Transformer configuration

Decoder:

$\rightarrow$ masked MH-SA

[Diagram of masked MH-SA]

Prefix LM:

[Diagram of partially masked MH-SA]

- $p_1$: complete
- $p_2$: sins
- $p_3$: phrase

- $c_1$: students
- $c_2$: opened
- $c_3$: their

Token we want to generate

Useful for text generation
Encoder:

- Students opened their books

- Useful for computing representations of a sequence of text that can then be used in other applications

- Cannot generate text!

- Ex: BERT, RoBERTa, ELECTRA
Encoder / Decoder model:
(sequence-to-sequence model)

Cross-attention always uses the representations from the final layer of the encoder.
1. Pretraining
   - Self-supervised objective
   - Language modeling
   - Use as much data as you can find
   - Biggest model you can afford
   - Goal: a model that understands many linguistic properties
     - Grammar
     - World knowledge
     - "The President of the USA is ---"
     - "Emergent properties"
   - We aren't focusing on a specific task or application

2. Fine-tuning
   - Smaller labeled dataset corresponding to a single task/domain of interest
   - Goal: maximize perf on this task/domain
   - Parameter adaptation
   - Parameter-efficient adaptation
Step 1: Pretraining:

Randomly initialized Transformer decoder $\xrightarrow{\text{train on}}$ pre-trained Transformer decoder $\xrightarrow{\text{transfer learning}}$

Step 2: Fine-tuning:

Pre-trained Transformer decoder $\xrightarrow{\text{train on}}$ small task/domain specific dataset $\Rightarrow$ Finetuned Transformer decoder specialized for our task
BERT:

→ example of the encoder paradigm
→ pretraining
→ masked LM
→ fine-tuning
→ adapt to "downstream" task

Pretraining BERT:

```
[MASK] their book

15%

→ usually 15% of tokens are masked

→ not good for text generation
```