Lecture 11: Pre-trained Language Models

Instructor: Jackie CK Cheung

COMP-550

Outline

Transfer learning

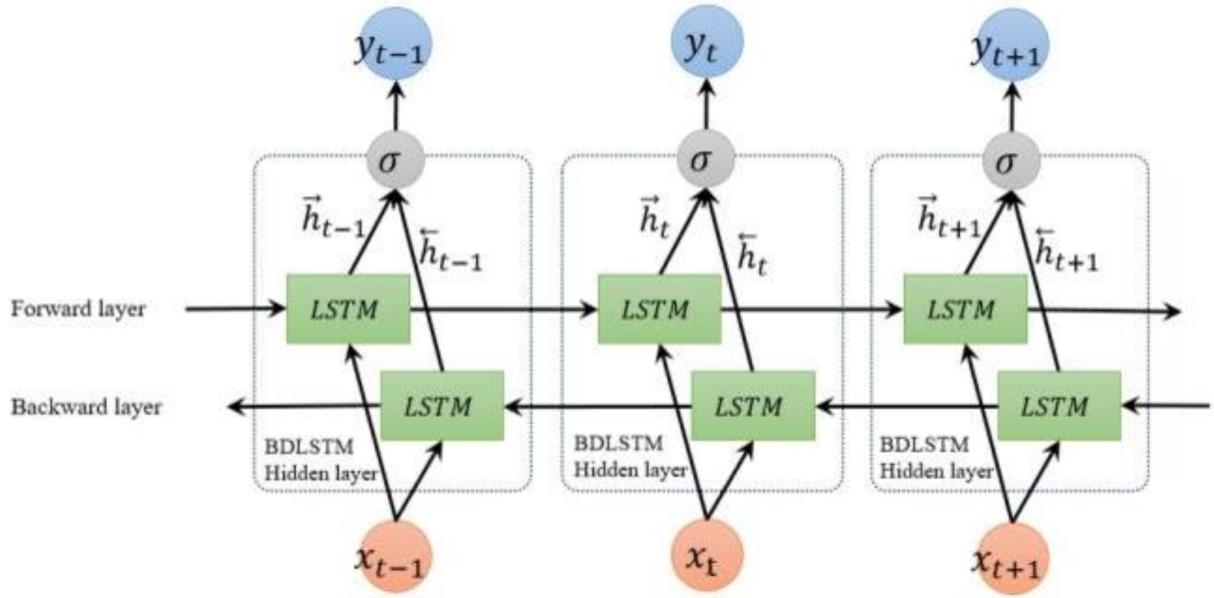
Transformer architecture

Large pre-trained language models

Limitations of approach?

Last Class: BiLSTMs

Have two LSTM layers, forward and backward in time



Concatenate their outputs to make final prediction

Where To Go From Here?

Two key ideas:

Transfer learning

Using knowledge gained from one task to improve performance on another task

Transformer architecture

Make different assumptions in the model architecture about how to model a sequence

Transfer Learning

When solving a new language task, people do not start from scratch!

- Knowledge about words
- Knowledge about syntax and other grammatical structures
- Knowledge about the world; what is likely or unlikely to happen

Why make NLP models relearn all this for each task?

Key question: what should be the **source task** to transfer knowledge from?

Language Modelling

Ideal as source task because:

- Captures a variety of competencies that are relevant to many NLP tasks
- Training data is cheap and plentiful (just need to crawl the web for English texts)

- Example:

  Chris Turner has been finding lost rings for 30 years, actor Jon Cryer couldn't be happier he found _____

  Source: CBC

  Answer: *his*

  Knowledge required? syntactic, world knowledge

ELMo (Peters et al., 2018)

ELMo – Embeddings from language models

1. Train a biLSTM for language modelling, using log-likelihood objective:

$$\sum_{k=1}^{N} \left(\log p(t_k \mid t_1, \ldots, t_{k-1}; \Theta_x, \overrightarrow{\Theta}_{LSTM}, \Theta_s) \right.$$
$$\left. + \log p(t_k \mid t_{k+1}, \ldots, t_N; \Theta_x, \overleftarrow{\Theta}_{LSTM}, \Theta_s) \right).$$

2. Use this language model to compute contextualized word representations in a model for a downstream task

Transfer in ELMo

Specifically, learn a linear combination of the hidden representations at multiple layers for a downstream task:

$$\mathbf{ELMo}_k^{task} = E(R_k; \Theta^{task}) = \gamma^{task} \sum_{j=0}^{L} s_j^{task} \mathbf{h}_{k,j}^{LM}.$$

γ^{task}: scalar

s_j^{task}: weight for layer j

This is then used to help initialize word representations in a new RNN that is specifically used for that downstream task.

ELMo Tested On

Question answering

Finding the answer to a natural language question in a passage

Natural language inference

Deciding if a span is entailed (i.e., necessarily follows from) another span, or is a contradiction, or neither

Semantic role labelling

Deciding what the agent, patient, location, time, … of a predicate are

Named entity recognition

Others…

Transformer Architecture (Vaswani et al., 2017)

Problem with LSTMs:

- Despite supposedly solving vanishing gradient problem, recurrence in LSTMs still make it difficult to look at patterns and information over long distances.
- Inherent nature of recurrence – need to pass information one step at a time

Idea behind Transformers:

- Allow information flow between any pair of words!

Attention

Sentence: $w_1 \;\; w_2 \;\; \ldots \;\; w_n$

Embeddings: $x_1 \;\; x_2 \;\; \ldots \;\; x_n$

Goal is to compute next layer of word representations at layer l:

$$z_1^l \;\; z_2^l \;\; \ldots z_n^l$$

Attention learn a distribution over words to decide how important each word is in order to compute the representations at the next layer

Values, Keys, and Queries

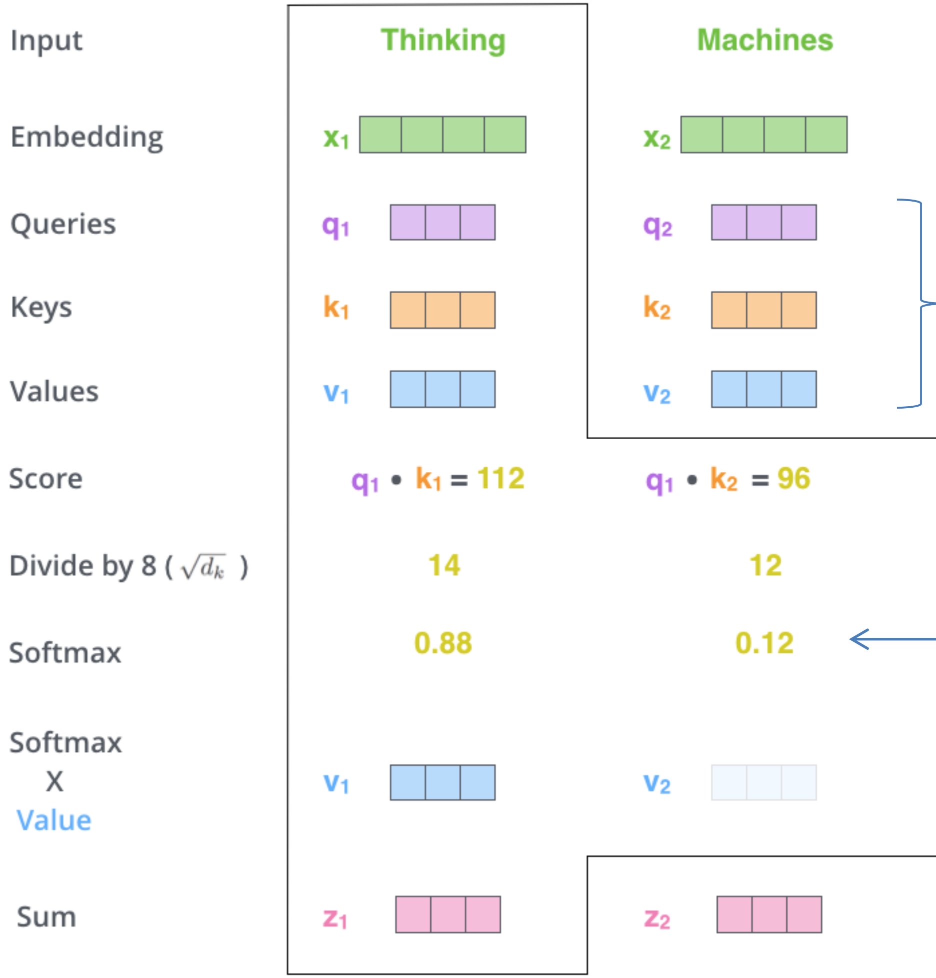
Three views of a word:

query	use of this word as a query, because we want to compute its representation at the next layer
key	use of this word as a key; we use this vector to decide how important the word is to another word as part of the attention computation
value	this vector stores the value associated with the key, once you've done the attention computation

Each view is associated with its own vector

Example: Two word sentence

Computing the representation of the first word at the next layer:



Done by projecting x_i into subspaces

These are the attention scores

$O(N^2)$ connections!

Source: http://jalammar.github.io/illustrated-transformer/

Transformer Architecture

There are a number of other bells and whistles.

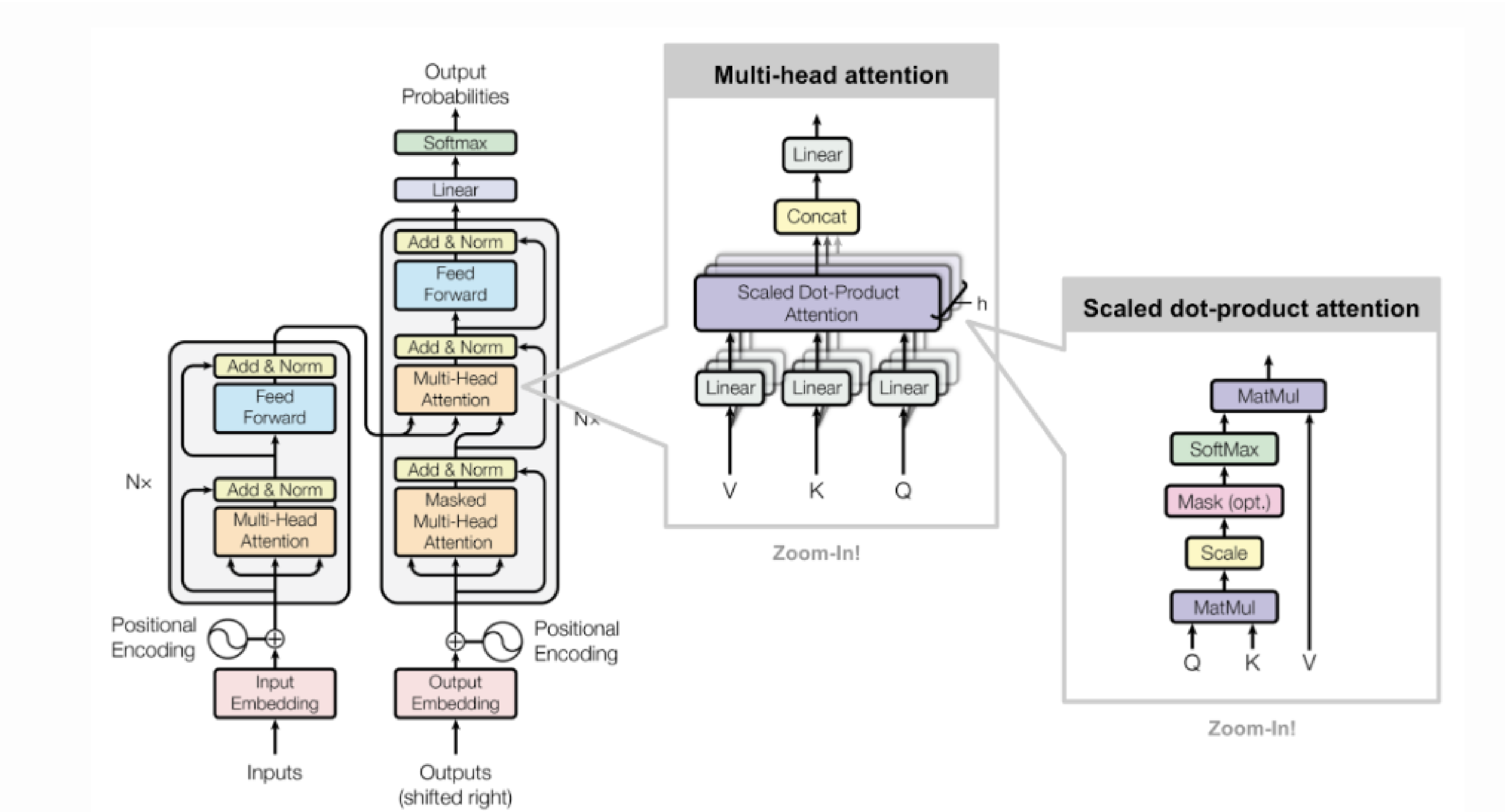


Fig. 17. The full model architecture of the transformer. (Image source: Fig 1 & 2 in Vaswani, et al., 2017.)

BERT (Devlin et al., 2018)

A transformer model trained on:

- masked language modelling

e.g., *There is a word [MASKED] in this sentence*.

- next sentence prediction

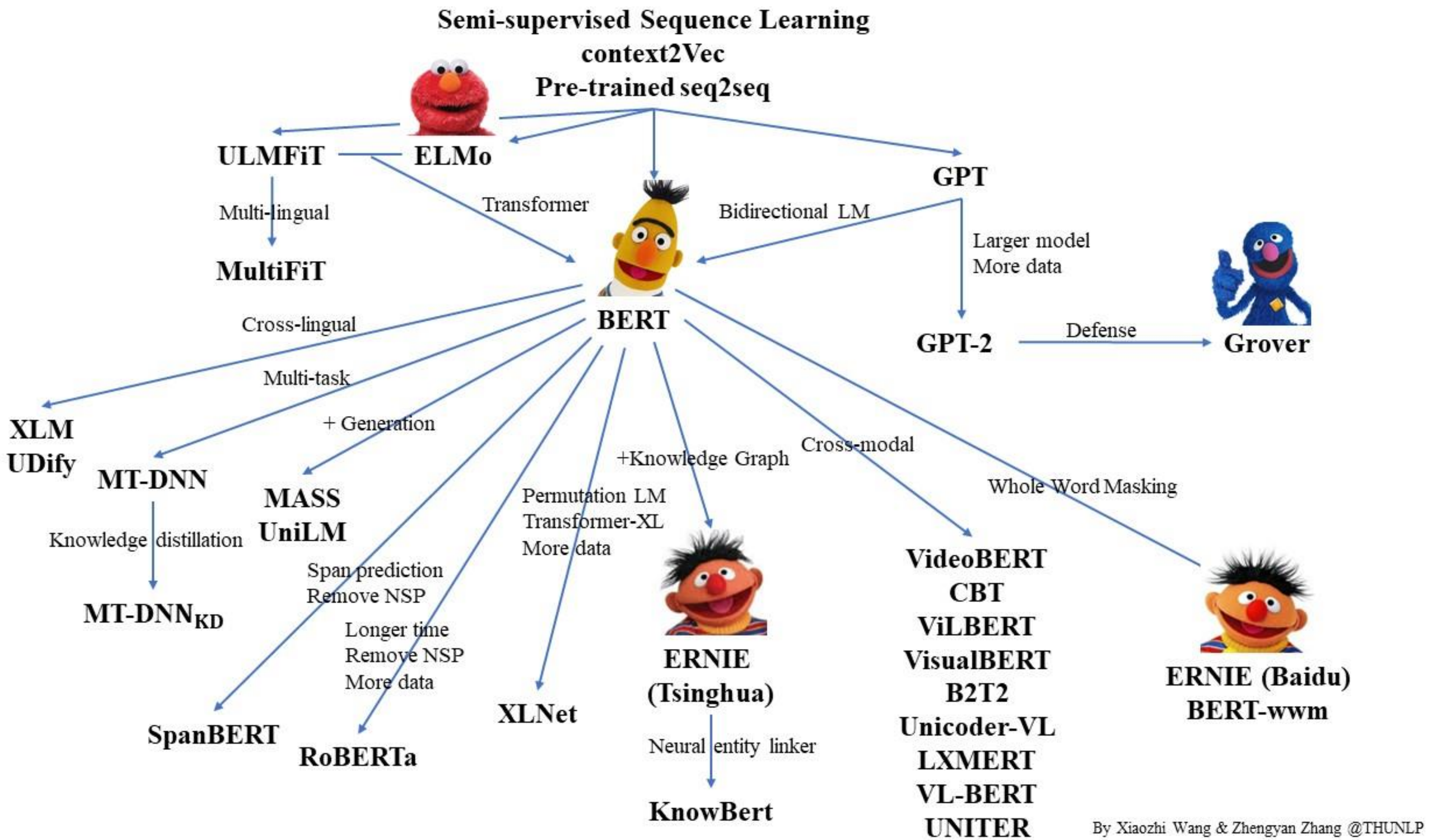
Given, s1 s2 -> Does s2 follow s1?

Training corpora:

- Books (800M words), English Wikipedia (2500M words)

Up to 340M parameters!

Many Other Variants



Scaling Up Even More

GPT-3 from OpenAI (Brown et al., 2020):

- Training: ~500B words (web crawled data, books, Wikipedia)
- Up to 175B model parameters

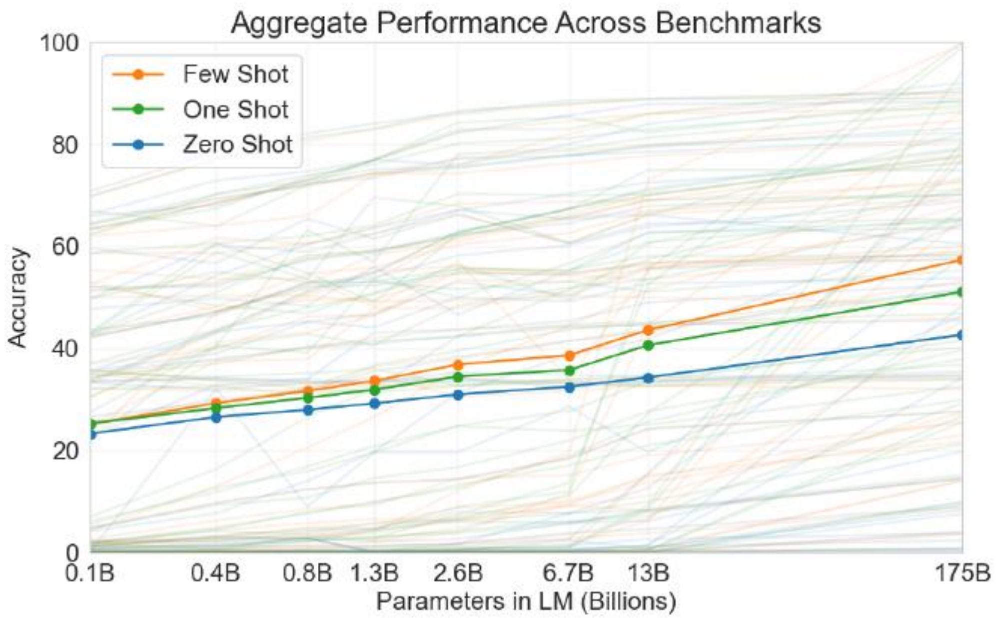


Figure 1.3: Aggregate performance for all 42 accuracy-denominated benchmarks While zero-shot performance improves steadily with model size, few-shot performance increases more rapidly, demonstrating that larger models are more proficient at in-context learning. See Figure 3.8 for a more detailed analysis on SuperGLUE, a standard NLP benchmark suite.

Successes

BERT + variants are the basis of modern NLP systems in the past 2 years.

- Many new SOTA results which start by fine-tuning one of these pre-trained Transformer models

GPT-3 also shows some success at few-shot or zero-shot learning:

few-shot	give a small number (<100) of examples to finetune on
zero-shot	give no new examples; usually need to give some other natural language prompt as side input

Limitations

The largest models have read such a large number of texts; may have memorized all common situations

- Have seen much much more text than any single human ever would!
- Recent results suggest they may not generalize as people do

Problems with:

- Fine-grained semantic understanding of world as expressed through text
- Long-range coherence of texts; e.g., repetition of texts
- Reasoning about physical relations and common sense

Points to return to in future classes

Social Impacts

Misuse of language models – spamming and generating fake news

Fairness and bias of language models – LMs may pick up on biases in training data (e.g., related to gender, race, religion) and make decisions that are unfair; in fact they may even amplify biases in the training data

Cost – very expensive to train! In terms of time, money, and energy usage. Who gets access to the model?

Future Lectures

Return to pre-neural world to investigate basic NLP ideas and algorithms involving *structure*

- **Syntax**: grammar formalisms and parsing
- **Semantics**: meaning representations – what does meaning even mean?