# RAG, LORA, QLORA, Summarisation

## Intro to NLP

**Rahul Mishra**
IIIT-Hyderabad
Apr 16 & 19, 2024

# LLMs are superb

General purpose seq2seq models are getting really powerful

- Capture world knowledge in parameters
- Strong results on loads of tasks
- Applicable for almost everything!

But

- Hallucinate
- Struggle to access and apply knowledge
- Difficult to update

# Retrieval is also superb

Externally-retrieved knowledge/text is useful for a huge variety of NLP tasks

- Precise and accurate knowledge access mechanism
- Trivial to update at test time
- **Dense retrieval** starting to outperform traditional IR.

But often limited applicability because usually:

- Need retrieval supervision
- Or “heuristics”-based retrieval (e.g. TF-IDF)
- Need some (usually task specific) way to integrate into downstream models

How can  we combine the strengths of explicit knowledge retrieval and Seq2Seq?

# Retrieval Augmented Generation (RAG)

**Jointly** learn to **retrieve** and **generate** end2end.

**Latent retrieval** - no labels needed for retrieved docs

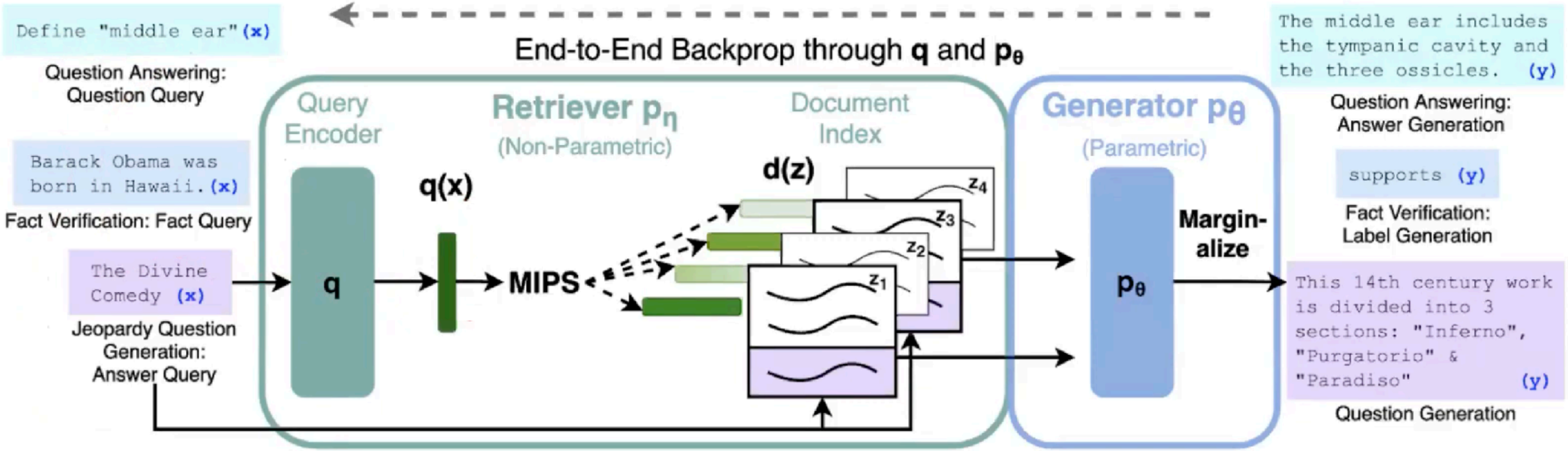
**General recipe** for any seq2seq task

Needs 3 things:

- A (pretrained) generator model P(y|...) e.g. **BART**, GPT2, T5
- A (pretrained) retriever model P(z|x) e.g. **DPR**, ICT
- An indexed KB of text documents Z e.g. **Wikipedia**, CommonCrawl, tweets, ++

RAG models combine **parametric** and ***non-parametric*** memory and work well for **knowledge intensive tasks**

# RAG


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 pθ
Query Encoder
Retriever pη
(Non-Parametric)
Document Index
q
q(x)
MIPS
d(z)
z1
z2
z3
z4
Generator pθ
(Parametric)
pθ
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

# RAG

$$p_{\text{RAG-Sequence}}(y|x) \approx \sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) p_\theta(y|x,z) = \sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z|x) \prod_i^N p_\theta(y_i|x,z,y_{1:i-1})$$

Retriever

Generator

## *RAG-Token* Model

$$p_{\text{RAG-Token}}(y|x) \approx \prod_i^N \sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z_i|x) p_\theta(y_i|x,z_i,y_{1:i-1})$$

*Both trained by directly minimising -log p(y|x)*

# Retriever: Dense Passage Retrieval

$$p_\eta(z|x) \propto \exp\left(\mathbf{d}(z)^\top \mathbf{q}(x)\right) \qquad \mathbf{d}(z) = \mathrm{BERT}_d(z), \quad \mathbf{q}(x) = \mathrm{BERT}_q(x)$$

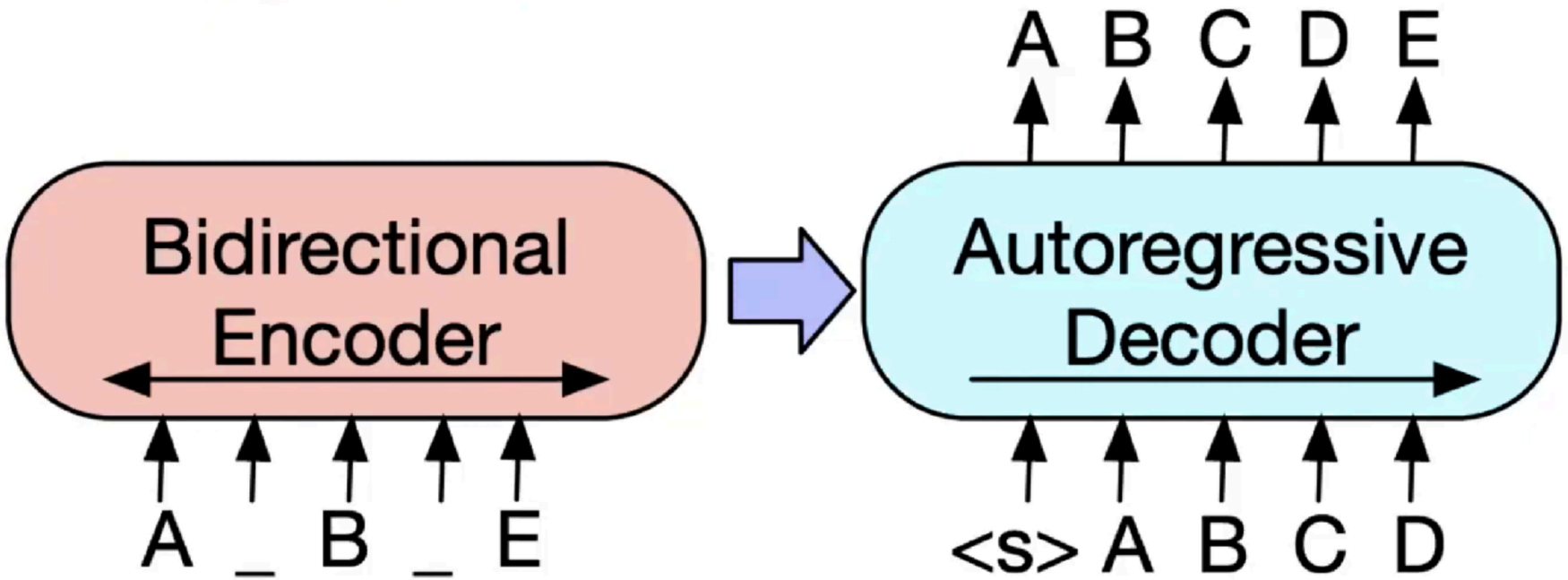
Bi-Encoder Architecture

Document Encoder

Query Encoder

Get a pretrained **Bi-Encoder**
Encode Wikipedia Documents Once with **Document Encoder**
Finetune **Query Encoder** end-to-end with RAG

# Generator: BART




- Pretrained Seq2seq model
- RAG Simply concatenates Latent Document z to Input x

# Decoding in RAG

## *RAG-Token* Model

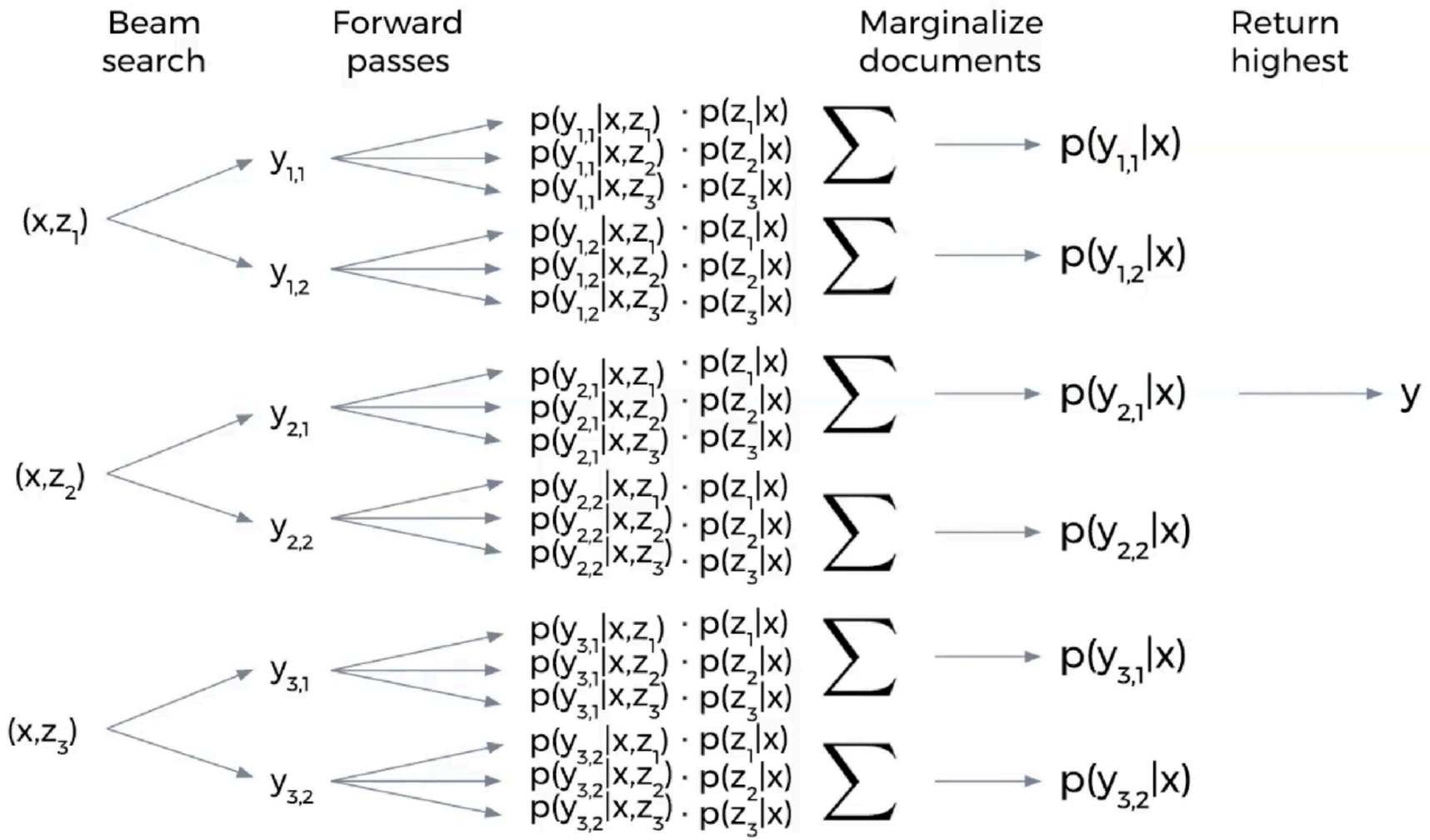
Standard Beam Search with transition probability:

$$p'_\theta(y_i|x, y_{1:i-1}) = \sum_{z \in \text{top-}k(p(\cdot|x))} p_\eta(z_i|x) p_\theta(y_i|x, z_i, y_{1:i-1})$$

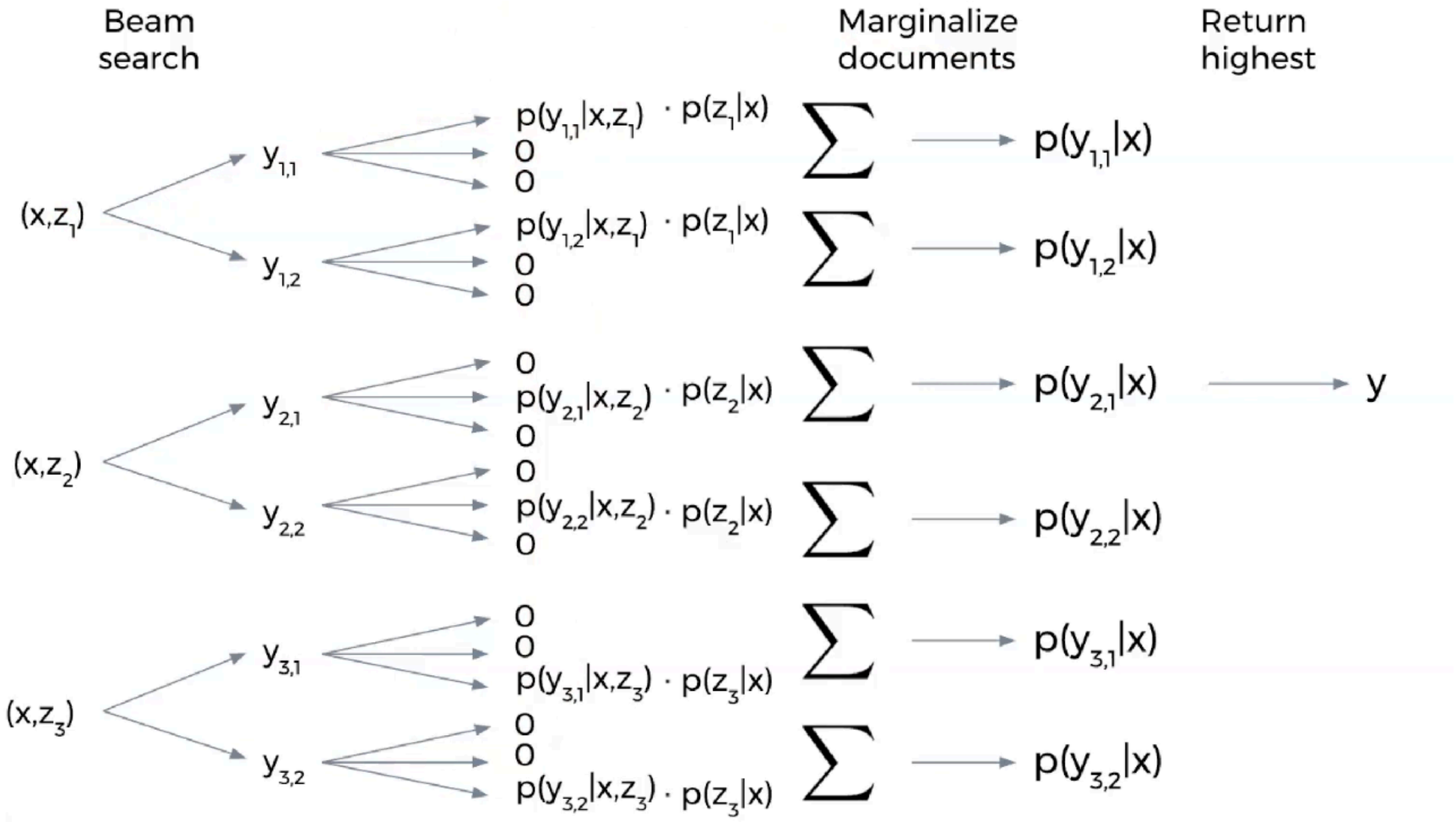
## *RAG-Sequence* Model

...

# RAG


Beam search
Forward passes
Marginalize documents
Return highest
$(x,z_1)$
$(x,z_2)$
$(x,z_3)$
$y_{1,1}$
$y_{1,2}$
$y_{2,1}$
$y_{2,2}$
$y_{3,1}$
$y_{3,2}$
$p(y_{1,1}|x,z_1) \cdot p(z_1|x)$
$p(y_{1,1}|x,z_2) \cdot p(z_2|x)$
$p(y_{1,1}|x,z_3) \cdot p(z_3|x)$
$p(y_{1,2}|x,z_1) \cdot p(z_1|x)$
$p(y_{1,2}|x,z_2) \cdot p(z_2|x)$
$p(y_{1,2}|x,z_3) \cdot p(z_3|x)$
$p(y_{2,1}|x,z_1) \cdot p(z_1|x)$
$p(y_{2,1}|x,z_2) \cdot p(z_2|x)$
$p(y_{2,1}|x,z_3) \cdot p(z_3|x)$
$p(y_{2,2}|x,z_1) \cdot p(z_1|x)$
$p(y_{2,2}|x,z_2) \cdot p(z_2|x)$
$p(y_{2,2}|x,z_3) \cdot p(z_3|x)$
$p(y_{3,1}|x,z_1) \cdot p(z_1|x)$
$p(y_{3,1}|x,z_2) \cdot p(z_2|x)$
$p(y_{3,1}|x,z_3) \cdot p(z_3|x)$
$p(y_{3,2}|x,z_1) \cdot p(z_1|x)$
$p(y_{3,2}|x,z_2) \cdot p(z_2|x)$
$p(y_{3,2}|x,z_3) \cdot p(z_3|x)$
$\sum$
$p(y_{1,1}|x)$
$p(y_{1,2}|x)$
$p(y_{2,1}|x)$
$p(y_{2,2}|x)$
$p(y_{3,1}|x)$
$p(y_{3,2}|x)$
y

# RAG


Beam search
Marginalize documents
Return highest
$(x,z_1)$
$(x,z_2)$
$(x,z_3)$
$y_{1,1}$
$y_{1,2}$
$y_{2,1}$
$y_{2,2}$
$y_{3,1}$
$y_{3,2}$
$p(y_{1,1}|x,z_1) \cdot p(z_1|x)$
$p(y_{1,2}|x,z_1) \cdot p(z_1|x)$
$p(y_{2,1}|x,z_2) \cdot p(z_2|x)$
$p(y_{2,2}|x,z_2) \cdot p(z_2|x)$
$p(y_{3,1}|x,z_3) \cdot p(z_3|x)$
$p(y_{3,2}|x,z_3) \cdot p(z_3|x)$
0
$\sum$
$p(y_{1,1}|x)$
$p(y_{1,2}|x)$
$p(y_{2,1}|x)$
$p(y_{2,2}|x)$
$p(y_{3,1}|x)$
$p(y_{3,2}|x)$
y

# Applications

RAG can be applied to **any task with input and output sequences**.

Focus on tasks with a clear need for precisely accessing knowledge

**Open-domain QA**
Natural Questions, TriviaQA, WebQuestions, CuratedTREC
**Abstractive open-domain QA**:
"Open" MS MARCO
**Question Generation:**
Jeopardy questions
**Fact Verification:**
FEVER

# Results

| | Model | NQ | TQA | WQ | CT |
|---|---|---|---|---|---|
| Closed Book | T5-11B [49] | 34.5 | - /50.1 | 37.4 | - |
| | T5-11B+SSM[49] | 36.6 | - /60.5 | 44.7 | - |
| Open Book | REALM [19] | 40.4 | - / - | 40.7 | 46.8 |
| | DPR [23] | 41.5 | **57.9**/ - | 41.1 | 50.6 |
| | RAG-Token | 44.1 | 55.2/66.1 | **45.5** | 50.0 |
| | RAG-Seq. | **44.5** | 56.1/**68.0** | 45.2 | **52.2** |

Question: who sings does he love me with reba?

BART: The Beatles
RAG: Linda Kaye Davies
GOLD: Linda Davies

# Jeopardy Question Generation

Input: Washington

Gold: Florida's in the southeast corner of the 48 contiguous states; this state is in the northwest corner

BART: This state has the largest number of counties in the U.S.

RAG: Its the only U.S. state named for a U.S. President

Input: The Divine Comedy

BART: This epic poem by Dante is divided into three parts: the Inferno, The Purgatorio & the Purgatorio

RAG: This 14th Century work is divided into 3 sections: "inferno", "Purgatorio" & "Paradiso"

# Jeopardy Question Generation

- Challenging knowledge intensive generation task
- Unlike other tasks **RAG-Token performs best** here
- Task requires integrating facts from different documents

| Model | Jeopardy B-1 | Jeopardy QB-1 |
|---|---|---|
| SotA | - | - |
| BART | 15.1 | 19.7 |
| RAG-Tok. | **17.3** | **22.2** |
| RAG-Seq. | 14.7 | 21.4 |

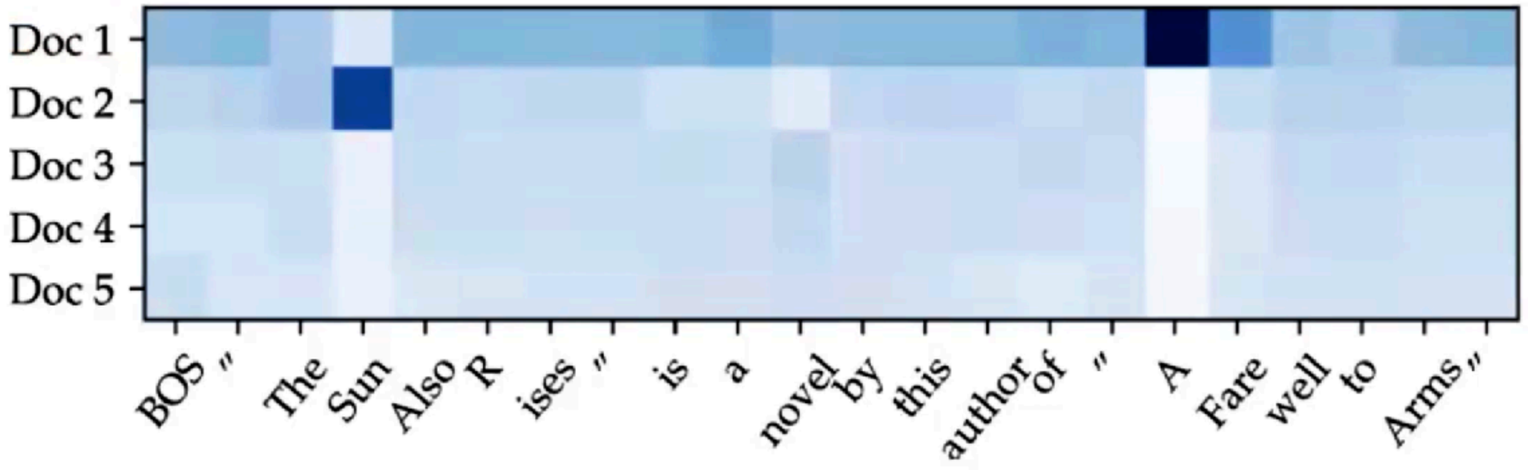
Automatic metrics

| | Factuality | Specificity |
|---|---|---|
| BART better | 7.1% | 16.8% |
| RAG better | **42.7%** | **37.4%** |
| Both good | 11.7% | 11.8% |
| Both poor | 17.7% | 6.9% |
| No majority | 20.8% | 20.1% |

Human evaluation

# Interaction b/w parametric and non-parametric memory

*RAG-Token* document probability $p(z|x,y_{1:t-1})$ for 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

# Interaction b/w parametric and non-parametric memory

Retrieved documents **cue correct responses** from BART:

Feed BART with input `Hemingway` and partial decoding `"The Sun:`

> Completion: `"The Sun also Rises" is a novel by this author of "the Sun Also Rises"`

Feed BART with input `Hemingway` and partial decoding `"The Sun Also Rises" is a novel by this author of "A:`

> Completion: `"The Sun also Rises" is a novel by this author of "A Farewell to Arms"`

# Generation diversity

Table 5: Ratio of distinct to total tri-grams for generation tasks

| | MSMARCO | Jeopardy QGen |
|---|---|---|
| Gold | 89.6% | 90.0% |
| BART | 70.7% | 32.4% |
| RAG-Token | 77.8% | 46.8% |
| RAG-Seq. | 83.5% | 53.8% |

# FEVER: Fact Checking

- RAG can also be used for classification
- 3-way task:
  - 4.3% behind SOTA
  - RAG trained only on (claim, label) pairs
  - SOTA models use complex pipeline and strong retrieval supervision
- 2-way task:
  - 2.7% RoBERTa model using gold evidence sentence at test time.

| Model | FVR3 Label Acc. | FVR2 Acc. |
|---|---|---|
| SotA | **76.8** | **92.2*** |
| BART | 64.0 | 81.1 |
| RAG-Tok. RAG-Seq. | 72.5 | <u>89.5</u> |

# Document index hot swapping

Update model's memory on the fly by swapping the document index

Compare generated answers using **2016** vs. **2018** *Wikipedia* index.

Query RAG models "`who is the {position}`"? for world leaders who have changed between **2016** and **2018**

- E.g., "`Who is the President of Peru?`"
  - **2016** leaders and **2016** index: 70%
  - **2016** leaders and **2018** index: 12%
  - **2018** leaders and **2016** index: 4%
  - **2018** leaders and **2018** index: 68%

# Ablations

Table 6: Ablations on the dev set. As FEVER is a classification task, both RAG models are equivalent

| Model | NQ | TQA | WQ | CT | Jeopardy-QGen | | MSMarco | | FVR-3 | FVR-2 |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Exact Match | | | B-1 | QB-1 | R-L | B-1 | Label Accuracy | |
| RAG-Token-BM25 | 29.7 | 41.5 | 32.1 | 33.1 | 17.5 | 22.3 | 55.5 | 48.4 | **75.1** | **91.6** |
| RAG-Sequence-BM25 | 31.8 | 44.1 | 36.6 | 33.8 | 11.1 | 19.5 | 56.5 | 46.9 | | |
| RAG-Token-Frozen | 37.8 | 50.1 | 37.1 | 51.1 | 16.7 | 21.7 | 55.9 | 49.4 | 72.9 | 89.4 |
| RAG-Sequence-Frozen | 41.2 | 52.1 | 41.8 | 52.6 | 11.8 | 19.6 | 56.7 | 47.3 | | |
| RAG-Token | 43.5 | 54.8 | **46.5** | 51.9 | **17.9** | **22.6** | 56.2 | **49.4** | 74.5 | 90.6 |
| RAG-Sequence | **44.0** | **55.8** | 44.9 | **53.4** | 15.3 | 21.5 | **57.2** | 47.5 | | |

- Retrieval-finetuning always helps, even when using supervised
- MIPS retrieval usually outperforms BM25 (FEVER is the exception)

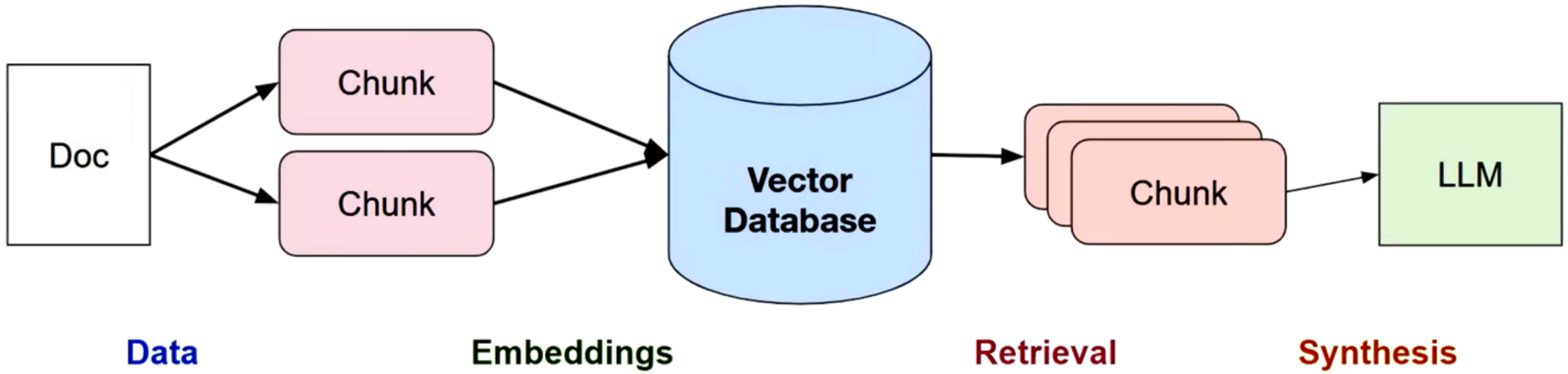
- Robust Retrieval
- Multi-hop reasoning RAG
- Better retriever initialization
- Multimodal RAGs
- End2End Pretrained RAGs

# Challenges with RAG

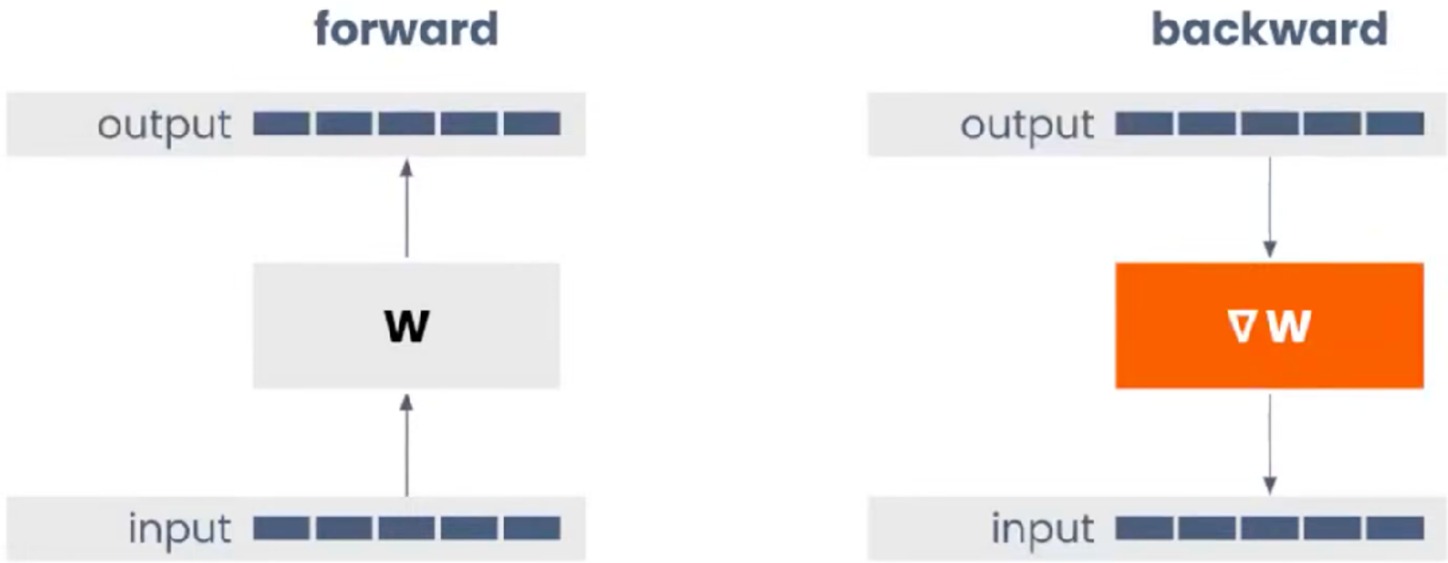
- Bad Retrieval
  - **Low Precision:** Not all chunks in retrieved set are relevant
    - Hallucination + Lost in the Middle Problems
  - **Low Recall:** Now all relevant chunks are retrieved.
    - Lacks enough context for LLM to synthesize an answer
  - **Outdated information:** The data is redundant or out of date.
- Bad Response Generation
  - **Hallucination:** Model makes up an answer that isn't in the context.
  - **Irrelevance:** Model makes up an answer that doesn't answer the question.
  - **Toxicity/Bias:** Model makes up an answer that's harmful/offensive.

# Challenges with RAG

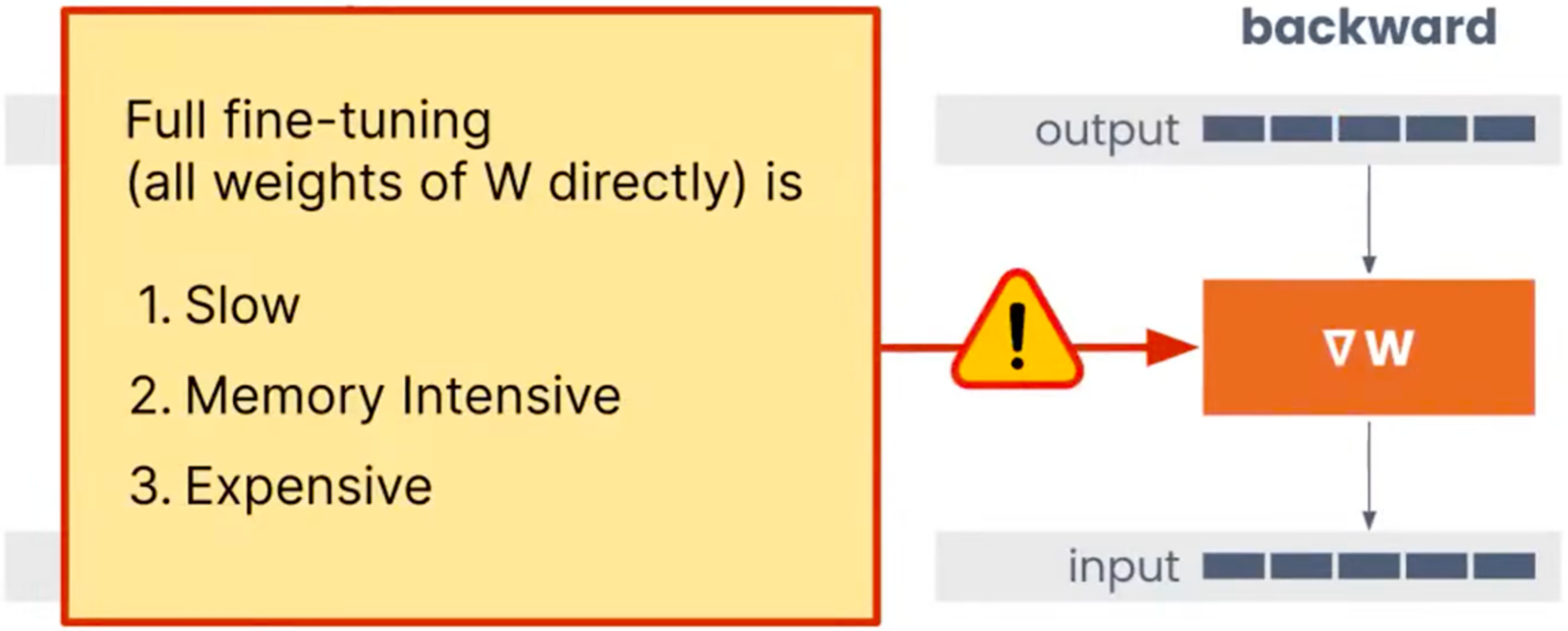
- **Data:** Can we store additional information beyond raw text chunks?
- **Embeddings:** Can we optimize our embedding representations?
- **Retrieval:** Can we do better than top-k embedding lookup?
- **Synthesis:** Can we use LLMs for more than generation?



# Parameter Efficient Training: LORA

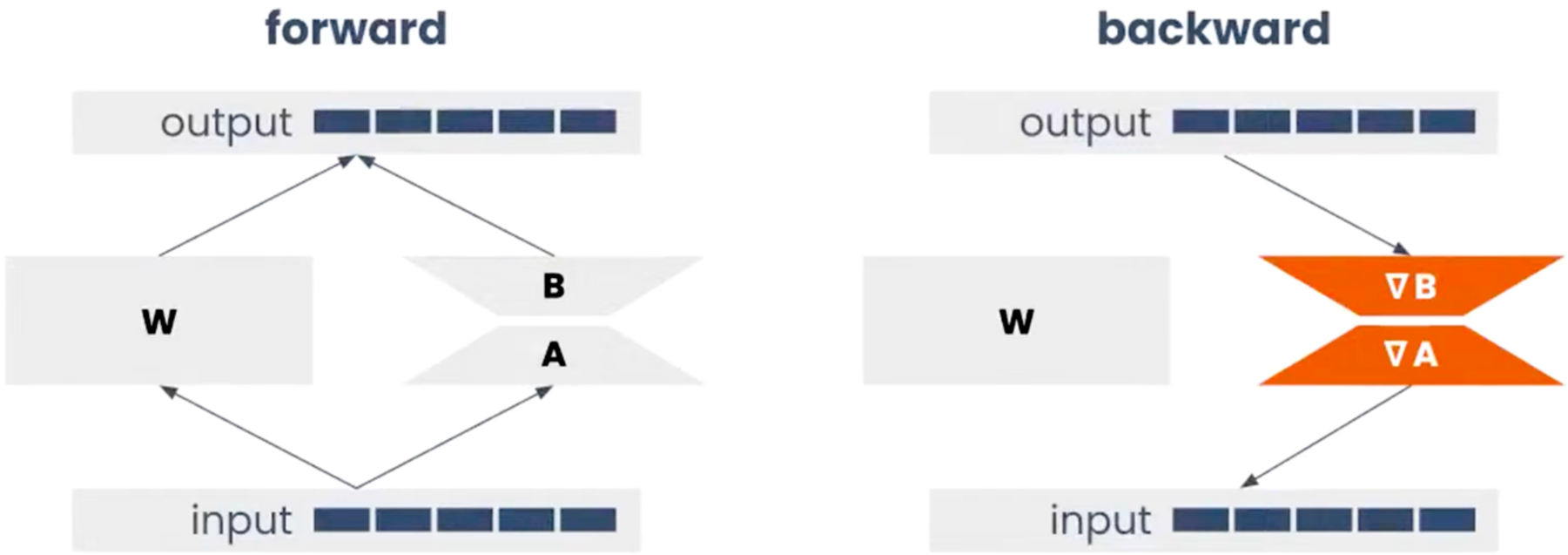


# Parameter Efficient Training: LORA



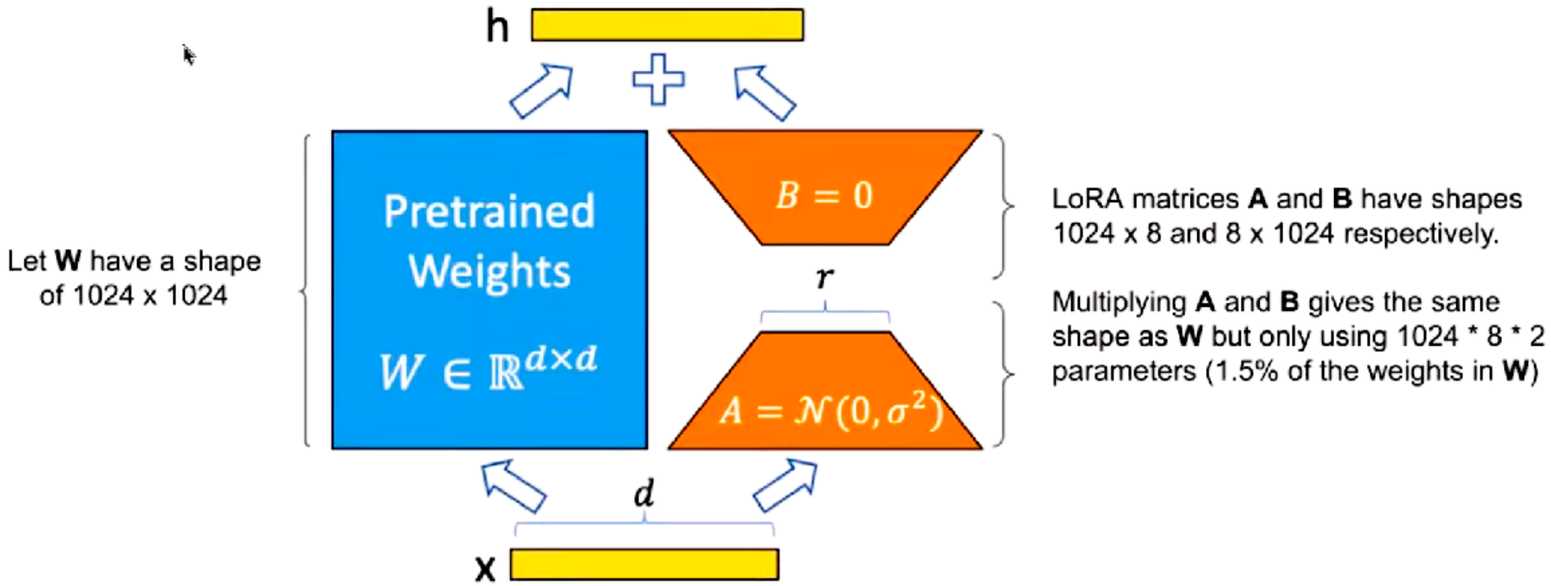
# Parameter Efficient Training: LORA

Freezing the LLM's original weights, surgically inserting new ones



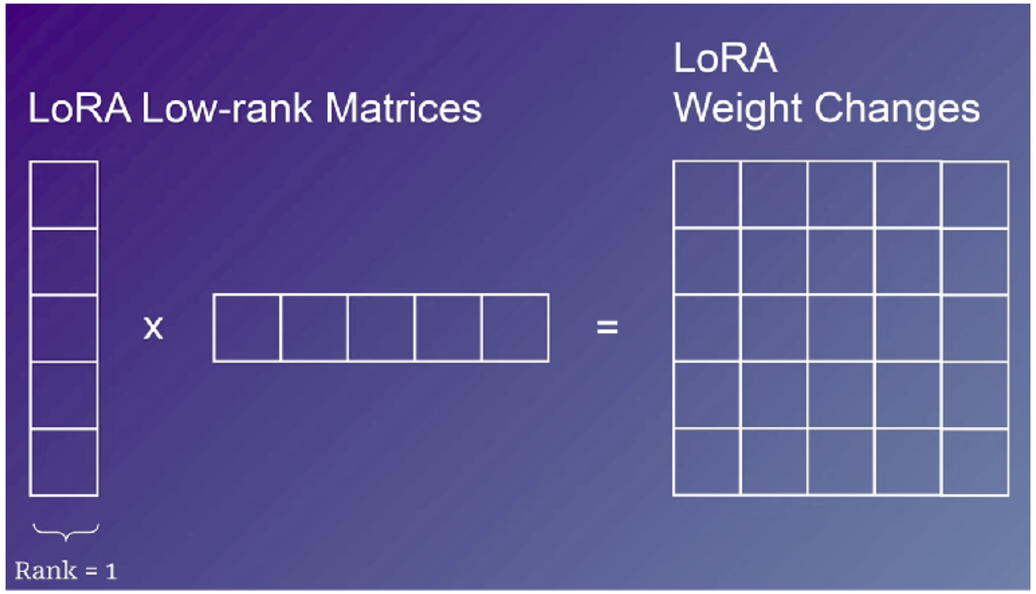

LoRA: https://arxiv.org/abs/2106.09685

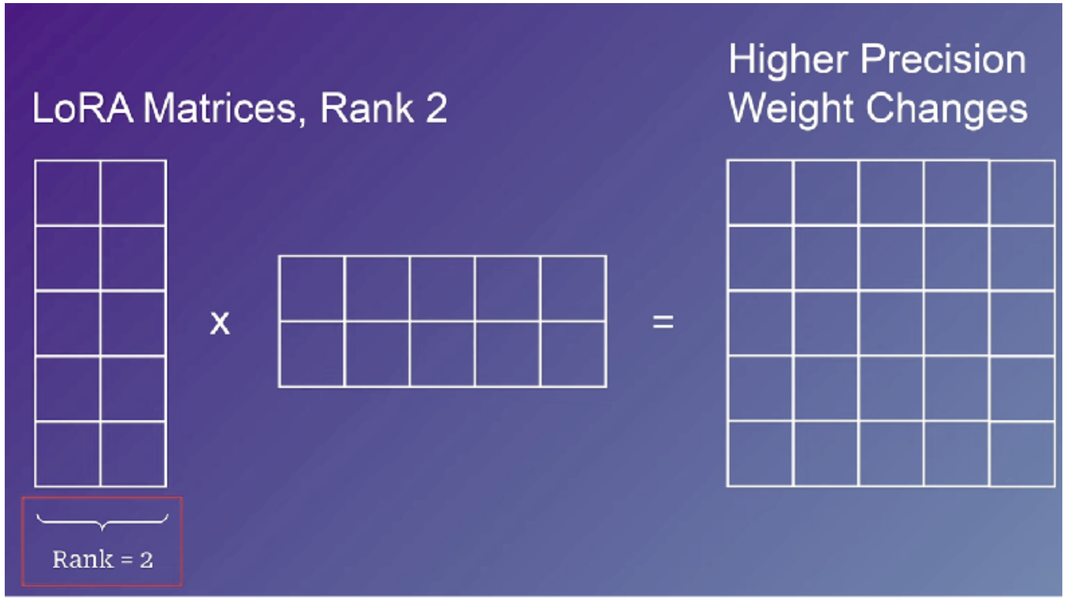
# Parameter Efficient Training: LORA



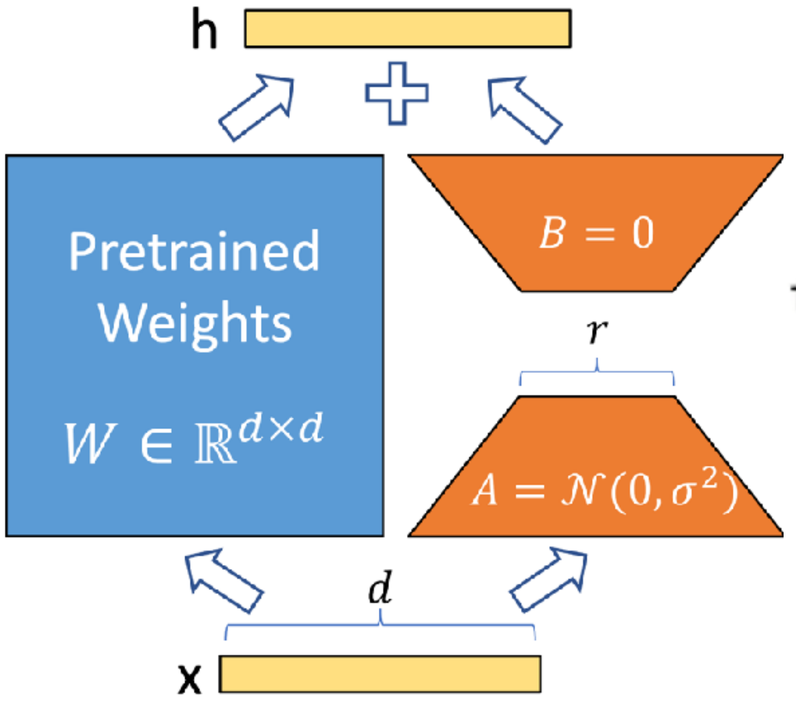

**Compress "fine-tunable" parameters to 0.5% - 10 % of total parameters in your LLM**

# Parameter Efficient Training: LORA

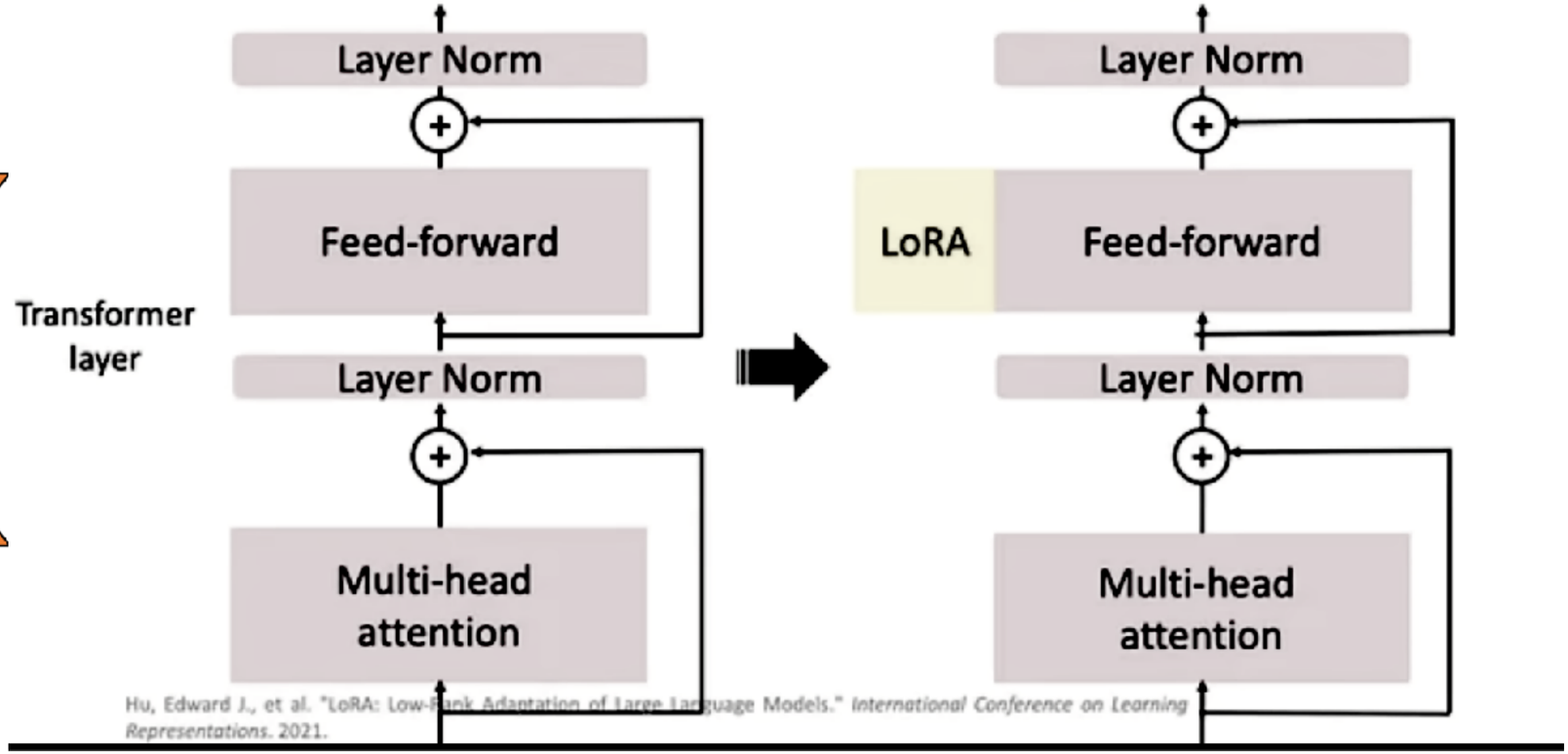


# Parameter Efficient Training: LORA




- LoRA: Low-Rank Adaptation of Large Language Models




Hu, Edward J., et al. "LoRA: Low-Rank Adaptation of Large Language Models." *International Conference on Learning Representations*. 2021.

# Parameter Efficient Training: LORA

LoRA fine-tuned model performance ≈ Full fine-tuned model performance

| Model & Method | MNLI | SST-2 | MRPC | CoLA | QNLI | QQP | RTE | STS-B | Avg. |
|---|---|---|---|---|---|---|---|---|---|
| $\text{RoB}_{\text{large}}$ (FT)* | 90.2 | **96.4** | **90.9** | 68.0 | 94.7 | **92.2** | 86.6 | 92.4 | 88.9 |
| $\text{RoB}_{\text{large}}$ (LoRA) | **90.6**$_{\pm.2}$ | 96.2$_{\pm.5}$ | **90.9**$_{\pm1.2}$ | **68.2**$_{\pm1.9}$ | **94.9**$_{\pm.3}$ | 91.6$_{\pm.1}$ | **87.4**$_{\pm2.5}$ | **92.6**$_{\pm.2}$ | **89.0** |
| $\text{DeB}_{\text{XXL}}$ (FT)* | 91.8 | **97.2** | 92.0 | 72.0 | **96.0** | 92.7 | 93.9 | 92.9 | 91.1 |
| $\text{DeB}_{\text{XXL}}$ (LoRA) | **91.9**$_{\pm.2}$ | 96.9$_{\pm.2}$ | **92.6**$_{\pm.6}$ | **72.4**$_{\pm1.1}$ | **96.0**$_{\pm.1}$ | **92.9**$_{\pm.1}$ | **94.9**$_{\pm.4}$ | **93.0**$_{\pm.2}$ | **91.3** |

# Parameter Efficient Training: LORA

0.2% of weights trained ⇒ lower memory footprint and faster fine-tuning jobs

| Model & Method | # Trainable Parameters | Avg. |
|---|---|---|
| $\text{RoB}_{\text{large}}$ (FT)* | 355.0M | 88.9 |
| $\text{RoB}_{\text{large}}$ (LoRA) | 0.8M | **89.0** |
| $\text{DeB}_{\text{XXL}}$ (FT)* | 1500.0M | 91.1 |
| $\text{DeB}_{\text{XXL}}$ (LoRA) | 4.7M | **91.3** |

1. Less parameters to train and store: if $d = 1000, k = 5000, (d \times k) = 5{,}000{,}000$; using $r = 5$, we get $(d \times r) + (r \times k) = 5{,}000 + 25{,}000 = 30{,}000$. Less than 1% of the original.
2. Less parameters = less storage requirements
3. Faster backpropagation, as we do not need to evaluate the gradient for most of the parameters
4. We can easily switch between two different fine-tuned model (one for SQL generation and one for Javascript code generation) just by changing the parameters of the A and B matrices instead of reloading the W matrix again.

# Parameter Efficient Training: LORA

| Rank | 7B | 13B | 70B | 180B |
|---|---|---|---|---|
| 1 | 167k | 228k | 529k | 849k |
| 2 | 334k | 456k | 1M | 2M |
| 8 | 1M | 2M | 4M | 7M |
| 16 | 3M | 4M | 8M | 14M |
| 512 | 86M | 117M | 270M | 434M |
| 1,024 | 171M | 233M | 542M | 869M |
| 8,192 | 1.4B | 1.8B | 4.3B | 7.0B |

*In reality, LLMs are made up of multiple layers of differing sizes. This is a generalization as if the model were a single layer.*

| Rank | 7B | 13B | 70B | 180B |
|---|---|---|---|---|
| 1 | 0.00% | 0.00% | 0.00% | 0.00% |
| 2 | 0.01% | 0.00% | 0.00% | 0.00% |
| 8 | 0.02% | 0.01% | 0.01% | 0.00% |
| 16 | 0.04% | 0.03% | 0.01% | 0.01% |
| 512 | 1.22% | 0.90% | 0.39% | 0.24% |
| 1,024 | 2.45% | 1.80% | 0.77% | 0.48% |
| 8,192 | 19.58% | 14.37% | 6.19% | 3.86% |

*Percentages are understated since models are made up of different layers, but you get the idea.*

# Why does it work?

### 4.1 Low-Rank-Parametrized Update Matrices

A neural network contains many dense layers which perform matrix multiplication. The weight matrices in these layers typically have full-rank. When adapting to a specific task, Aghajanyan et al. (2020) shows that the pre-trained language models have a low "instrisic dimension" and can still learn efficiently despite a random projection to a smaller subspace. Inspired by this, we hypothesize the updates to the weights also have a low "intrinsic rank" during adaptation. For a pre-trained weight matrix $W_0 \in \mathbb{R}^{d\times k}$, we constrain its update by representing the latter with a low-rank decomposition $W_0 + \Delta W = W_0 + BA$, where $B \in \mathbb{R}^{d\times r}$, $A \in \mathbb{R}^{r\times k}$, and the rank $r \ll \min(d, k)$. During training, $W_0$ is frozen and does not receive gradient updates, while $A$ and $B$ contain trainable parameters. Note both $W_0$ and $\Delta W = BA$ are multiplied with the same input, and their respective output vectors are summed coordinate-wise. For $h = W_0 x$, our modified forward pass yields:

$$h = W_0 x + \Delta W x = W_0 x + BAx \tag{3}$$

# QLORA

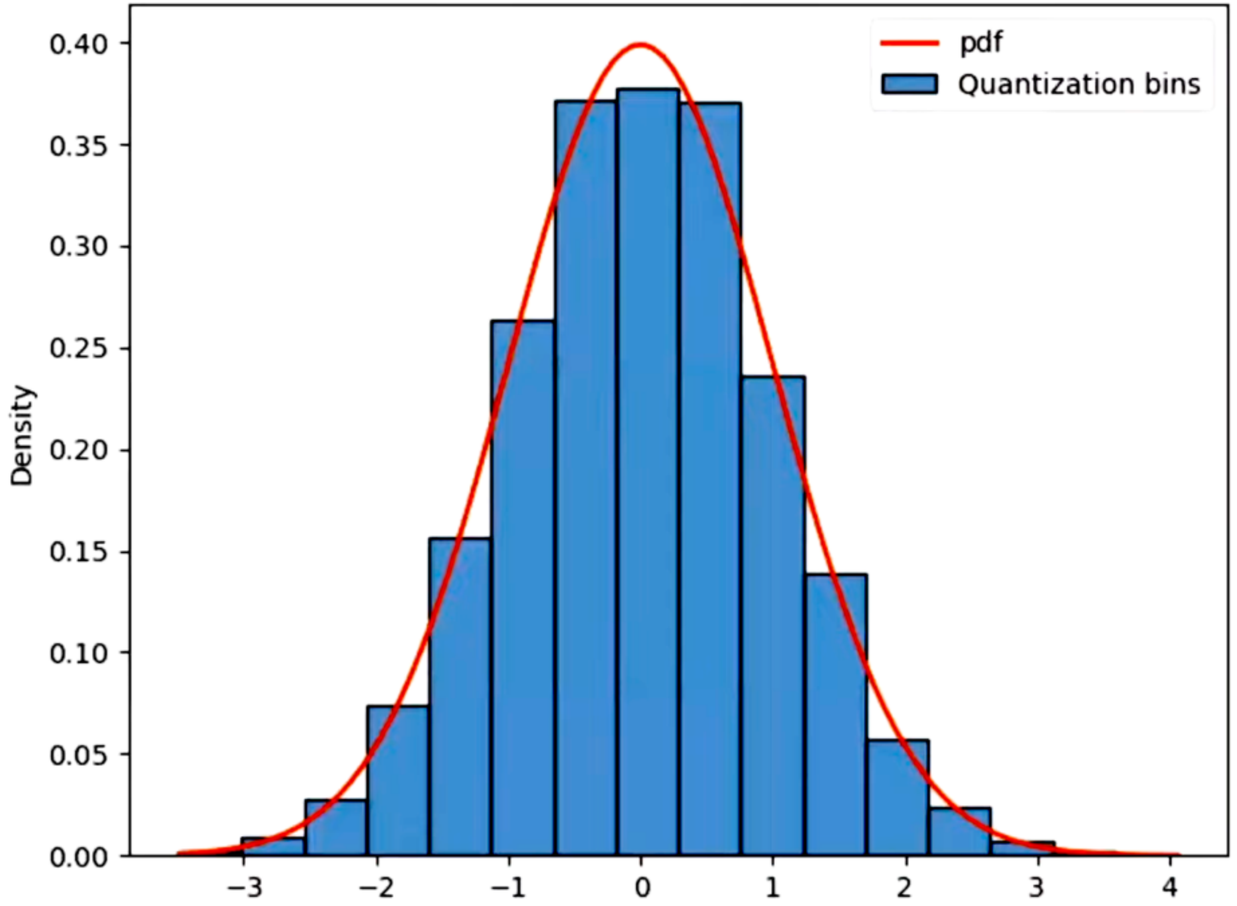
| Model | Inference memory | Fine-tuning memory |
|---|---|---|
| T5-11B | 22 GB | 176 GB |
| LLaMA-33B | 66 GB | 396 GB |
| LLaMA-65B | 130 GB | 780 GB |

SpQR

QLoRA

| Model | Inference memory | Fine-tuning memory |
|---|---|---|
| T5-11B | 5 GB | 6 GB |
| LLaMA-33B | 14 GB | 23 GB |
| LLaMA-65B | 27 GB | 44 GB |

# Quantization




Most general form of describe quantization is through a mapping from integers to float values normalized to the range -1.0 and 1.0.

Int4 0 1 2 3 4 …
-7 -6 -5 -4 -3 …

FP4 0 1 2 3 4 5 …
-12 -8 -6 -4 -3 …

Int4 0 1 2 3 4 …
-7 -6 -5 -4 -3 …
-1 -0.86 -0.71 -0.57 -0.43 …

FP4 0 1 2 3 4 …
-12 -8 -6 -4 -3 …
-1 -0.67 -0.5 -0.33 -0.25 …

The mapping format { index : float value} generalizes to all data types.

# Quantization

Given a tensor X of any real data type. We can apply quantization as follows:

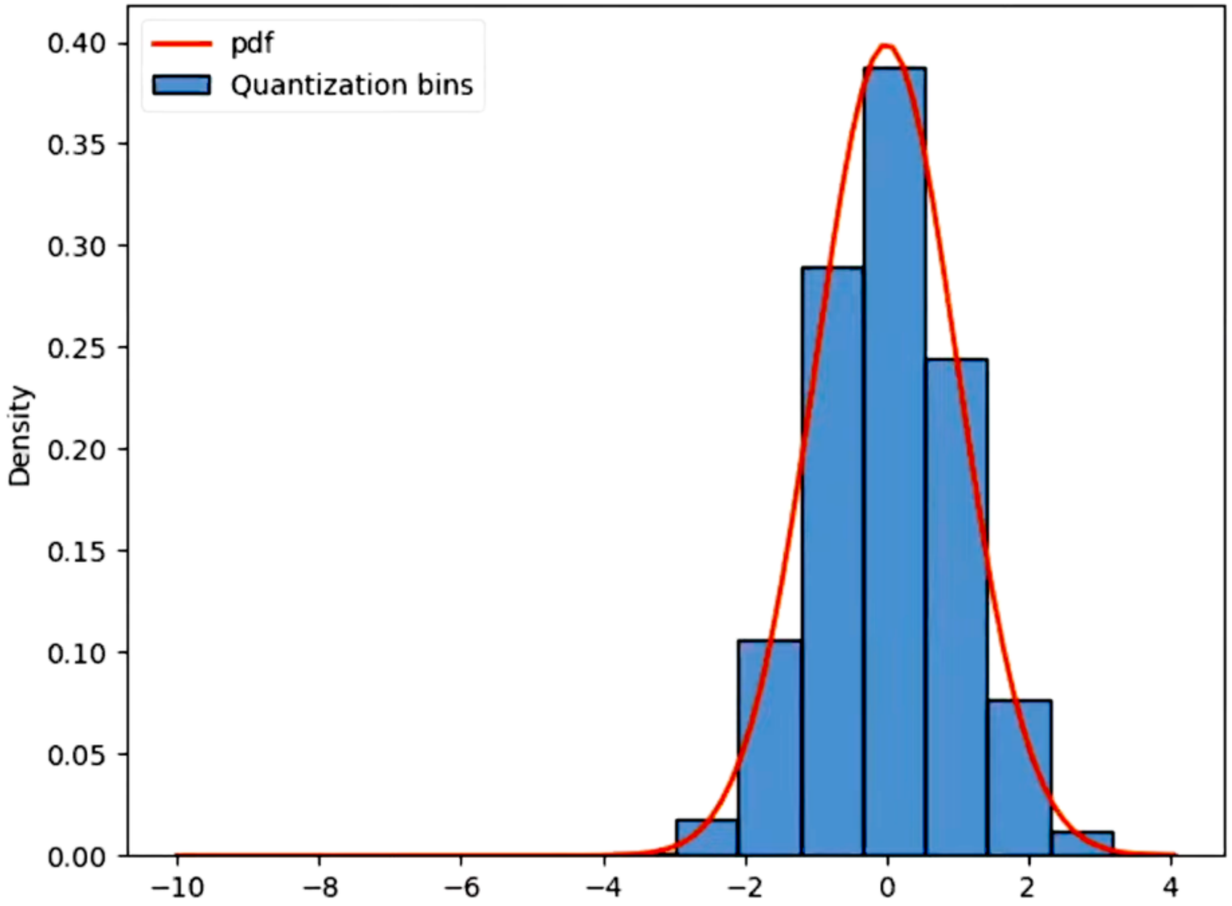
1. Normalize X into the range [-1.0, 1.0] by dividing by absmax(X)
2. Find the closest value in the data type (rounding for integers; in general binary search)

Map: {Index: 0, 1, 2, 3 -> Values: -1.0, 0.3, 0.5, 1.0}

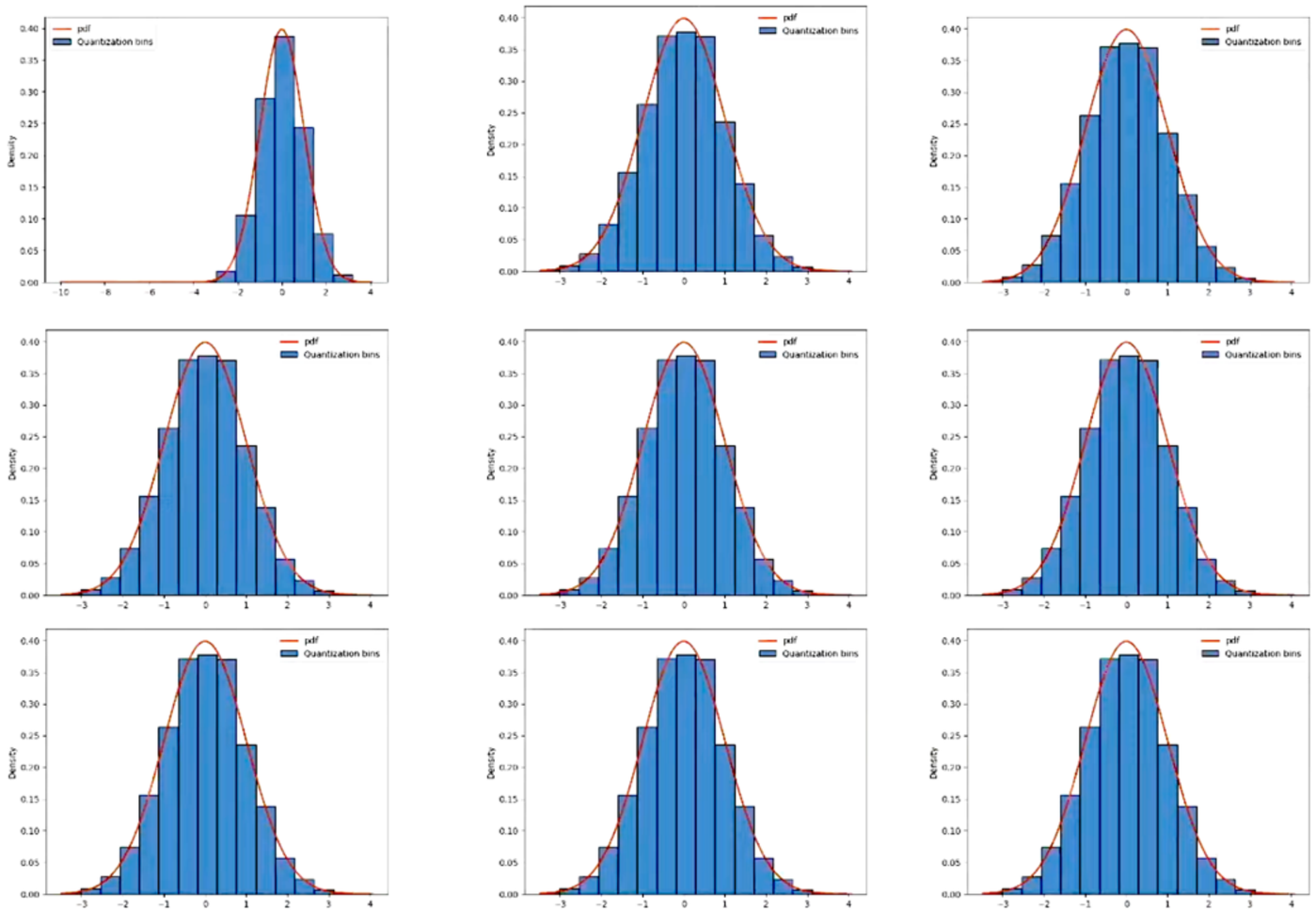
Input tensor: [10, -3, 5, 4]

1. Normalize with absmax: [10, -3, 5, 4] -> [1, -0.3, 0.5, 0.4]
2. Find closest value: [1, -0.3, 0.5, 0.4] -> [1.0, 0.3, 0.5, 0.5]
3. Find the associated index: [1.0, 0.3, 0.5, 0.5] -> [3, 1, 2, 2] -> store
4. Dequantization: load -> [3, 1, 2, 2] -> lookup -> [1.0, 0.3, 0.5, 0.5] -> denormalize -> [10, 3, 5, 5]

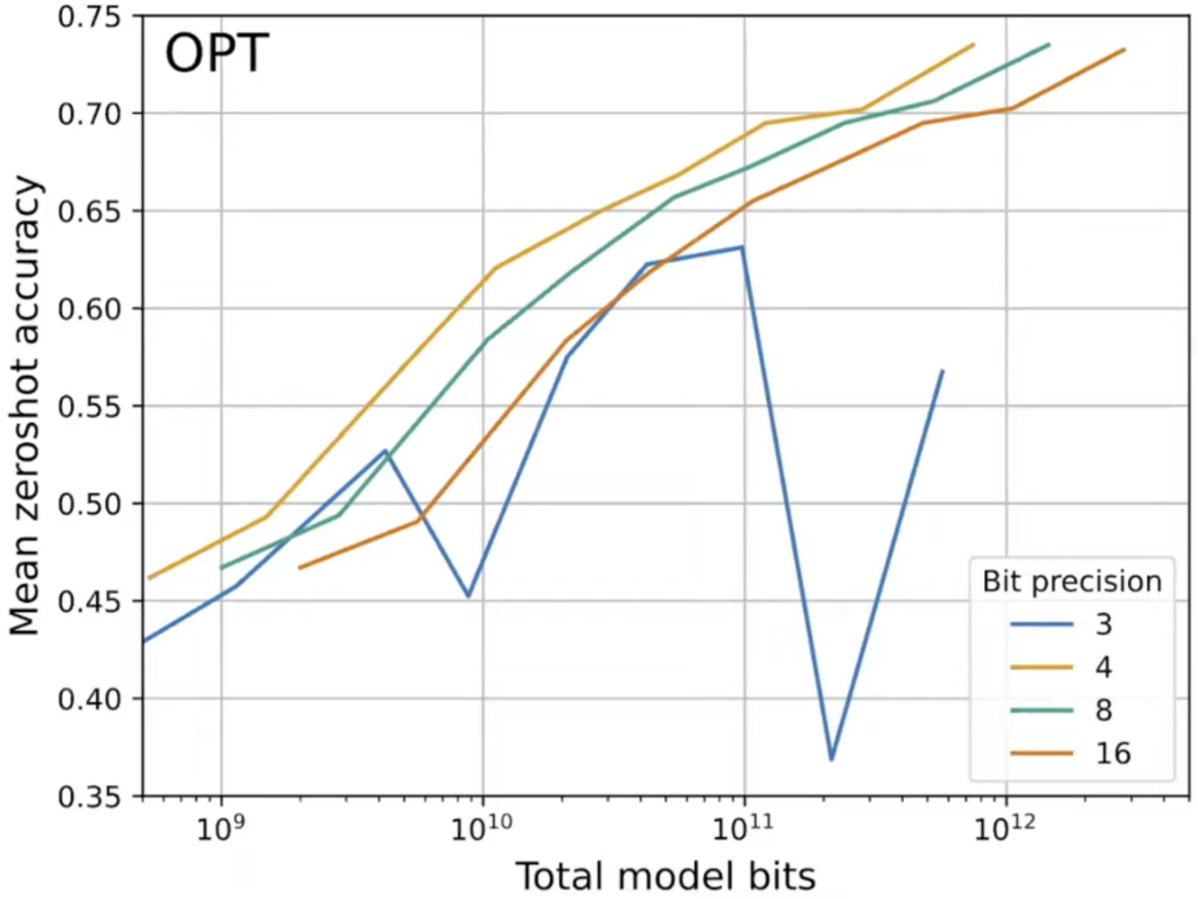
# Outliers


pdf
Quantization bins
Density
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
−10
−8
−6
−4
−2
0
2
4

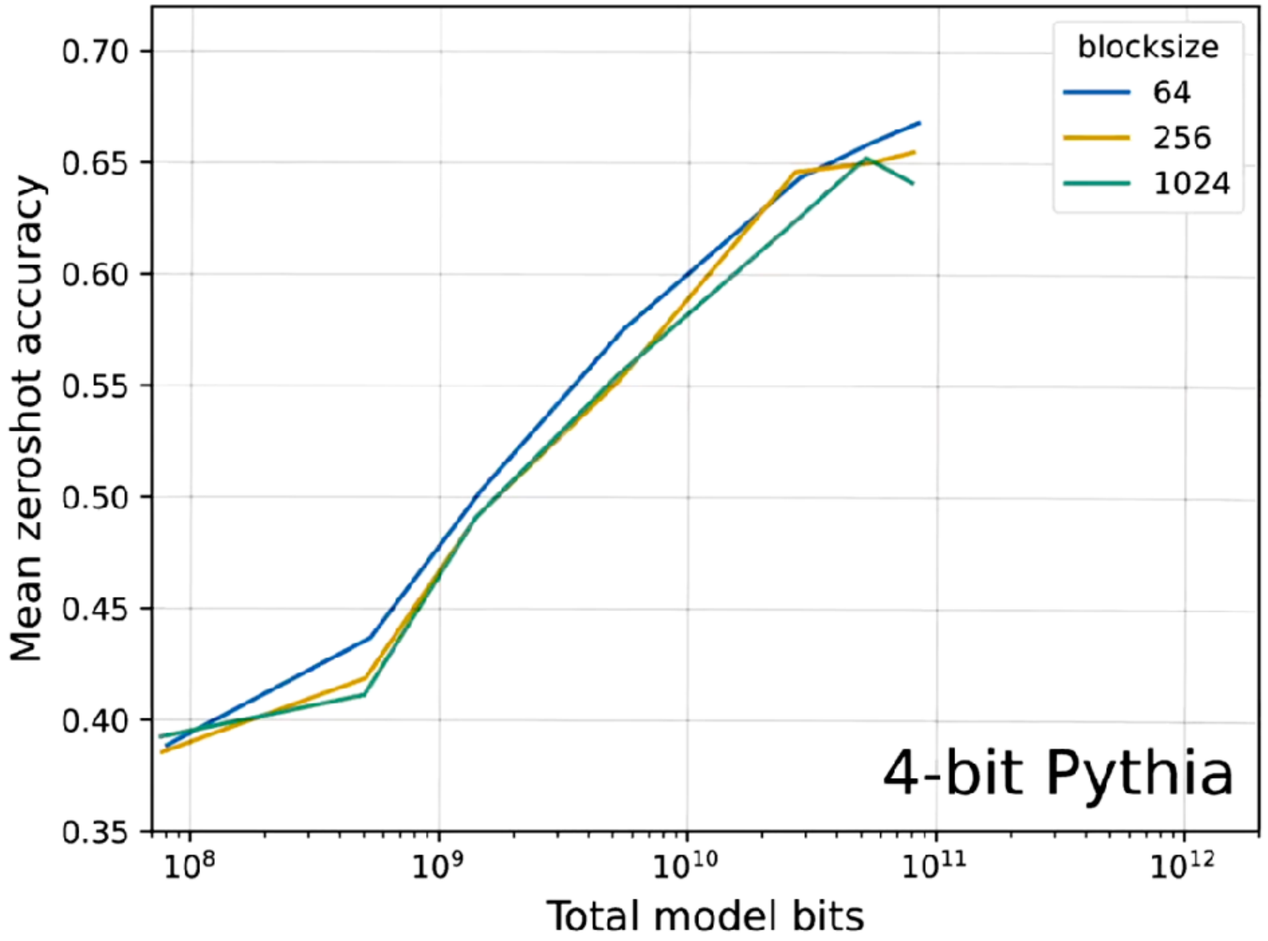
# Block-wise Quantization



# Quantization



# Quantization



# Typical Fine-tuning Cost

Finetuning cost per parameter:

- Weight: 2 byte
- Weight gradient: 2 byte
- Optimizer state: 8 byte
- 12 byte per parameter
- 65B model -> 780 GB of GPU memory -> 17x data center GPUs

Finetuning cost per parameter:

- Weight: 16 bits
- Weight gradient: ~0.4bit
- Optimizer state:  ~0.8bit
- Adapter weights: ~0.4bit
- 17.6 bits per parameter
- 65B model -> 143 GB of GPU memory -> 4x data center GPUs or 8x consumer GPUs (RTX 3090 / RTX 4090)

**With LoRA**

# QLoRA: 4 bit quantized LoRA

Finetuning cost per parameter:

- Weight: 4 bit
- Weight gradient: ~0.4 bit
- Optimizer state: ~0.8 bit
- Adapter weights: ~0.4 bit
- 5.2 bit per parameter
- 65B model -> ~~780 GB~~

42 GB of GPU memory -> ~~17x~~

1x data center GPUs

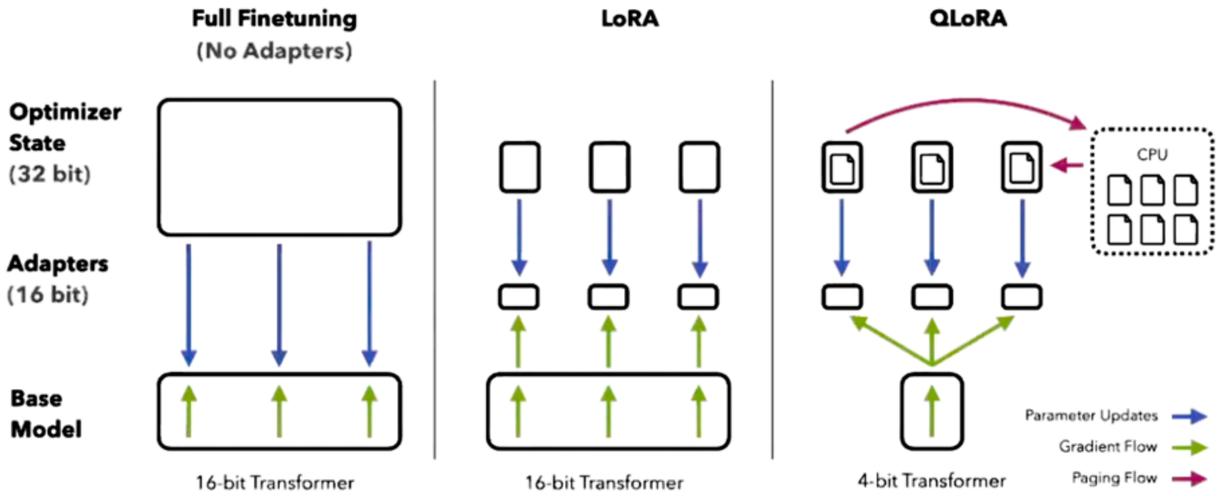
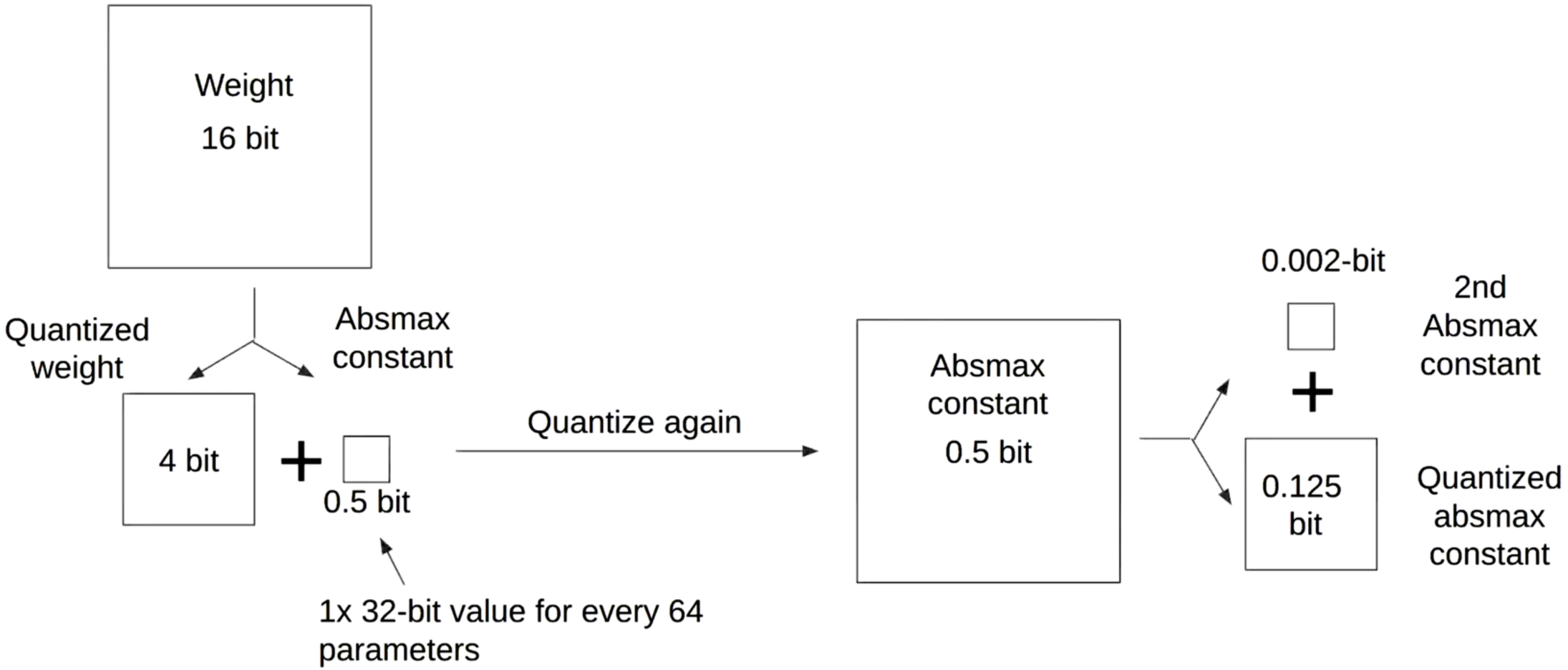
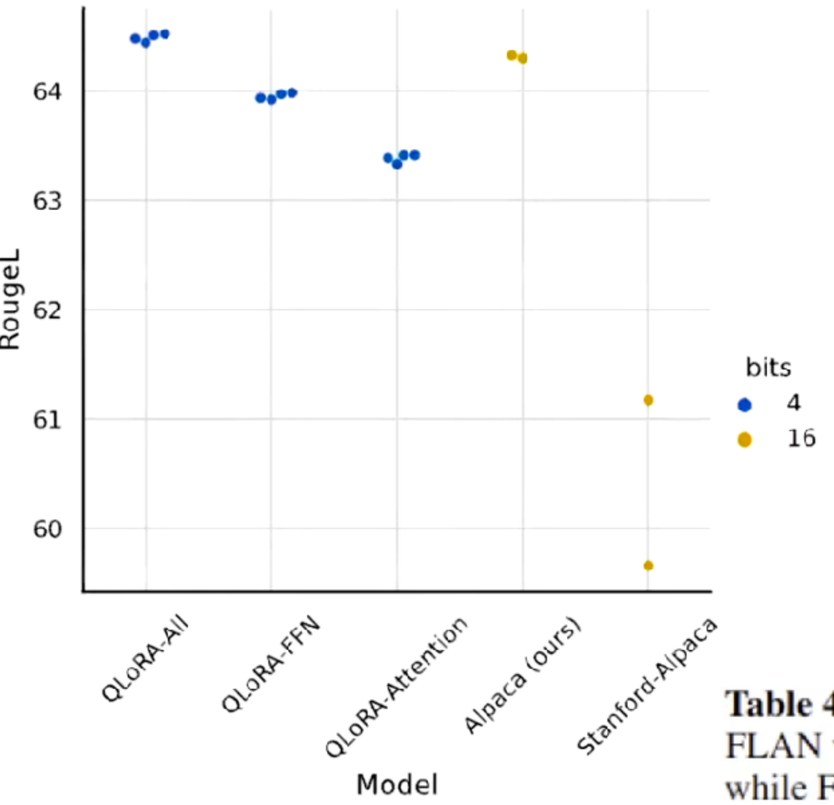



**Figure 1:** Different finetuning methods and their memory requirements. QLoRA improves over LoRA by quantizing the transformer model to 4-bit precision and using paged optimizers to handle memory spikes.

# Reduce absmax: Double Quantization



# Results



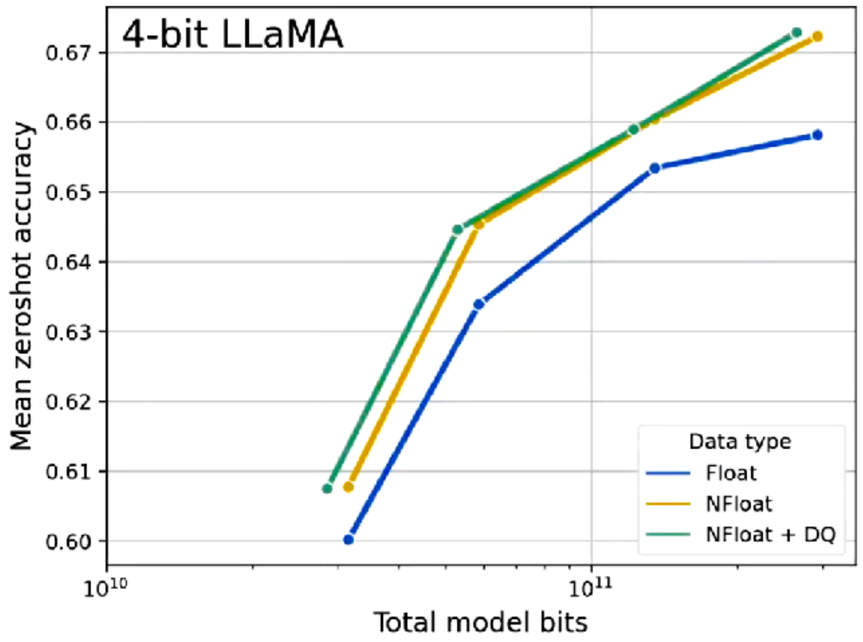




**Table 2:** Pile Common Crawl mean perplexity for different data types for 125M to 13B OPT, BLOOM, LLaMA, and Pythia models.

| Data type | Mean PPL |
|---|---|
| Int4 | 34.34 |
| Float4 (E2M1) | 31.07 |
| Float4 (E3M0) | 29.48 |
| NFloat4 + DQ | **27.41** |

**Table 4:** Mean 5-shot MMLU test accuracy for LLaMA 7-65B models finetuned with adapters on Alpaca and FLAN v2 for different data types. Overall, NF4 with double quantization (DQ) matches BFloat16 performance, while FP4 is consistently one percentage point behind both.

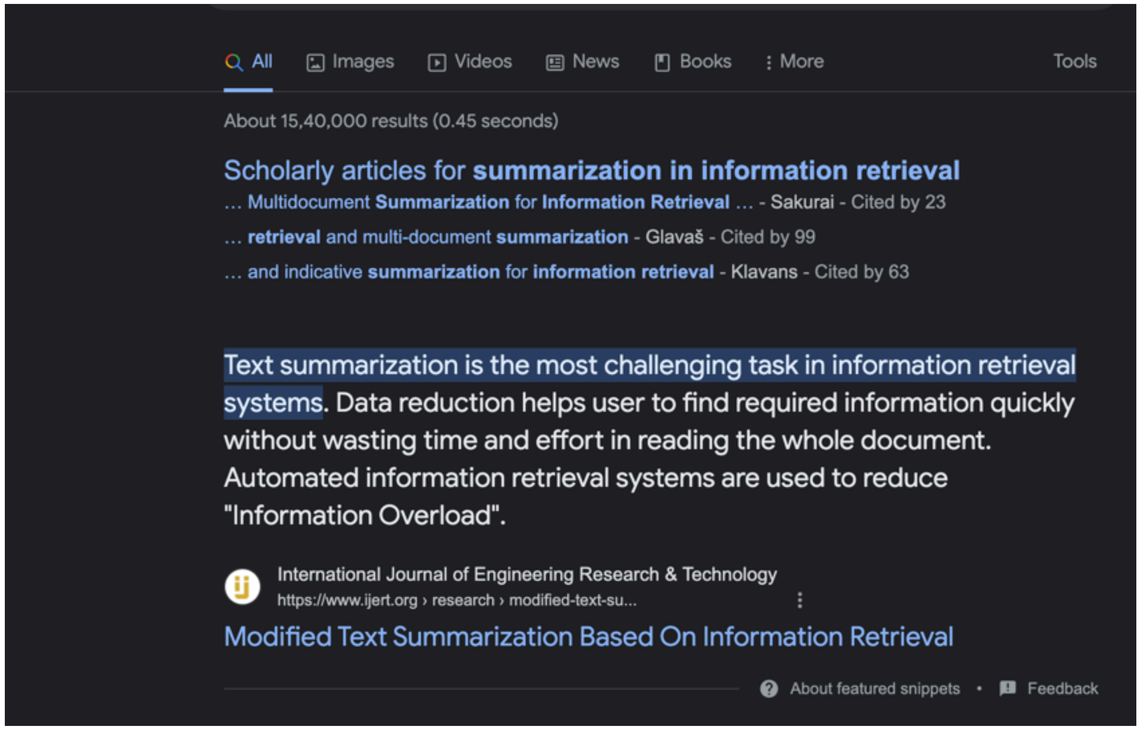
| | Mean 5-shot MMLU Accuracy | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| LLaMA Size | 7B | | 13B | | 33B | | 65B | | Mean |
| Dataset | Alpaca | FLAN v2 | Alpaca | FLAN v2 | Alpaca | FLAN v2 | Alpaca | FLAN v2 | |
| BFloat16 | 38.4 | 45.6 | 47.2 | 50.6 | 57.7 | 60.5 | 61.8 | 62.5 | 53.0 |
| Float4 | 37.2 | 44.0 | 47.3 | 50.0 | 55.9 | 58.5 | 61.3 | 63.3 | 52.2 |
| NFloat4 + DQ | 39.0 | 44.5 | 47.5 | 50.7 | 57.3 | 59.2 | 61.8 | 63.9 | 53.1 |

# Text Summarization

Task: given input text *x*, write a summary *y* which is shorter and contains the main information of *x*.

Summarization can be single-document or multi-document.

- Single-document means we write a summary *y* of a single document *x*.
- Multi-document means we write a summary *y* of *multiple* documents $x_1,\ldots,x_n$



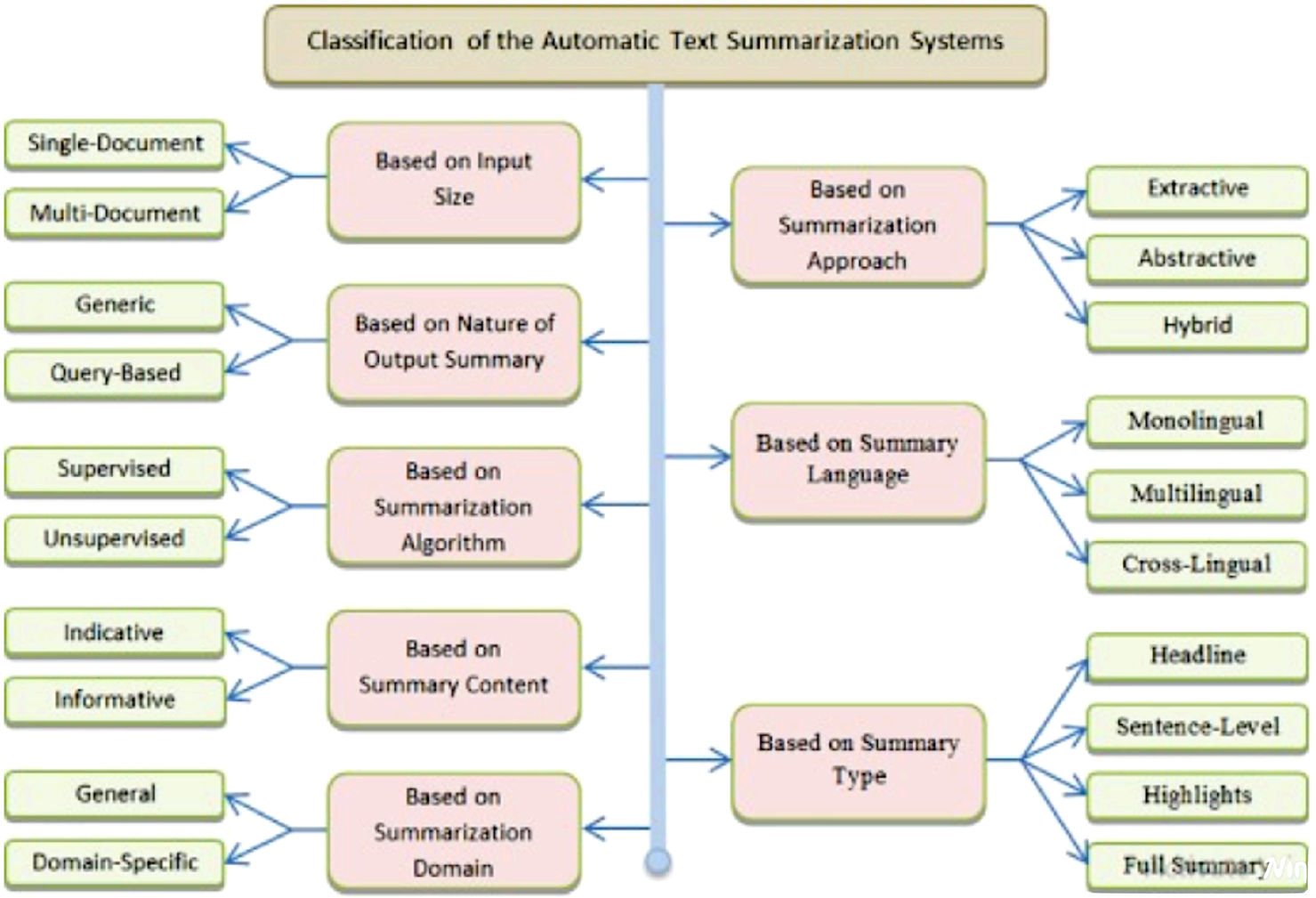
# Another Classification of Summarisation techniques

- **Extractive Summarisation**
  - In extractive summarization, the system selects and extracts sentences, phrases, or passages directly from the source text to form the summary.
  - The selected content is usually chosen based on various criteria, such as sentence importance, relevance, or informativeness.
  - Extractive summarization does not involve paraphrasing or rewriting the text; it relies on directly using existing text segments.
  - Extractive methods often employ techniques like sentence scoring, keyword extraction, and graph algorithms to identify the most salient content.

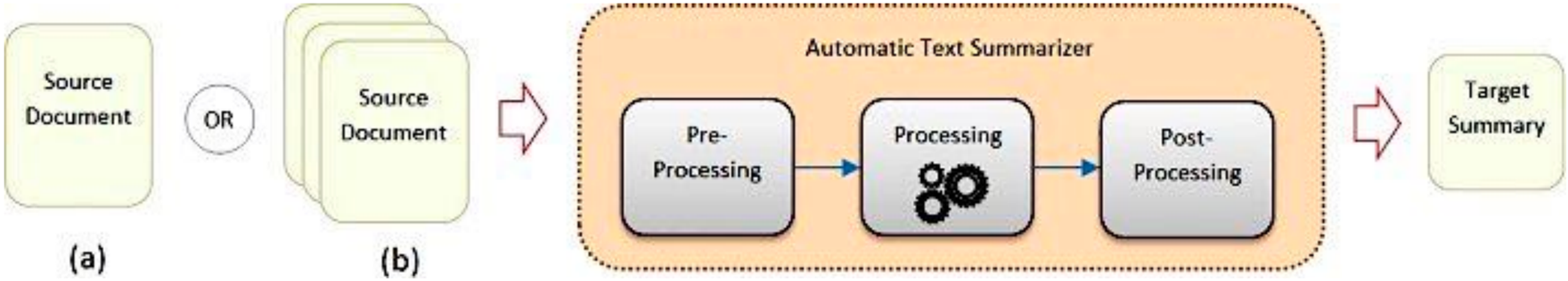
# Another Classification of Summarisation techniques

- **Abstractive Summarisation**
  - The system generates a concise summary in its own words, capturing the essence of the source text while possibly using different sentence structures and words.
  - Abstractive summarization typically requires a deeper understanding of the source text, as it involves natural language generation, paraphrasing, and potentially even incorporating additional context.
  - It can be more creative and flexible in summarizing content but also presents greater challenges in terms of language fluency, coherence, and maintaining factual accuracy.
  - Abstractive methods often employ neural network-based techniques, including sequence-to-sequence models and transformer-based models, which have shown significant advancements in generating abstractive summaries.

# Another Classification of Summarisation techniques



# Text Summarisation pipeline




**Pre-Processing:** producing a structured representation of the original text using many linguistic techniques like sentences segmentation, words tokenization, removal of stop-words, part-of-speech tagging, stemming, etc.

**Processing:** using one of the text summarization approaches

**Post-Processing:** solving some problems in the generated summary sentences like anaphora resolution and reordering the selected sentences before generating the final summary.

# Evaluation

**ROUGE Metric:** it is the most commonly used tool for automatic evaluation of the automatically generated summaries

**ROUGE** (Recall-Oriented Understudy for Gisting Evaluation)

ROUGE-N

$$= \frac{\sum_{S \in \{ReferemceSummaries\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in \{ReferenceSummaries\}} \sum_{gram_n \in S} Count(gram_n)} \quad (1)$$

**ROUGE-1:** it is based on the uni-gram measure between a candidate summary and a reference summary.

**ROUGE-2:** is an bi-gram measure between a candidate summary and reference summaries.

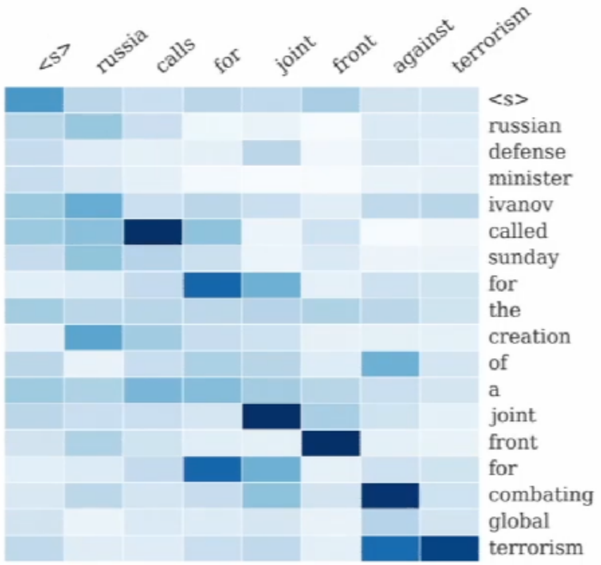
**ROUGE-L:** it is based on the longest common subsequences between a candidate summary and a reference summary.

# Models

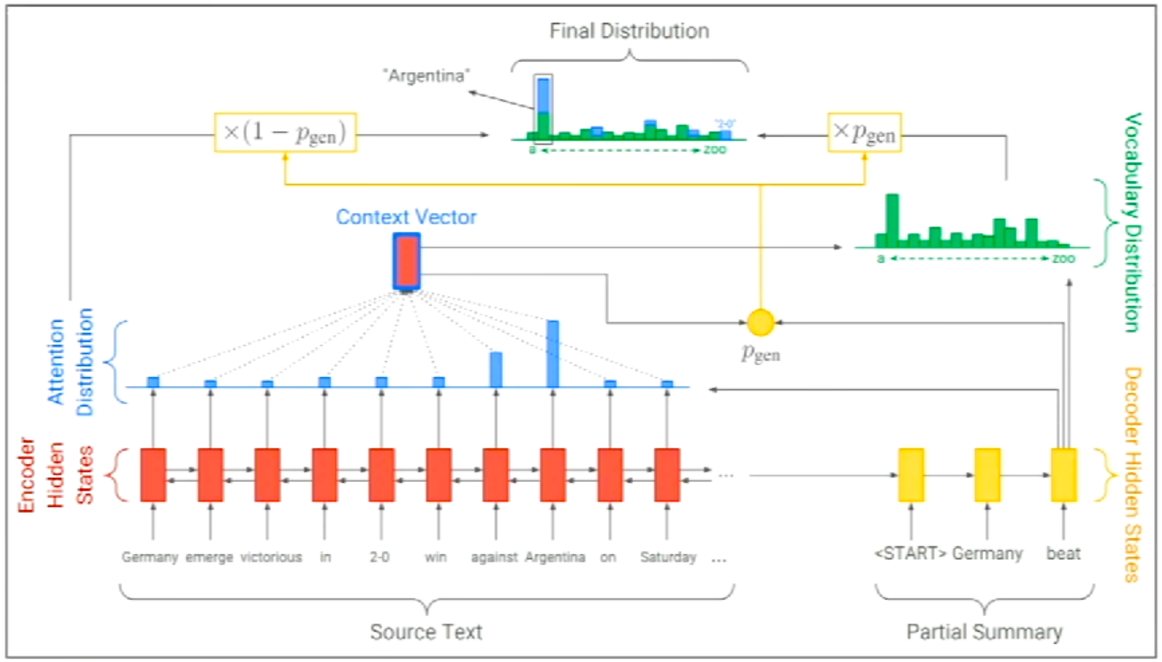
2015: Rush et al. publish the first seq2seq summarization paper

Single-document abstractive summarization is a translation task!

Thus we can apply standard seq2seq + attention NMT methods



# Models: Pointer Generator Networks



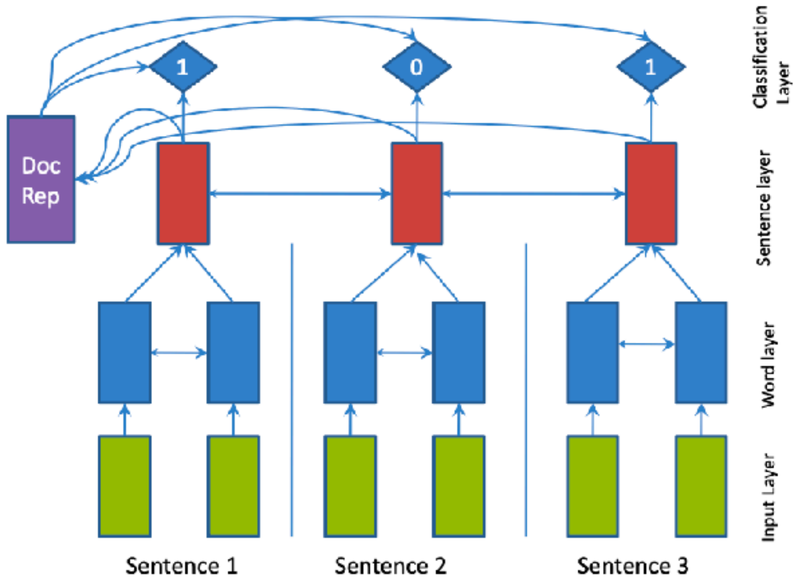

<u>One example of how to do a copying mechanism:</u>

On each decoder step, calculate $p_{gen}$, the probability of *generating* the next word (rather than copying it). The final distribution is a mixture of the generation (aka "vocabulary") distribution, and the copying (i.e. attention) distribution:

$$P(w) = p_{\text{gen}} P_{\text{vocab}}(w) + (1 - p_{\text{gen}}) \sum_{i:w_i=w} a_i^t$$

*Get To The Point: Summarization with Pointer-Generator Networks*, See et al, 2017 https://arxiv.org/pdf/1704.04368.pdf

# SummaRuNNer




$$\mathbf{d} = \tanh(W_d \frac{1}{N_d} \sum_{j=1}^{N^d} [\mathbf{h}_j^f, \mathbf{h}_j^b] + \mathbf{b}),$$

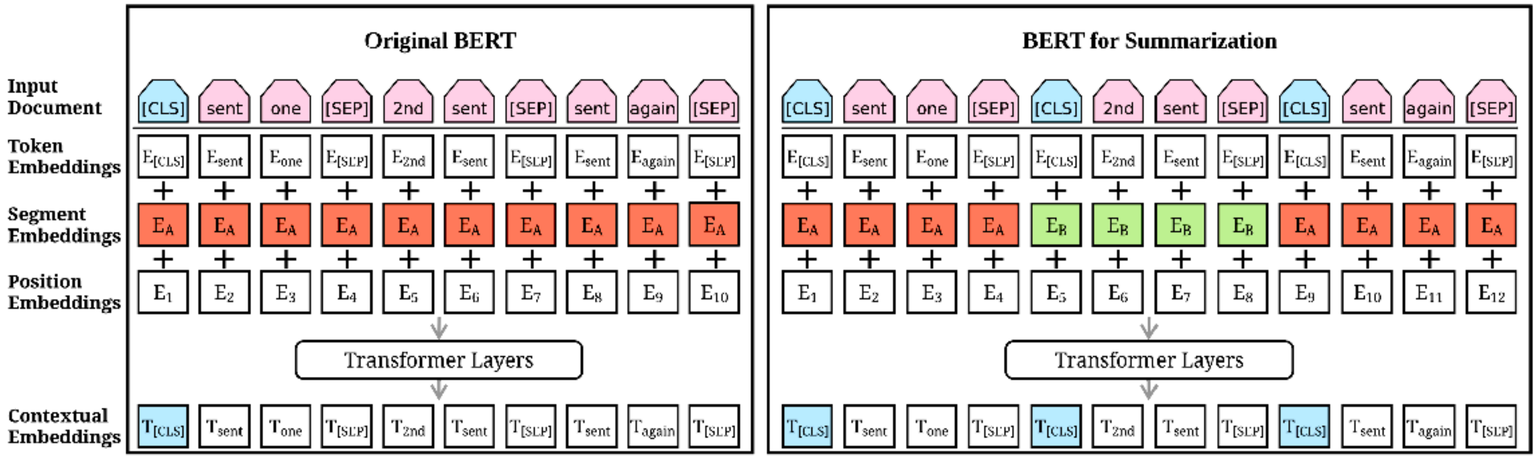
$$\begin{aligned} P(y_j = 1|\mathbf{h}_j, \mathbf{s}_j, \mathbf{d}) = \sigma(&W_c \mathbf{h}_j && \#\,\texttt{(content)} \\ &+ \mathbf{h}_j^T W_s \mathbf{d} && \#\,\texttt{(salience)} \\ &- \mathbf{h}_j^T W_r \tanh(\mathbf{s_j}) && \#\,\texttt{(novelty)} \\ &+ W_{ap} \mathbf{p}_j^a && \#\,\texttt{(abs. pos. imp.)} \\ &+ W_{rp} \mathbf{p}_j^r && \#\,\texttt{(rel. pos. imp.)} \\ &+ b), && \#\,\texttt{(bias term)} \end{aligned}$$

| **Gold Summary:** Redpath has ended his eight-year association with Sale Sharks. Redpath spent five years as a player and three as a coach at sale. He has thanked the owners, coaches and players for their support. | Salience | Content | Novelty | Position | Prob. |
|---|---|---|---|---|---|
| Bryan Redpath has left his coaching role at Sale Sharks with immediate effect. | 0.1 | **0.1** | 0.9 | 0.1 | 0.3 |
| The 43 - year - old Scot ends an eight-year association with the Aviva Premiership side, having spent five years with them as a player and three as a coach. | 0.9 | **0.6** | 0.9 | 0.9 | 0.7 |
| Redpath returned to Sale in June 2012 as director of rugby after starting a coaching career at Gloucester and progressing to the top job at Kingsholm . | 0.8 | **0.5** | 0.5 | 0.9 | 0.6 |
| Redpath spent five years with Sale Sharks as a player and a further three as a coach but with Sale Sharks struggling four months into Redpath's tenure, he was removed from the director of rugby role at the Salford-based side and has since been operating as head coach . | 0.8 | **0.9** | 0.7 | 0.8 | **0.9** |
| 'I would like to thank the owners, coaches, players and staff for all their help and support since I returned to the club in 2012. | 0.4 | **0.1** | 0.1 | 0.7 | 0.2 |
| Also to the supporters who have been great with me both as a player and as a coach,' Redpath said. | 0.6 | **0.0** | 0.2 | 0.3 | 0.2 |

*Document:* **today , the foreign ministry said that control operations carried out by the corvette spiro against a korean-flagged as received ship fishing illegally in argentine waters were carried out " in accordance with international law and in coordination with the foreign ministry " .** the foreign ministry thus approved the intervention by the argentine corvette when it discovered the korean ship chin yuan hsing violating argentine jurisdictional waters on 00 may . ... **the korean ship , which had been fishing illegally in argentine waters , was sunk by its own crew after failing to answer to the argentine ship 's warnings .** the crew was transferred to the chin chuan hsing , which was sailing nearby and approached to rescue the crew of the sinking ship .....

*Gold Summary:* the korean-flagged fishing vessel chin yuan hsing was scuttled in waters off argentina on 00 may 0000 . adverse weather conditions prevailed when the argentine corvette spiro spotted the korean ship fishing illegally in restricted argentine waters . the korean vessel did not respond to the corvette 's warning . instead , the korean crew sank their ship , and transferred to another korean ship sailing nearby . in accordance with a uk-argentine agreement , the argentine navy turned the surveillance of the second korean vessel over to the british when it approached within 00 nautical miles of the malvinas ( falkland ) islands .

# BERTSum




## Extractive Summarisation

$$\tilde{h}^l = \mathrm{LN}(h^{l-1} + \mathrm{MHAtt}(h^{l-1}))$$
$$h^l = \mathrm{LN}(\tilde{h}^l + \mathrm{FFN}(\tilde{h}^l))$$

where $h^0 = \mathrm{PosEmb}(T)$; $T$ denotes the sentence vectors output by BERTSUM

The final output layer is a sigmoid classifier:

$$\hat{y}_i = \sigma(W_o h_i^L + b_o)$$

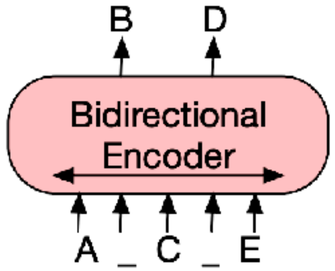
where $h_i^L$ is the vector for $sent_i$ from the top layer (the $L$-th layer ) of the Transformer.

## Abstractive Summarisation

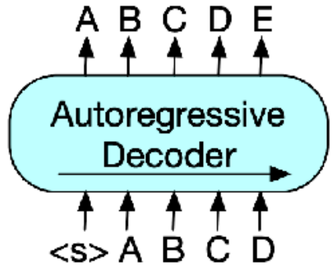
**Strategy 1**
encoder is the pretrained BERTSUM and the decoder is a 6-layered Transformer initialized randomly.

**Strategy 2** a two-stage fine-tuning approach, where we first fine-tune the encoder on the extractive summarization task (Section 3.2) and then fine-tune it on the abstractive summarization task

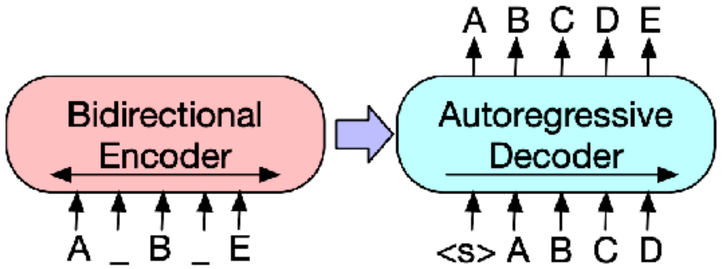
# BART




(a) BERT: Random tokens are replaced with masks, and the document is encoded bidirectionally. Missing tokens are predicted independently, so BERT cannot easily be used for generation.



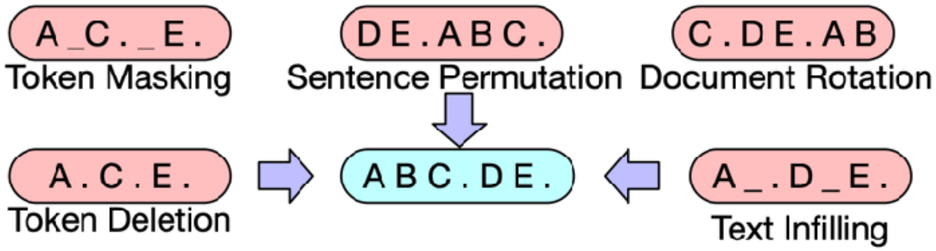

(b) GPT: Tokens are predicted auto-regressively, meaning GPT can be used for generation. However words can only condition on leftward context, so it cannot learn bidirectional interactions.



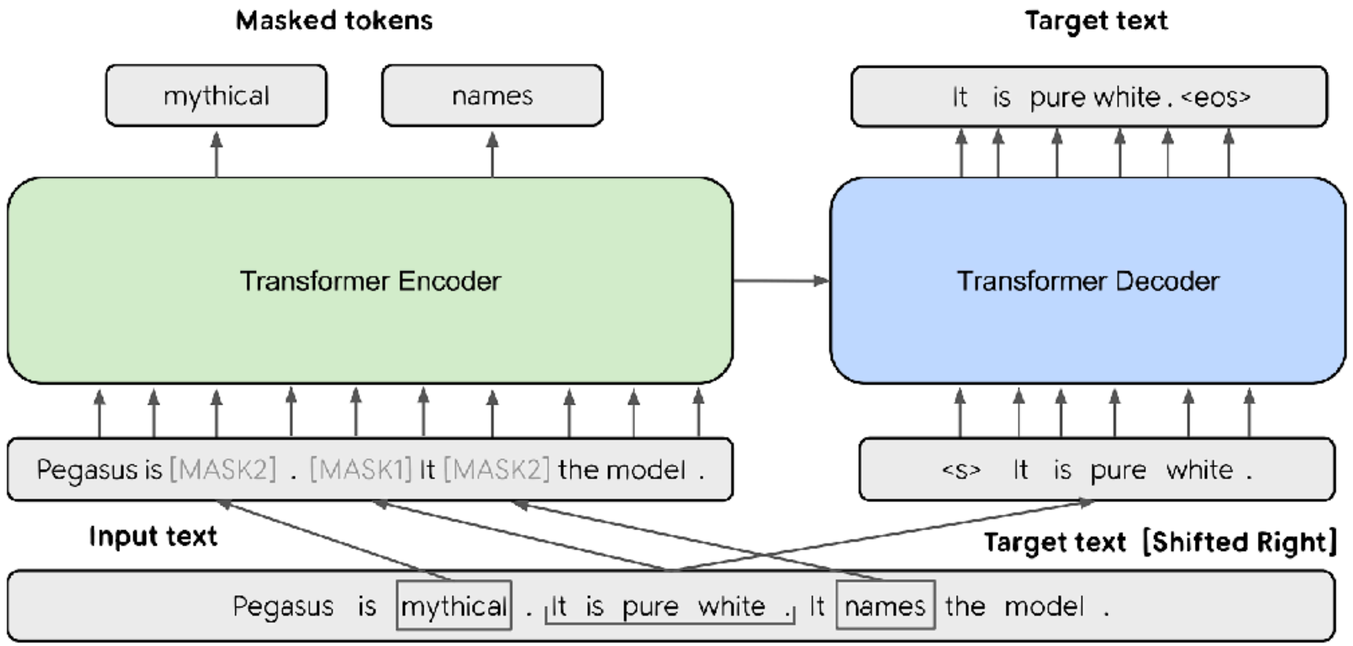

(c) BART: Inputs to the encoder need not be aligned with decoder outputs, allowing arbitary noise transformations. Here, a document has been corrupted by replacing spans of text with mask symbols. The corrupted document (left) is encoded with a bidirectional model, and then the likelihood of the original document (right) is calculated with an autoregressive decoder. For fine-tuning, an uncorrupted document is input to both the encoder and decoder, and we use representations from the final hidden state of the decoder.

BART, a denoising autoencoder for pretraining sequence-to-sequence models. BART is trained by (1) corrupting text with an arbitrary noising function, and (2) learning a model to reconstruct the original text.

BART is particularly effective when fine tuned for text generation but also works well for comprehension tasks.



# PEGASUS




**Random** Uniformly select $m$ sentences at random.

**Lead** Select the first $m$ sentences.

**Principal** Select top-$m$ scored sentences according to importance. As a proxy for importance we compute ROUGE1-F1 (Lin, 2004) between the sentence and the rest of the document, $s_i = rouge(x_i, D \setminus \{x_i\}), \forall i$.

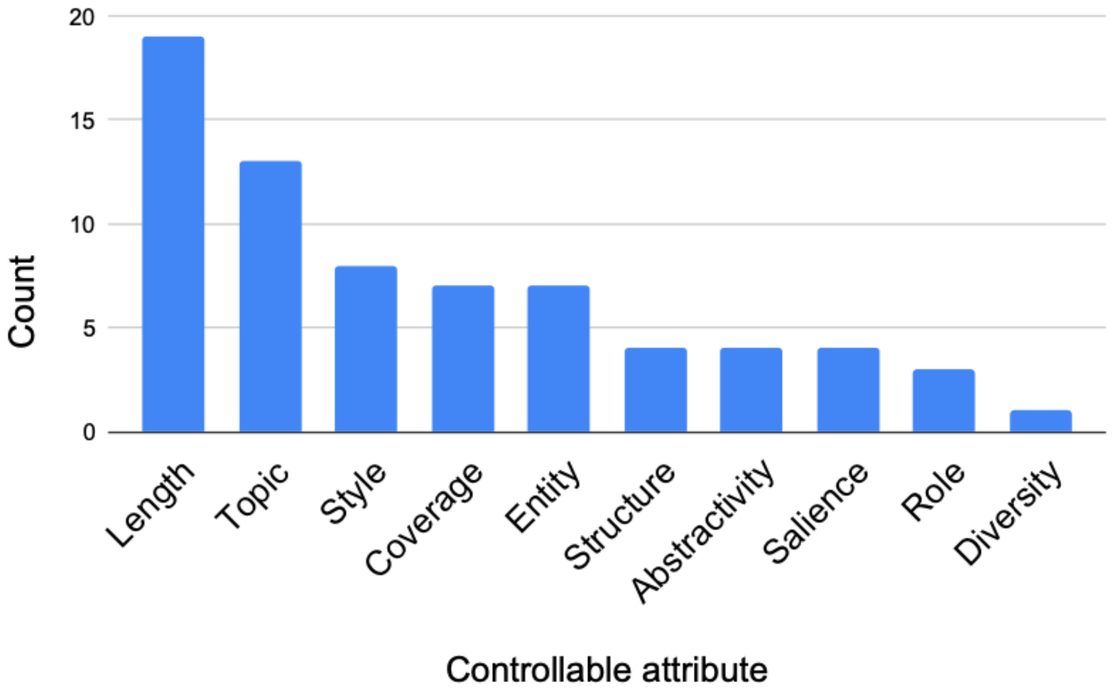
## Gap Sentences Generation (GSG)

We hypothesize that using a pre-training objective that more closely resembles the downstream task leads to better and faster fine-tuning performance.

INVITATION ONLY We are very excited to be co-hosting a major drinks reception with our friends at Progress. This event will sell out, so make sure to register at the link above. Speakers include Rajesh Agrawal, the London Deputy Mayor for Business, Alison McGovern, the Chair of Progress, and Seema Malhotra MP. Huge thanks to the our friends at the ACCA, who have supported this event. The Labour Business Fringe at this year's Labour Annual Conference is being co-sponsored by Labour in the City and the Industry Forum. Speakers include John McDonnell, Shadow Chancellor, and Rebecca Long-Bailey, the Shadow Chief Secretary to the Treasury, and our own Chair, Kitty Ussher. Attendance is free, and refreshments will be provided.

Figure 2: An example of sentences (from the C4 corpus) selected by Random, Lead and Ind-Orig respectively. Best viewed in color.

# CTS: Controllable Text Summarization




| Attribute | Definition |
|---|---|
| ***Length*** | Controlling the length of the summary |
| ***Style*** | Controlling the readability levels, politeness, humor, and emotion |
| ***Coverage*** | Controlling the salient information in summary |
| ***Entity*** | Summary specific to pre-defined entities |
| ***Structure*** | Create summaries with predefined structure or order |
| ***Abstractivity*** | Controlling the novelty in sentence formation |
| ***Salience*** | Adjusting the presence of prominent information |
| ***Role*** | Providing role-specific summaries |
| ***Diversity*** | Generating semantically diverse summaries |
| ***Topic*** | Controlling topic-focused summary generation |

Table 2: Controllable attributes definitions.