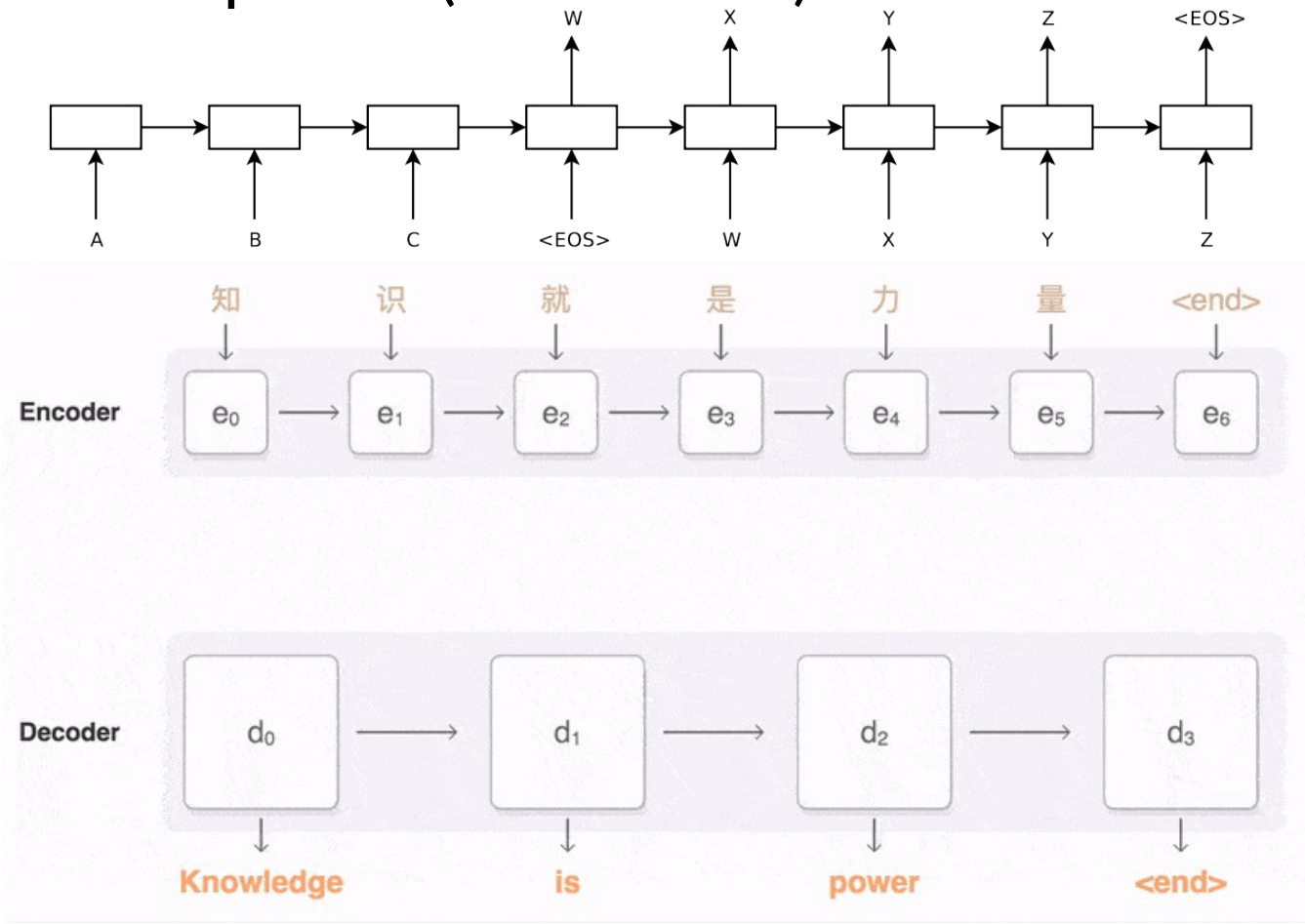
Natural Language Generation

Translation (NMT: neural machine translation)

- Sequence to Sequence Learning (Seq2Seq)
- begin or end of sequence (BOS or EOS) tokens



Summarization

- Extractive Summarization
  - assigns scores to sentences based on their significance
  - selects the top-ranked sentences for the summary
- Abstractive Summarization
  - generates new sentences

Question Answering (QA)

- Close-book
  - Q->A
- Reading Comprehension
  - Context+Q -> A
  - Extractive
    - Answer is within context
  - Abstractive
    - Answer is not in context
- Closed-domain VS Open-domain

Seq2Seq

- Goal

$$p(y_1, \ldots, y_{T'} | x_1, \ldots, x_T) = \prod_{t=1}^{T'} p(y_t | v, y_1, \ldots, y_{t-1})$$

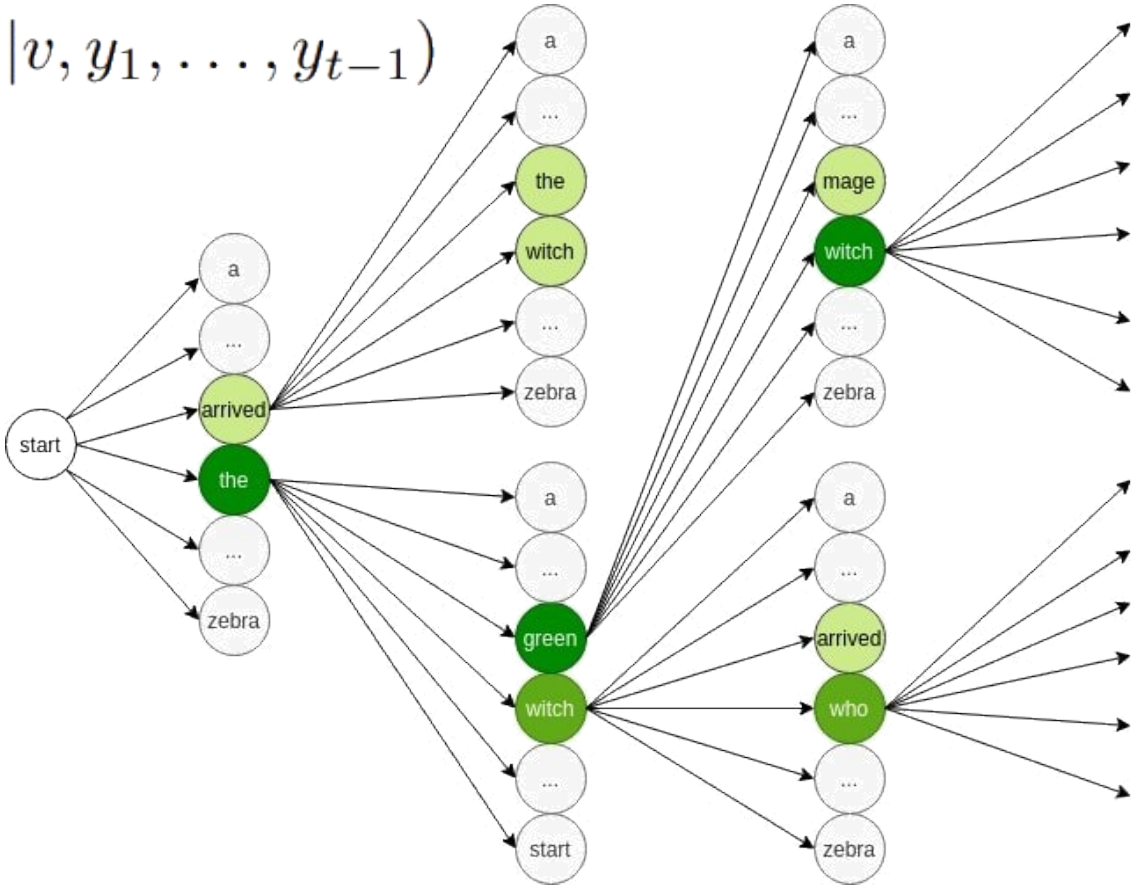
- Training objective

$$1/|\mathcal{S}| \sum_{(T,S) \in \mathcal{S}} \log p(T|S)$$

  - S: source sentence
  - T: correct translation
  - :training set

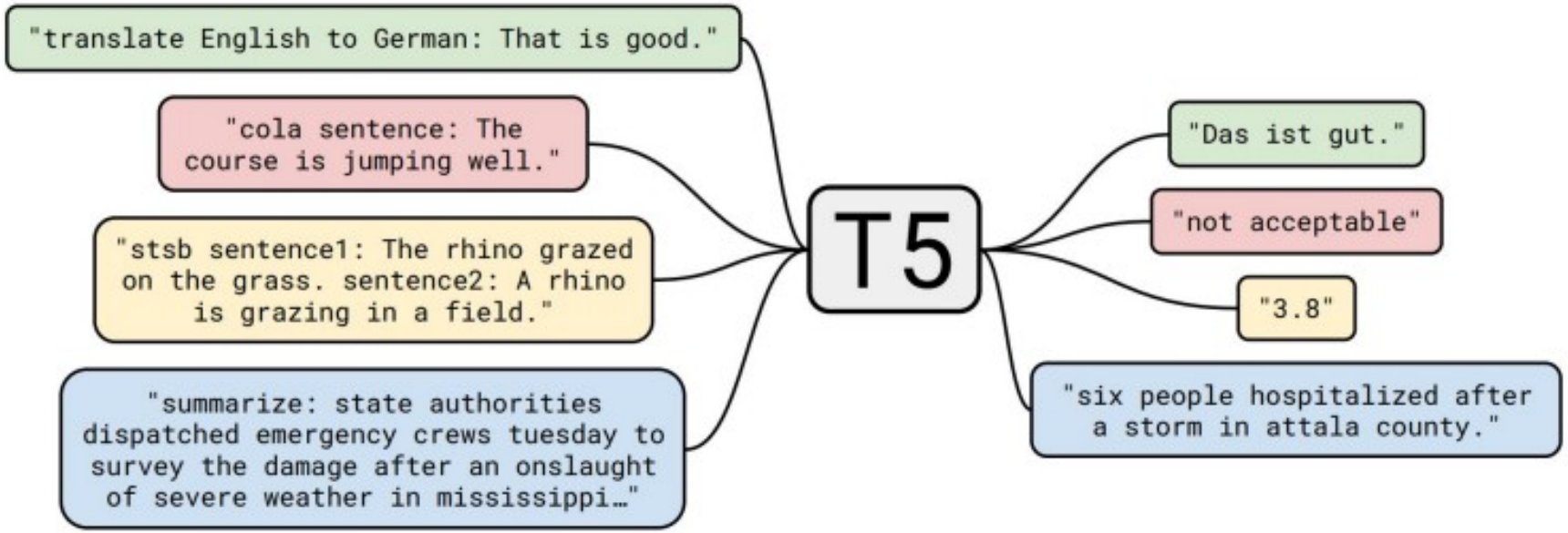
- Inference (beam search)

$$\hat{T} = \arg\max_T p(T|S)$$



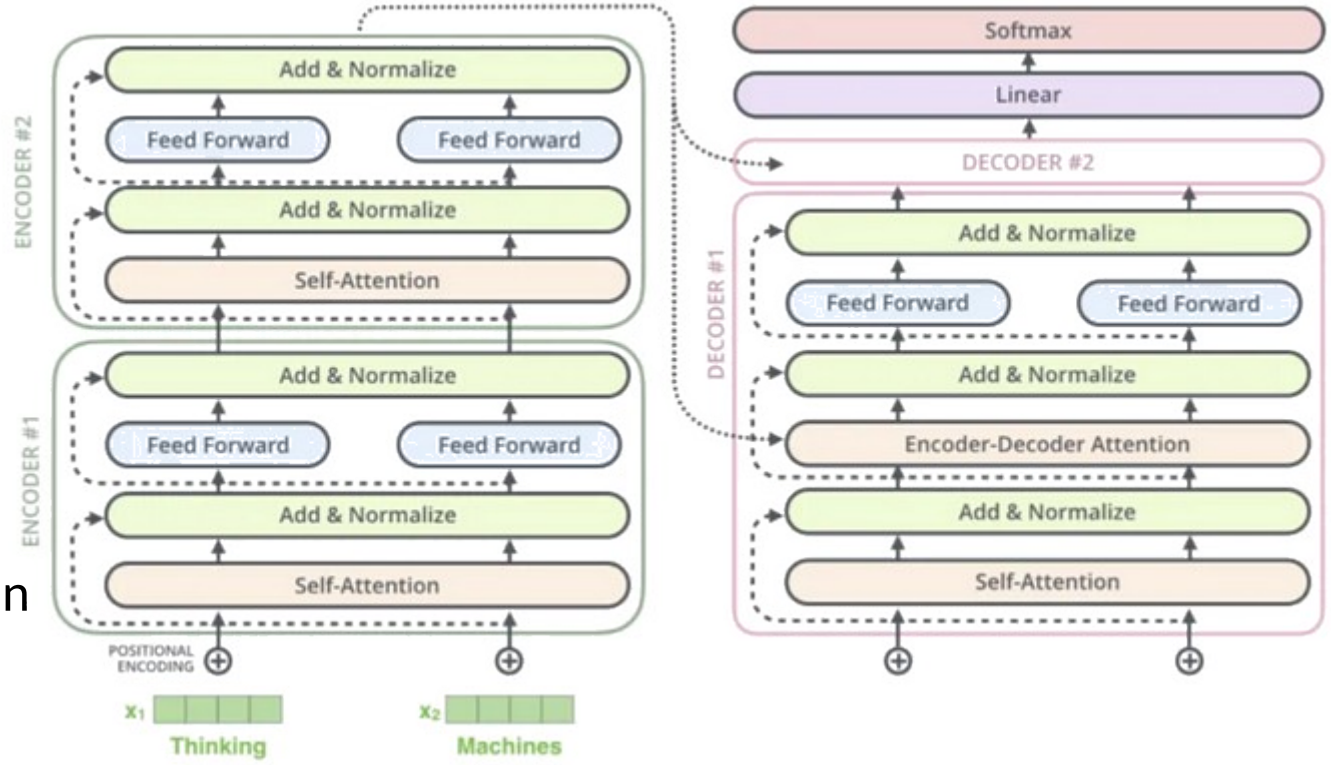
T5 Model (google)

- Text-to-Text Transfer Transformer
- Encoder-decoder model
- “[Task-specific prefix]: [Input text]” -> “[output text]”



Model Architecture

- Roughly equivalent to the original Transformer
  - removing the Layer Norm bias
  - placing the layer normalization outside the residual path
  - different position embedding
    - relative position embeddings
    - a different learned embedding according to the offset between the “key” and “query” being compared in the self-attention mechanism



Data

- Colossal Clean Crawled Corpus (C4)
  - text scraped from the web
  - With a lot of preprocessing
    - Remove pages with dirty/naughty/obscene/bad words
    - Remove code
    - Only English
    - …
  - April 2019
- mC4 (multilingual)
  - 101 languages

Data set	Size
★ C4	745GB
C4, unfiltered	6.1TB
RealNews-like	35GB
WebText-like	17GB
Wikipedia	16GB
Wikipedia + TBC	20GB

Testing on Downstream Tasks

- Sentence acceptability judgment
- Sentiment analysis
- Paraphrasing/sentence similarity
- Natural language inference
- Coreference resolution
- Sentence completion
- Word sense disambiguation
- Question answering
- machine translation
- text summarization

Pretraining

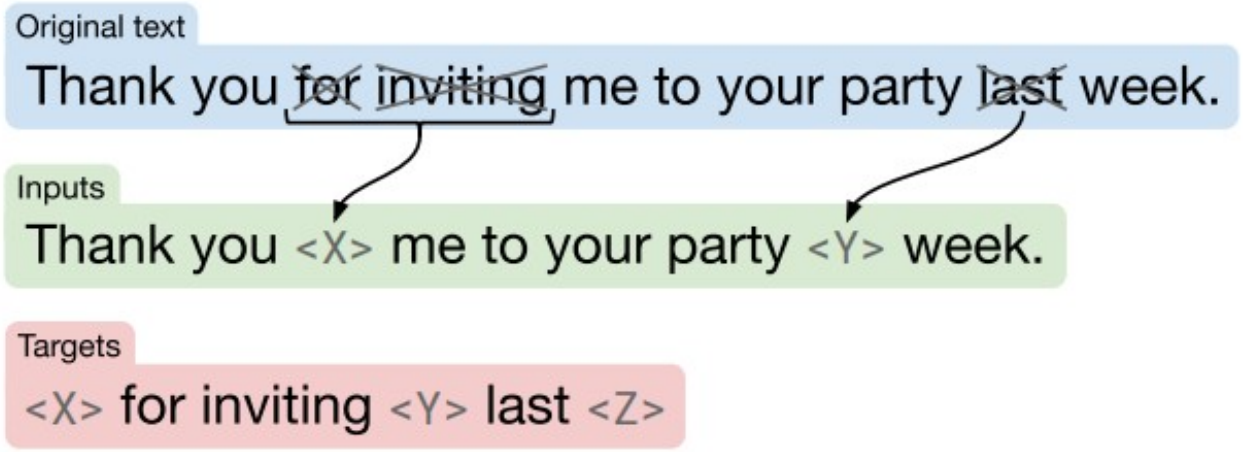
- Even Google has a budget…
- pre-train each model for steps on C4 before fine-tuning
- maximum sequence length of 512
- batch size of 128 sequences (about tokens)
- pre-training on ≈ 34B tokens
  - BERT: 137B tokens
  - RoBERTa: 2.2T tokens
- tokens only covers a fraction of C4
  - never repeat any data during pre-training

Finetuning

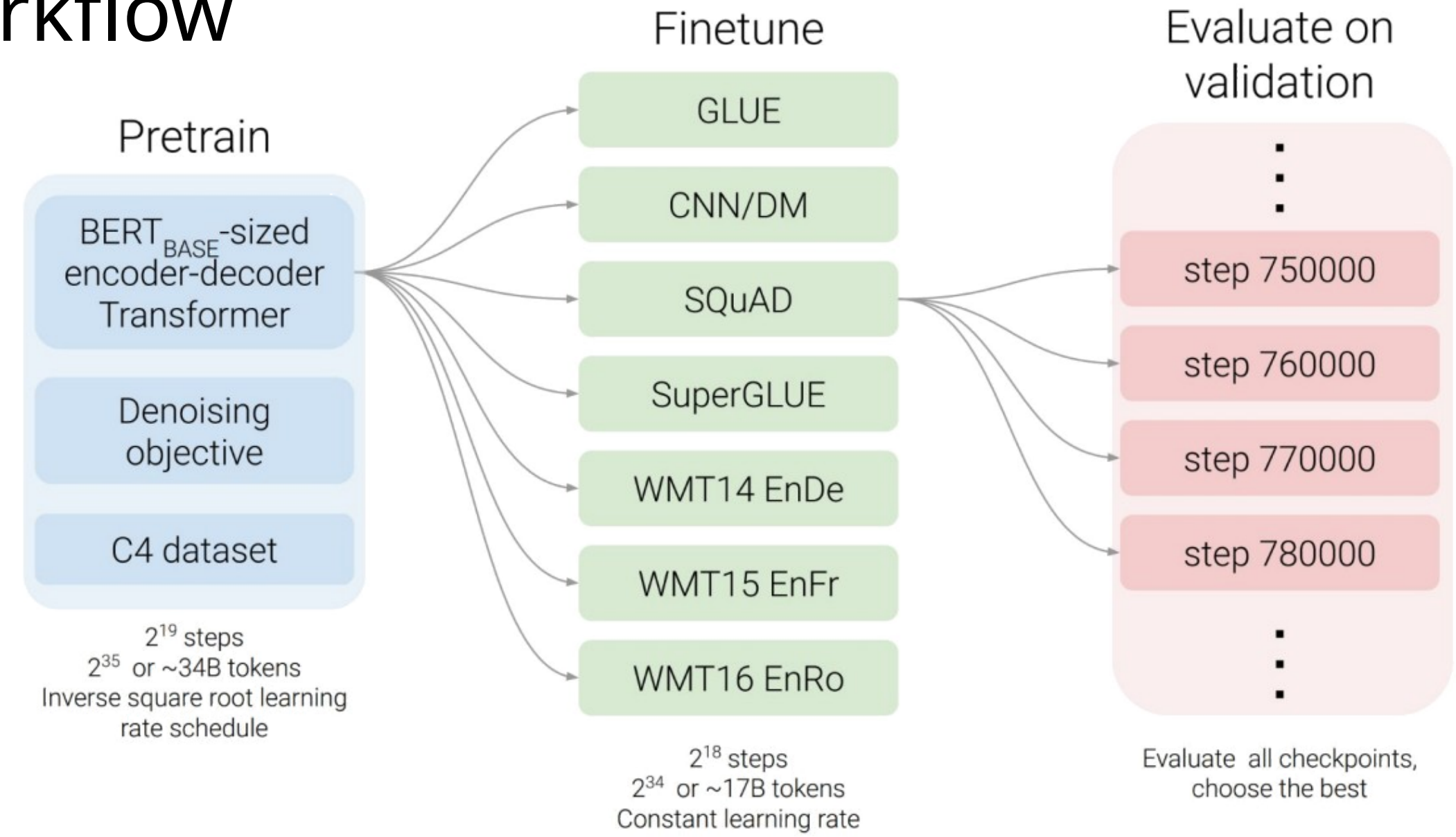
- Fine-tuned for steps on all tasks
- continue using batches with 128 length-512 sequences
- a constant learning rate of 0.001 when fine-tuning
- a checkpoint every 5,000 steps and report results on the model checkpoint corresponding to the highest validation performance

Baseline Objective

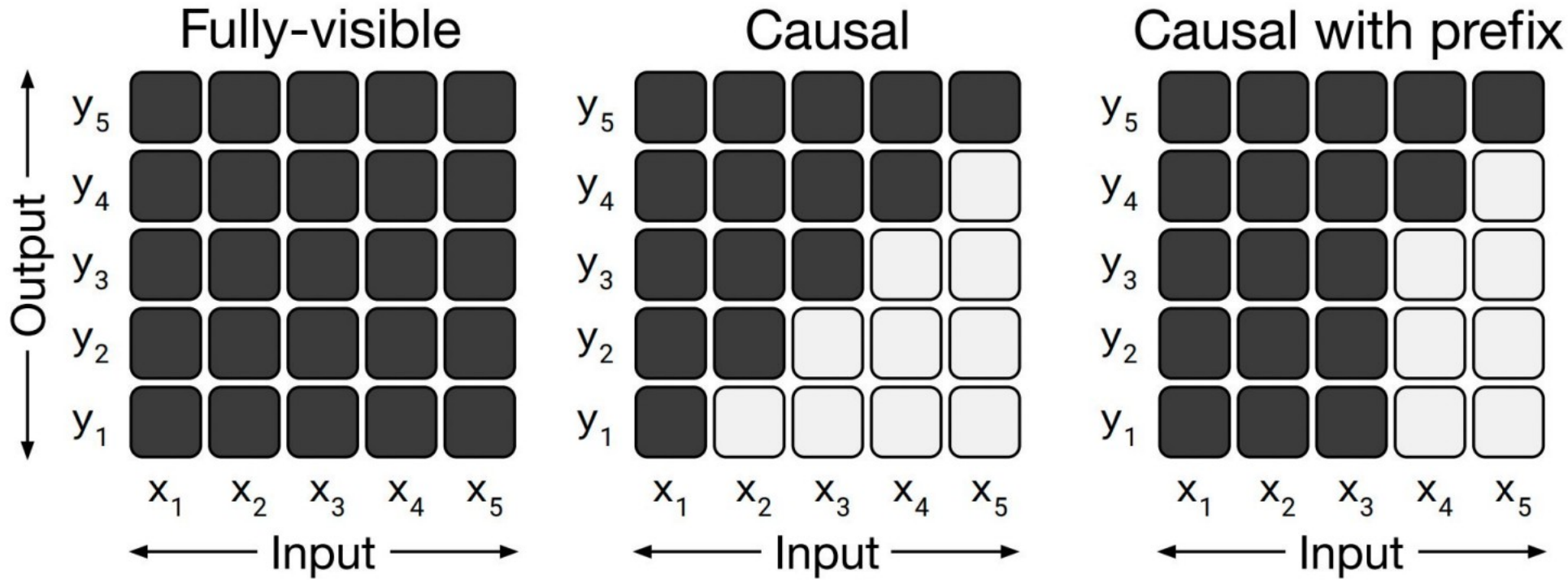
- masked language modeling
- AKA “denoising” objectives



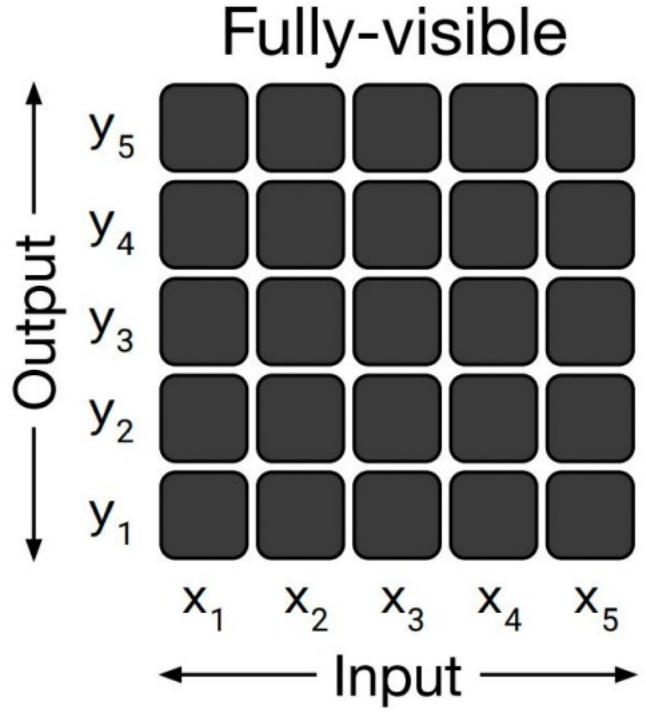
Workflow


Pretrain
BERT BASE -sized encoder-decoder Transformer
Denoising objective
C4 dataset
2^19 steps
2^35 or ~34B tokens
Inverse square root learning rate schedule
Finetune
GLUE
CNN/DM
SQuAD
SuperGLUE
WMT14 EnDe
WMT15 EnFr
WMT16 EnRo
2^18 steps
2^34 or ~17B tokens
Constant learning rate
Evaluate on validation
step 750000
step 760000
step 770000
step 780000
Evaluate all checkpoints, choose the best

Different Attention Mask Patterns

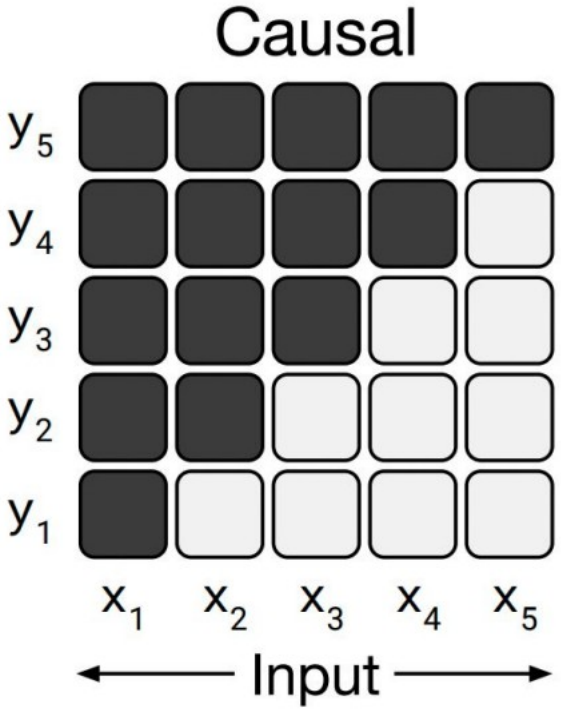


Different Attention Mask Patterns



Fully visible mask allows the self attention mechanism to attend to the full input.

Different Attention Mask Patterns



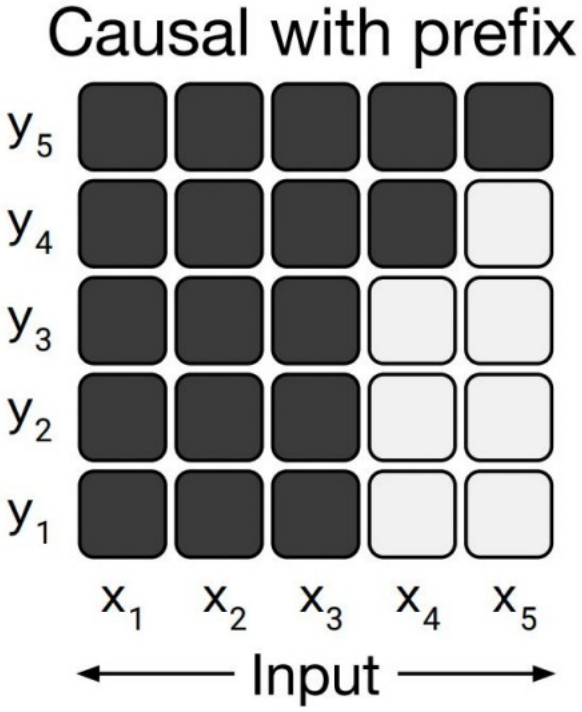
is the scalar weight produced by the self-attention mechanism as a function of and

- causal mask

if

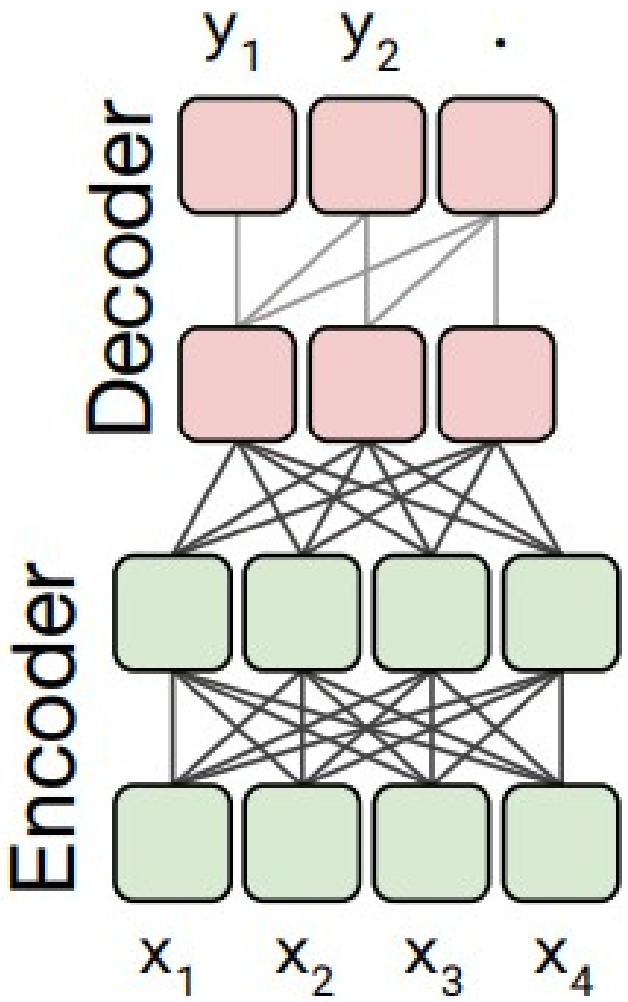
A causal mask doesn’t allow output elements to look into the future

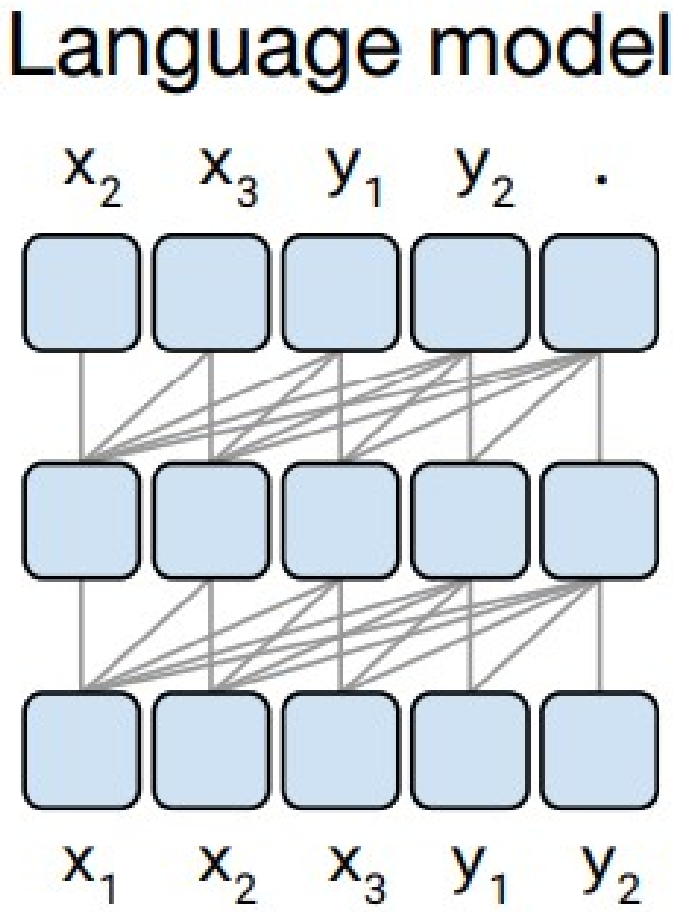
Different Attention Mask Patterns

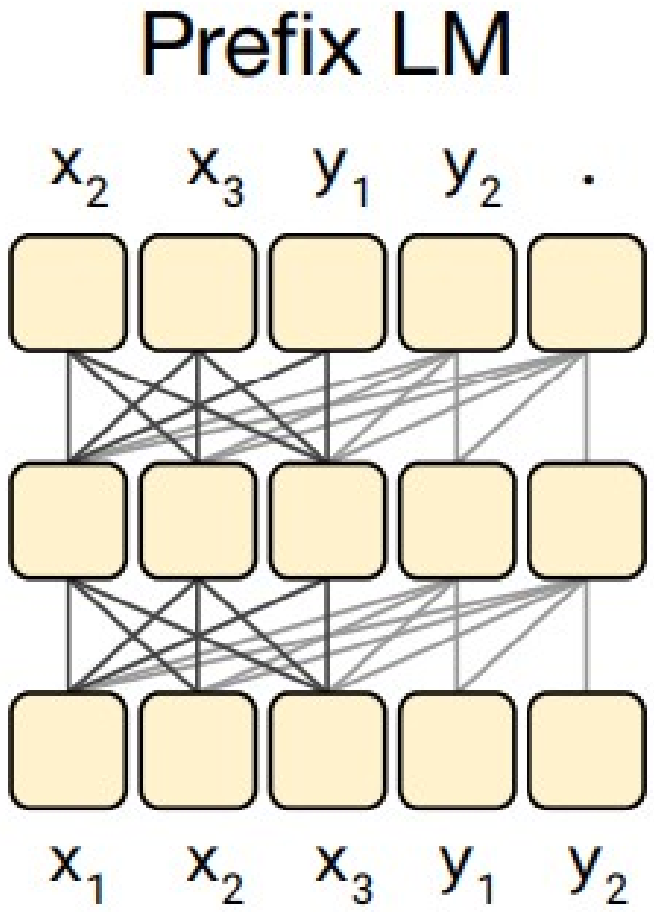


Causal mask with prefix allows to fully-visible masking on a portion of input.

Architectural Variants

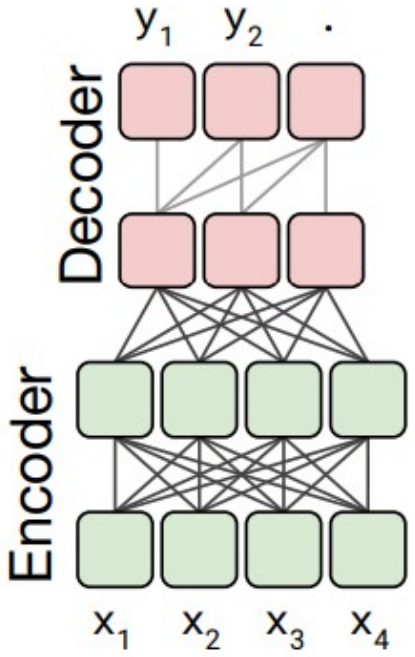






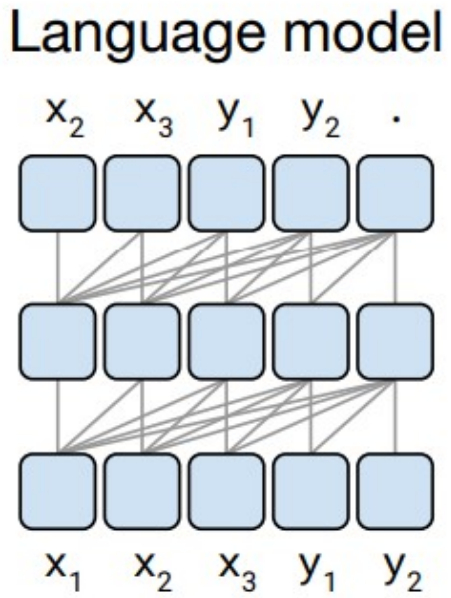
Architectural Variants

- A standard encoder-decoder architecture uses fully visible masking in the encoder and the encoder-decoder attention, with causal masking in the decoder



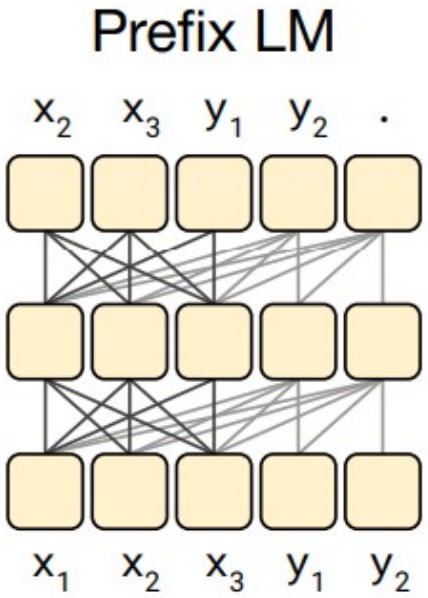
Architectural Variants

- A language model consists of a single Transformer layer stack and is fed the concatenation of the input and target, using a causal mask throughout.



Architectural Variants

- Adding a prefix to a language model corresponds to allowing fully-visible masking over the input



Performance of different Architectural Variants

Architecture	Objective	Params	Cost	GLUE	CNNDM	SQuAD	SGLUE	EnDe	EnFr	EnRo
★ Encoder-decoder	Denoising	$2P$	M	**83.28**	**19.24**	**80.88**	**71.36**	**26.98**	**39.82**	**27.65**
Enc-dec, shared	Denoising	P	M	82.81	18.78	**80.63**	**70.73**	26.72	39.03	**27.46**
Enc-dec, 6 layers	Denoising	P	$M/2$	80.88	18.97	77.59	68.42	26.38	38.40	26.95
Language model	Denoising	P	M	74.70	17.93	61.14	55.02	25.09	35.28	25.86
Prefix LM	Denoising	P	M	81.82	18.61	78.94	68.11	26.43	37.98	27.39

1. Sharing parameters in encoder and decoder models perform nearly as well as the baseline.
2. Halving the number of layers in encoder and decoder hurts the performance.
3. Performance of Encoder and Decoder with shared parameters is better than decoder only LM and prefix LM.

Objective Variants

Objective	Inputs	Targets
Prefix language modeling	Thank you for inviting	me to your party last week .
BERT-style Devlin et al. (2018)	Thank you <M> <M> me to your party apple week .	*(original text)*
Deshuffling	party me for your to . last fun you inviting week Thank	*(original text)*
MASS-style Song et al. (2019)	Thank you <M> <M> me to your party <M> week .	*(original text)*
I.i.d. noise, replace spans	Thank you <X> me to your party <Y> week .	<X> for inviting <Y> last <Z>
I.i.d. noise, drop tokens	Thank you me to your party week .	for inviting last
Random spans	Thank you <X> to <Y> week .	<X> for inviting me <Y> your party last <Z>

- BERT-style objective performs best
  - All the variants perform similarly.
- Prefix LM works well on translation tasks
- Deshuffling objective is significantly worse.

Final choice

- Architecture: Encoder-decoder
- Prediction objective: span-corruption objective
- Pre-training Data set: C4 dataset
- Multi-task pre-training: Unsupervised pre-training + fine-tuning
- Bigger model trained longer

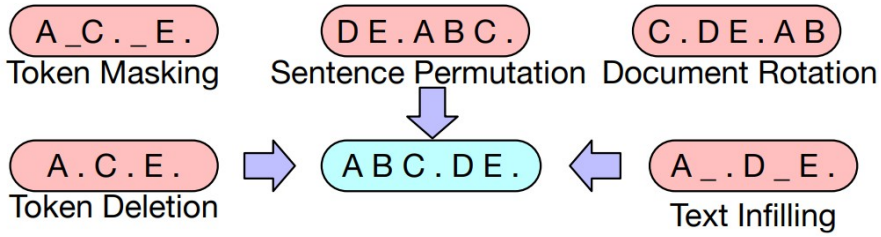
Model Variant

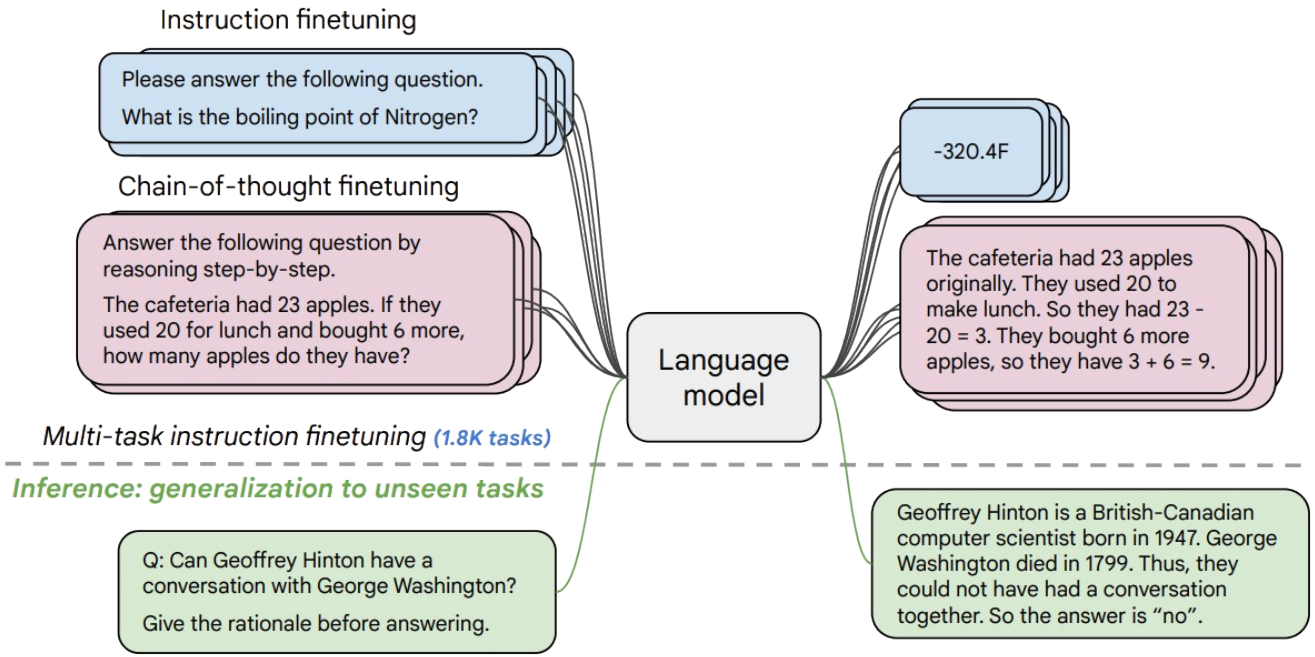
Model	Parameters	No. of layers	d_{model}	d_{ff}	d_{kv}	No. of heads
Small	60M	6	512	2048	64	8
Base	220M	12	768	3072	64	12
Large	770M	24	1024	4096	64	16
3B	3B	24	1024	16384	128	32
11B	11B	24	1024	65536	128	128

Model	GLUE	CNNDM	SQuAD	SGLUE	EnDe	EnFr	EnRo
Previous best	89.4	20.30	95.5	84.6	**33.8**	**43.8**	**38.5**
T5-Small	77.4	19.56	87.24	63.3	26.7	36.0	26.8
T5-Base	82.7	20.34	92.08	76.2	30.9	41.2	28.0
T5-Large	86.4	20.68	93.79	82.3	32.0	41.5	28.1
T5-3B	88.5	21.02	94.95	86.4	31.8	42.6	28.2
T5-11B	**89.7**	**21.55**	**95.64**	**88.9**	32.1	43.4	28.1

Other Variants

- BART (Lewis et al. 2020) Facebook AI
  - Similar Architecture as T5
  - corrupting documents and then optimizing a reconstruction loss
  - Different tokenizers, training data, initialization, position embeddings
  - https://arxiv.org/pdf/1910.13461
- Flan-T5
  - Instruction-Finetuned
  - 1.8K tasks phrased as instructions
  - https://arxiv.org/pdf/2210.11416





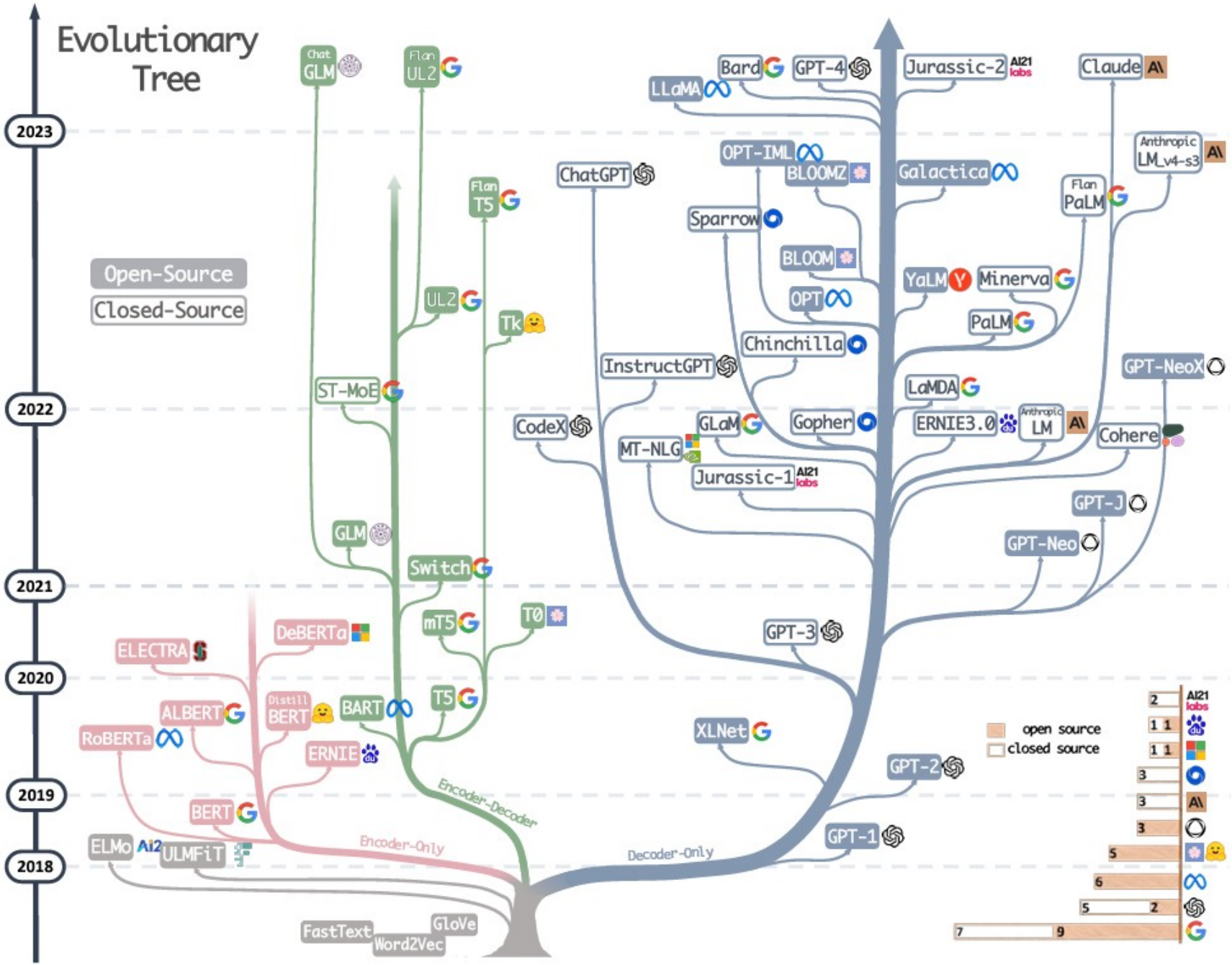


Figure from Yang et al. 2023. Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond. arXiv:2304.13712v2

Evaluation

- BLEU (bilingual evaluation understudy)
  - Machine translation
  - [0,1], the higher the better
  - https://aclanthology.org/P02-1040.pdf
- ROUGE (Recall-Oriented Understudy for Gisting Evaluation)
  - Summarization (and translation)
  - [0,1], the higher the better
  - https://aclanthology.org/W04-1013.pdf
- Both are n-gram based
  - BLEU emphasizes the precision
  - ROUGE focuses on recall

Evaluation

- Reference-based
  - BLEU
  - ROUGE
  - BERTScore
- Reference-free
  - Training with human rating – regression model
  - Training with reference - use reference as good, use corruption as bad
  - Use LLM

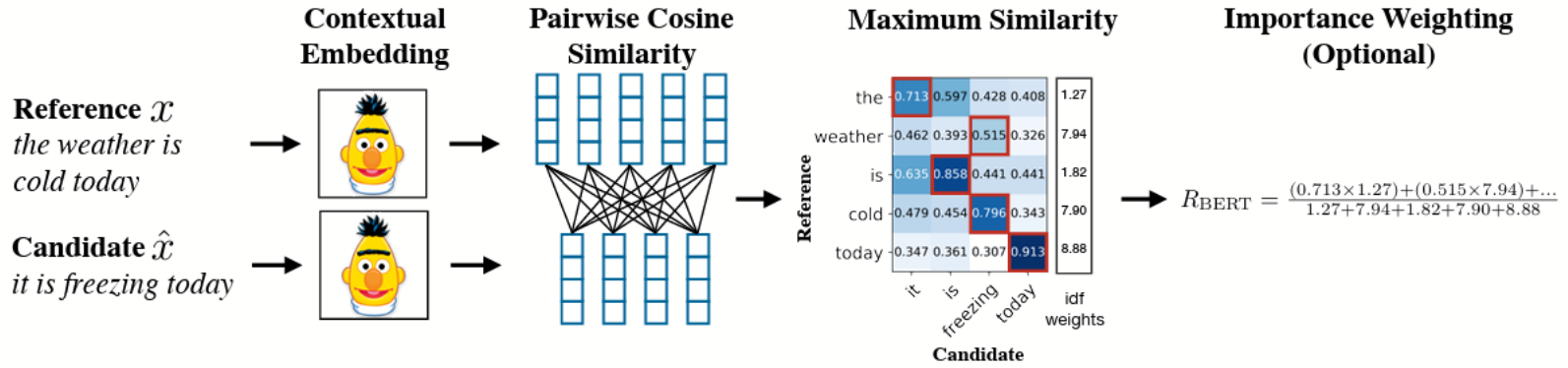


Figure 1: Illustration of the computation of the recall metric R_{BERT}. Given the reference x and candidate $\hat{x}$, we compute BERT embeddings and pairwise cosine similarity. We highlight the greedy matching in red, and include the optional idf importance weighting.

NLI based

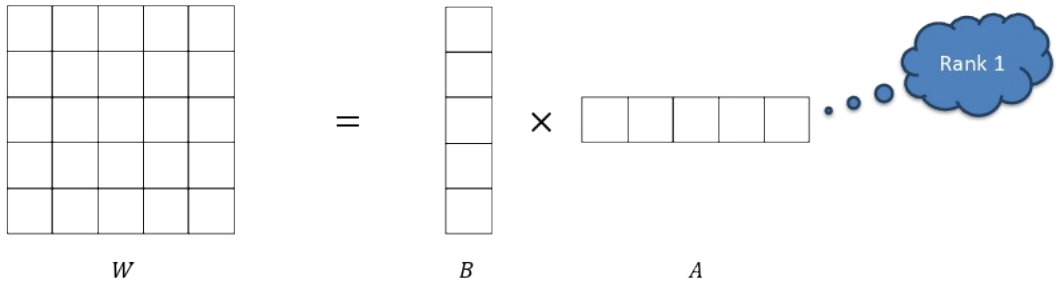
- Reference: 5000 Russian soldiers killed in Ukraine
- Candidate: 5 Ukrainian soldiers wounded in Russia
- BERTScore is high
- NLI: natural language inference
- core upstream tasks in NLP
- Many benchmark datasets

- source should entail summary
- May need adaption

Premise	Relation	Hypothesis
A turtle danced.	entails	A turtle moved.
turtle	contradicts	linguist
Every reptile danced.	neutral	A turtle ate.
Some turtles walk.	contradicts	No turtles move.
James Byron Dean refused to move without blue jeans.	entails	James Dean didn't dance without pants.
Mitsubishi Motors Corp's new vehicle sales in the US fell 46 percent in June.	contradicts	Mitsubishi's sales rose 46 percent.
Acme Corporation reported that its CEO resigned.	entails	Acme's CEO resigned.

LoRA: Low-Rank Adaptation of Large Language Models

- Intuition
  - Low rank matrix decomposition



- Low rank matrix decomposition
  - Sparse Matrix Completion

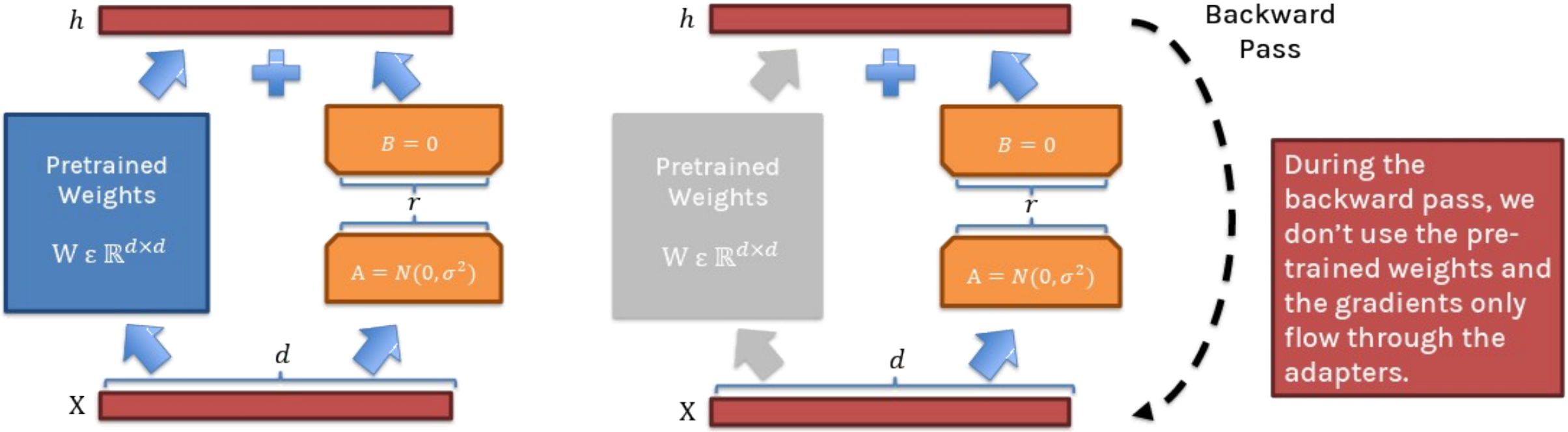
- For fine-tuning
  - Update weight
  - Decompose the update matrix

$$W_0 + \boxed{\Delta W} = W_0 + \frac{\alpha}{r} BA$$

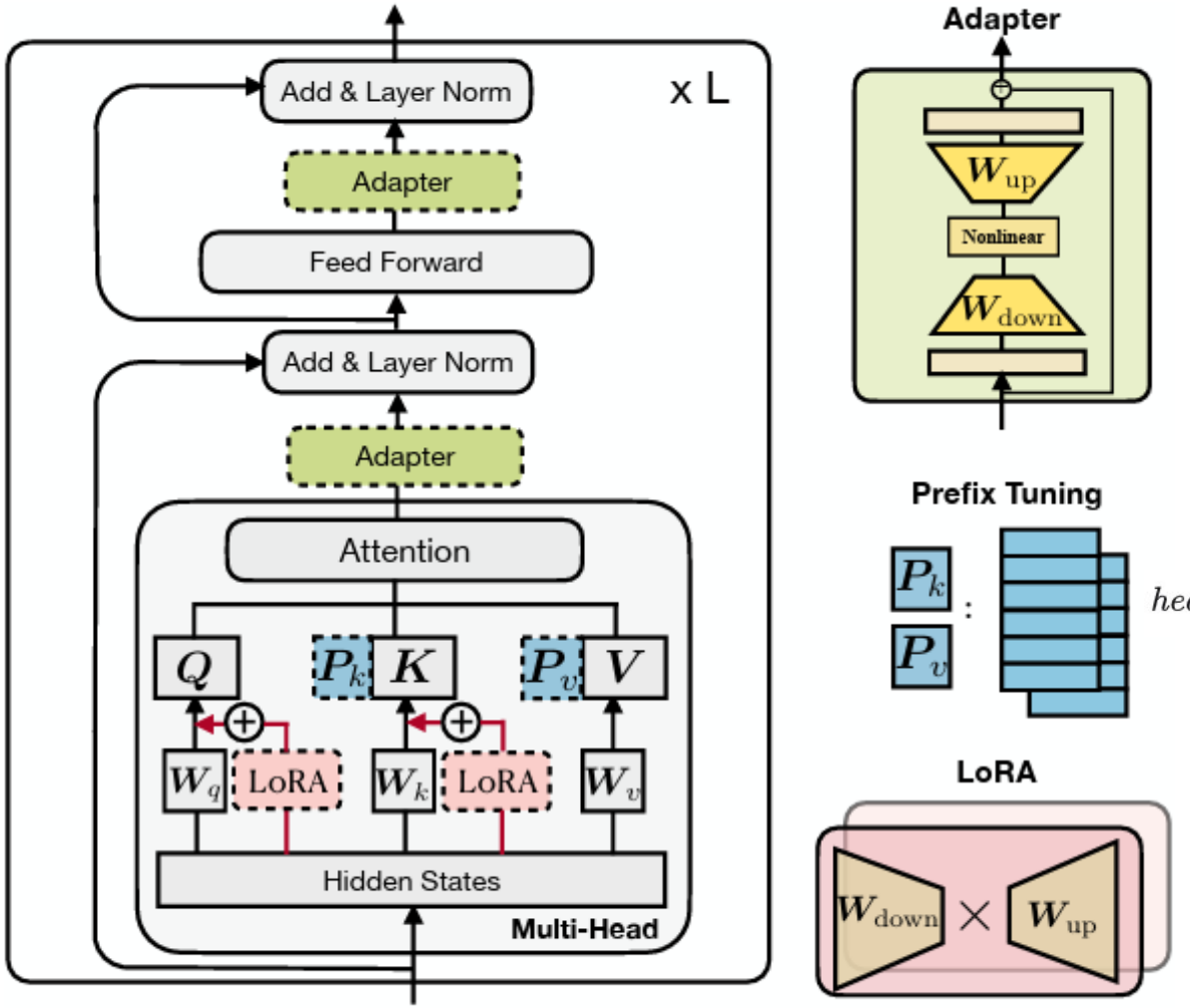
https://arxiv.org/abs/2106.09685
Hu, Edward J., et al. "LoRA: Low-Rank Adaptation of Large Language Models." *ICLR*

LoRA

- the reparameterization (LoRA) runs parallel to the original model.



PEFT



$$h \leftarrow h + f(hW_{down})W_{up}$$

$$head_i = Attn(xW_q^{(i)}, concat(P_k^{(i)}, CW_k^{(i)}), concat(P_v^{(i)}, CW_v^{(i)}))$$

$$h \leftarrow h + s \cdot xW_{down}W_{up}$$

Figure 1: Illustration of the transformer architecture and several state-of-the-art parameter-efficient tuning methods. We use blocks with dashed borderlines to represent the added modules by those methods.

https://arxiv.org/pdf/2110.04366

He, Junxian, et al. "Towards a Unified View of Parameter-Efficient Transfer Learning." *ICLR*

Model & Method	# Trainable Parameters	MNLI	SST-2	MRPC	CoLA	QNLI	QQP	RTE	STS-B	Avg.
RoB_{base} (FT)*	125.0M	**87.6**	94.8	90.2	**63.6**	92.8	**91.9**	78.7	91.2	86.4
RoB_{base} (BitFit)*	0.1M	84.7	93.7	**92.7**	62.0	91.8	84.0	81.5	90.8	85.2
RoB_{base} (Adpt^{D})*	0.3M	$87.1_{\pm.0}$	$94.2_{\pm.1}$	$88.5_{\pm1.1}$	$60.8_{\pm.4}$	$93.1_{\pm.1}$	$90.2_{\pm.0}$	$71.5_{\pm2.7}$	$89.7_{\pm.3}$	84.4
RoB_{base} (Adpt^{D})*	0.9M	$87.3_{\pm.1}$	$94.7_{\pm.3}$	$88.4_{\pm.1}$	$62.6_{\pm.9}$	$93.0_{\pm.2}$	$90.6_{\pm.0}$	$75.9_{\pm2.2}$	$90.3_{\pm.1}$	85.4
RoB_{base} (LoRA)	0.3M	$87.5_{\pm.3}$	$\mathbf{95.1}_{\pm.2}$	$89.7_{\pm.7}$	$63.4_{\pm1.2}$	$\mathbf{93.3}_{\pm.3}$	$90.8_{\pm.1}$	$\mathbf{86.6}_{\pm.7}$	$\mathbf{91.5}_{\pm.2}$	**87.2**
$\text{RoB}_{\text{large}}$ (FT)*	355.0M	90.2	**96.4**	**90.9**	68.0	94.7	**92.2**	86.6	92.4	88.9
$\text{RoB}_{\text{large}}$ (LoRA)	0.8M	$\mathbf{90.6}_{\pm.2}$	$96.2_{\pm.5}$	$\mathbf{90.9}_{\pm1.2}$	$\mathbf{68.2}_{\pm1.9}$	$\mathbf{94.9}_{\pm.3}$	$91.6_{\pm.1}$	$\mathbf{87.4}_{\pm2.5}$	$\mathbf{92.6}_{\pm.2}$	**89.0**
$\text{RoB}_{\text{large}}$ (Adpt^{P})†	3.0M	$90.2_{\pm.3}$	$96.1_{\pm.3}$	$90.2_{\pm.7}$	$\mathbf{68.3}_{\pm1.0}$	$\mathbf{94.8}_{\pm.2}$	$\mathbf{91.9}_{\pm.1}$	$83.8_{\pm2.9}$	$92.1_{\pm.7}$	88.4
$\text{RoB}_{\text{large}}$ (Adpt^{P})†	0.8M	$\mathbf{90.5}_{\pm.3}$	$\mathbf{96.6}_{\pm.2}$	$89.7_{\pm1.2}$	$67.8_{\pm2.5}$	$\mathbf{94.8}_{\pm.3}$	$91.7_{\pm.2}$	$80.1_{\pm2.9}$	$91.9_{\pm.4}$	87.9
$\text{RoB}_{\text{large}}$ (Adpt^{H})†	6.0M	$89.9_{\pm.5}$	$96.2_{\pm.3}$	$88.7_{\pm2.9}$	$66.5_{\pm4.4}$	$94.7_{\pm.2}$	$92.1_{\pm.1}$	$83.4_{\pm1.1}$	$91.0_{\pm1.7}$	87.8
$\text{RoB}_{\text{large}}$ (Adpt^{H})†	0.8M	$90.3_{\pm.3}$	$96.3_{\pm.5}$	$87.7_{\pm1.7}$	$66.3_{\pm2.0}$	$94.7_{\pm.2}$	$91.5_{\pm.1}$	$72.9_{\pm2.9}$	$91.5_{\pm.5}$	86.4
$\text{RoB}_{\text{large}}$ (LoRA)†	0.8M	$\mathbf{90.6}_{\pm.2}$	$96.2_{\pm.5}$	$\mathbf{90.2}_{\pm1.0}$	$68.2_{\pm1.9}$	$\mathbf{94.8}_{\pm.3}$	$91.6_{\pm.2}$	$\mathbf{85.2}_{\pm1.1}$	$\mathbf{92.3}_{\pm.5}$	**88.6**
DeB_{XXL} (FT)*	1500.0M	91.8	**97.2**	92.0	72.0	**96.0**	92.7	93.9	92.9	91.1
DeB_{XXL} (LoRA)	4.7M	$\mathbf{91.9}_{\pm.2}$	$96.9_{\pm.2}$	$\mathbf{92.6}_{\pm.6}$	$\mathbf{72.4}_{\pm1.1}$	$\mathbf{96.0}_{\pm.1}$	$\mathbf{92.9}_{\pm.1}$	$\mathbf{94.9}_{\pm.4}$	$\mathbf{93.0}_{\pm.2}$	**91.3**

Model & Method	# Trainable Parameters	E2E NLG Challenge				
		BLEU	NIST	MET	ROUGE-L	CIDEr
GPT-2 M (FT)*	354.92M	68.2	8.62	46.2	71.0	2.47
GPT-2 M (Adapter^{L})*	0.37M	66.3	8.41	45.0	69.8	2.40
GPT-2 M (Adapter^{L})*	11.09M	68.9	8.71	46.1	71.3	2.47
GPT-2 M (Adapter^{H})	11.09M	$67.3_{\pm.6}$	$8.50_{\pm.07}$	$46.0_{\pm.2}$	$70.7_{\pm.2}$	$2.44_{\pm.01}$
GPT-2 M (FT^{Top2})*	25.19M	68.1	8.59	46.0	70.8	2.41
GPT-2 M (PreLayer)*	0.35M	69.7	8.81	46.1	71.4	2.49
GPT-2 M (LoRA)	0.35M	$\mathbf{70.4}_{\pm.1}$	$\mathbf{8.85}_{\pm.02}$	$\mathbf{46.8}_{\pm.2}$	$\mathbf{71.8}_{\pm.1}$	$\mathbf{2.53}_{\pm.02}$
GPT-2 L (FT)*	774.03M	68.5	8.78	46.0	69.9	2.45
GPT-2 L (Adapter^{L})	0.88M	$69.1_{\pm.1}$	$8.68_{\pm.03}$	$46.3_{\pm.0}$	$71.4_{\pm.2}$	$\mathbf{2.49}_{\pm.0}$
GPT-2 L (Adapter^{L})	23.00M	$68.9_{\pm.3}$	$8.70_{\pm.04}$	$46.1_{\pm.1}$	$71.3_{\pm.2}$	$2.45_{\pm.02}$
GPT-2 L (PreLayer)*	0.77M	70.3	8.85	46.2	71.7	2.47
GPT-2 L (LoRA)	0.77M	$\mathbf{70.4}_{\pm.1}$	$\mathbf{8.89}_{\pm.02}$	$\mathbf{46.8}_{\pm.2}$	$\mathbf{72.0}_{\pm.2}$	$2.47_{\pm.02}$

LoRA Summary

- Advantages
  - Much faster
  - Finetuning can be achieved using less GPU memory
  - Cost efficient
  - Less prone to “catastrophic forgetting” since the original model weights are kept the same.
- QLoRA, an extended version of LoRA
  - Q: quantization. Think of quantization as ‘splitting range into buckets’
  - More memory reduction, slightly slower speed
  - https://arxiv.org/abs/2305.14314