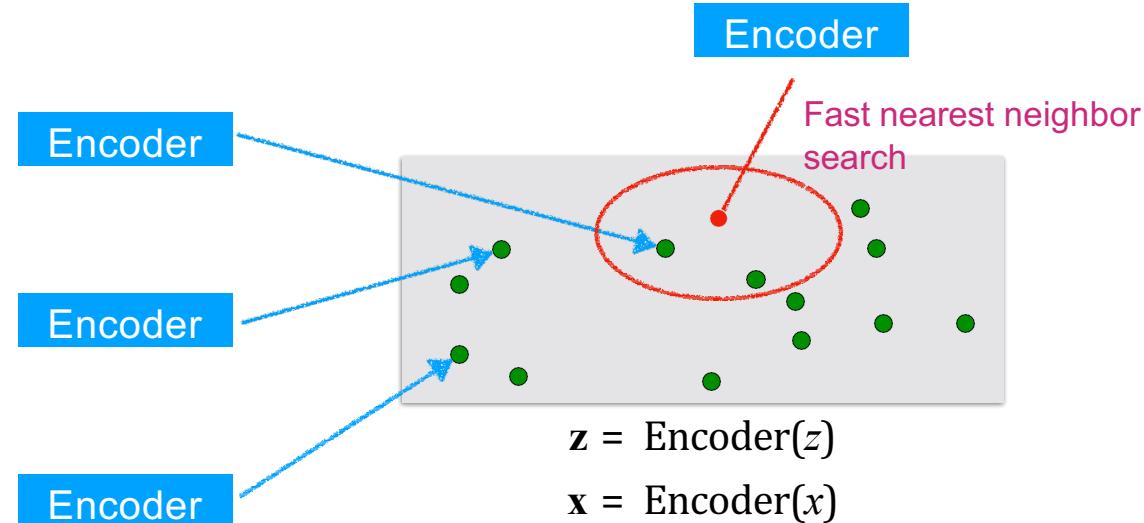
Retrieval-Augmented LM

x = World Cup 2022 was ... the increase to [MASK] in 2026.

FIFA World Cup 2026 will expand to 48 teams.

In 2022, the 32 national teams involved in the tournament.

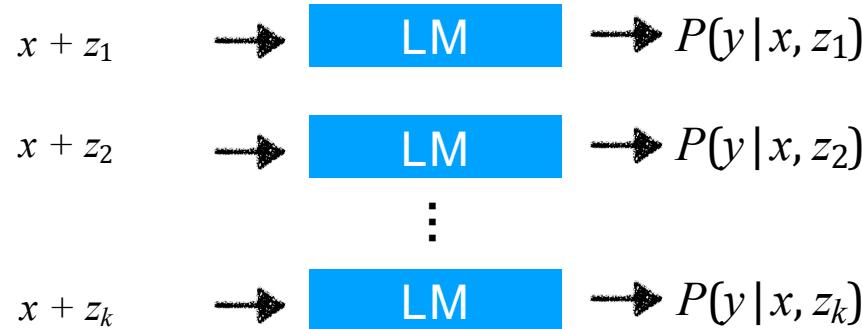
Team USA celebrated after winning its match against Iran ...



$$z_1, \dots, z_k = \text{argTop-}k(\mathbf{x} \cdot \mathbf{z})$$

k retrieved chunks

Retrieval-Augmented LM



Need to approximate
→ Consider top k chunks only

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

0 if not one of top k

from the retrieve stage from the read stage

Retrieval-Augmented LM: Variants

What to retrieve?

- Chunks
- Tokens
- Others

How to use retrieval?

- Input layer
- Intermediate layers
- Output layer

When to retrieve?

- Once
- Every n tokens ($n > 1$)
- Every token

Retrieval-Augmented LM: Common Variant

What to retrieve?

- Chunks
- Tokens
- Others

How to use retrieval?

- Input layer
- Intermediate layers
- Output layer

When to retrieve?

- Once
- Every n tokens ($n > 1$)
- Every token

IR in the Middle of LM

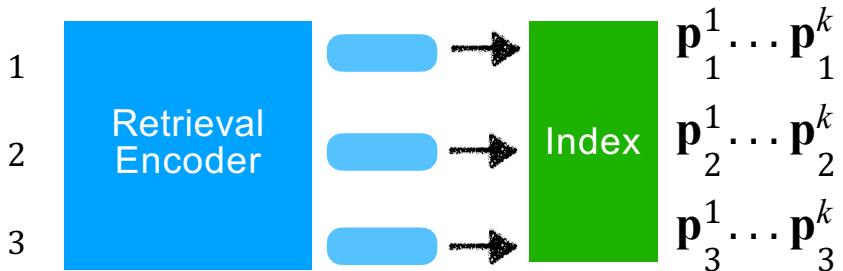
$x = \text{World Cup 2022 was}$ /the last with 32 teams,/ before the increase to

1

2

3

(k chunks of text per split)



IR in the Middle of LM

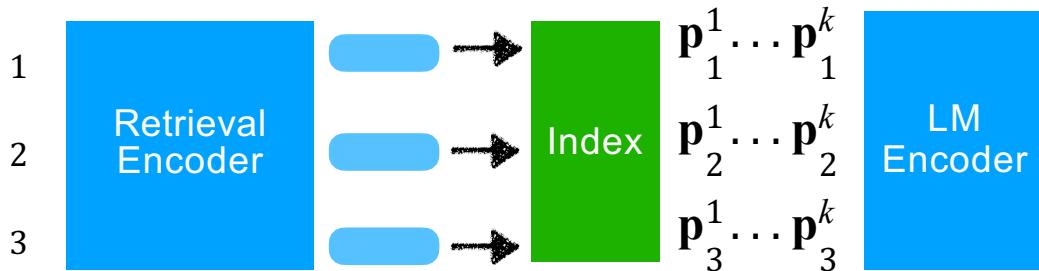
$x = \text{World Cup 2022 was}$ /the last with 32 teams,/ before the increase to

1

2

3

(k chunks of text per split)



IR in the Middle of LM

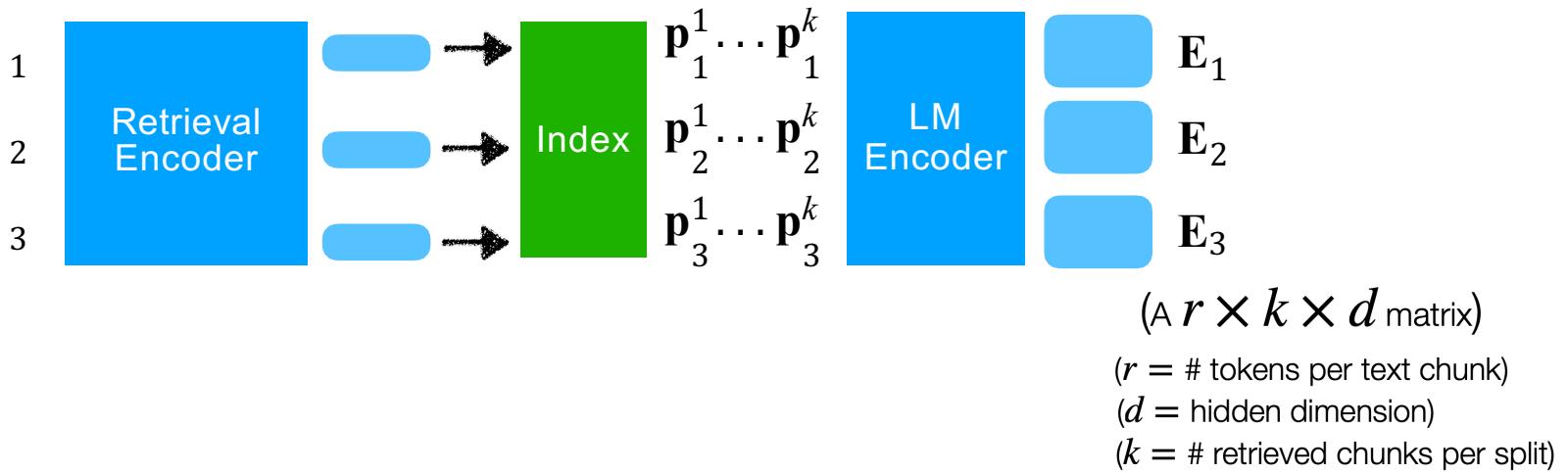
$x = \text{World Cup 2022 was}$ /the last with 32 teams,/ before the increase to

1

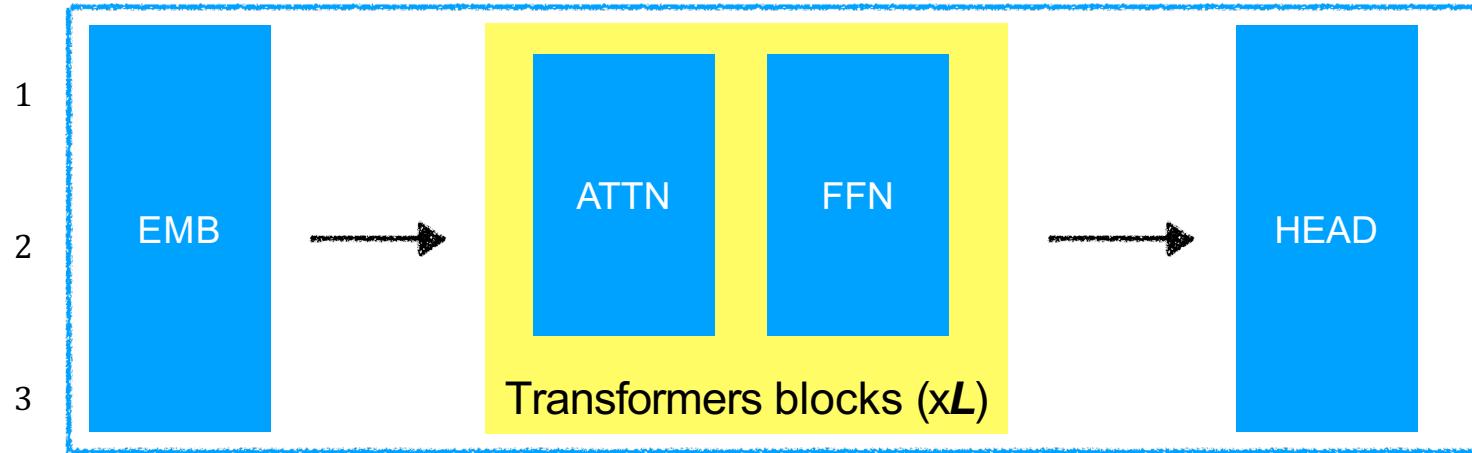
2

3

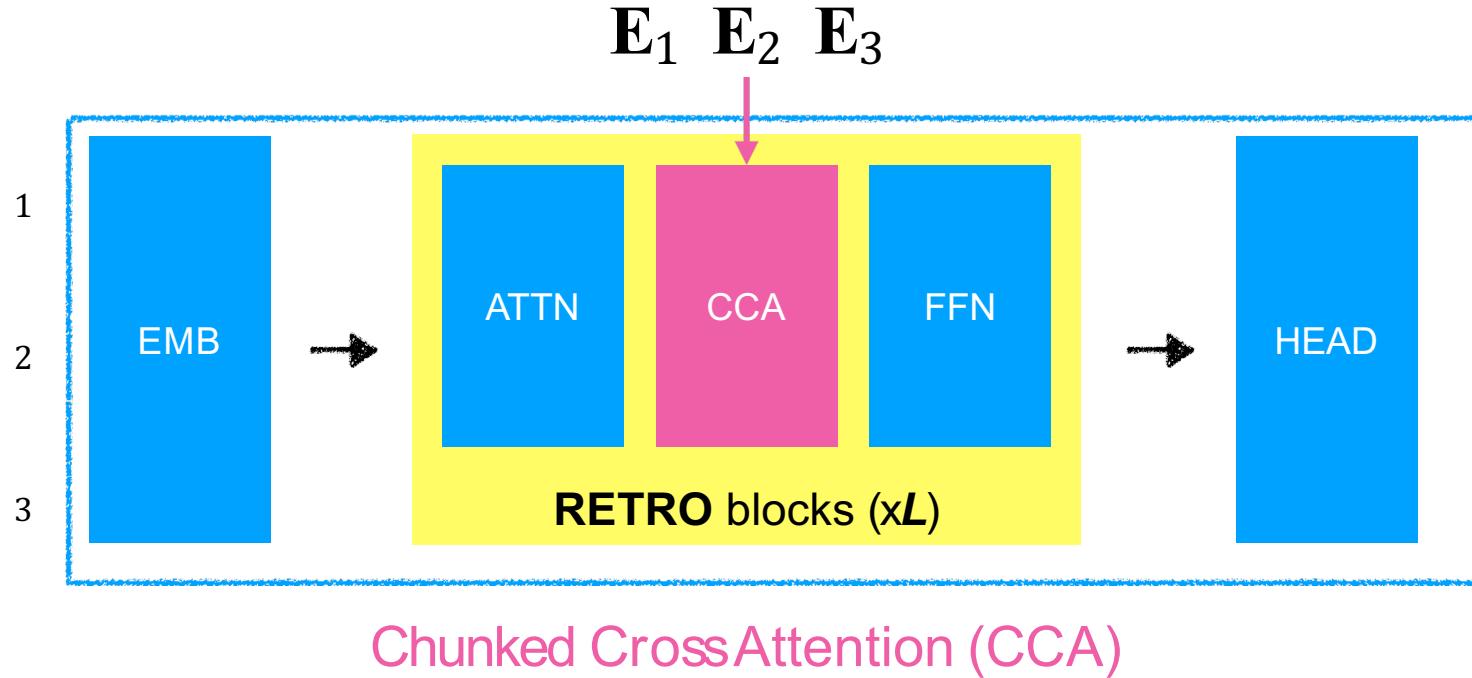
(k chunks of text per split)



Regular Decoder



Regular Decoder with IR Embeddings



Results

Model	Retrieval Set	#Database tokens	#Database keys	Perplexity: The lower the better	
				Valid	Test
Adaptive Inputs (Baevski and Auli, 2019)	-	-	-	17.96	18.65
SPALM (Yogatama et al., 2021)	Wikipedia	3B	3B	17.20	17.60
kNN-LM (Khandelwal et al., 2020)	Wikipedia	3B	3B	16.06	16.12
Megatron (Shoeybi et al., 2019)	-	-	-	-	10.81
Baseline transformer (ours)	-	-	-	21.53	22.96
kNN-LM (ours)	Wikipedia	4B	4B	18.52	19.54
RETRO	Wikipedia	4B	0.06B	18.46	18.97
RETRO	C4	174B	2.9B	12.87	10.23
RETRO	MassiveText (1%)	18B	0.8B	18.92	20.33
RETRO	MassiveText (10%)	179B	4B	13.54	14.95
RETRO	MassiveText (100%)	1792B	28B	3.21	3.92

Significant improvements by retrieving from 1.8 trillion tokens

Results

Perplexity: The lower the better

Model	Retrieval Set	#Database tokens	#Database keys	Valid	Test
Adaptive Inputs (Baevski and Auli, 2019)	-	-	-	17.96	18.65
SPALM (Yogatama et al., 2021)	Wikipedia	3B	3B	17.20	17.60
kNN-LM (Khandelwal et al., 2020)	Wikipedia	3B	3B	16.06	16.12
Megatron (Shoeybi et al., 2019)	-	-	-	-	10.81
Baseline transformer (ours)	-	-	-	21.53	22.96
kNN-LM (ours)	Wikipedia	4B	4B	18.52	19.54
RETRO	Wikipedia	4B	0.06B	18.46	18.97
RETRO	C4	174B	2.9B	12.87	10.23
RETRO	MassiveText (1%)	18B	0.8B	18.92	20.33
RETRO	MassiveText (10%)	179B	4B	13.54	14.95
RETRO	MassiveText (100%)	1792B	28B	3.21	3.92

Significant improvements by retrieving from 1.8 trillion tokens

Retrieval-Augmented LM

What to retrieve?

- Chunks
- Tokens
- Others

How to use retrieval?

- Input layer
- **Intermediate layers**
- Output layer

When to retrieve?

- Once
- **Every n tokens ($n > 1$)**
- Every token

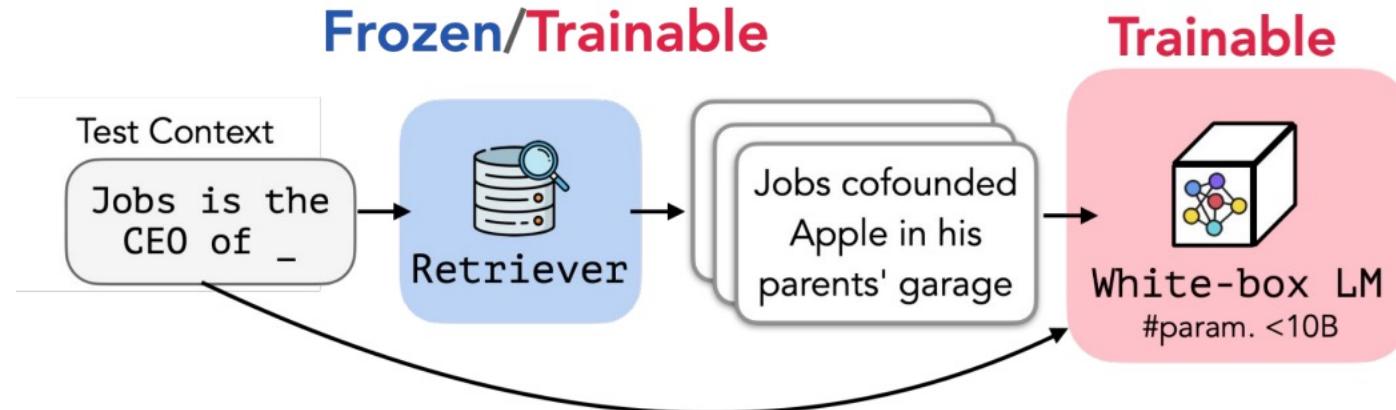


How do you train these models?



End-to-end Training

- There are various ideas in the literature for how to train these models efficiently and in an end-to-end fashion.



Main takeaways

- How do we enable LMs to utilize external knowledge?
 - Retrieval-augmented language models
- A retriever is a function, $f(\text{input}, \text{memory}) \rightarrow \text{score}$
- What we did not discuss:
 - Attribution: Tracing decisions to the source knowledge
 - How to modify the knowledge
 - Conflicting knowledge
 - Editing knowledge
 - More efficient scaling
 -