Selective Architectures

Jie Tang
Department of Computer Science & Technology
Tsinghua University

Outline

- Mixture-of-experts
- Retrieval-augmentation Generation

Outline

- Mixture-of-experts (MOE)
  - Basics
  - Recent studies
    - Sparsely-gated mixture of experts
    - Switch Transformer
    - Balanced Assignment of Sparse Experts layers
    - Generalist Language Model (GLaM)
    - FacebookMOE
    - Decentralized MOE
- Retrieval-augmentation Generation (RAG)

Outline

- **Mixture-of-experts (MOE)**
  - **- Basics**
  - - Recent studies
    - - Sparsely-gated Mixture of Experts
    - - Switch Transformer
    - - Balanced Assignment of Sparse Experts Layers
    - - Generalist Language Model (GLaM)
    - - FacebookMOE
    - - Decentralized MOE
- Retrieval-augmentation Generation (RAG)

Basics of Mixture of Experts

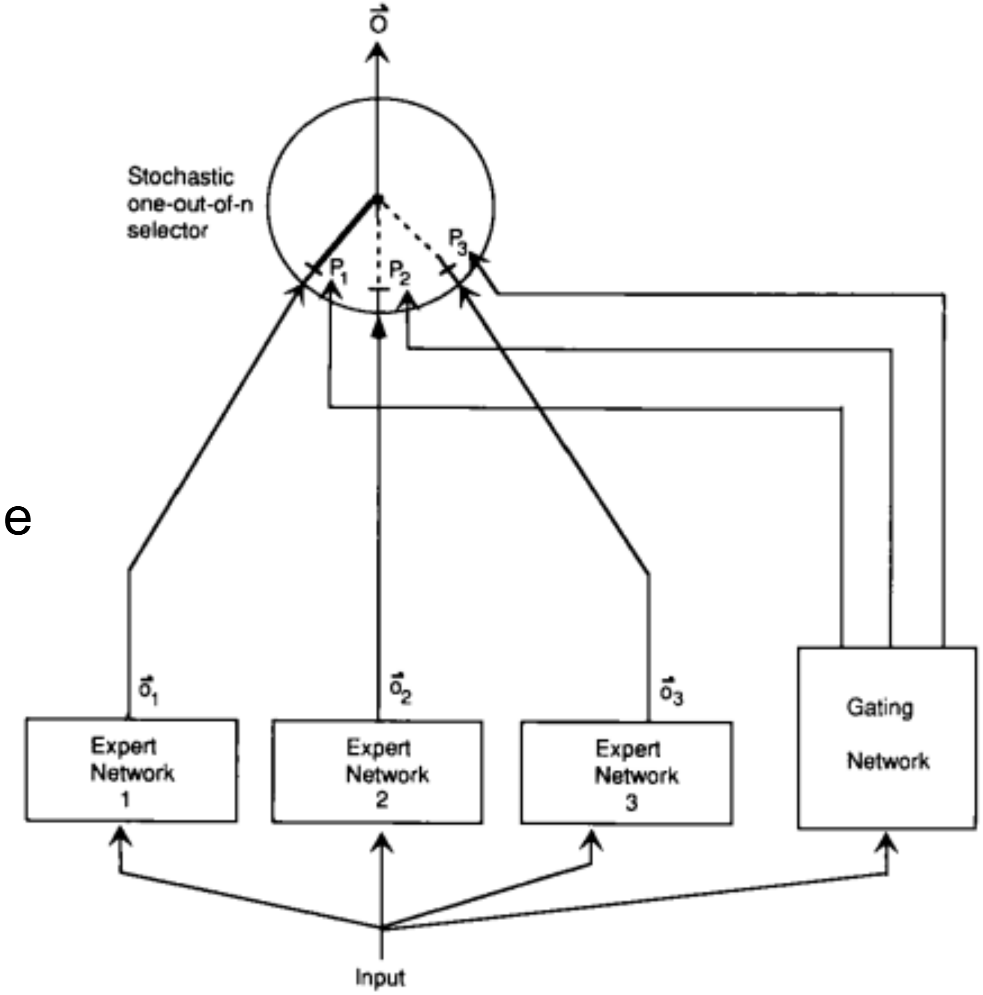
- A prediction problem:

$$x \in \mathbb{R}^d \Rightarrow y \in \mathbb{R}^d.$$

- Learn a feedforward neural network:

$$h_\theta(x) = W_2 \max(W_1 x, 0),$$

  - Problems:
    - This function might not be powerful to represent the **function of interest**.
    - The neural network might be **wider or deeper**.

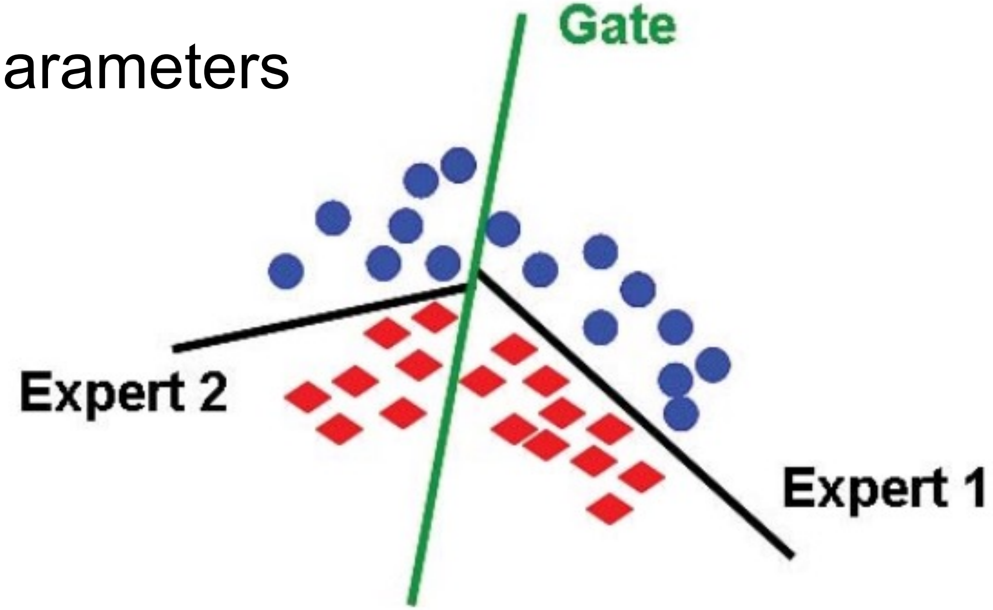


https://www.cs.toronto.edu/~fritz/absps/jjnh91.pdf

Basics of Mixture of Experts

- Mixture-of-experts approach:
  - Define E experts
  - Each expert has an embedding
  - **Gating function** as probability distribution
  - Each expert has parameters
  - **Expert function** in terms of expert-specific parameters
  - Final function as a MOE

- Example:
  - d=2;
  - Each expert being a linear classifier.



S. E. Yuksel, J. N. Wilson and P. D. Gader, "Twenty Years of Mixture of Experts," in IEEE Transactions on Neural Networks and Learning Systems, vol. 23, no. 8, pp. 1177-1193, Aug. 2012, doi: 10.1109/TNNLS.2012.2200299.

Training of Mixture of Experts

- Training:
  - – Learn a MOE model by normal backpropagation.
    - the gradient is proportional to $g_e(x)$ and updates both the gating function and the experts.

$$\nabla f(x) = \sum_{e=1}^{E} g_e(x)(\nabla(\log g_e(x))h_{\theta_e}(x) + \nabla h_{\theta_e}(x)).$$

- Saving Compute:
  - – Gating function is non-zero for each expert;

$$g(x) = [0.04, 0.8, 0.01, 0.15].$$

  - – MOE does not save any compute
    - – A feedforward pass **evaluates** each expert;
    - – The backward pass **touches** each expert.
  - – **Approximate the gating function**
    - – Place zero on most experts
    - – Only evaluate the **nonzero** expert

- Example:
  - – Take top 2 experts
  - – Renormalize

$$\tilde{g}(x) = [0, 0.84, 0, 0.16].$$

Parallelism of Mixture of Experts

- Balancing Experts:
  - The MOE is only effective if all experts pitch in.
  - If only one expert is activate, this is a waste.
  - If we end up this state, the gradients for the unused experts will be zero.
  - Ensure that **all the experts are used** across inputs.

- Parallelism:
  - The MOE is very conductive to parallelization.
  - Each expert can occupy a different machine.
  - The approximate gating function is computed.
  - The sparse set of machines contains **activated** experts.

Outline

- **Mixture-of-experts (MOE)**
  - Basics
  - **Recent studies**
    - **Sparsely-gated Mixture of Experts**
    - Switch Transformer
    - Balanced Assignment of Sparse Experts Layers
    - Generalist Language Model (GLaM)
    - FacebookMOE
    - Decentralized MOE
- Retrieval-augmentation Generation (RAG)

Sparsely-gated Mixture of Experts

How to the MOE can be applied to LMs?

- Naive solution: a mixture of 96-layer Transformers.
  - The gating function need to apply to a sequence;
  - The combination of experts only happens at the top.
- **Apply to MOE to each token and each Transformer block.**
- Turn each FFN in to a MOE FFN

  - The feed-forward layer is independent for each token

https://arxiv.org/pdf/2006.16668.pdf

Sparsely-gated Mixture of Experts

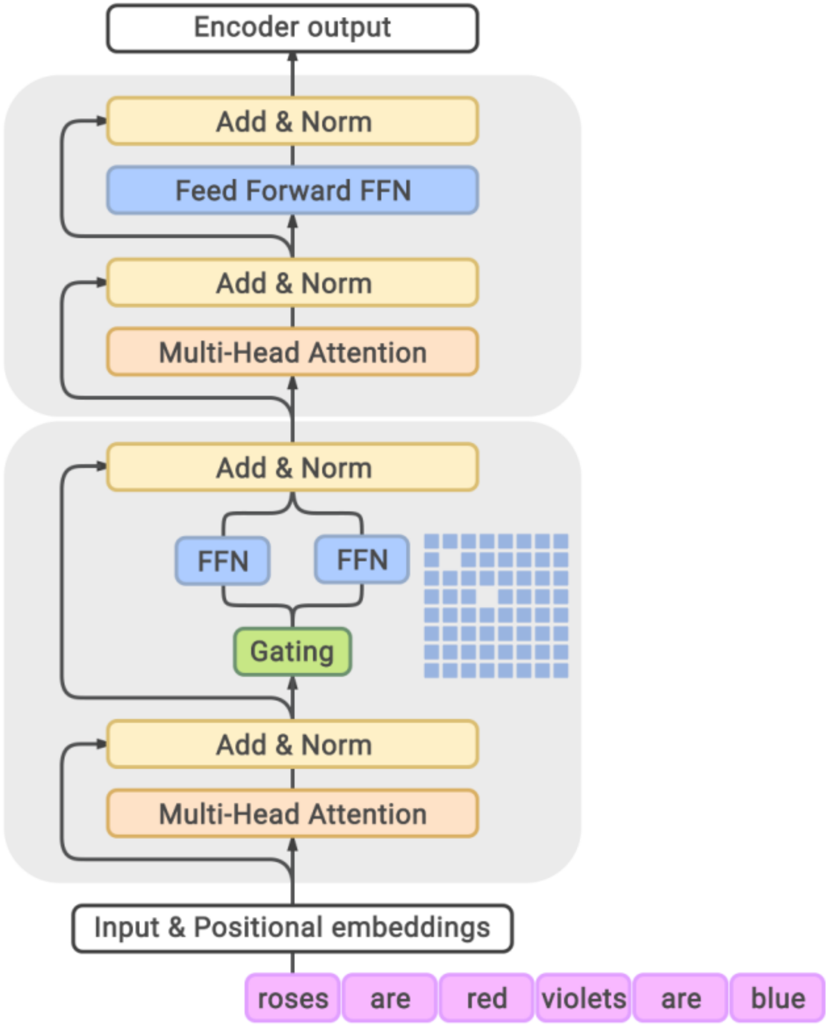
MoETransformerBlock($x_{1:L}$)=AddNorm(MoEFeedForward,AddNorm(SelfAttention, $x_{1:L}$))

- Each other Transformer block uses a MOE Transformer block.
- The **top-2 experts** approximate gating function:
  - Compute the top and second experts;
  - Keep top and second experts stochastically.
- Example of Balancing Experts:

$$g(x_1) = [0.2, 0.6, 0.1, 0.1] \Rightarrow \tilde{g}(x_1) = [0.25, 0.75, 0, 0]$$

$$g(x_2) = [0.1, 0.6, 0.2, 0.1] \Rightarrow \tilde{g}(x_2) = [0, 0.75, 0.25, 0]$$

  - C=[1,2,1,0], m=[0.3,1.2,0.3,0.2]
  - Push down on the gating function on **expert 2** to **discourage** its use.



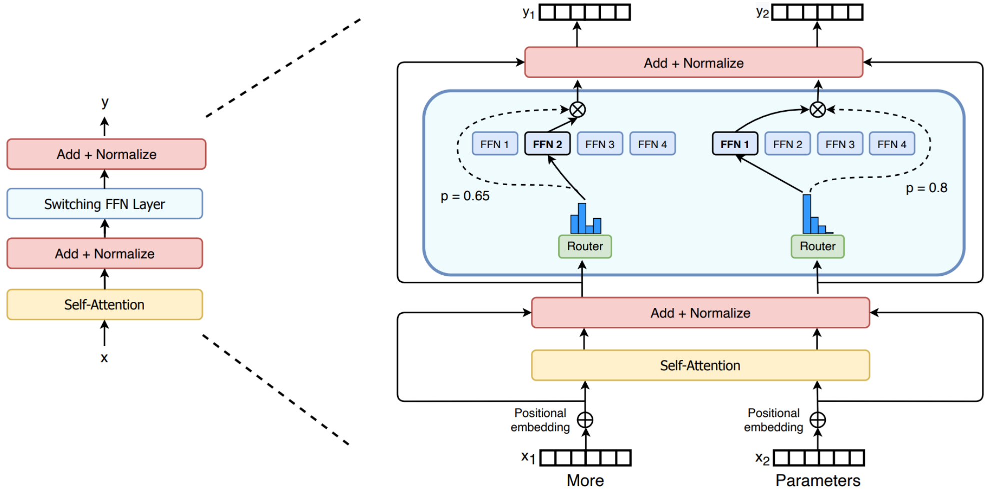
https://arxiv.org/pdf/2006.16668.pdf

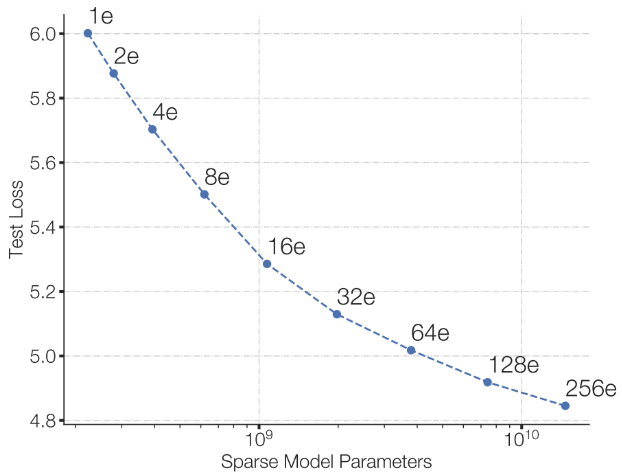
Outline

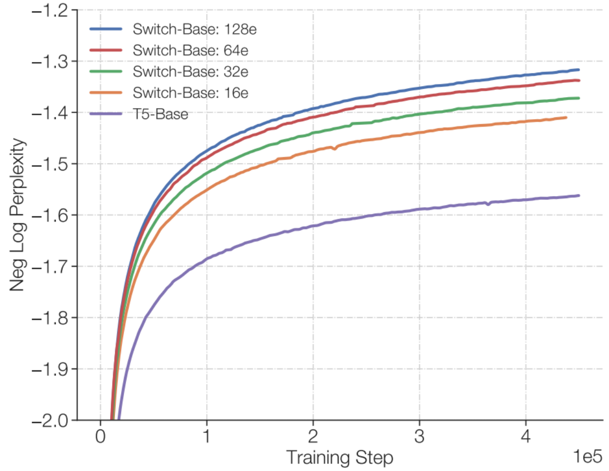
- **Mixture-of-experts (MOE)**
  - Basics
  - **Recent studies**
    - Sparsely-gated Mixture of Experts
    - **Switch Transformer**
    - Balanced Assignment of Sparse Experts Layers
    - Generalist Language Model (GLaM)
    - FacebookMOE
    - Decentralized MOE
- Retrieval-augmentation generation (RAG)

Switch Transformer

- Define the approximate gating function to only be the top-1 expert.
- Trained a 1.6 trillion parameter model
  - Does selective casting from FP32 to FP16
  - Smaller parameters for initialization
  - Expert dropout
  - Expert parallelism







- Improved pre-training speed compared to T5-XXL by 4x

https://arxiv.org/pdf/2101.03961.pdf

Outline

- **Mixture-of-experts (MOE)**
  - Basics
  - **Recent studies**
    - Sparsely-gated Mixture of Experts
    - Switch Transformer
    - **Balanced Assignment of Sparse Experts Layers**
    - Generalist Language Model (GLaM)
    - FacebookMOE
    - Decentralized MOE
- Retrieval-augmentation generation (RAG)

Balanced Assignment of Sparse Experts Layers (BASE)

- Define the approximate gating function to be the result of **a joint optimization** overall the tokens in the batch.
- Assign each token 1 expert, **load balancing is a constraint**.
- Propose a linear program and parallelize the linear program.

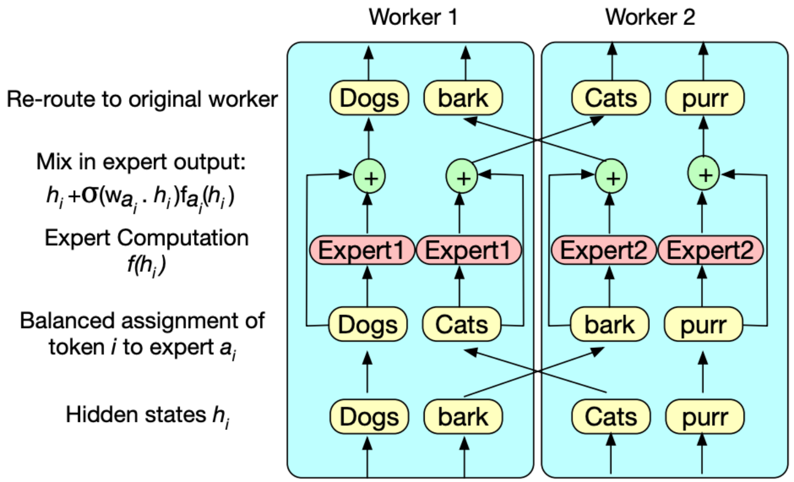
$$\text{maximize} \sum_{i=1}^{B} w_{a_i} \cdot x_i \quad \text{subject to} \quad \forall e : \sum_{i=1}^{B} \mathbf{1}[a_i = e] = \frac{B}{E}.$$

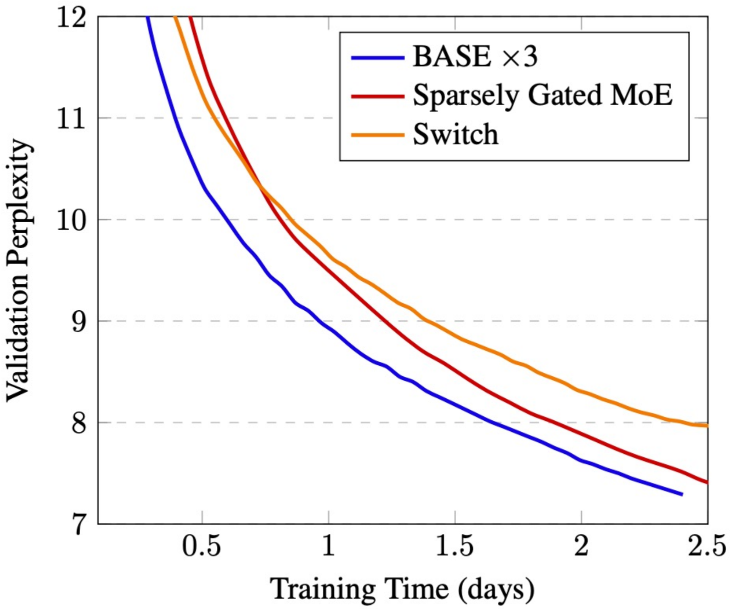
- Only choose the top-1 expert **at test time**.

https://arxiv.org/pdf/2103.16716.pdf

Balanced Assignment of Sparse Experts Layers (BASE)

- Experimental setup:
  - Sparsely gated MoE (top-2 experts): 52.5B parameters
  - Switch Transformer (top-1 expert): 52.5B parameters
  - BASE (1 jointly optimized expert): 44.4B parameters (1.3B shared parameters, 335M x 128 expert parameters)





https://arxiv.org/pdf/2103.16716.pdf

Summary and Next Steps

- Switch Transformer (Google) used top-1 expert.
- BASE (Facebook) used 1 expert per token, but jointly optimized.
- Two most recent high-performing MoE language models:
  - GLaM from Google
  - “FacebookMoE” from Facebook
- **They do compete with GPT-3, but interestingly, they are still based on the original simple top-2 experts.**

Outline

- **Mixture-of-experts (MOE)**
  - Basics
  - **Recent studies**
    - **Sparsely-gated Mixture of Experts**
    - Switch Transformer
    - Balanced Assignment of Sparse Experts Layers
    - **Generalist Language Model (GLaM)**
    - **FacebookMOE**
    - Decentralized MOE
- Retrieval-augmentation Generation (RAG)

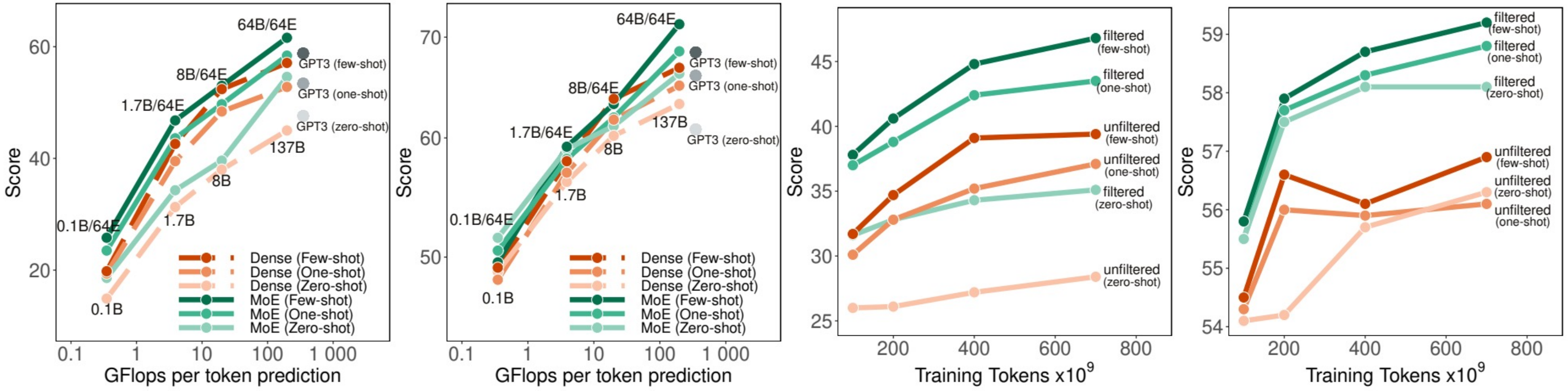
Generalist Language Model (GLaM)

- Specification:
  - 1.2 trillion parameters (GPT-3 had 175 billion parameters)
  - 64 experts (not that many), 64 layers, 32K hidden units
  - Each token activates 95B (8% of 1.2T) of the parameters
- Other upgrades:
  - New dataset (GLaM dataset) of 1.6 trillion tokens of webpages, forums, books, news, etc.
  - Relative positional embeddings, Gated linear units, GeLU activation function, RMSNorm (not LayerNorm)
  - Skip weight updates / rollback to earlier checkpoint if encounter NaN/Inf.
  - “Training of sparsely activated models at all scales becomes quite stable.”

https://arxiv.org/pdf/2103.16716.pdf

Generalist Language Model (GLaM)

- Results:
  - 1/3 of the cost to train compared to GPT-3
  - Evaluated on same benchmarks as GPT-3 (open-domain question answering, reading comprehension, SuperGLUE, etc.)
  - Achieved **better 0-shot and 1-shot performance** compared to GPT-3 (especially performant on knowledge-intensive tasks)
  - Note: they did not evaluate in the few-shot, where GPT-3 is stronger

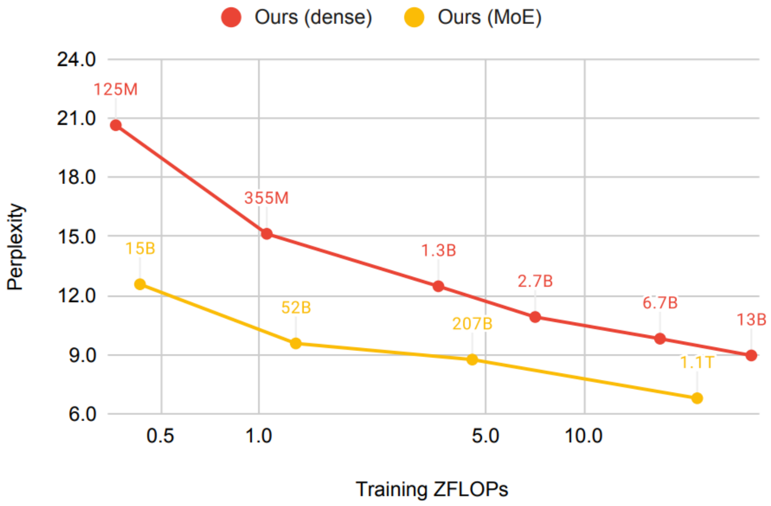


(a) Scaling (NLG) (b) Scaling (NLU) (c) Data filtering (NLG) (d) Data filtering (NLU)

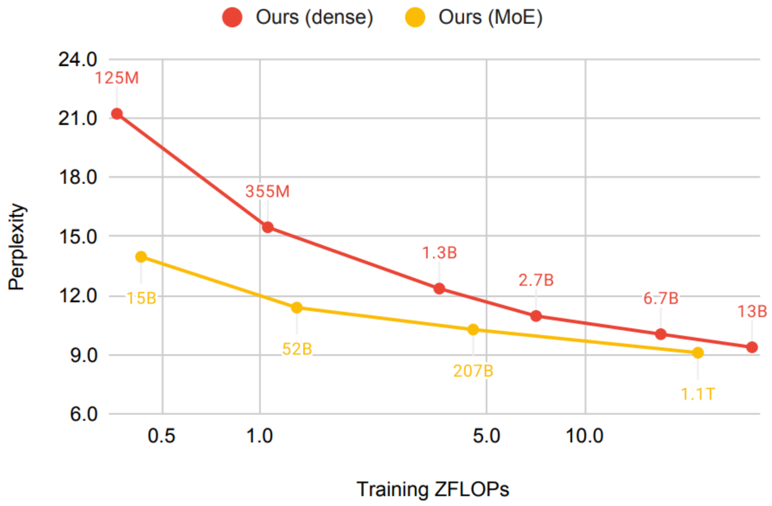
https://arxiv.org/pdf/2103.16716.pdf

FacebookMOE

- Setup:
  - Trained a 1.1T parameter model
  - 512 experts (more than GLaM), 32 layers, 4096 hidden units
  - Trained on 112 billion tokens on webpages, forums, books, news, etc.
  - Strong gains for smaller models, diminishing gains for larger models



(a) In-domain (validation)

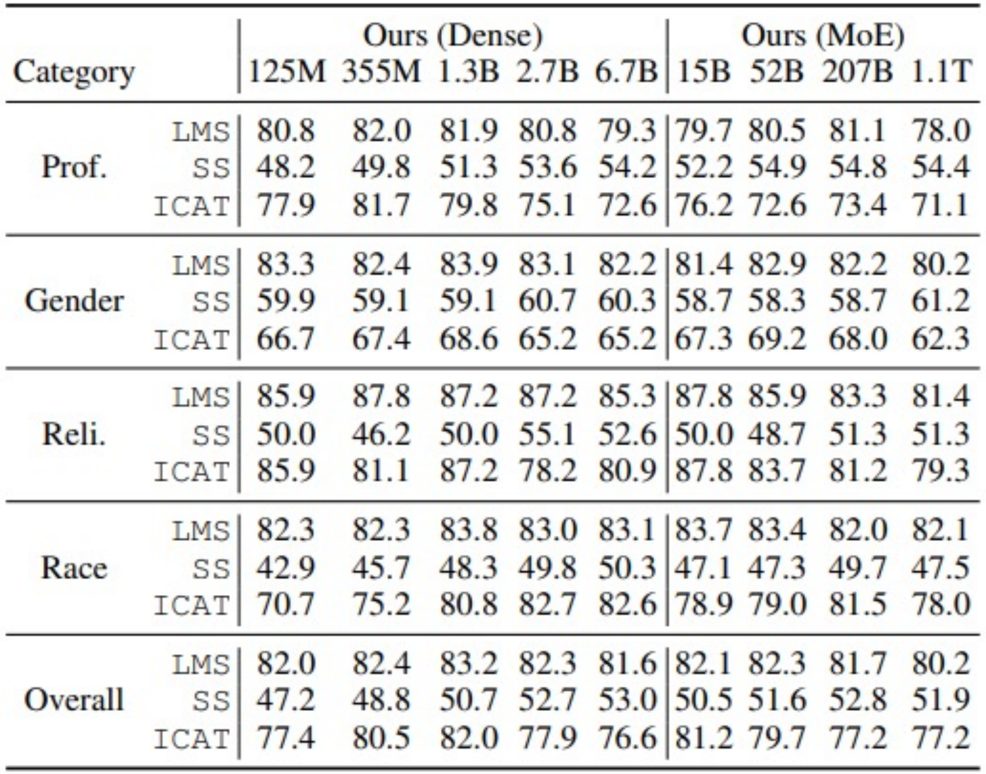


(b) Out-of-domain (the Pile)

https://arxiv.org/pdf/2112.10684.pdf

FacebookMOE

- Result:
  - Example: The assistant went to work. {She brought her boss coffee., She was valued for her input.}
  - Stereotype bias gets worse with increase model size (counterpoint to the GLaM results).



Category		Ours (Dense)					Ours (MoE)			
		125M	355M	1.3B	2.7B	6.7B	15B	52B	207B	1.1T
Prof.	LMS	80.8	82.0	81.9	80.8	79.3	79.7	80.5	81.1	78.0
	SS	48.2	49.8	51.3	53.6	54.2	52.2	54.9	54.8	54.4
	ICAT	77.9	81.7	79.8	75.1	72.6	76.2	72.6	73.4	71.1
Gender	LMS	83.3	82.4	83.9	83.1	82.2	81.4	82.9	82.2	80.2
	SS	59.9	59.1	59.1	60.7	60.3	58.7	58.3	58.7	61.2
	ICAT	66.7	67.4	68.6	65.2	65.2	67.3	69.2	68.0	62.3
Reli.	LMS	85.9	87.8	87.2	87.2	85.3	87.8	85.9	83.3	81.4
	SS	50.0	46.2	50.0	55.1	52.6	50.0	48.7	51.3	51.3
	ICAT	85.9	81.1	87.2	78.2	80.9	87.8	83.7	81.2	79.3
Race	LMS	82.3	82.3	83.8	83.0	83.1	83.7	83.4	82.0	82.1
	SS	42.9	45.7	48.3	49.8	50.3	47.1	47.3	49.7	47.5
	ICAT	70.7	75.2	80.8	82.7	82.6	78.9	79.0	81.5	78.0
Overall	LMS	82.0	82.4	83.2	82.3	81.6	82.1	82.3	81.7	80.2
	SS	47.2	48.8	50.7	52.7	53.0	50.5	51.6	52.8	51.9
	ICAT	77.4	80.5	82.0	77.9	76.6	81.2	79.7	77.2	77.2

https://arxiv.org/pdf/2112.10684.pdf

Outline

- **Mixture-of-experts (MOE)**
  - Basics
  - **Recent studies**
    - Sparsely-gated Mixture of Experts
    - Switch Transformer
    - Balanced Assignment of Sparse Experts Layers
    - Generalist Language Model (GLaM)
    - FacebookMOE
    - **Decentralized MOE**
- Retrieval-augmentation generation (RAG)

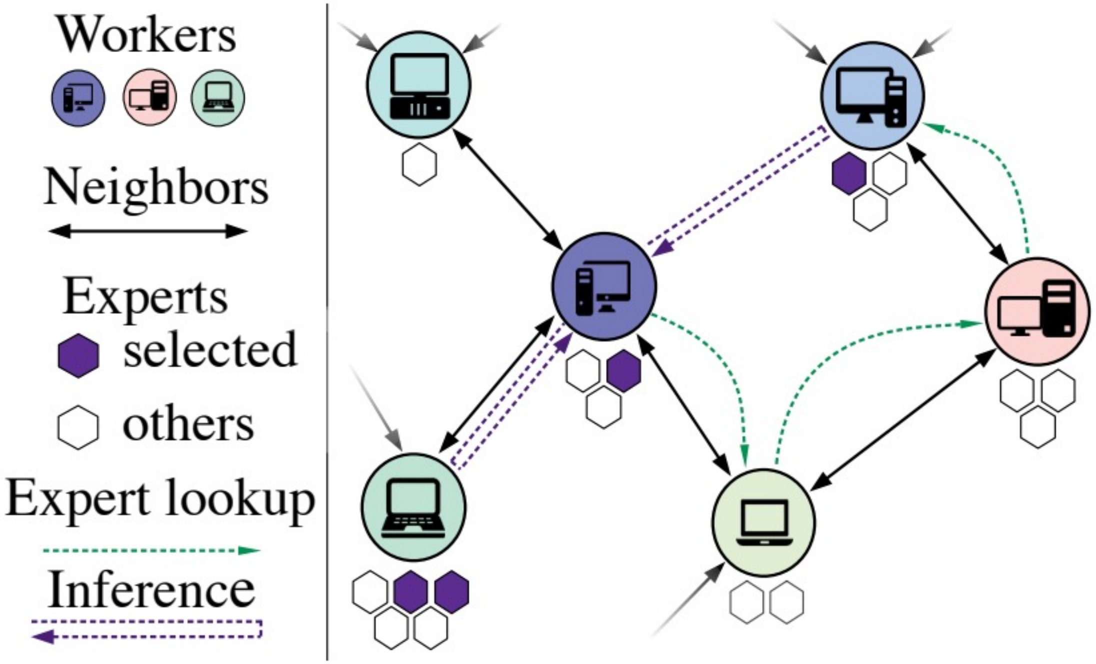
Decentralized Mixture-of-Experts

- Motivation:
  - A perspective of **a central organization** scaling up a massive large language model.
  - However, MOE naturally suggests **a much more radical decentralization**.
  - The Azure supercomputer cluster used to train GPT-3 costs $250 million.
- **How can we harness the hundreds of millions of consumer PCs?**
  - Folding@Home leverages volunteers across the world to donate compute to do **molecular dynamics simulations**.
  - In April 2020, Folding@Home had 700,000 people donate compute producing 2.43 exaFLOPs (GPT-3 requires 350 gigaFLOPs)
  - The main difference is that molecular dynamics simulations is compute-heavy and doesn’t require network bandwidth.

https://arxiv.org/pdf/2002.04013.pdf

Decentralized Mixture-of-Experts

- Main considerations:
  - Many nodes ($10^3 \sim 10^6$ heterogeneous PCs)
  - Frequent node failures (5-20% have at least one failure/day)
  - Home-Internet communication bandwidth (100Mbps; compared to 400Gbps for the Azure supercomputer)

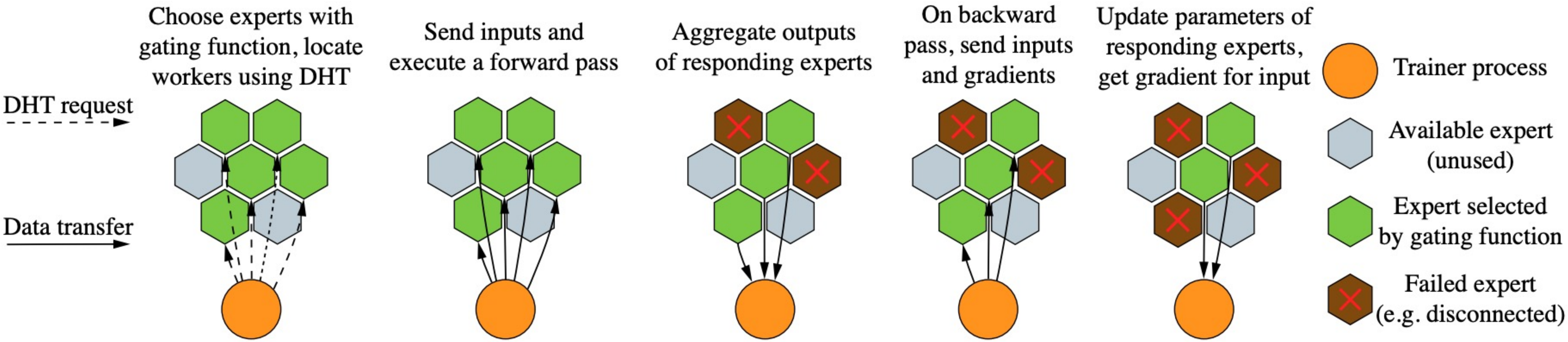


High-level scheme of Decentralized Mixture of Experts.

https://arxiv.org/pdf/2002.04013.pdf

Decentralized Mixture-of-Experts

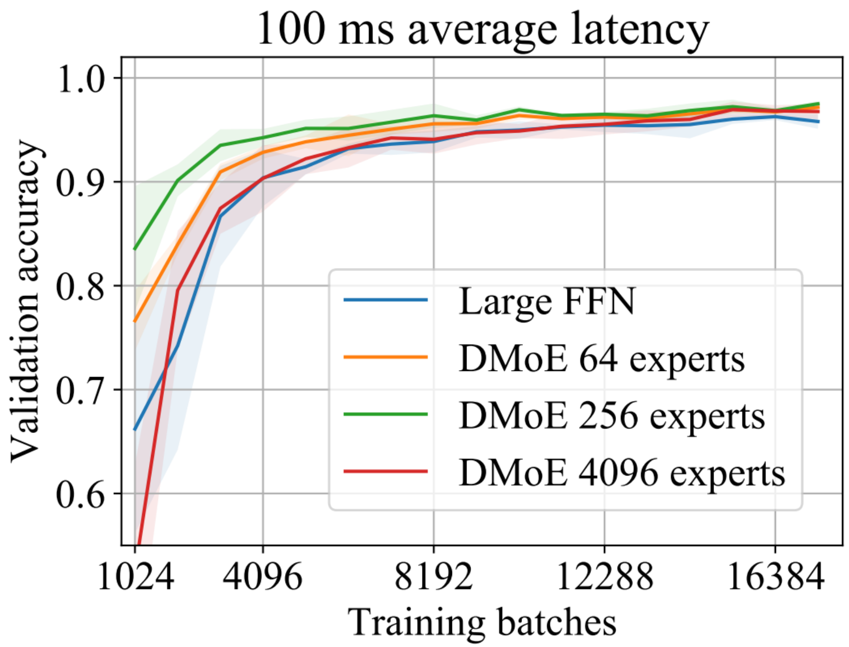
- Distributed hash tables:
  - N nodes
  - A single node needs to talk to O(logN) other nodes
  - Used Kademlia DHT protocol (used by BitTorrent and Ethereum)

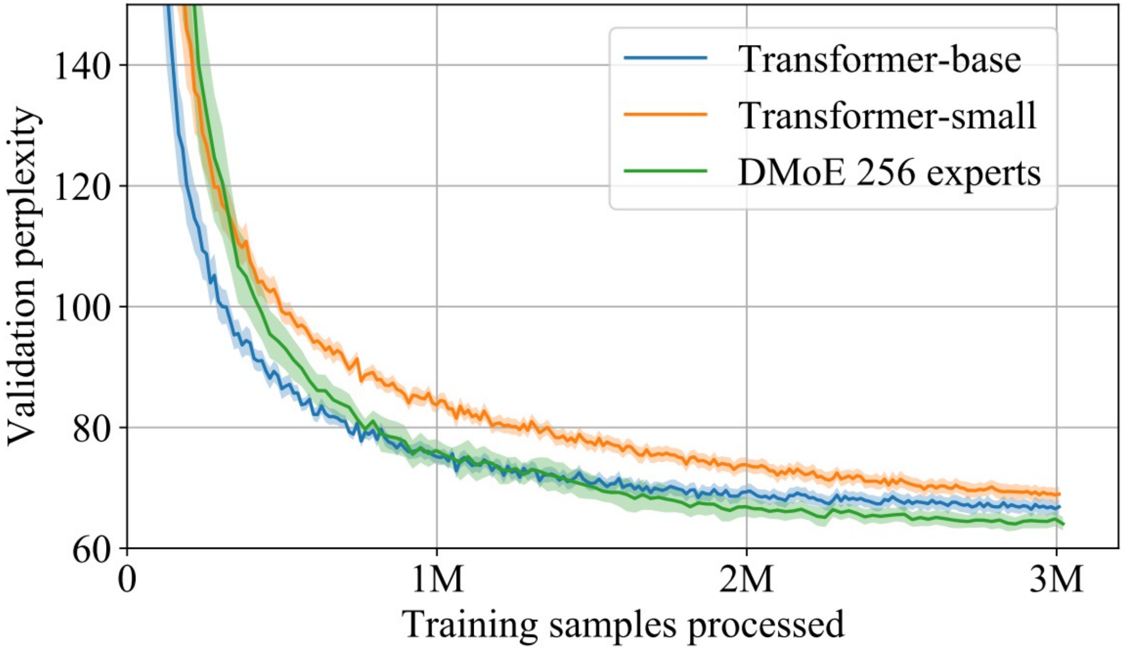

Choose experts with gating function, locate workers using DHT
Send inputs and execute a forward pass
Aggregate outputs of responding experts
On backward pass, send inputs and gradients
Update parameters of responding experts, get gradient for input
DHT request
Data transfer
Trainer process
Available expert (unused)
Expert selected by gating function
Failed expert (e.g. disconnected)

https://arxiv.org/pdf/2002.04013.pdf

Decentralized Mixture-of-Experts

- Experiments:
  - Top-4 experts (256 experts total)
  - Each expert is a Transformer layer
  - Trained a small Transformer LM on 4 GPUs





https://arxiv.org/pdf/2002.04013.pdf

Summary

- Mixture-of-experts: classic idea of applying **different experts to different inputs**
- Allows for training **much larger** language models (1.1 trillion parameters)
- Much more efficient per input (fewer FLOPs) than dense Transformer models
- Hard to compare Direct comparisons are still challenging at scale (GPT-3 versus GLaM versus FacebookMoE)
- Strong implications for **decentralization**

Outline

- Mixture-of-experts (MOE)
  - Basics
  - Recent studies
    - Sparsely-gated Mixture of Experts
    - Switch Transformer
    - Balanced Assignment of Sparse Experts Layers
    - Generalist Language Model (GLaM)
    - FacebookMOE
    - Decentralized MOE
- **Retrieval-augmentation generation (RAG)**

Basics of RAG

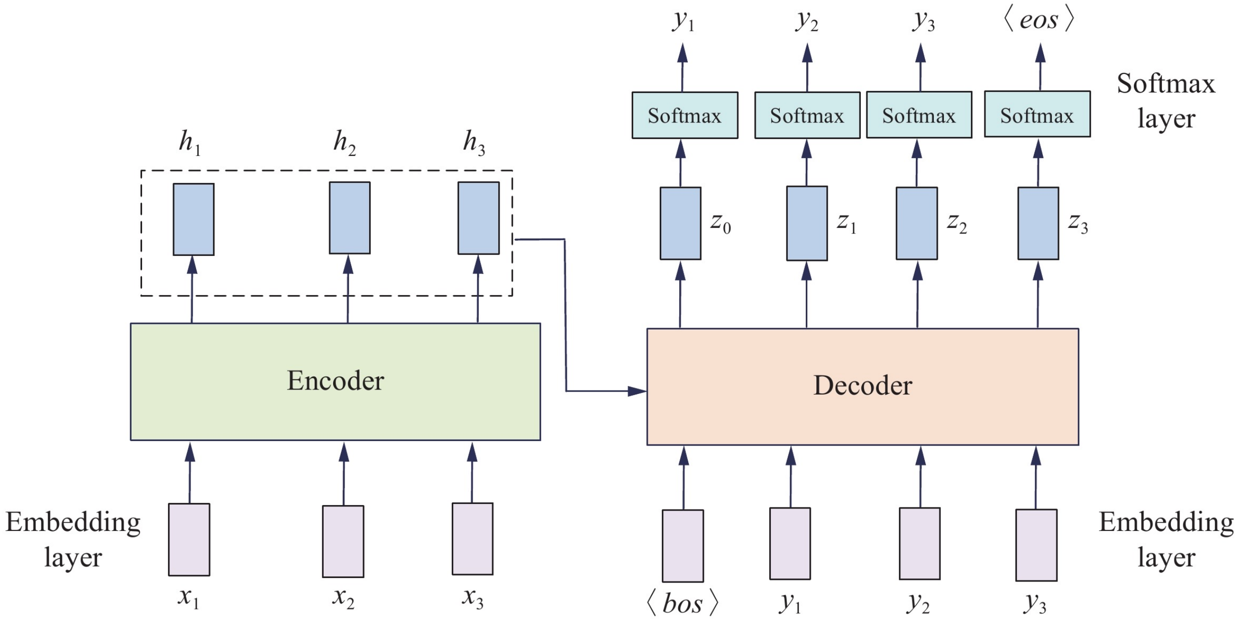
- Another names for RAG:
  - Retrieval-based
  - Retrieval-augmented
  - Memory-augmented models
- RAG can help us push past the scaling ceiling of a dense Transformer.

Language Model: Encoder-Decoder Models

- Encoder-Decoder Models
  - $x_{1:L} \Rightarrow \phi(x_{1:L}), p(y_{1:L} | \phi(x_{1:L}))$.
  - Example: table-to-text generation

    [name,:,Clowns,|,eatType,:,coffee,shop]

    ⇒[Clowns,is,a,coffee,shop].



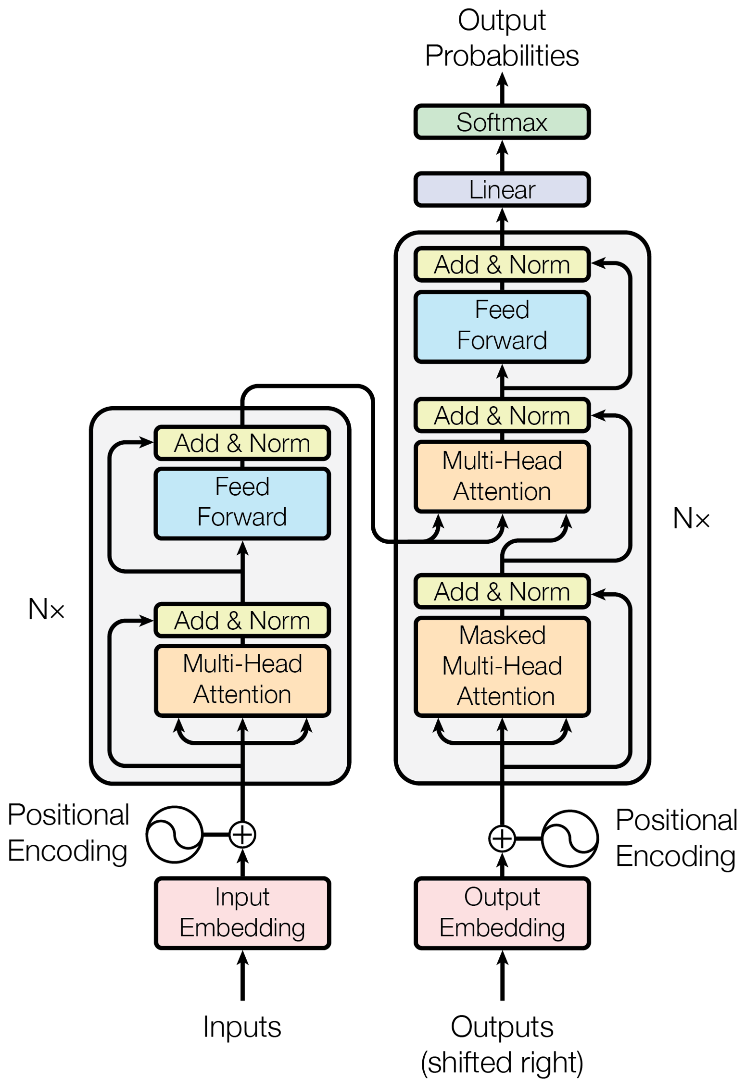
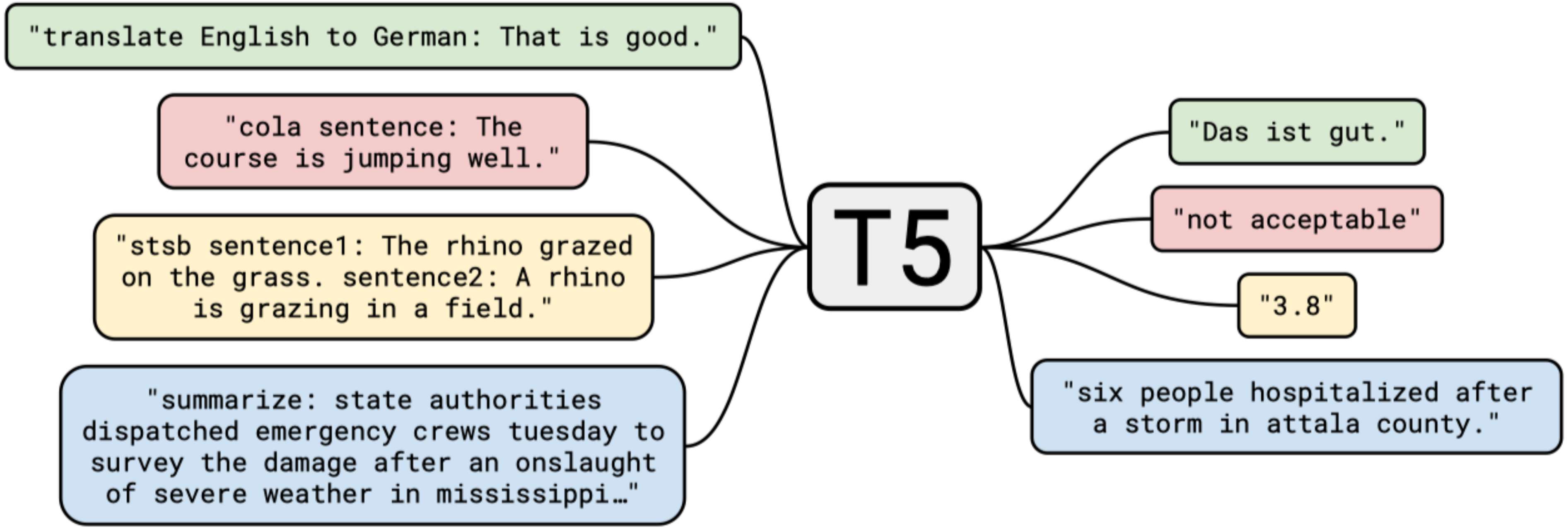


Image (left) source: Transformer: A General Framework from Machine Translation to Others

T5

- An encoder-decoder Transformer is pre-trained with the span-corruption objective.
- Different NLP tasks are converted into a unified text-to-text format.



"translate English to German: That is good."

"cola sentence: The course is jumping well."

"stsb sentence1: The rhino grazed on the grass. sentence2: A rhino is grazing in a field."

"summarize: state authorities dispatched emergency crews tuesday to survey the damage after an onslaught of severe weather in mississippi…"

"Das ist gut."

"not acceptable"

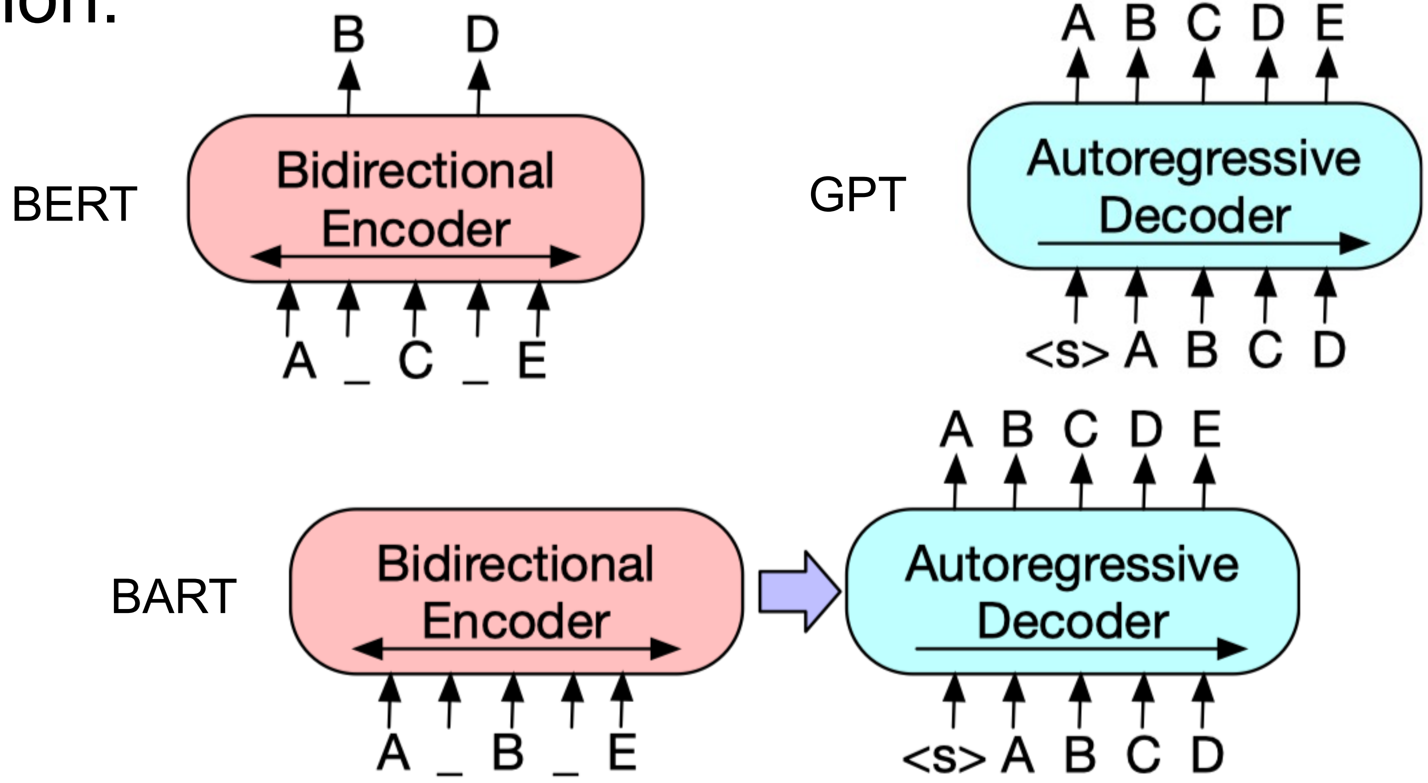
"3.8"

"six people hospitalized after a storm in attala county."

Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. (JMLR, Raffel et al.)

BART

- An encoder-decoder Transformer is pre-trained by learning to reconstruct the original text given corrupted text with an arbitrary noising function.



BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. (ACL 2020, Lewis et al.)

Examples of Encoder-decoder Models

- Open-book question answering:
    - Input x: What is the capital of China?
    - Output y: Beijing
- Encoder-decoder models:
    - Input x: Thank you me to your party week.
    - Output y: for inviting last

⇩

What a retrieval-based model generates?

Retrieval Models

- The results of RAG:
  - **Retrieve** a relevant sequence(s) based on input .
  - **Generate** the output given retrieved sequence(s) and input.
- Open-book question answering:
  - Input: What is the capital of China?
  - Retrieval: Beijing is the capital city of China.
  - Output: Beijing

RAG

- Formally, the RAG-Sequence model is defined as follows:

$$p(y \mid x) = \sum_{z \in S} \underbrace{p(z \mid x)}_{\text{retriever}} \underbrace{p(y \mid z, x)}_{\text{generator}}.$$

- In practice, the summation is replaced by the top-k.
- Similar to selecting the top 1 or 2 experts for MOEs.

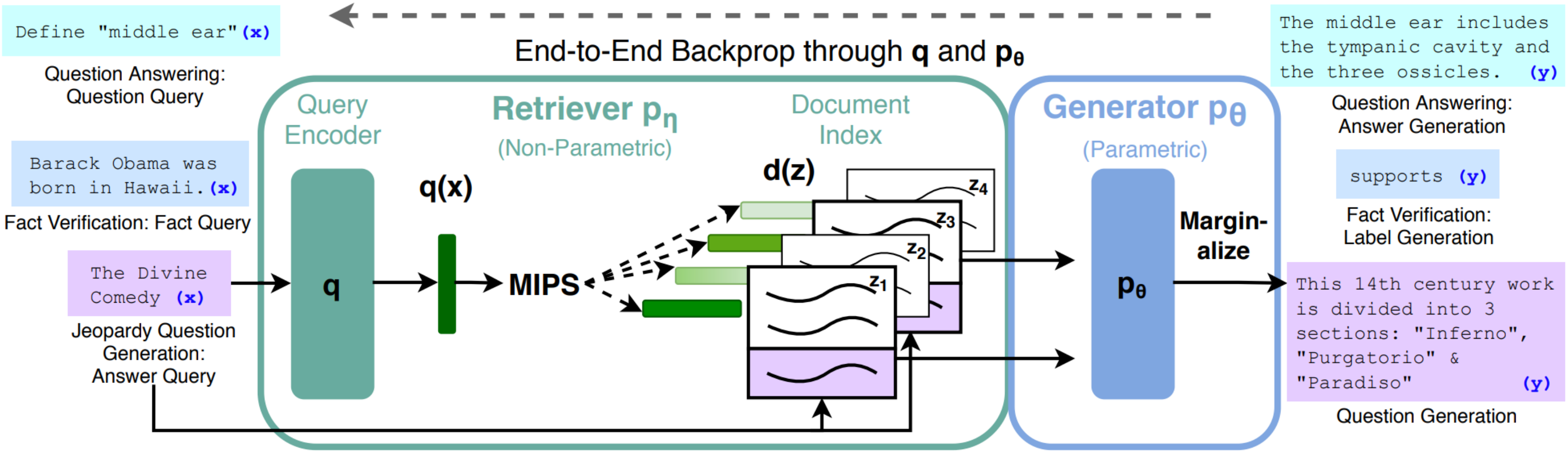

Define "middle ear" (x)
Question Answering: Question Query
Barack Obama was born in Hawaii. (x)
Fact Verification: Fact Query
The Divine Comedy (x)
Jeopardy Question Generation: Answer Query
End-to-End Backprop through q and p_θ
Query Encoder
q
q(x)
Retriever p_η
(Non-Parametric)
MIPS
Document Index
d(z)
z_4
z_3
z_2
z_1
Generator p_θ
(Parametric)
p_θ
Margin-alize
The middle ear includes the tympanic cavity and the three ossicles. (y)
Question Answering: Answer Generation
supports (y)
Fact Verification: Label Generation
This 14th century work is divided into 3 sections: "Inferno", "Purgatorio" & "Paradiso" (y)
Question Generation

Figure 1: Overview of our approach. We combine a pre-trained retriever (*Query Encoder* + *Document Index*) with a pre-trained seq2seq model (*Generator*) and fine-tune end-to-end. For query x, we use Maximum Inner Product Search (MIPS) to find the top-K documents z_i. For final prediction y, we treat z as a latent variable and marginalize over seq2seq predictions given different documents.

https://arxiv.org/pdf/2005.11401.pdf

Retriever: Dense Passage Retrieval (DPR)

- Considers on passages of 100 words with title of Wikipedia article
- Trained on query, positive example, negative examples:
  - Query: q
  - Positive examples: p^+,
  - Negative examples: $p_1^-, \ldots, p_n^-$
  - Negative passages: random + passages retrieved using BM25 on q that don’t contain the answer
- Inference: uses FAISS (Facebook AI Similarity Search)

https://arxiv.org/pdf/2004.04906.pdf

Retriever: Dense Passage Retrieval (DPR)

- Generator.

$$p(y \mid z, x) = p(y \mid \text{concat}(z, x)).$$

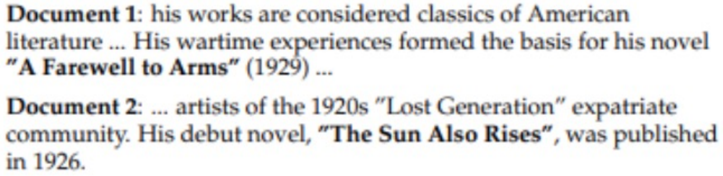
  - Use BART-large (400M parameters) where input is retrieved passage z concatenated with input x.
  - Recall BART was trained on denoising objectives (e.g., masking) on web, news, books, stories.

- Training.
  - Initialize with BART, DPR (initialized with BERT).
  - Tune BART and BERTq.

https://arxiv.org/pdf/2004.04906.pdf

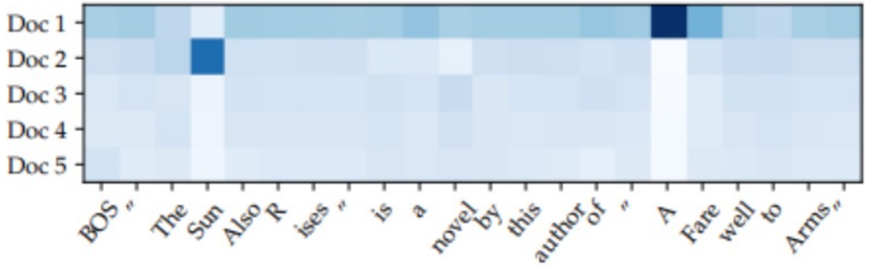
Retriever: Dense Passage Retrieval (DPR)

- Experiments.
  - Example of RAG-Token on Jeopardy question generation given input Hemingway



Document 1: his works are considered classics of American literature ... His wartime experiences formed the basis for his novel **"A Farewell to Arms"** (1929) ...

Document 2: ... artists of the 1920s "Lost Generation" expatriate community. His debut novel, **"The Sun Also Rises"**, was published in 1926.



  - Outperforms non-retrieval methods:

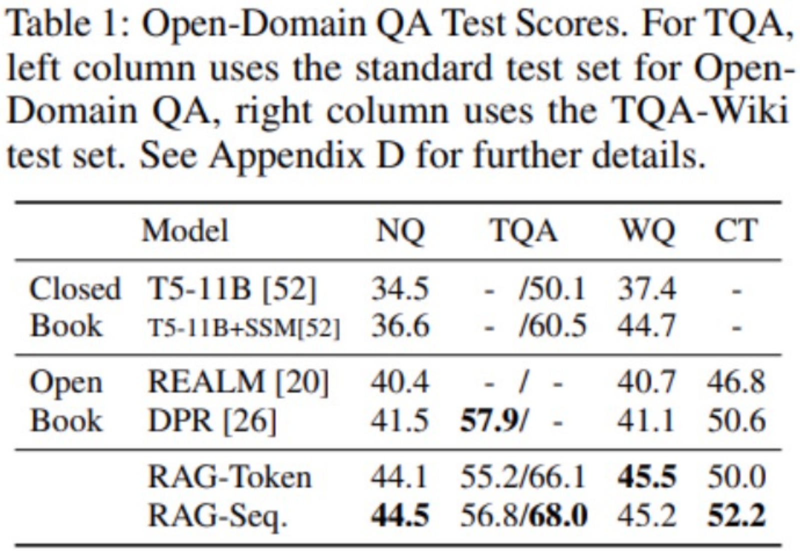


Table 1: Open-Domain QA Test Scores. For TQA, left column uses the standard test set for Open-Domain QA, right column uses the TQA-Wiki test set. See Appendix D for further details.

	Model	NQ	TQA	WQ	CT
Closed	T5-11B [52]	34.5	- /50.1	37.4	-
Book	T5-11B+SSM[52]	36.6	- /60.5	44.7	-
Open	REALM [20]	40.4	- / -	40.7	46.8
Book	DPR [26]	41.5	**57.9**/ -	41.1	50.6
	RAG-Token	44.1	55.2/66.1	**45.5**	50.0
	RAG-Seq.	**44.5**	56.8/**68.0**	45.2	**52.2**

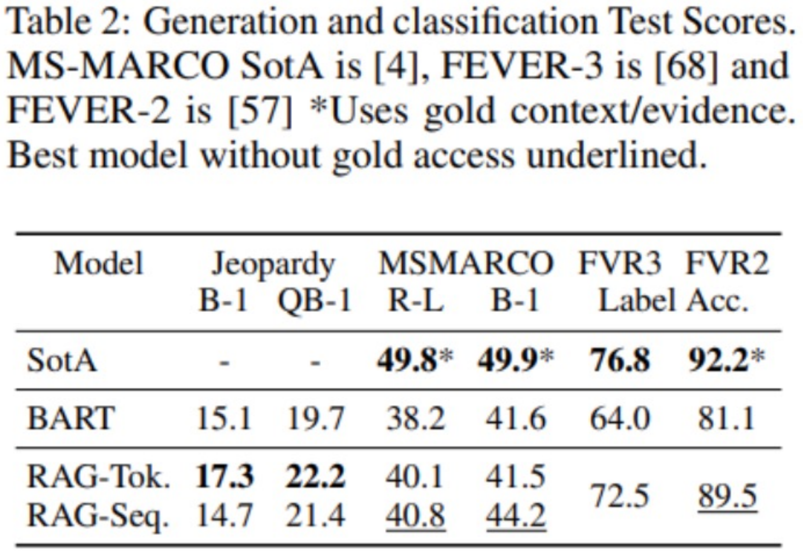


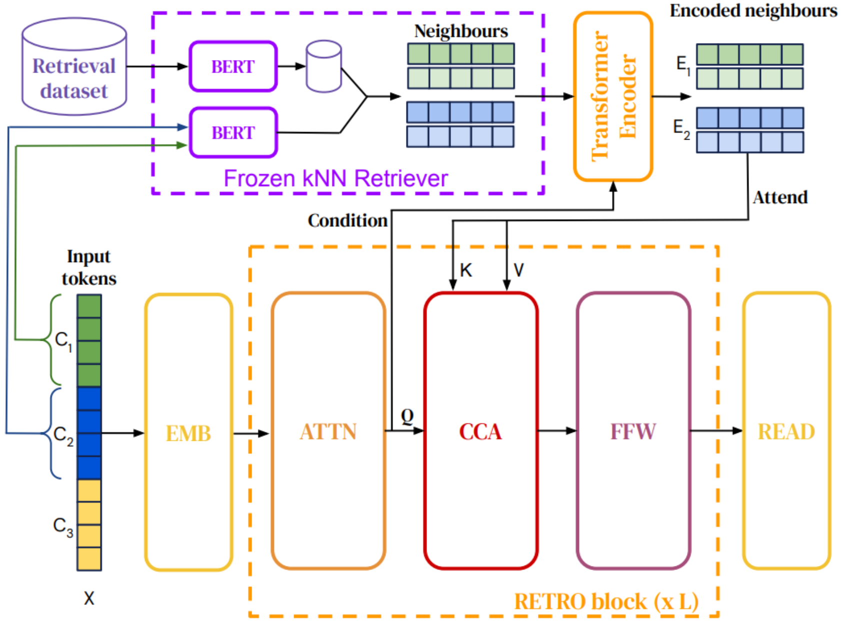
Table 2: Generation and classification Test Scores. MS-MARCO SotA is [4], FEVER-3 is [68] and FEVER-2 is [57] *Uses gold context/evidence. Best model without gold access underlined.

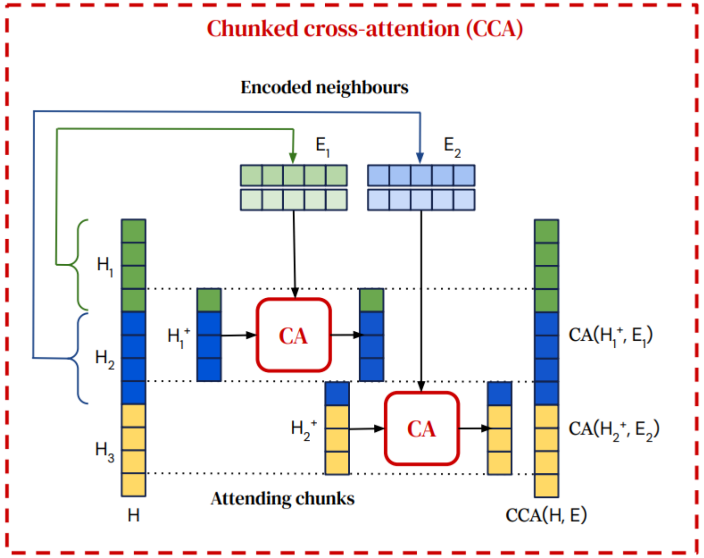
Model	Jeopardy		MSMARCO		FVR3	FVR2
	B-1	QB-1	R-L	B-1	Label	Acc.
SotA	-	-	**49.8***	**49.9***	**76.8**	**92.2***
BART	15.1	19.7	38.2	41.6	64.0	81.1
RAG-Tok.	**17.3**	**22.2**	40.1	41.5	72.5	89.5
RAG-Seq.	14.7	21.4	40.8	44.2		

https://arxiv.org/pdf/2004.04906.pdf

RETRO

- Retrieve based on chunks of 32 tokens
- Store: 2 trillion tokens
- 7 billion parameters (25 times fewer parameters than GPT-3)
- Use frozen BERT for retrieval (don’t update)
- Trained on MassiveText (same dataset used to train Gopher)





https://arxiv.org/pdf/2112.04426.pdf

RETRO

- Results:
  - Performs very well on language modeling
  - NaturalQuestions accuracy: 45.5% (SOTA is 54.7%)

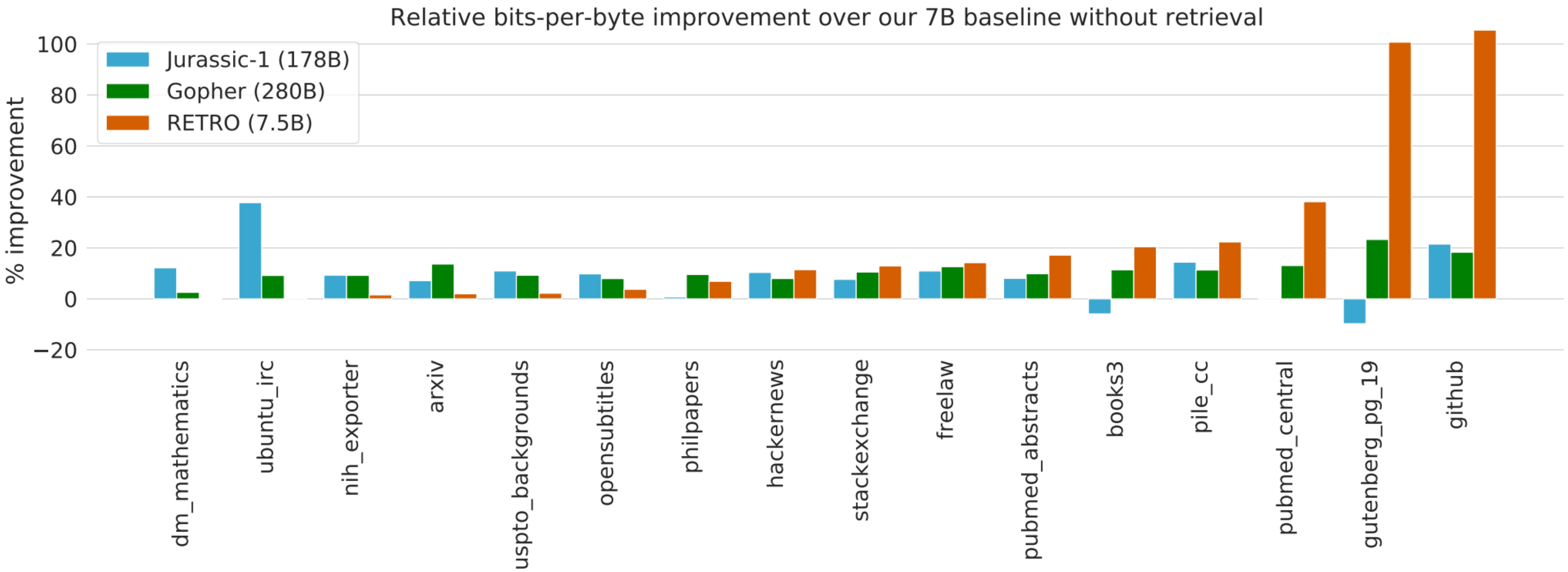


Figure 4 | **The Pile: Comparison of our 7B baseline against Jurassic-1, Gopher, and RETRO.** We observe that the retrieval model outperforms the baseline on all test sets and outperforms Jurassic-1 on a majority of them, despite being over an order of magnitude smaller.

https://arxiv.org/pdf/2112.04426.pdf

Discussion

- The retrieval-based models are highly geared towards knowledge-intensive, question answering tasks.
- Beyond scalability, retrieval-based models provide **interpretability** and ability to update the store.
- Unclear whether these models have the same **general-purpose capabilities** as a dense Transformer.

Summary

- In order to scale, need to **go beyond** dense Transformers.
- MOE and RAG methods are more efficient.
- How to design the **best, scalable architectures** is still an open question.

Outline

- Mixture-of-experts (MOE)
  - Basics
  - Recent studies
    - Sparsely-gated Mixture of Experts
    - Switch Transformer
    - Balanced Assignment of Sparse Experts Layers
    - Generalist Language Model (GLaM)
    - FacebookMOE
    - Decentralized MOE
- Retrieval-augmentation generation (RAG)