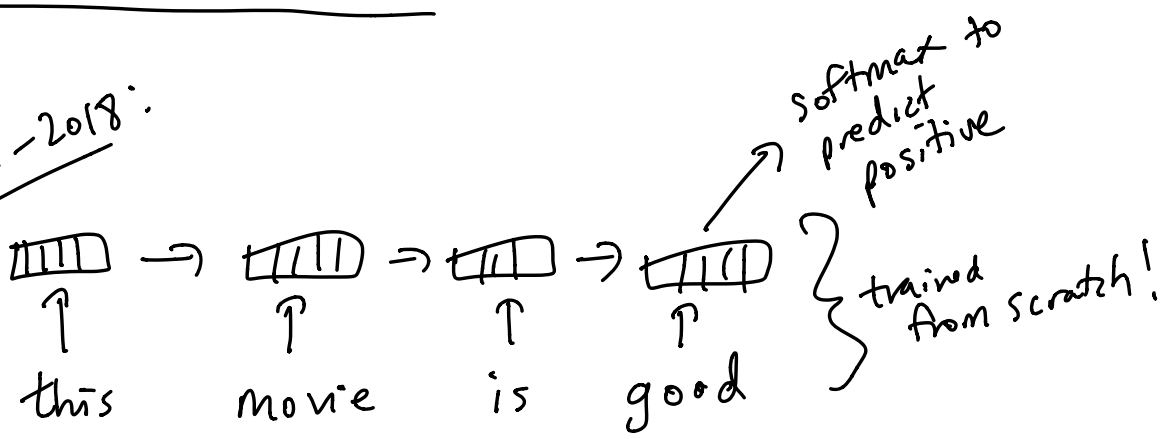
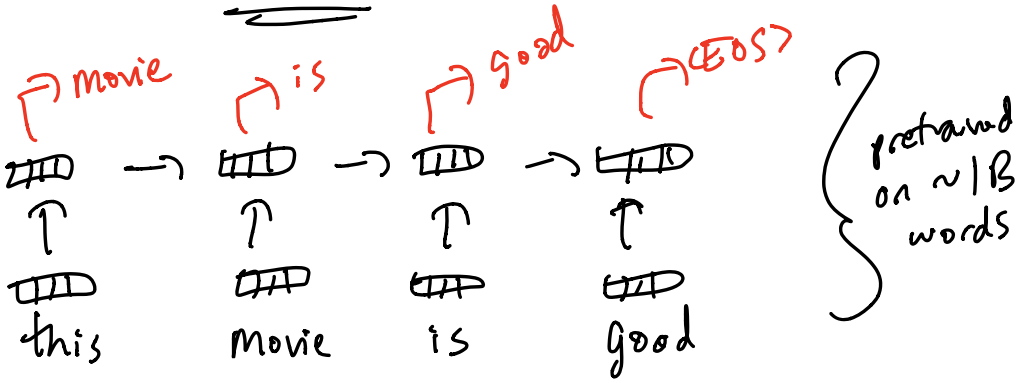
From ELMo to BERT:

pre-2018:

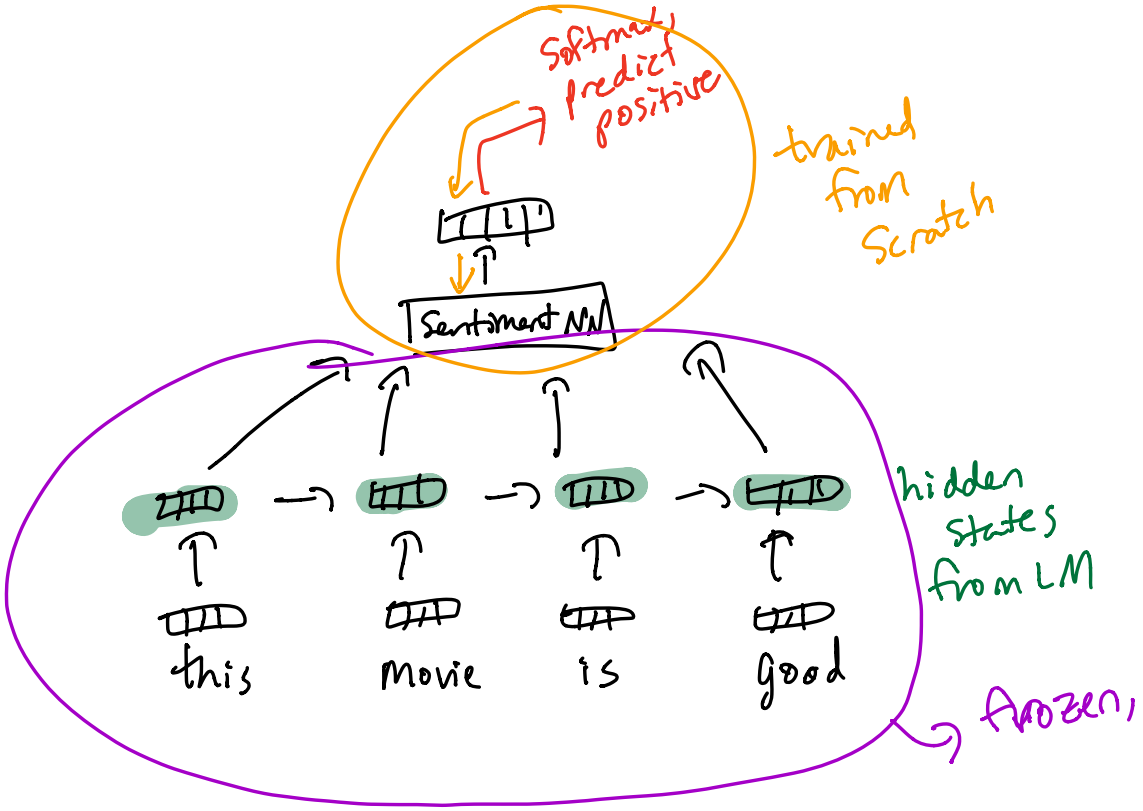


ELMo:

Step 1: pretrain an RNN LM on lots of unlabeled data

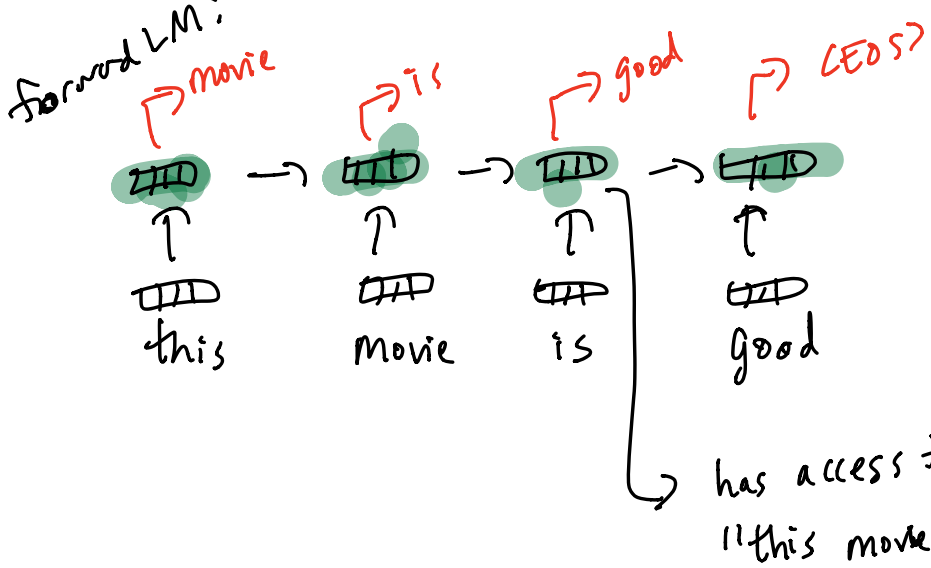


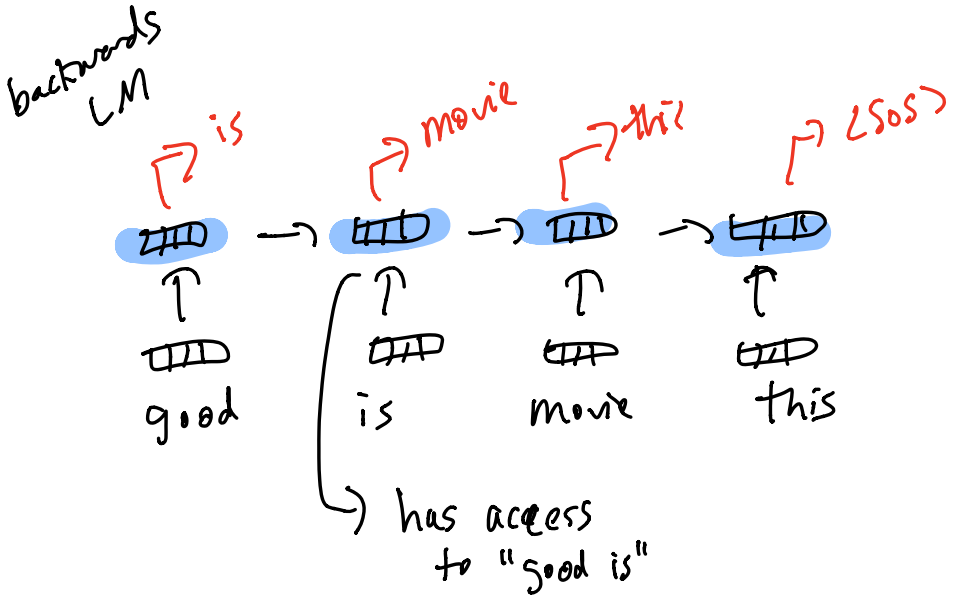
Step 2: freeze LM parameters, use its representations (hidden states) as input to a task-specific model

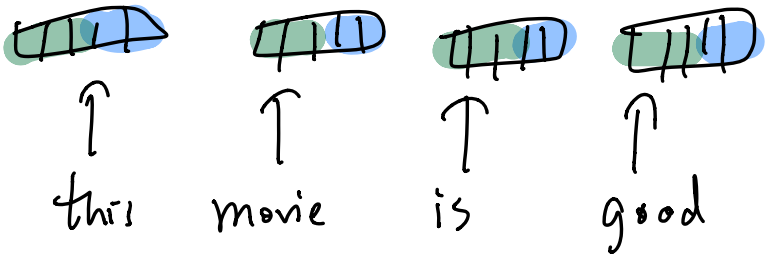


ELMo setup:

forward LM, backward LM ⇒ Combine via concatenation







---

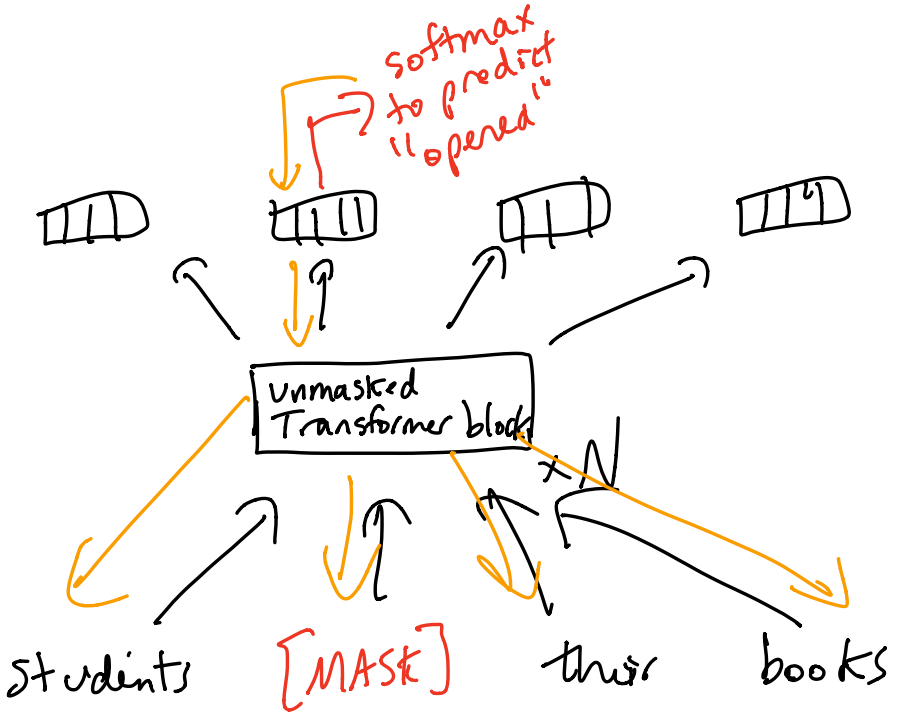
forward / backward LM is clumsy

Can we replace these w/ a single model?

ELMo (2018) ⇒ BERT (2019)

- 2 unidirectional LMs ⇒ 1 masked LM
- recurrent NNs to Transformers
- freezing the LM to fine-tuning LM
- pretrained LM on way more data, way bigger model

masked LM:
- input is a sequence where some tokens have been randomly masked out
- goal: predict identity of the masked tokens



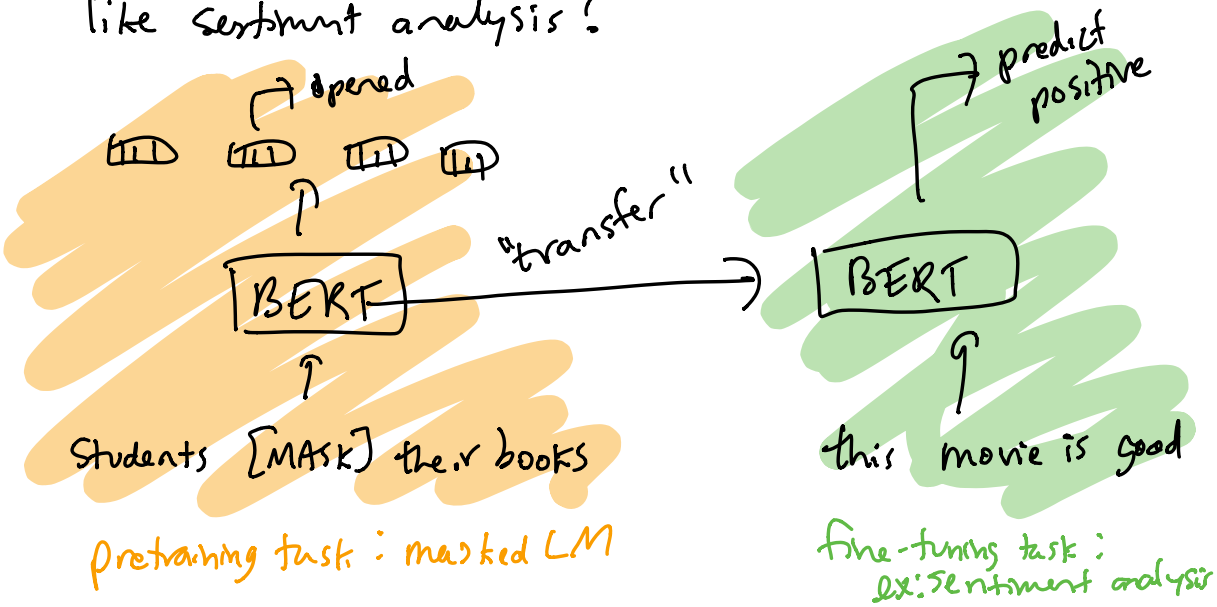
effect of increasing % [MASK]:

students [MASK] their books

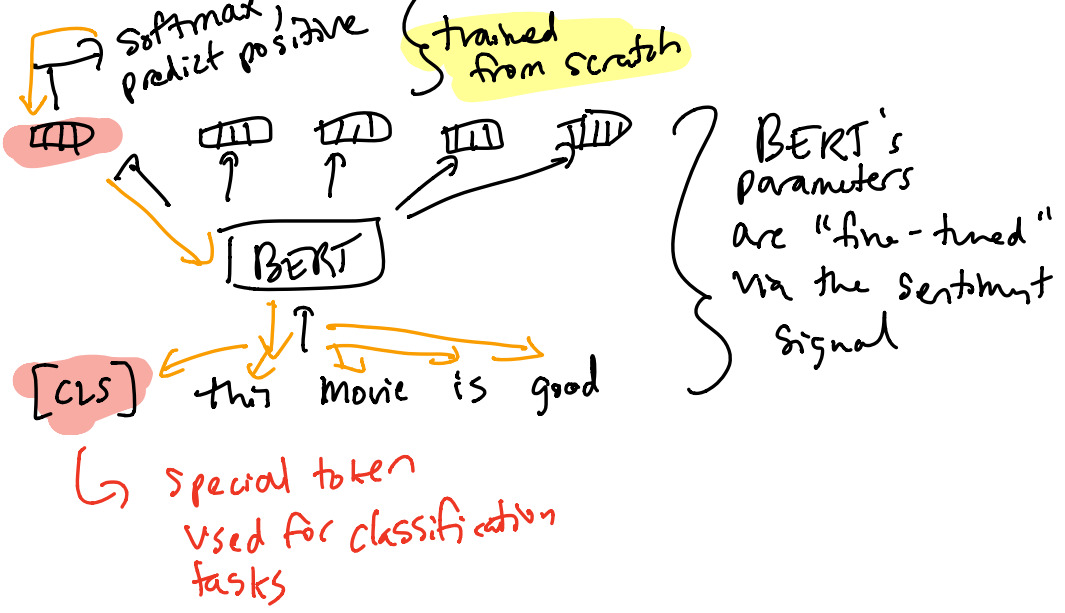
students [MASK] their [MASK]

BERT: [MASK] % of 15%

how do we use BERT to solve an NLP task like sentiment analysis?



Applying BERT for text classification



↳ special token used for classification tasks

terminology:

pretrain: start w/ randomly init. model, train it w/ a self-supervised obj.

↳ LM, masked LM

↳ data is free

↳ big models on big data

freeze: do not backprop into the params of the pretrained model using the downstream objective

fine-tuning: backprop into the pretrained model using task-specific signal; softmax is trained from scratch