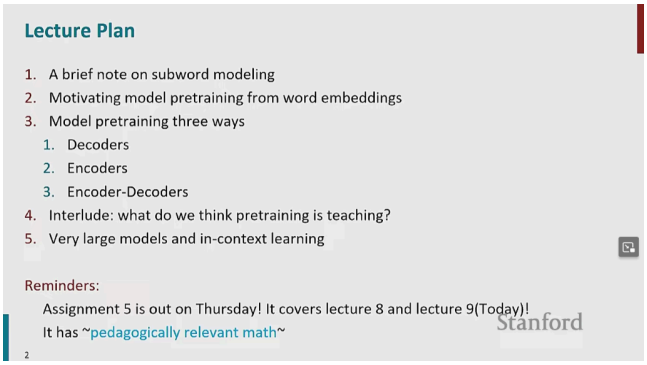
Notes

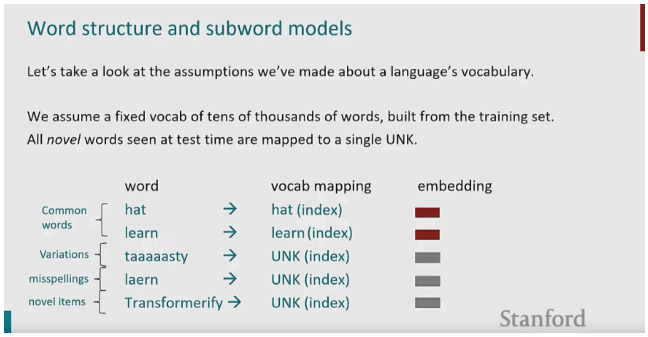
25 April 2024 06:44

Lecture Outline:



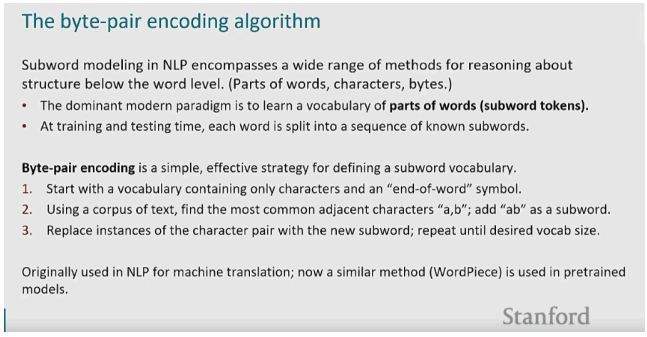
Word Structures and Subword modelling

- Finite Vocab assumption:
  - Initially a fixed-known set of words (called as vocab) were used in training. Any new/unknown word that comes in during test time would be mapped to UNK I.e. unknown. A big information loss



  - Such finite vocab assumption also does not help with languages having a complex word structure – because all the new words, even if they are conjugation/derivation of a known word will be mapped to UNK

- Byte-pair encoding: How do we get information below word level?



  - Most common words end up as subword in vocab but most unknown words are split in sub words (sometimes known, intuitive or sometimes unknown)

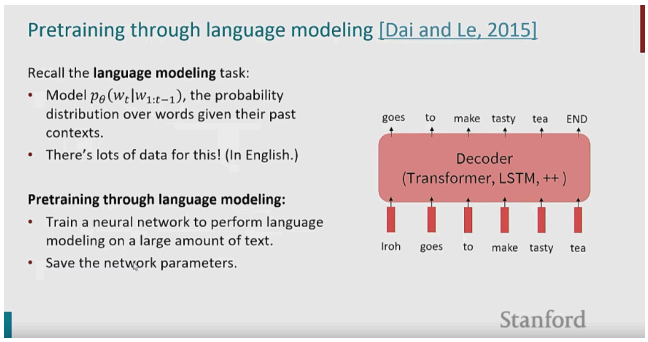
Before (end-to-end) pretraining:

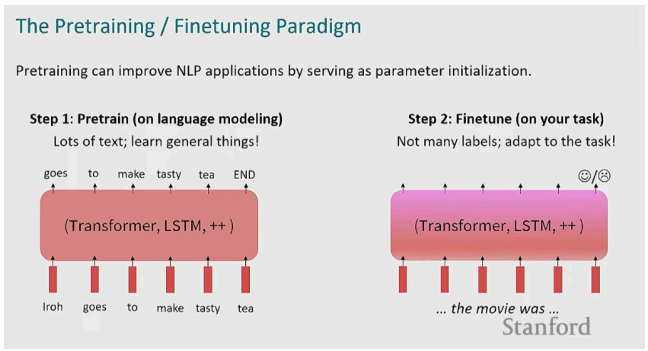
- We used learned **word embeddings** – the context of a particular use case was missing. I.e. How the words are being used in a sentence in a particular use case was not considered when the word embeddings were learnt.
- This forces user to have a huge training dataset of a particular use case for fine tuning.

(end-to-end) Pretraining:

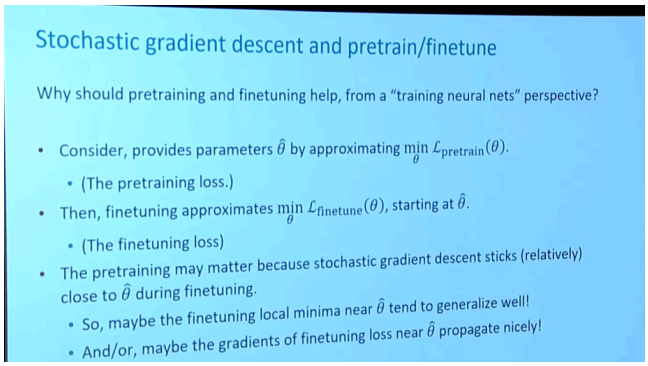
- How do we pretrain?
  - Masked-language modelling – by masking a specific word/part of the input sentence and asking the NN to recover the information

- What does this teach the network?
  - Geo-locations
    - e.g. Stanford University is located in _____, in California
  - Sense of physical world:
    - I put fork _____ the table
  - Entity relation
    - The woman walked across the street by checking for traffic over _____ shoulder
  - AND so on, we can teach the network to do sentiment analysis,

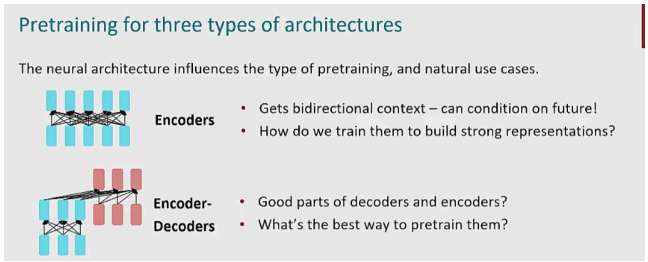


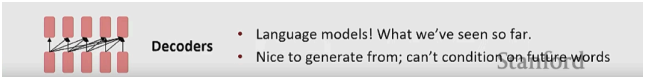


- Why to use pretrained NN parameters as starting point for fine tuning?
  - It offers just a better starting point for finding the minima in loss landscape



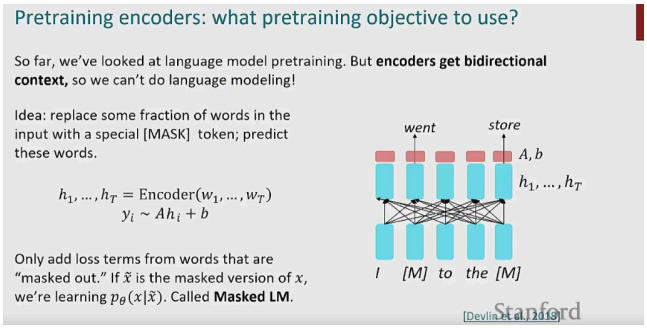
Three ways of pre-training:



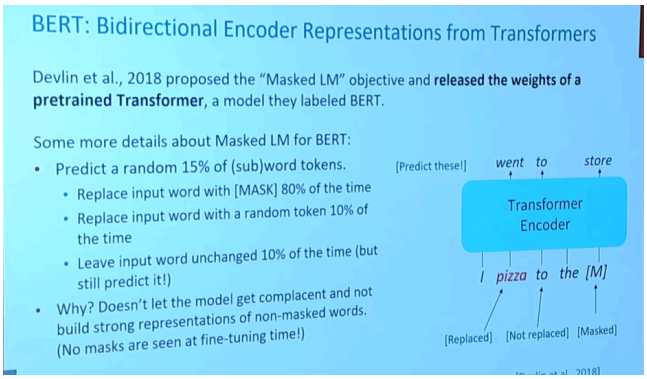


Encoder:

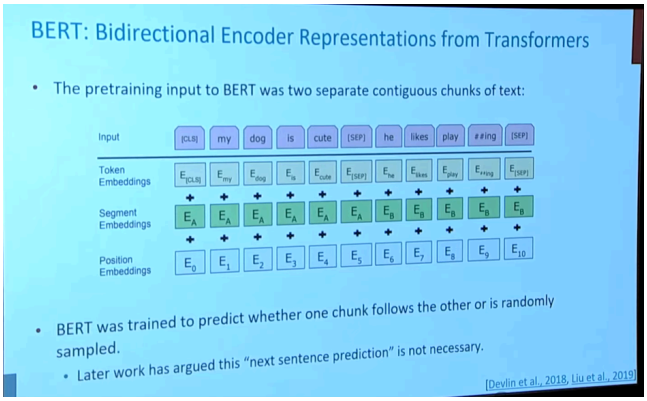
- As encoder have bi-directional context (remember RNN/LSTM lecture?) -- we can't do language modelling with them
- So –Hide some words from the model and ask model to predict the words



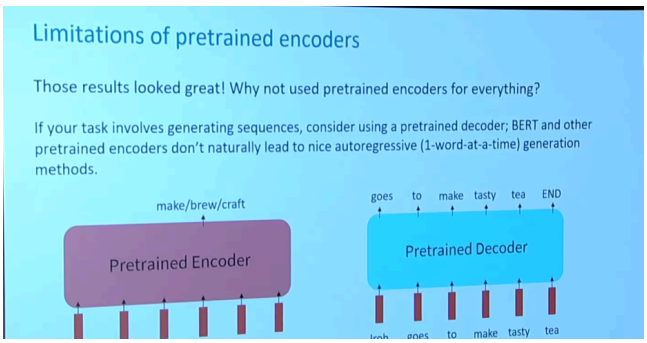
- e.g. of an (all) encoder model – **BERT** I.e. Bidirectional Encoder Representation from Transformers



- Adding long-distance dependency/coherence to BERT – by feeding two segments A and B as input and asking model if segment B is logical continuation of A

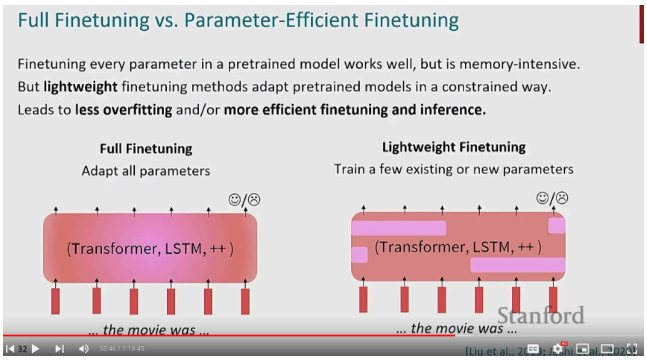


- **Limitations of (all) encoder models**: BERT is trained for filling the blanks, hence it works best when broader context is available and some words are to be predicted. This is not ideal for e.g. Sentence Generation (ChatGPT) like application

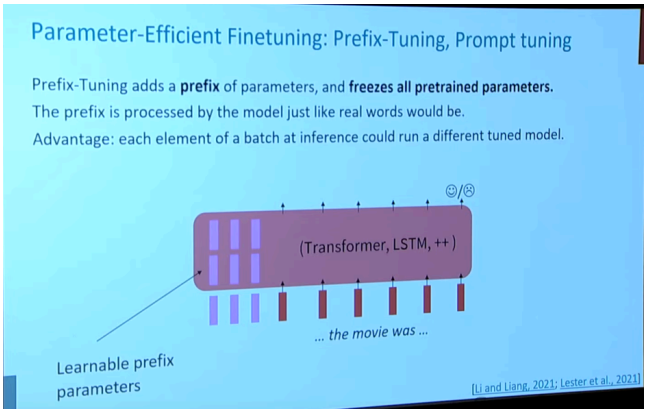


- Extensions of BERT:
  - RoBERT: trained BERT for long(er) duration and removed next sentence prediction.
  - SpanBERT: masking multiple contiguous words instead of masking individual words sporadically

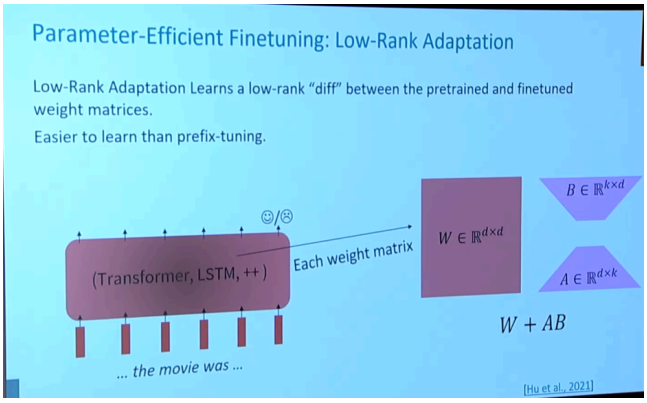
- Parameter Efficient fine tuning: AS WE ARE TALKING ABOUT FINE TUNING
  - How to efficiently fine tune a pretrained model?
    Instead of changing all the parameters during fine tuning, how about we (smartly) choose some parameters for fine tuning?



  - An example is prefix tuning:

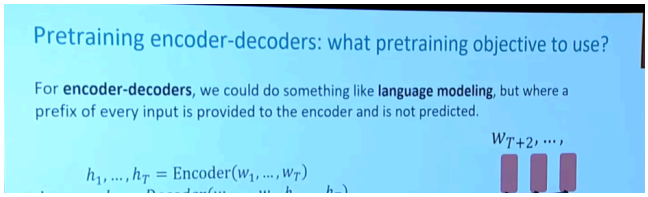


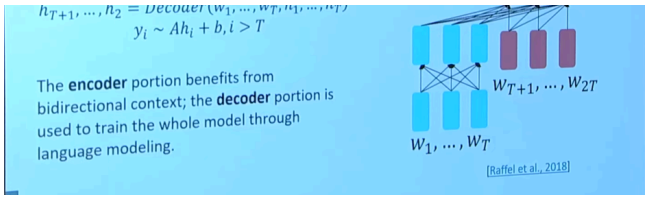
  - (the famous) LoRA – Low Rank Matrix Adaption



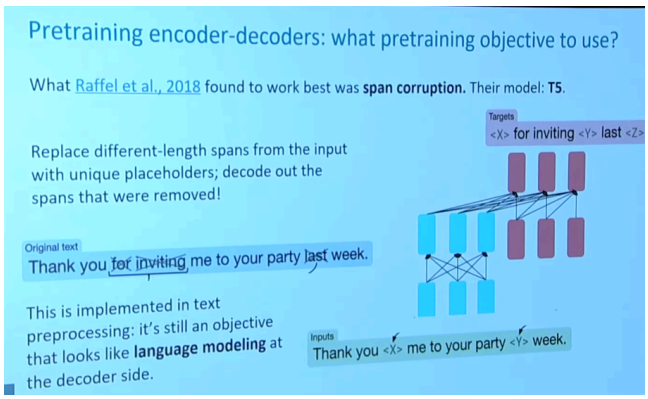
Encoder – Decoder:

- As encoder has bi-directional context here, we can use decoder for language modelling and then train whole model (encoder+decoder) by using the loss of decoder

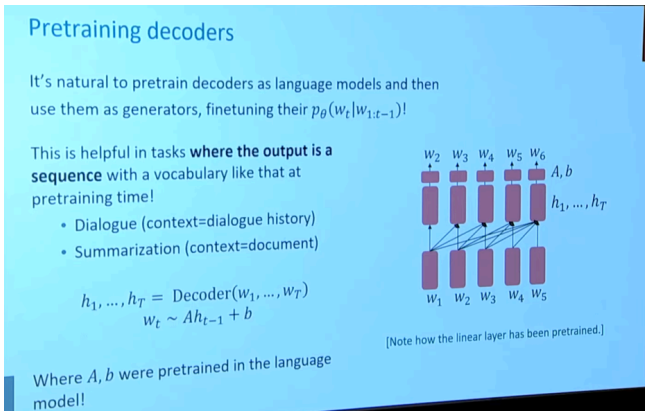




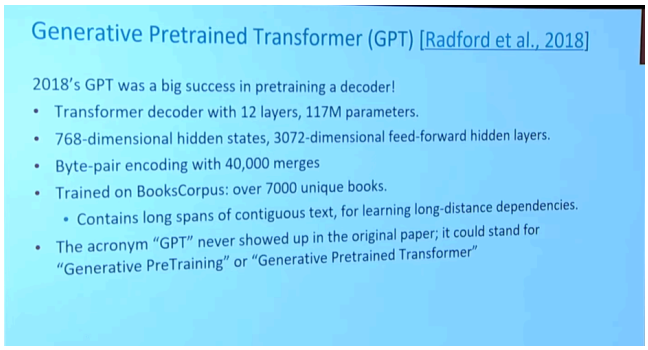
- **Masked LM** also works for encoder – decoder model



Decoder:

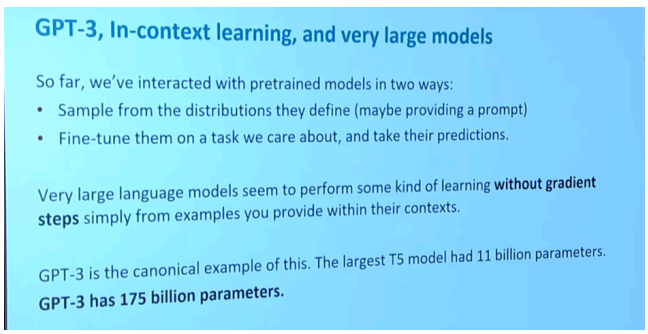


- The Famous GPT:



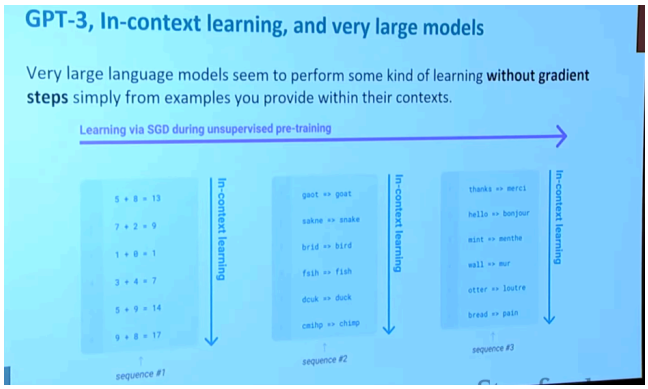
- GPT-2 -- Just a scaled version of original GPT. Trained on ~9B words of text and contains ~1.5B parameters

- GPT3 – A very large, 175B parameter model. Trained on ~300B words of text. Showed ability of in-context learning without explicit inputs or gradient step.

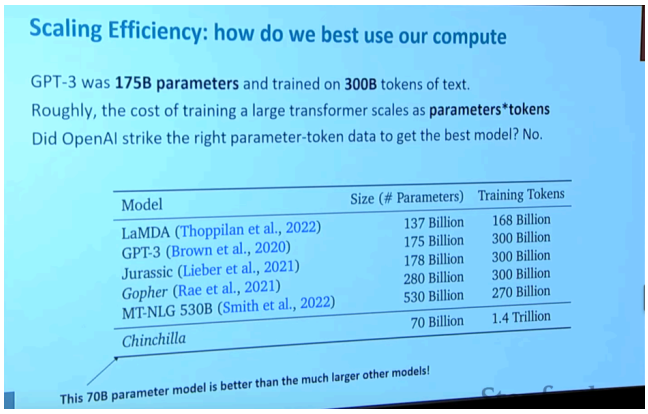


- In-context learning: learning to do a task from very minimal specific inputs or almost no explicit inputs. I.e. the model learnt to perform a task by looking at non-explicit inputs.

- The ability of in-context learning appeared only in large models (trained on large data) --- TO ME, this is a very sophisticated pattern recognition, without explicit training.



- Relation of number of parameters vs words in training data:



- Chain of thought prompting: (after pre-training)
  - Providing task specification (decomposition of steps to find answer) in the prompt