第13课：上下文相关表示 (Contextual Representations)

第14课：Transformer模型与生成式模型中的Self-Attention（客座讲座）

---

## 第13课：上下文相关表示 (Contextual Representations)

**未知词问题**

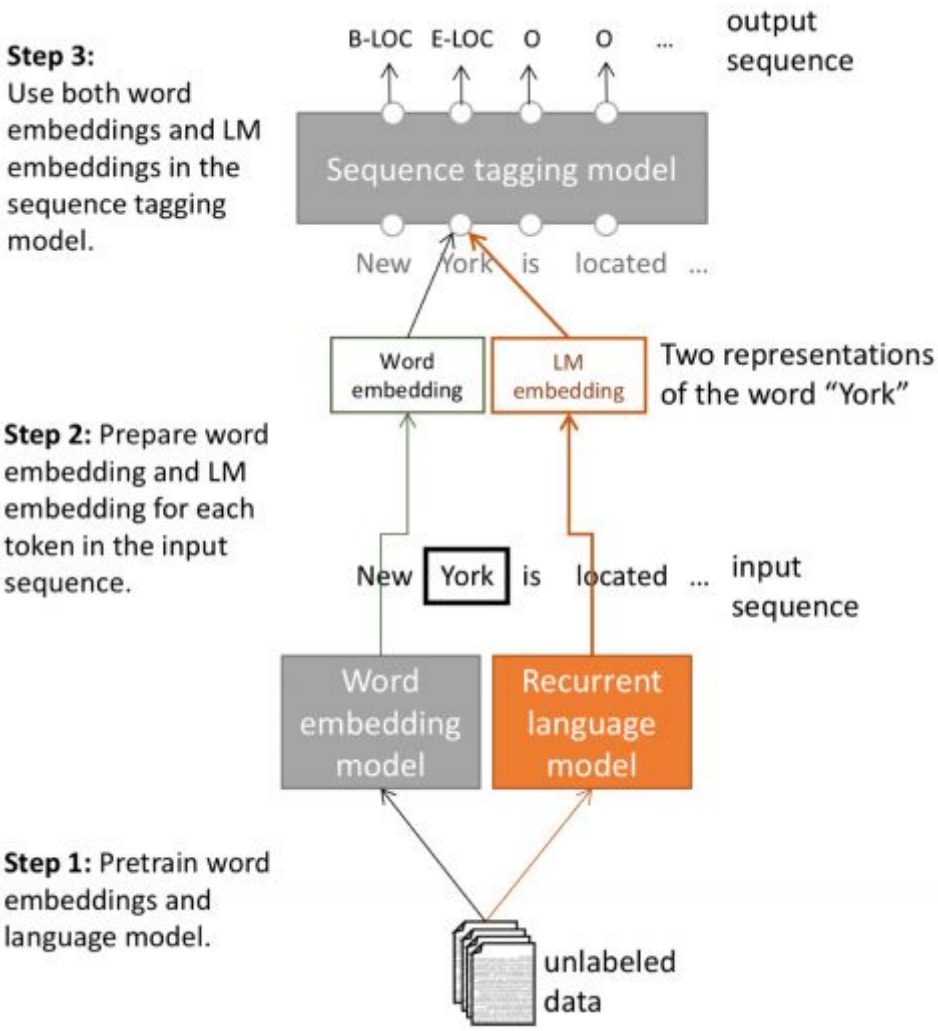
- 可以全部归为<unk>
- 可以用char-level模型
- 可以。。。。[ Dhingra, Liu, Salakhutdinov, Cohen 2017]

**预训练词向量 Pre-trained word vectors**

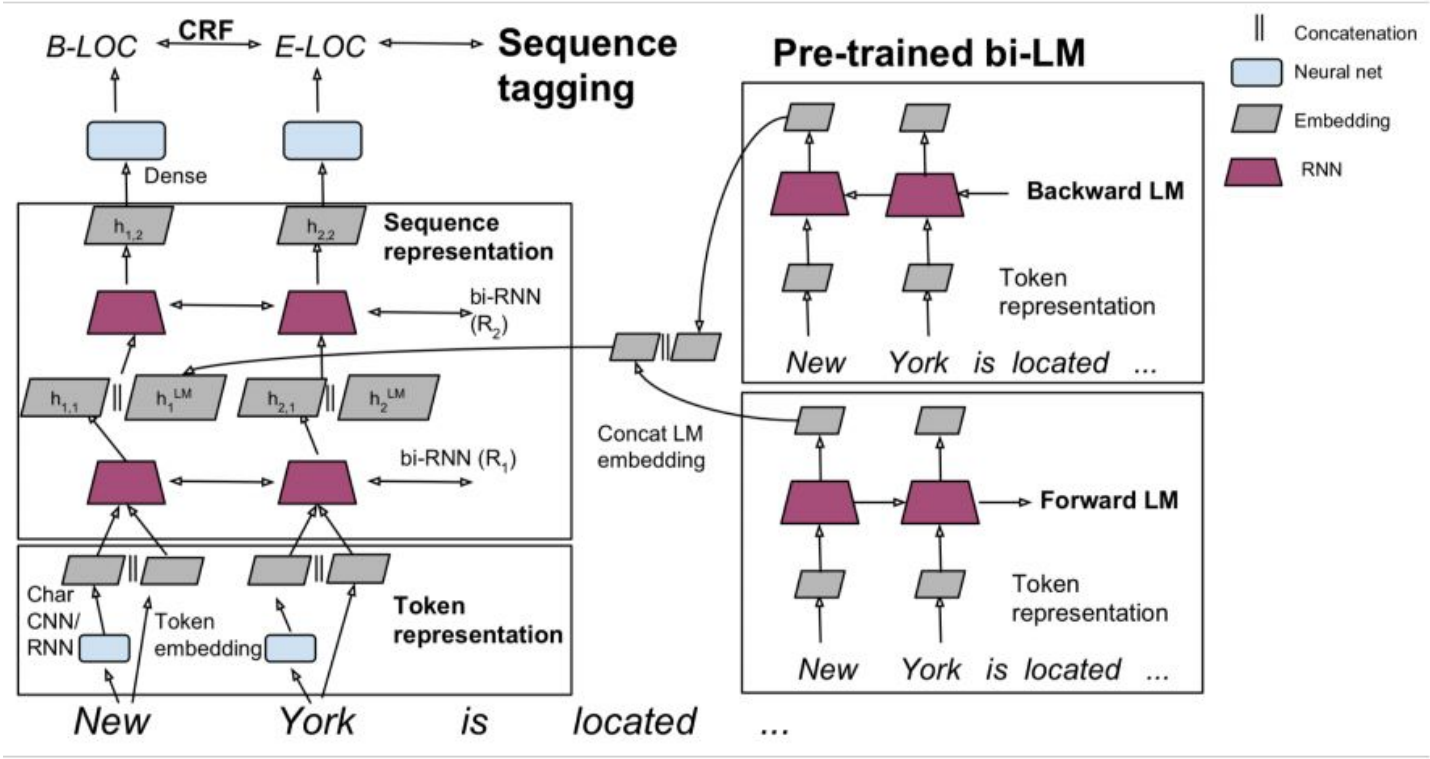
- 相比随机初始化，确实能够提高在各类任务各类模型上的效果
- 先前课程已经掌握：Word2vec, GloVe, fastText
- 问题：word表示不顾上下文，word有多个层面的语义变换
- 在NLM里，冻结住word vectors（上下文无关），然而LSTM产生的word representations则是与上下文相关的

Peters et al. (2017): **TagLM** – “Pre-ELMo”

- 想获得上下文之中的word含义。但标准的任务RNN只能在小规模标注数据集上运行。
- 为什么不能使用半监督方法，在无标注数据集上训练整个NLM，而不仅仅是word vectors呢?


Step 3:
Use both word embeddings and LM embeddings in the sequence tagging model.
B-LOC E-LOC O O ...
output sequence
Sequence tagging model
New York is located ...
Word embedding
LM embedding
Two representations of the word "York"
Step 2: Prepare word embedding and LM embedding for each token in the input sequence.
New York is located ...
input sequence
Word embedding model
Recurrent language model
Step 1: Pretrain word embeddings and language model.
unlabeled data

TagLM




$$\mathbf{h}_{k,1} = [\overrightarrow{\mathbf{h}}_{k,1}; \overleftarrow{\mathbf{h}}_{k,1}; \mathbf{h}_k^{LM}]$$

TagLM

McCann et al. 2017: **CoVe** Peters et al. (2018): **ELMo**: Embeddings from Language Models

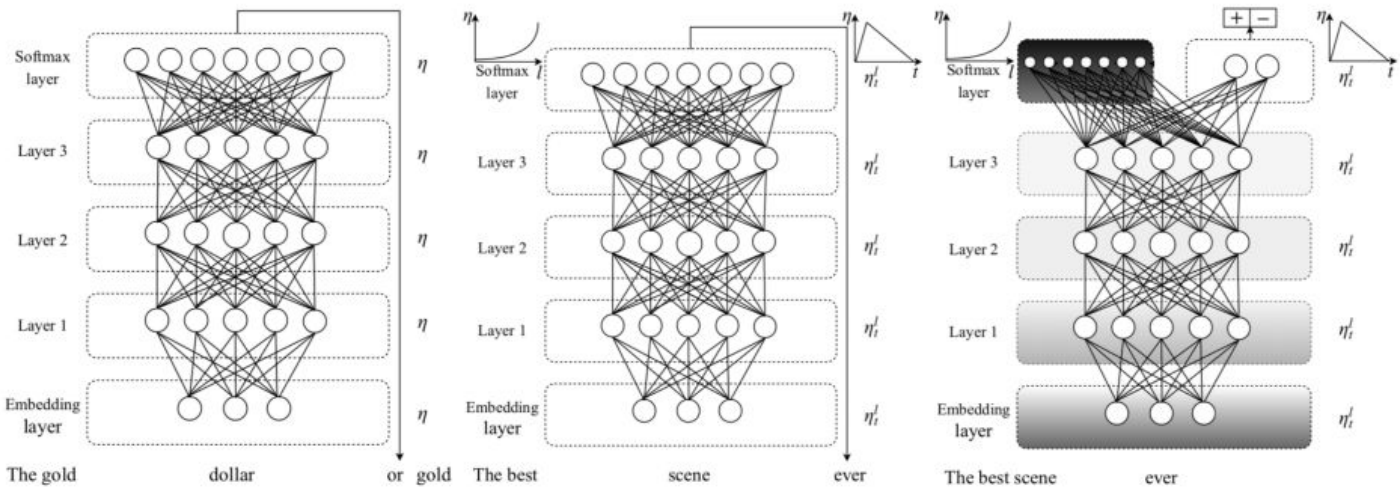
- word embedding最大的问题是多义词，不同上下文信息会编码到相同word embedding空间内
- ELMo的本质思想：先学好静态的word embedding，然后根据上下文语义调整word embedding

$$\begin{aligned} R_k &= \{\mathbf{x}_k^{LM}, \overrightarrow{\mathbf{h}}_{k,j}^{LM}, \overleftarrow{\mathbf{h}}_{k,j}^{LM} \mid j = 1, \ldots, L\} \\ &= \{\mathbf{h}_{k,j}^{LM} \mid j = 0, \ldots, L\}, \end{aligned}$$

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

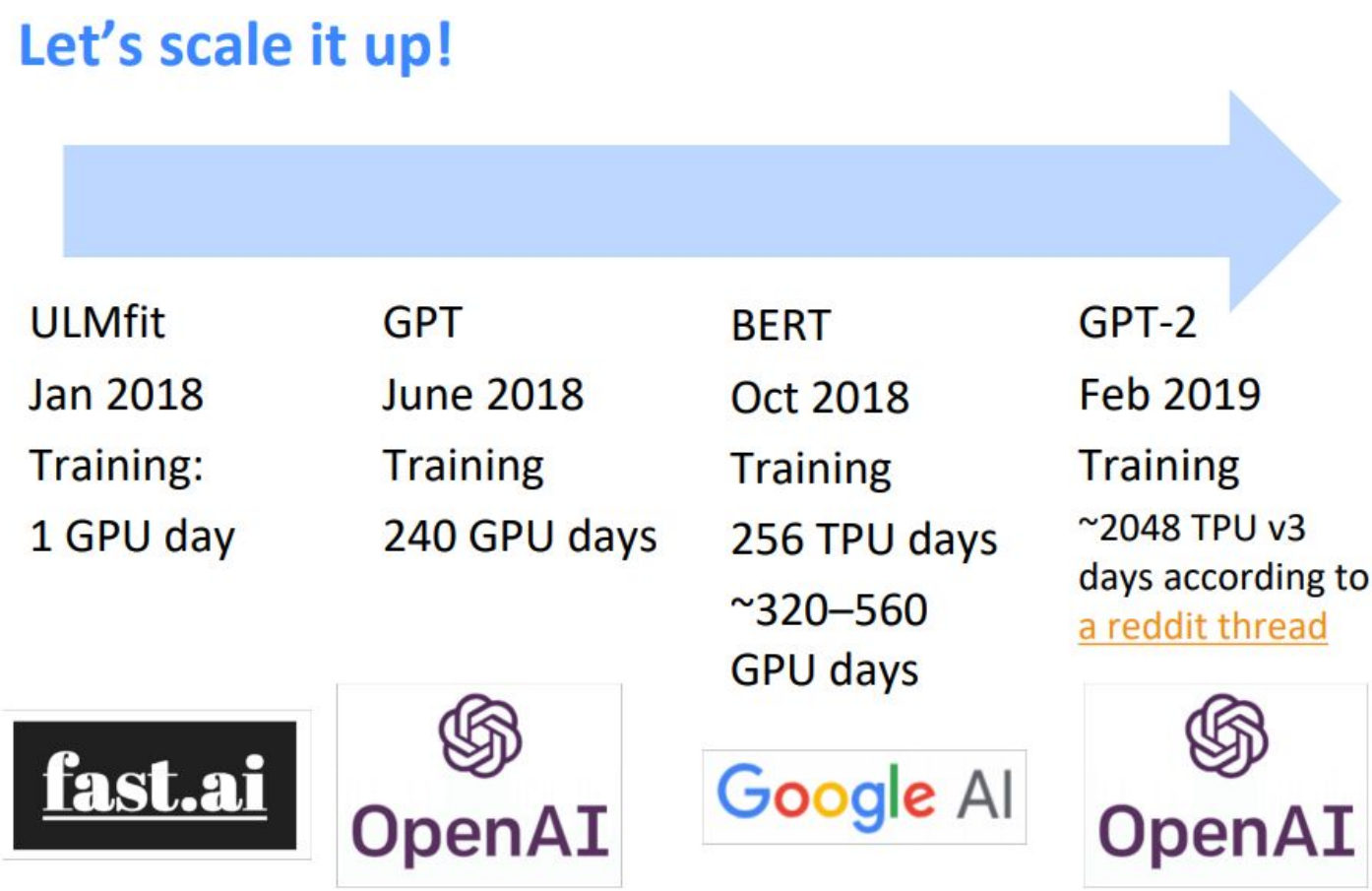
**ULFMiT** [Howard and Ruder (2018) Universal Language Model Fine-tuning for Text Classification.]

- 更加针对领域数据
- （没有详细看）



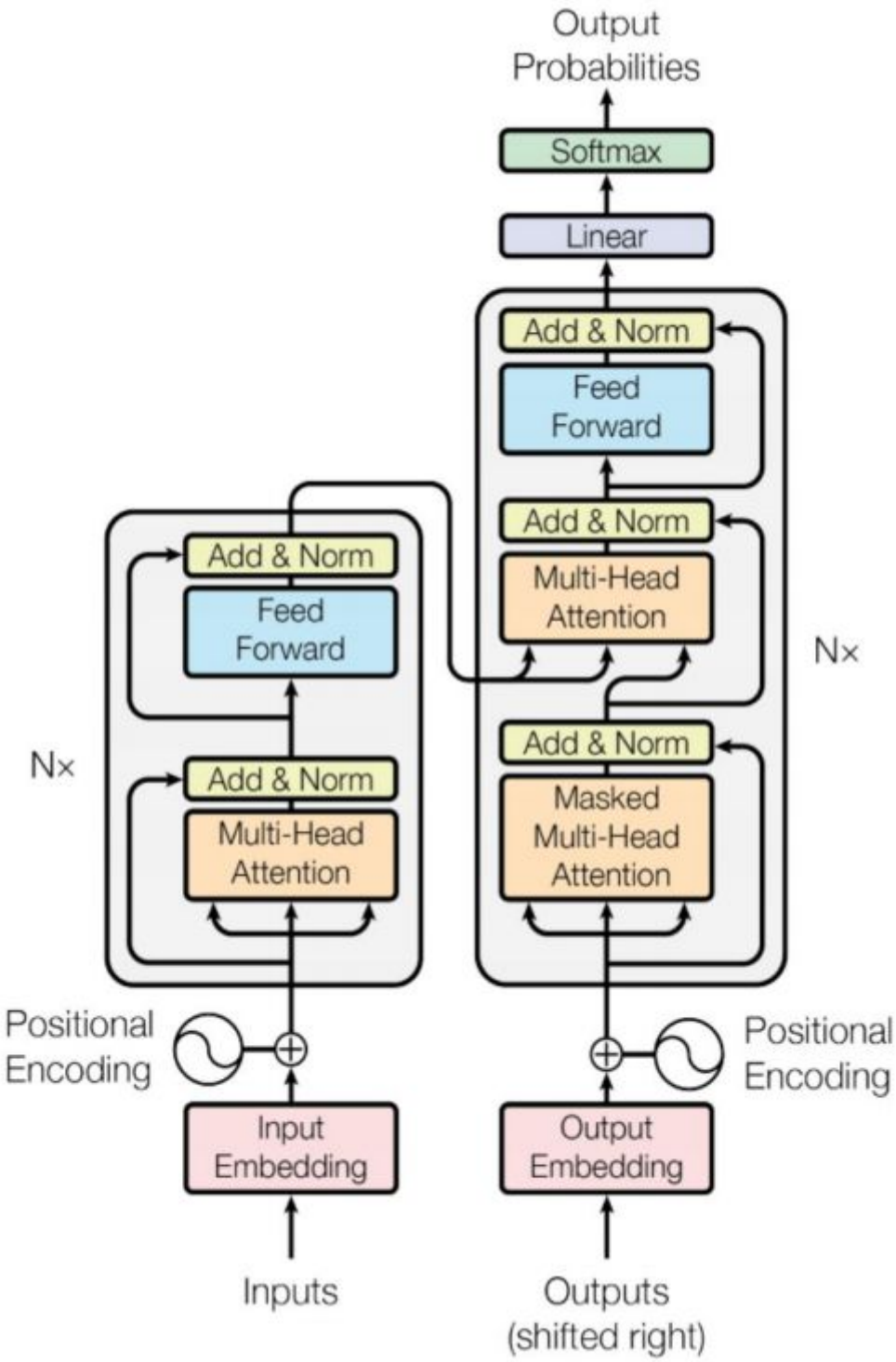

(a) LM pre-training (b) LM fine-tuning (c) Classifier fine-tuning

预训练规模发展历史：




**Transformer** model

- [Attention is all you need. 2017. Aswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Kaiser, Polosukhin]

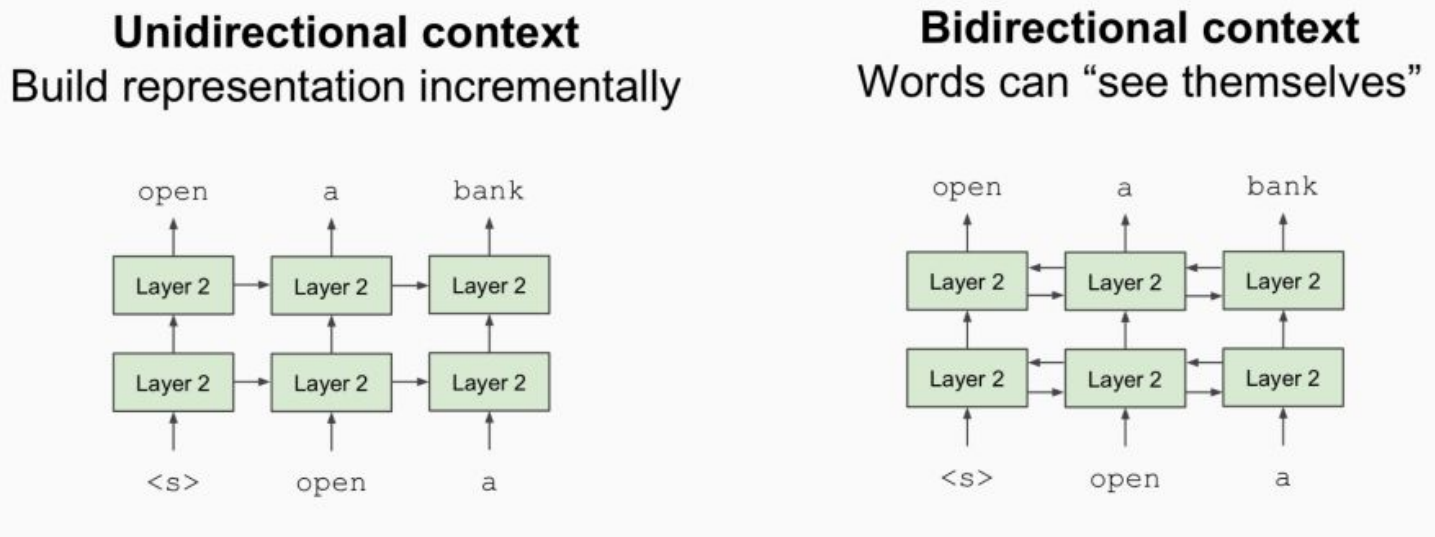



- Transformer Encoder，堆叠6层
  - Dot-Product Attention
  - Scaled Dot-Product Attention：除以 $\sqrt{d_k}$ 再softmax
  - Self-attention in encoder：采用相同的Q=K=V
  - Multi-head attention：同时采用多个attention，用来注意到不同层面的信息
- Input 输入
  - word采用byte-pair encoding
  - positional encoding：携带位置信息
- Transformer Decoder，同样堆叠6层
  - Masked decoder self-attention

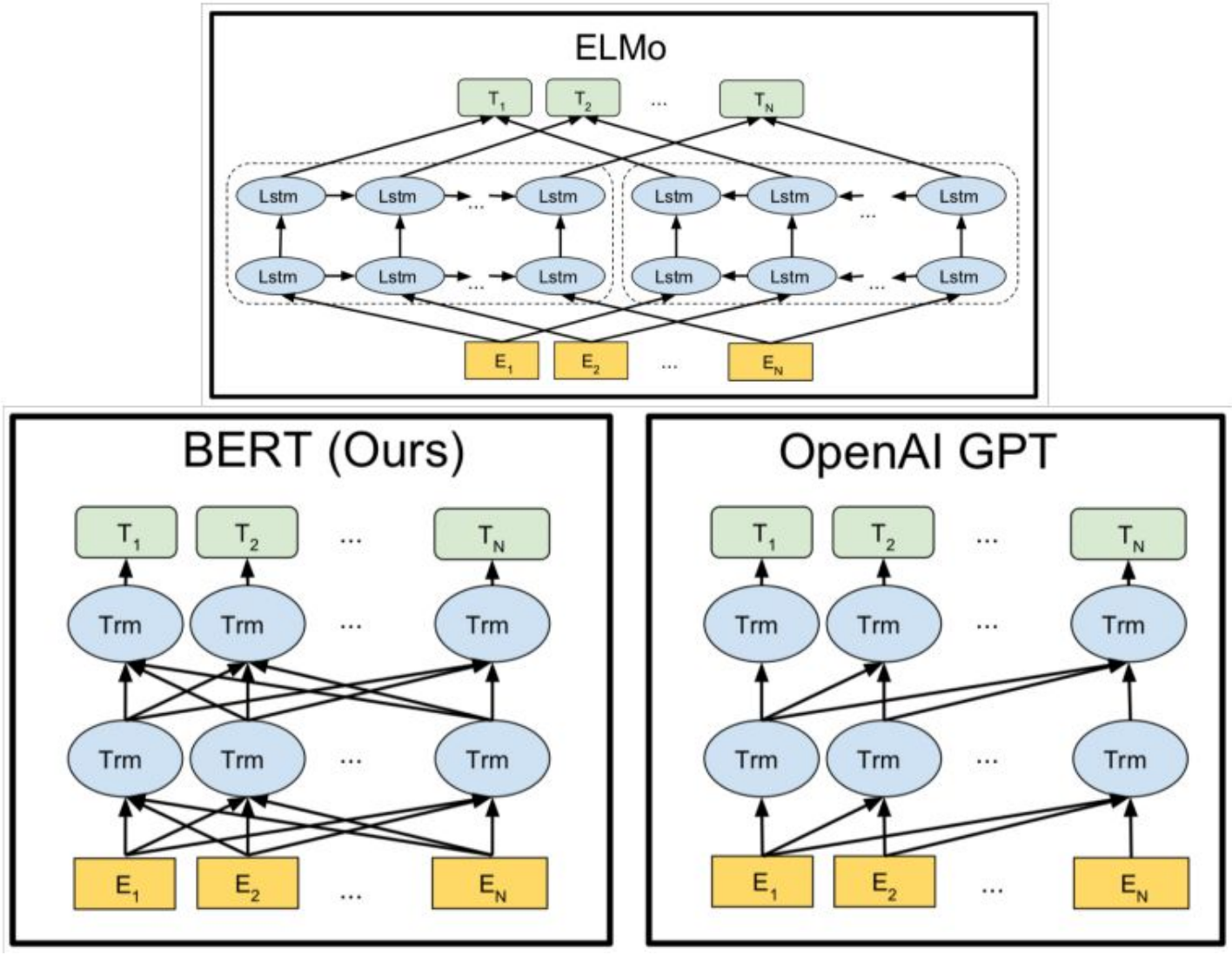
    - Encoder-Decoder Attention: Q来自前一层decoder，K、V来自encoder的输出
- Tips & tricks
    - Byte-pair encodings
    - Checkpoint averaging
    - ADAM optimizer with learning rate changes
    - Dropout during training at every layer just before adding residual
    - Label smoothing
    - Auto-regressive decoding with beam search and length penalties

**BERT** (Bidirectional Encoder Representations from Transformers): Pre-training of Deep Bidirectional Transformers for Language Understanding [Devlin, Chang, Lee, Toutanova (2018)]

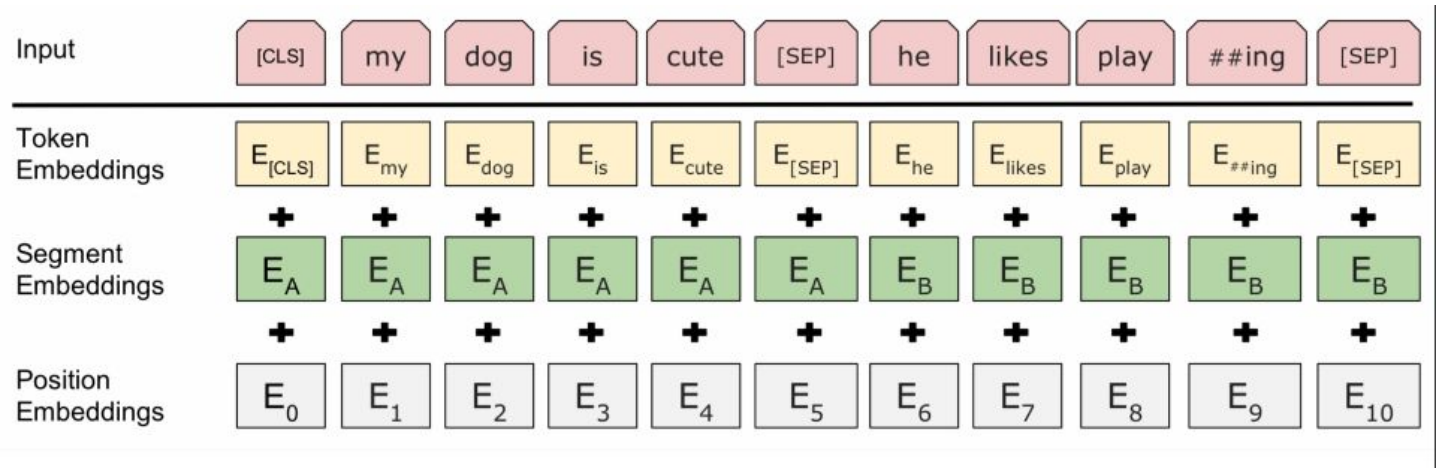
- 问题：先前的LM仅使用左边文字或者右边文字，但语言理解应该是双向的。为什么LM都是单向的呢?
- 需要方向来生成文本（与本任务无关）
- 在双向encoder里word可以看到自己
- 解决方案：把训练任务改成遮盖15%的输入文字，然后预测这个文字



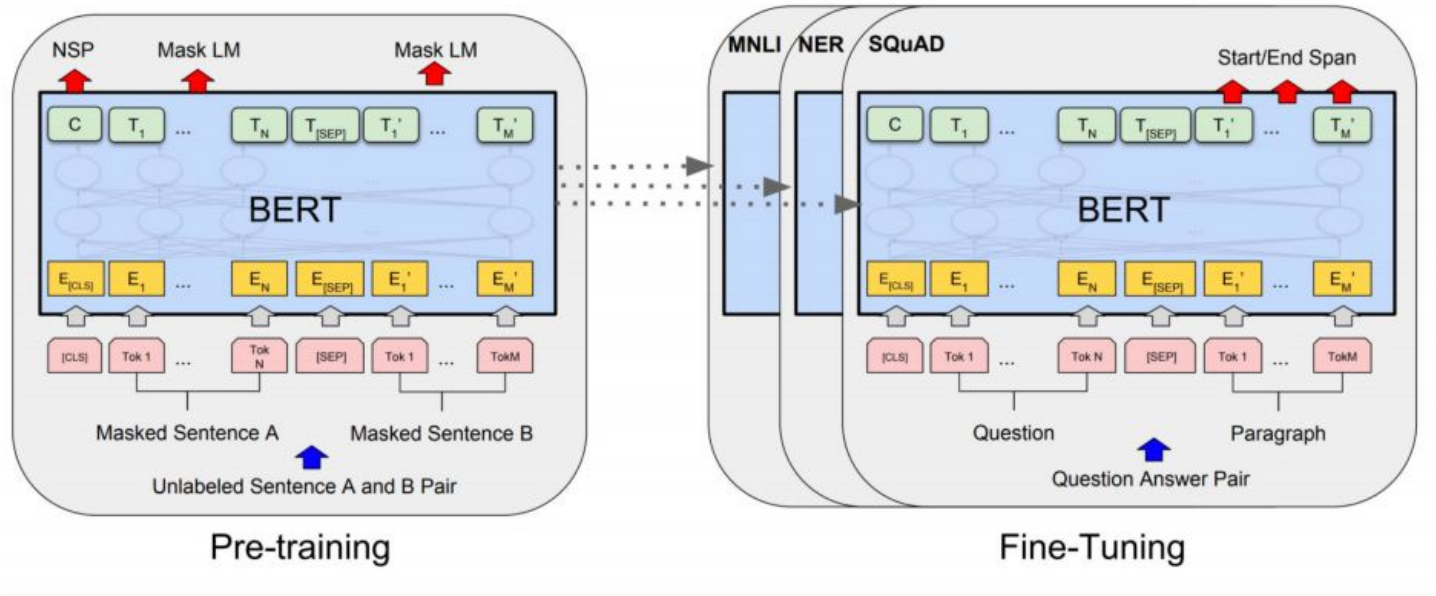

ELMo、BERT、GPT结构对比




BERT训练任务2：两个句子拼接，预测后者是否是前者的下一个句子




Fine-tune，将预训练模型再根据实际任务调整

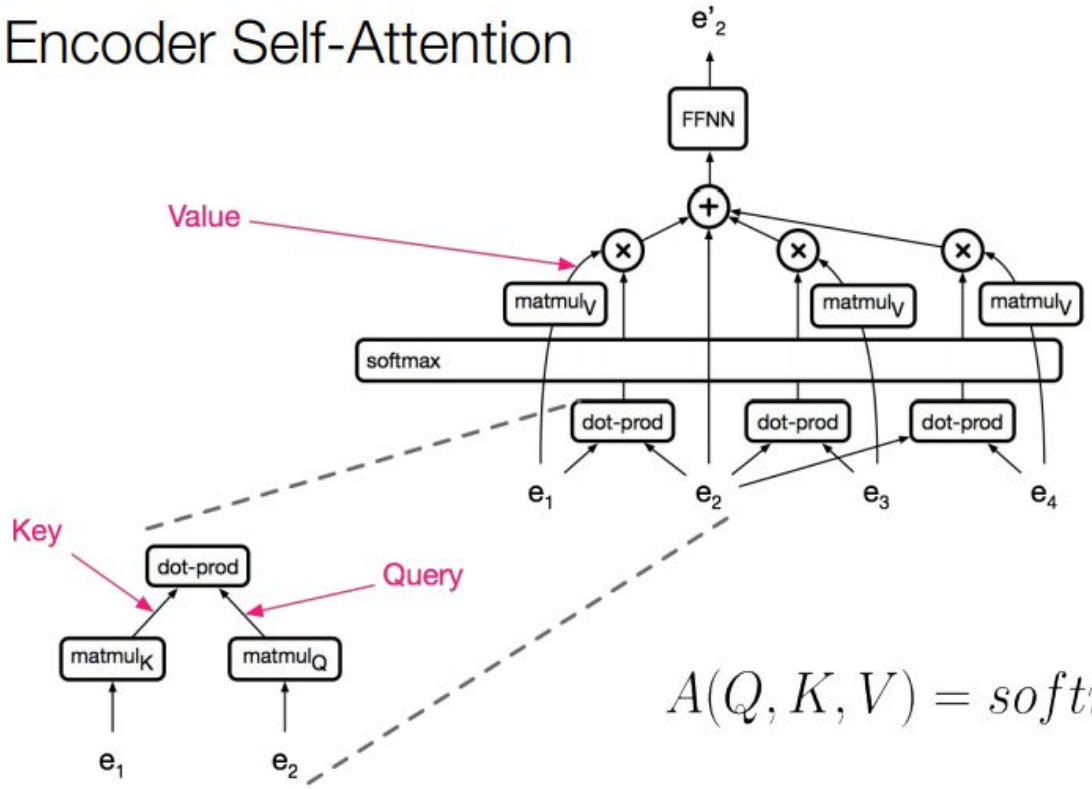



## 第14课：Transformer模型与生成式模型中的Self-Attention（客座讲座）

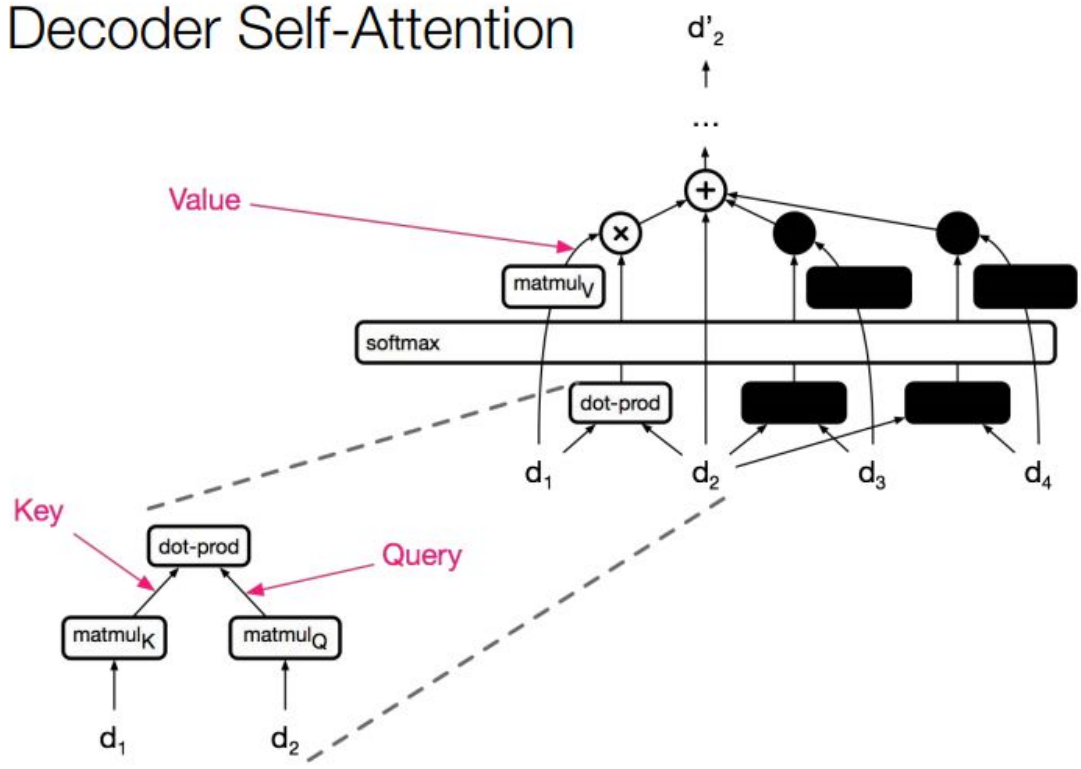
**学习序列数据的方法**

- RNN：LSTM、GRU等
- 计算无法并行化
- 没有对长依赖、短依赖的建模
- 缺乏层次建模
- CNN
- 很容易并行化
- 利用局部依赖
- 需要多层来实现长距离依赖
- Attention
- encoder和decoder之间的attention是NMT重要部分
- 那么是不是也可以用attention来representations?
- 那就是Self-Attention
- 不同位置之间有固定的“距离”
- 门控制/加权的相互作用
- 很容易并行化

**Transformer中的Attention**




$$A(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

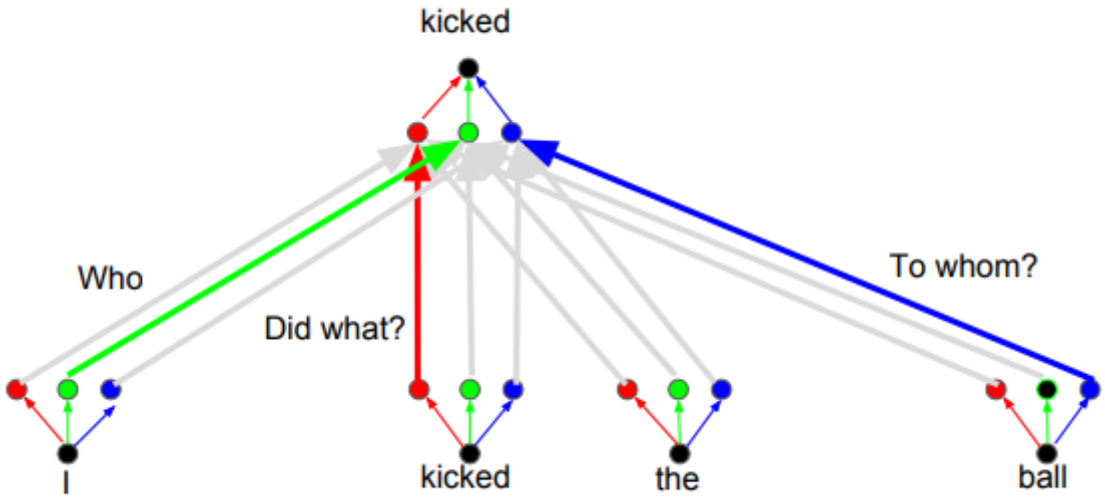



计算量对比

| | FLOPs |
|---|---|
| Self-Attention | $O(length^2 \cdot dim)$ $= 4 \cdot 10^9$ |
| RNN (LSTM) | $O(length \cdot dim^2)$ $= 16 \cdot 10^9$ |
| Convolution | $O(length \cdot dim^2 \cdot kernel\_width)$ $= 6 \cdot 10^9$ |

length=1000 dim=1000 kernel_width=3

**Multihead Attention**




**残差连接**的重要性：

- 残差连接可以携带位置信息到更高层次

**自相似（Self-Similarity）用于图片生成、音乐生成**

**概率图片生成**

- 对像素的联合分布进行建模
- 转换为序列模型问题
- RNN和CNN是最先进的（PixelRNN， PixelCNN）
- CNN加上门控制现在能够媲美RNN，并且并行化很好
- 图片需要具有长距离相关性（比如对称），那么CNN需要更多层、更大卷积核
- Texture Synthesis with Self-Similarity
- Non-local Means

**Self-Attention用于概率图像生成**

- Parikh et al. (2016), Lin et al. (2016), Vaswani et al. (2017)
- 增加图片局域性，让attention范围不是前面的一批像素，而是周围的一批像素
- Image Transformer [Parmar*, Vaswani*, Uszkoreit, Kaiser, Shazeer, Ku, and Tran. ICML 2018]
- 任务：超分辨率、无条件图像生成、有条件图像生成

**Relative Self-Attention用于音乐生成**

- Music Transformer (ICLR 2019) by Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Noam Shazeer, Ian Simon, Curtis Hawthorne, Andrew M. Dai, Matthew D. Hoffman, Monica Dinculescu and Douglas Eck.
- Multihead attention + convolution

**迁移学习**

- Improving Language Understanding by Generative Pre-Training (Radford, Narsimhan, Salimans, and Sutskever)
- BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (Devlin, Chang, Lee, and Toutanova)

**优化和大型模型**

- Adafactor: Adaptive Learning Rates with Sublinear Memory Cost (ICML 2018). Shazeer, Stern.
- Memory-Efficient Adaptive Optimization for Large-Scale Learning (2019). Anil, Gupta, Koren, Singer.
- Mesh-TensorFlow: Deep Learning for Supercomputers (NeurIPS 2019). Shazeer, Cheng, Parmar, Tran, Vaswani, Koanantakool, Hawkins, Lee, Hong, Young, Sepassi, Hechtman)

**self-attention在其他领域**

- Generating Wikipedia by Summarizing Long sequences. (ICLR 2018). Liu, Saleh, Pot, Goodrich, Sepassi, Shazeer, Kaiser.
- Universal Transformers (ICLR 2019). Deghiani*, Gouws*, Vinyals, Uszkoreit, Kaiser.
- Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context (2019). Dai, Yang, Yang, Carbonell, Le, Salakhutdinov.
- A Time-Restricted Self-Attention Layer for ASR (ICASSP 2018). Povey, Hadian, Gharemani, Li, Khudanpur.
- Character-Level Language Modeling with Deeper Self-Attention (2018). Roufou*, Choe*, Guo*, Constant*, Jones*