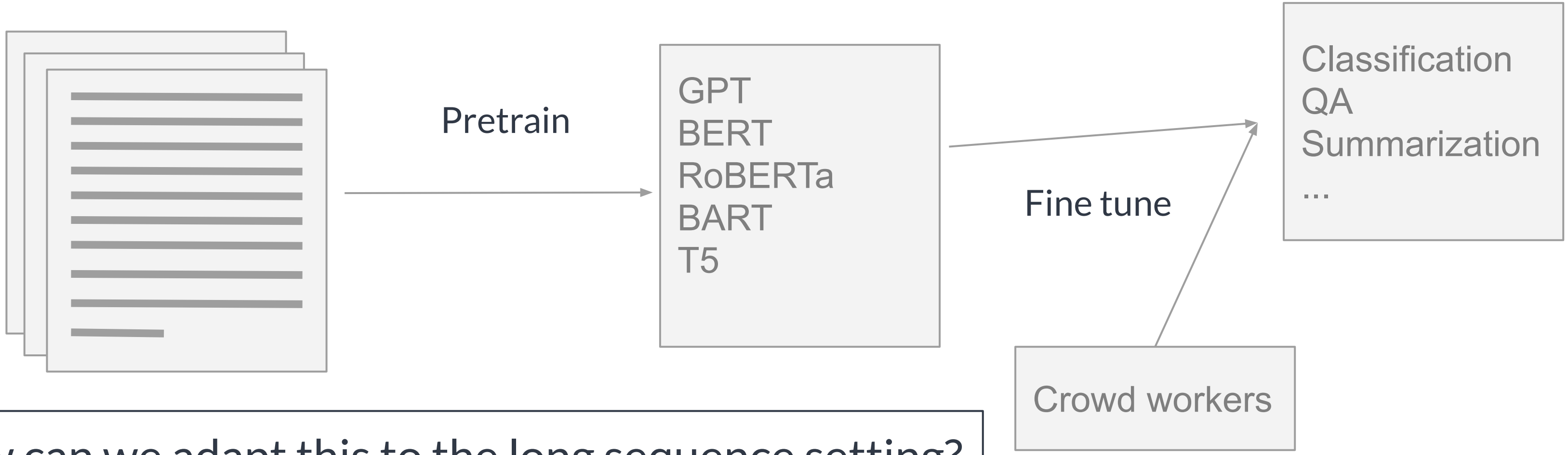
Pre-training and fine-tuning

Overview

- Significant recent advances in NLP with transfer learning
- Two steps:
  - Pretrain a large language model with self-supervised learning on a large dataset
  - Fine tune the model with supervised learning for a target task (or multiple tasks)



How can we adapt this to the long sequence setting?

Prior work

A lot of prior work in long sequence transformers does not evaluate in the transfer setting.

In transfer setting:

- Option 1: Pretraining from random initialization:
  - Linformer (Wang et al 2020)
  - Nyströmformer: (Xiong et al. 2021))
- Option 2: Adapt pretrained short model:
  - Longformer (Beltagy et. al 2020): Initializes from RoBERTa (Liu et al. 2019) and BART (Lewis et al. 2019)
  - ETC (Ainslie et al. 2020) and BigBird (Zaheer et al. 2020): Initializes from RoBERTa and Pegasus (Zhang et al. 2019)

Reusing shorter models

- Training language models (short or long) uses large amounts of compute (RoBERTa-large estimated $250,000 to train).
- We'd like to re-use a shorter model if possible instead of training from scratch.

Reusing shorter models

- Training language models (short or long) uses large amounts of compute (RoBERTa-large estimated $250,000 to train).
- We'd like to re-use a shorter model if possible instead of training from scratch.

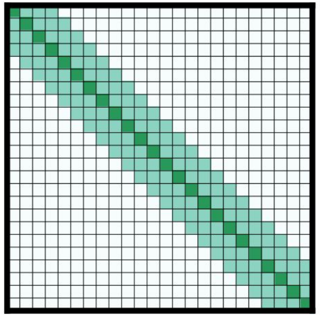
Carefully initialize a long sequence model from an existing short sequence one in a way that allows convergence of long model with minimal additional compute.

Gradually growing sequence length

- If training from scratch can gradually grow sequence length.
  - Shorter models more efficient then longer ones
  - Most language understanding uses local context, and before models can effectively use long context they must learn to use local context.
- Especially useful with relative position embeddings (e.g. Transformer XL Dai et. al 2019) or position infused attention (Shortformer, Press et al. 2020).
- Can also be combined with larger window sizes.

Gradually growing sequence length

Number of phases	5
Phase 1 window sizes	32 (bottom layer) - 8,192 (top layer)
Phase 5 window sizes	512 (bottom layer) - (top layer)
Phase 1 sequence length	2,048
Phase 5 sequence length	23,040 (gpu memory limit)
Phase 1 LR	0.00025
Phase 5 LR	000015625
Batch size per phase	32, 32, 16, 16, 16
#Steps per phase (small)	430K, 50k, 50k, 35k, 5k
#Steps per phase (large)	350K, 25k, 10k, 5k, 5k



Longformer used 5 phases gradually growing sequence length and window sizes to train autoregressive LM.

Gradually growing sequence length

	Train	Inference (Test)		
Model	Speed ↑	Mode	Speed ↑	PPL ↓
Baseline	13.9k	N.o.	**14.7k**	19.4
		S.W.	2.5k	18.70
Baseline + Staged Train.	17.6k	S.W.	2.5k	17.56
Shortformer	**22.9k**	N.o.	14.5k	18.15

- Shortformer used 2 phases, evaluation using Wikitext 103.
- Baseline model is $O(N^2)$ Transformer with 3,072 sequence length.
- Staged training using two stages with 128 length first stage.
- Full model combines staged training with position infused attention (adds position information to Q/K matrices instead of word embeddings).

Initializing from pre-trained model

- Significantly speed up convergence and improve performance by initializing from existing shorter pre-trained model such as BERT.

Initializing from pre-trained model

- Significantly speed up convergence and improve performance by initializing from existing shorter pre-trained model.
- Position embeddings: Longformer / ETC / BigBird initialized from RoBERTa with absolute learned position embeddings.
  - Longformer: used “copy initialization” for new position embeddings
  - ETC/BigBird uses relative position embeddings randomly initialized

Initializing from pre-trained model

- Significantly speed up convergence and improve performance by initializing from existing shorter pre-trained model.
- Position embeddings: Longformer / ETC / BigBird initialized from RoBERTa with absolute learned position embeddings.
  - Longformer: used "copy initialization" for new position embeddings
  - ETC/BigBird uses relative position embeddings randomly initialized
- Local/global Q/K/V attention projection matrix parameters initialized to pre-trained values.

Initializing from pre-trained model

Source	Tokens	Avg doc len
Books (Zhu et al., 2015)	0.5B	95.9K
English Wikipedia	2.1B	506
Realnews (Zellers et al., 2019)	1.8B	1.7K
Stories (Trinh and Le, 2018)	2.1B	7.8K

Important to select corpora with long documents for additional pre-training.

Importance of initialization

Model	base	large
RoBERTa (seqlen: 512)	1.846	1.496
Longformer (seqlen: 4,096)	10.299	8.738

- Random initialization leads to poor performance.

Importance of initialization

Model	base	large
RoBERTa (seqlen: 512)	1.846	1.496
Longformer (seqlen: 4,096)	10.299	8.738
+ copy position embeddings	1.957	1.597

- Random initialization leads to poor performance.
- Copy initialization significantly reduces loss.

Importance of initialization

Model	base	large
RoBERTa (seqlen: 512)	1.846	1.496
Longformer (seqlen: 4,096)	10.299	8.738
+ copy position embeddings	1.957	1.597
+ 2K gradient updates	1.753	1.414
+ 65K gradient updates	1.705	1.358

- Random initialization leads to poor performance.
- Copy initialization significantly reduces loss.
- Model converges quickly with a little fine tuning.
- 2K gradient updates = 0.5B tokens = 1/4000 pre-training compute of RoBERTa.
- ETC: 260B tokens = ⅛ pre-training compute of RoBERTa.

Initializing from pre-trained model

Model	*Input length*	*Configuration*	*#Params*	*Long answer F1*	*Short answer F1*
ETC-large	4096		539M	0.761	0.565
ETC-large	4096	lifting from RoBERTa	558M	**0.782**	0.585

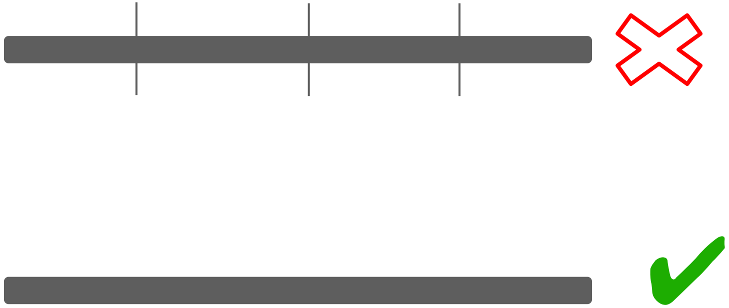
Initializing from pre-trained models improves overall performance.

ETC (Ainslie et al. 2020) evaluation on Natural Questions with randomly initialized model, and one initialized from RoBERTa (with randomly initialized position embeddings).

Both models pre-trained for 260B tokens (=⅛ RoBERTa pre-training).

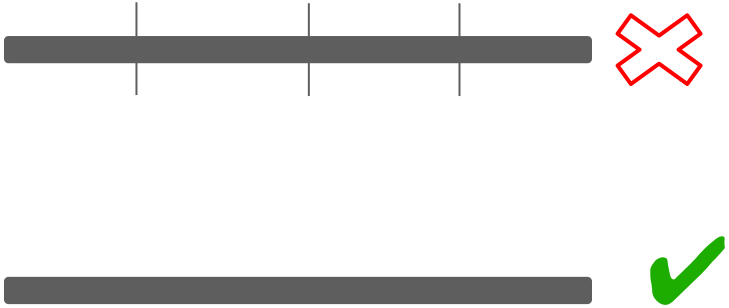
Fine-tuning for downstream tasks

The ability to process long sequence lengths can significantly reduce complexity by removing need for chunking.

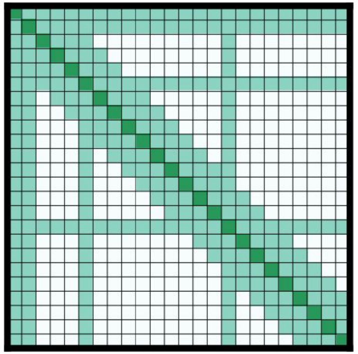


Fine-tuning for downstream tasks

The ability to process long sequence lengths can significantly reduce complexity by removing need for chunking.



Longformer / ETC / BigBird need to specify the global attention pattern, and it does not need to follow pre-training pattern.



UWNLP AI2

Fine-tuning for classification

For document classification, apply global attention to [CLS] or other special tokens used for prediction.

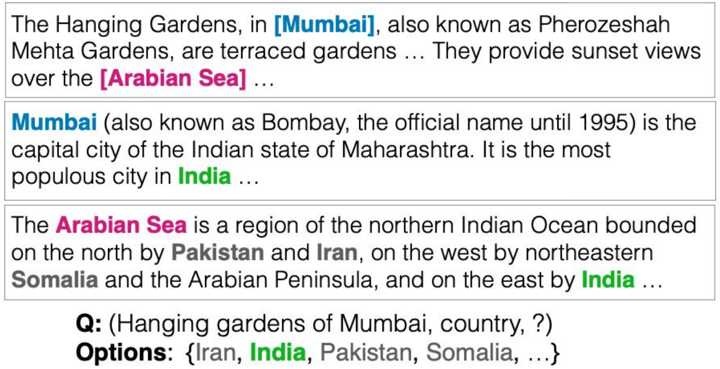
predict

Global attn.

Local attn.

<s> The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments … a small fraction of the training costs of the best models from the literature. </s>

Fine-tuning for multi-hop QA



WikiHop QA ([Welbl et. al 2018](#))

Each instance has question, list of answers, and many support paragraphs.

Avg. support len=1500, 95th percentile=3600

predict   predict

<s> [q] question [/q] [ent] choice$_1$ [/ent] . . . [ent] choice$_N$ [/ent] </s> context$_1$ </s> . . . </s> context$_M$ </s>

Global attn.   Local attn.

Pretraining global attention

BigBird-ETC and Longformer have separate projection matrices for global vs. local attention. Can pretrain global attention with additional objective function.

- BigBird uses CPC objective (modified from Oord et al. 2018) to pretrain global attention. Assign masked segment of long input to a global attention token and match representation vs. encoding unmasked segment.

Pretraining global attention

BigBird-ETC and Longformer have separate projection matrices for global vs. local attention. Can pretrain global attention with additional objective function.

- BigBird uses CPC objective (modified from Oord et al. 2018) to pretrain global attention. Assign masked segment of long input to a global attention token and match representation vs. encoding unmasked segment.
- CDLM (Caciularu et al. 2021) applies global attention to masked tokens during training → learn long range dependencies, improved results for cross-document tasks.