Representations for a word

我们已经学到

- word2vec
- GloVe
- FastText

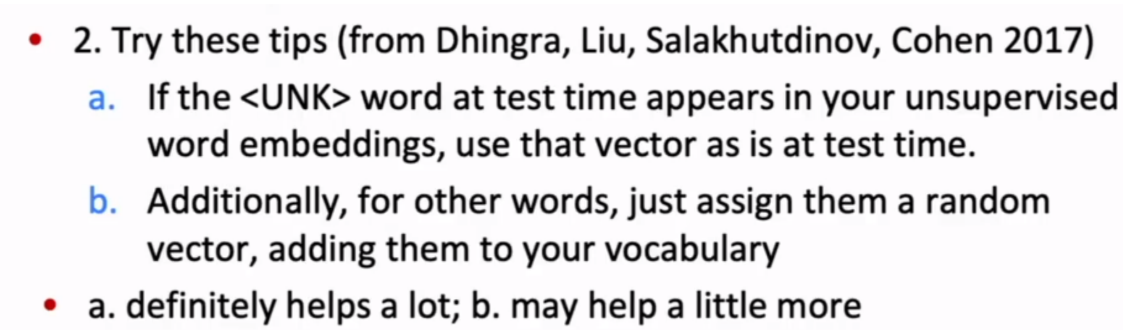
在2014年，普遍人为pre-training比random的wordvec对下游任务更有效。

目前wordvec存在的问题

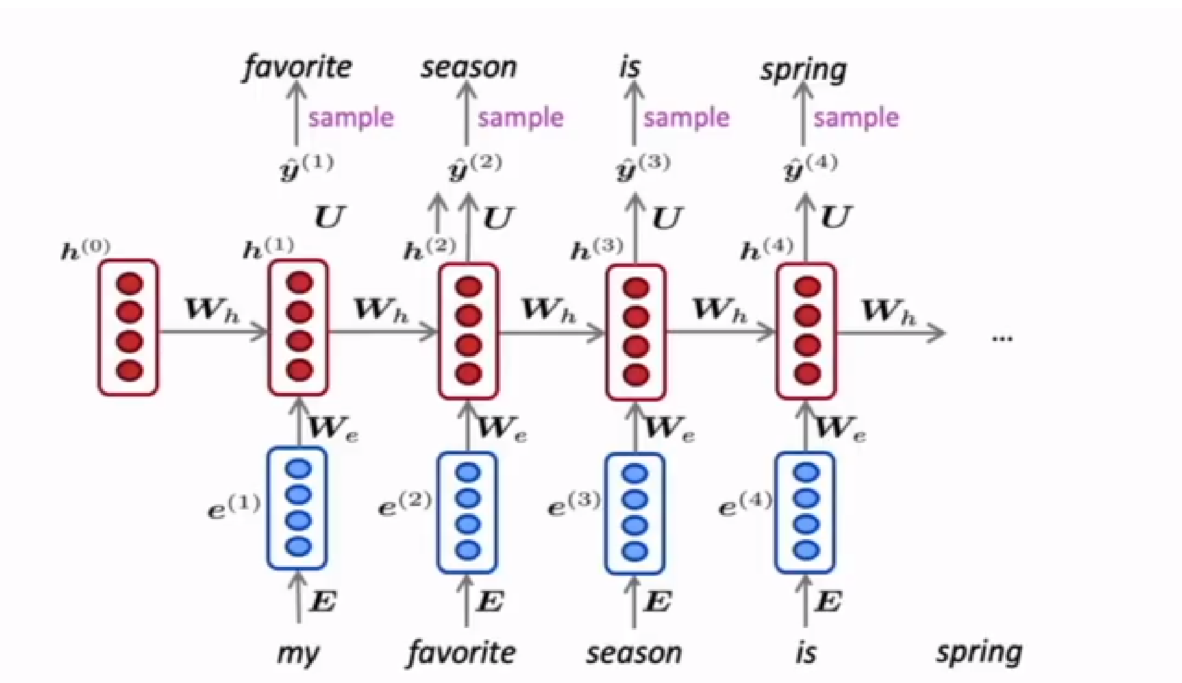
- 同一个word type("人")向量相同，没有考虑word token("人次")的context
- 同义词/派生词也有不同的适用场景，要根据context加以区分

Tips for unknown words with word vectors

1. char-level
2. Dhingra 为什么有unsupervised wordvec?#因为word2vec可以无监督学习corpus中的word embedding，但最后下游任务的vocab不包含corpus中所有单词，出现unk。



Did we solve the problem?



在lstm中，每层输出h作为预测单词的contextual word embedding。

核心思想

可以利用学到的contextual word embedding！

Contextual word embeddings

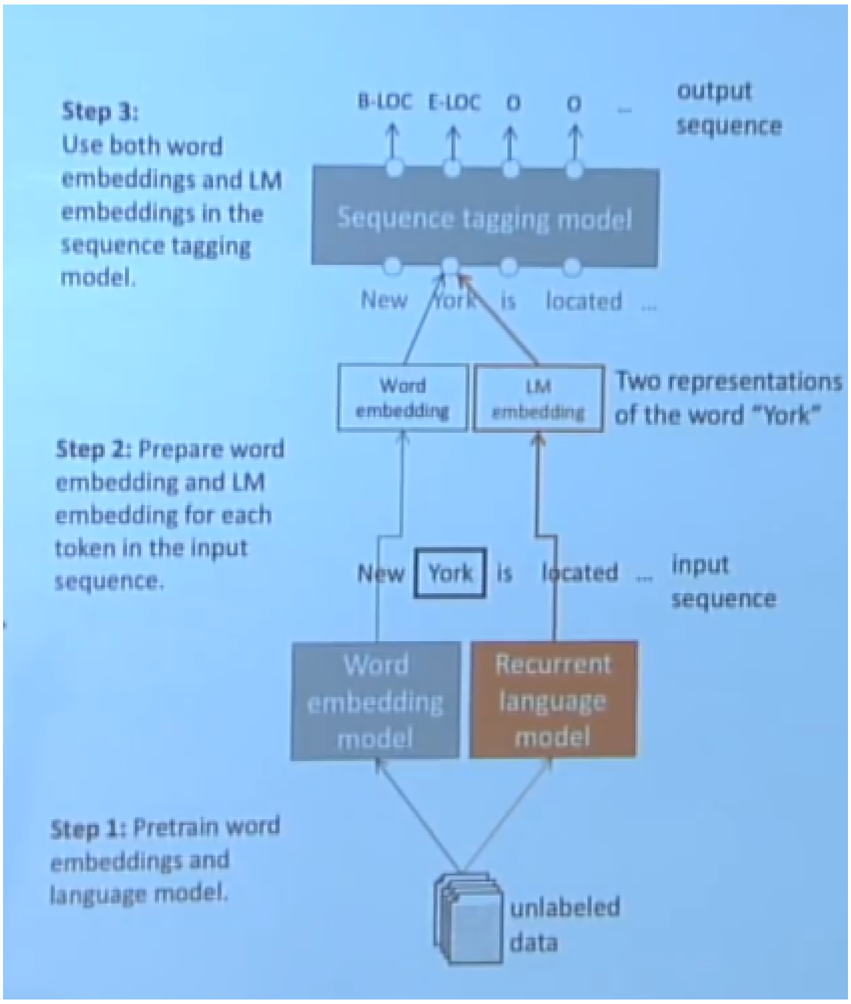
TagLM

用word embedding+LM embedding

https://arxiv.org/pdf/1705.00108.pdf

- Idea: Want meaning of word in context, but standardly learn task RNN only on small task-labeled data (e.g., NER)
- Why don't we do semi-supervised approach where we train NLM on large unlabeled corpus, rather than just word vectors?

如何无监督地pretrain wordvec和LM？#word2vec就是无监督的。LM可以通过"预测下一个单词"任务来完成，这是无监督的。通常pretrain的参数不会在train时改变。



ELMo (Embedding from Language Models)

ELMo是对TagLM的改进，包括

- 两 层bi-LSTM
- combine每层的特征而不是最后一层

- First run biLM to get representations for each word
- Then let (whatever) end-task model use them
  - Freeze weights of ELMo for purposes of supervised model
  - Concatenate ELMo weights into task-specific model
    - Details depend on task
      - Concatenating into intermediate layer as for TagLM is typical
      - Can provide ELMo representations again when producing outputs, as in a question answering system

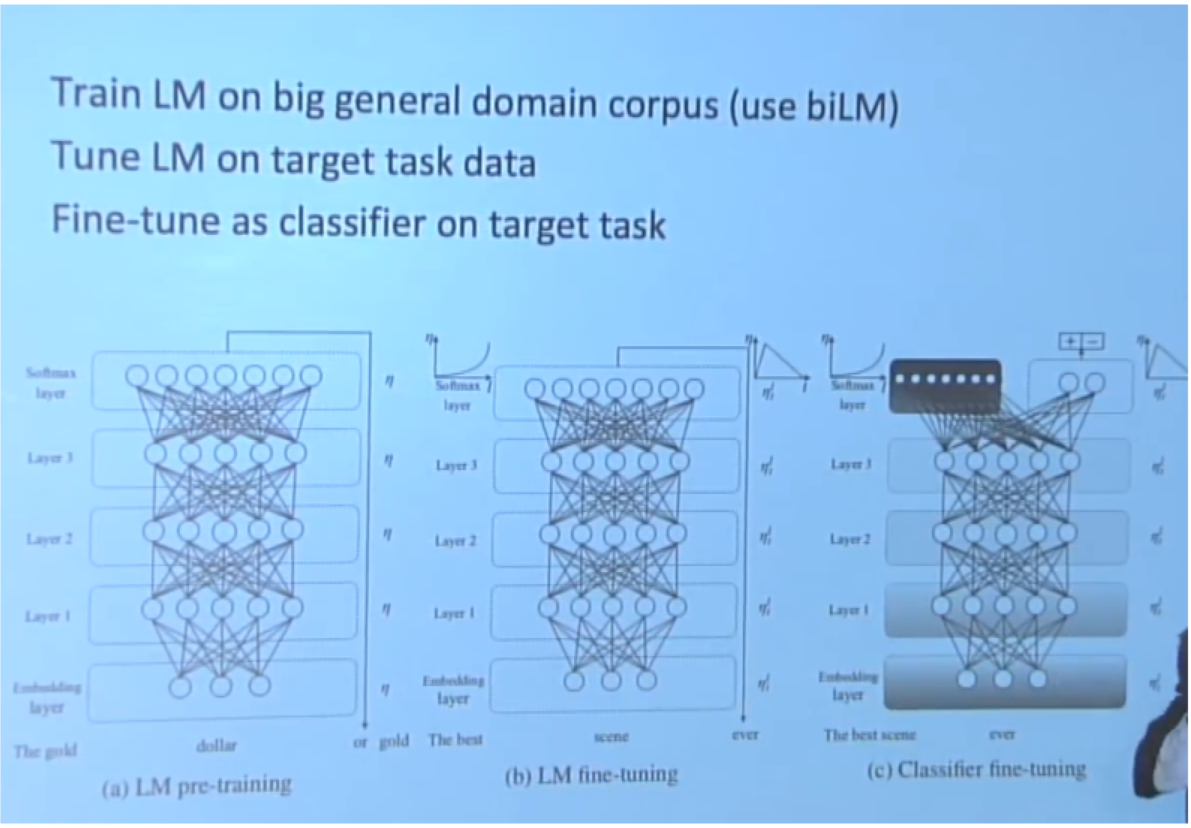
ELMo的最大贡献是，这种word embedding可以被用在不同下游任务上。

ULMfit (Universal Language Model Fine-tuning for Text Classification)

用transfer learning的思想，把一个任务/数据集的embedding迁移到另一个任务/数据集。

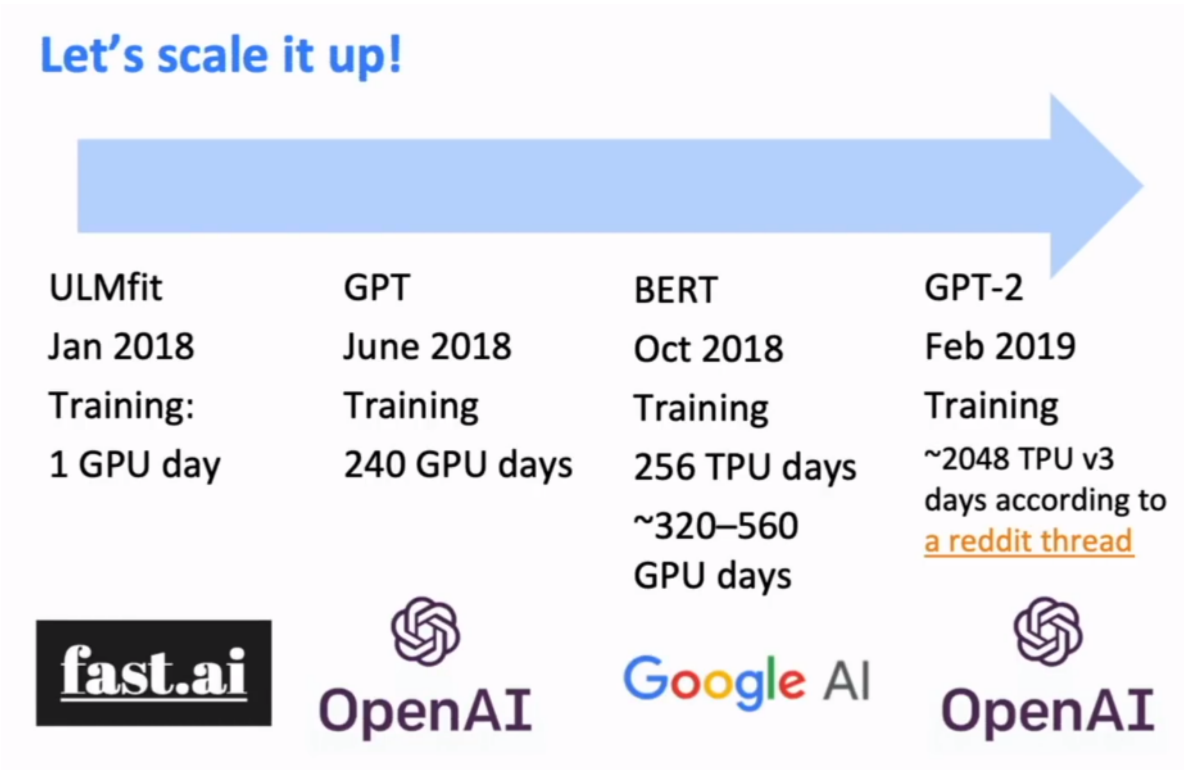
流程:

1. 在大的corpus上训练embedding
2. 在小的vocab上fine tune
3. 训练下游分类器



ULMfit提供了一个有效思路: 在大量数据上预训练，在少量数据上fine-tune。

由此出现了大公司的算力竞争，在巨大数据量的corpus上预训练LM：



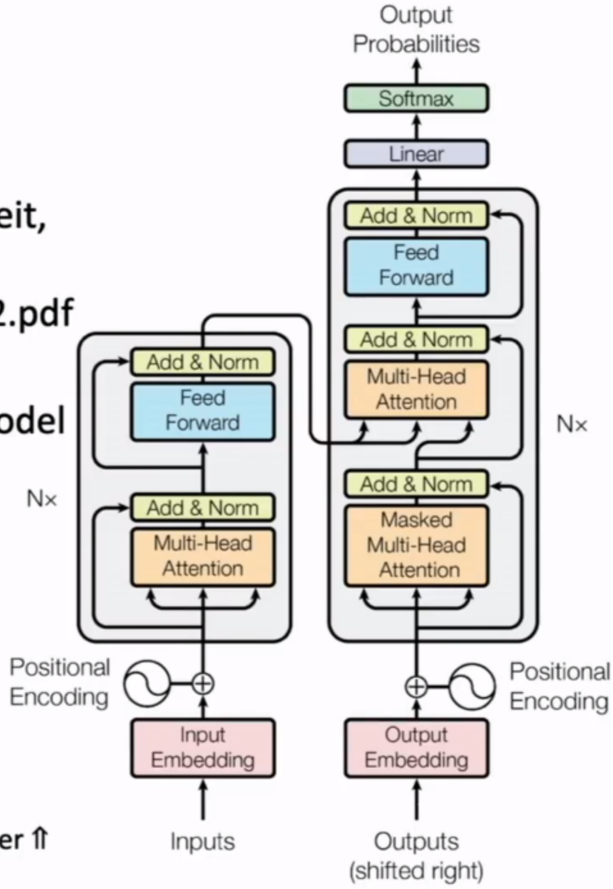
Transformers

Attention可以表示各个timestep之间的关系，可以只用attention。

Transformer Overview

Attention is all you need. 2017.
Aswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Kaiser, Polosukhin
https://arxiv.org/pdf/1706.03762.pdf

- Non-recurrent sequence-to-sequence encoder-decoder model
- Task: machine translation with parallel corpus
- Predict each translated word
- Final cost/error function is standard cross-entropy error on top of a softmax classifier



This and related figures from paper ⇑

38

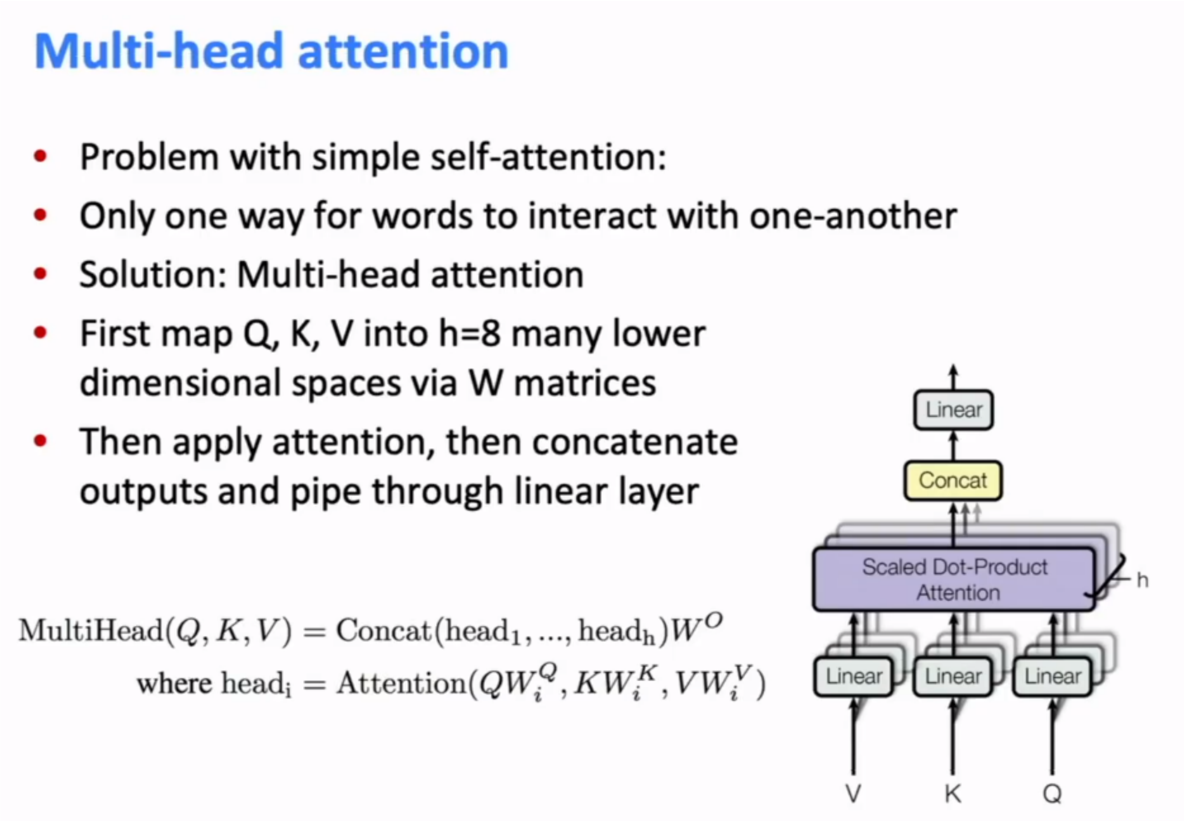
attention的q, K, V的解释。

Dot-Product Attention (Extending our previous def.)

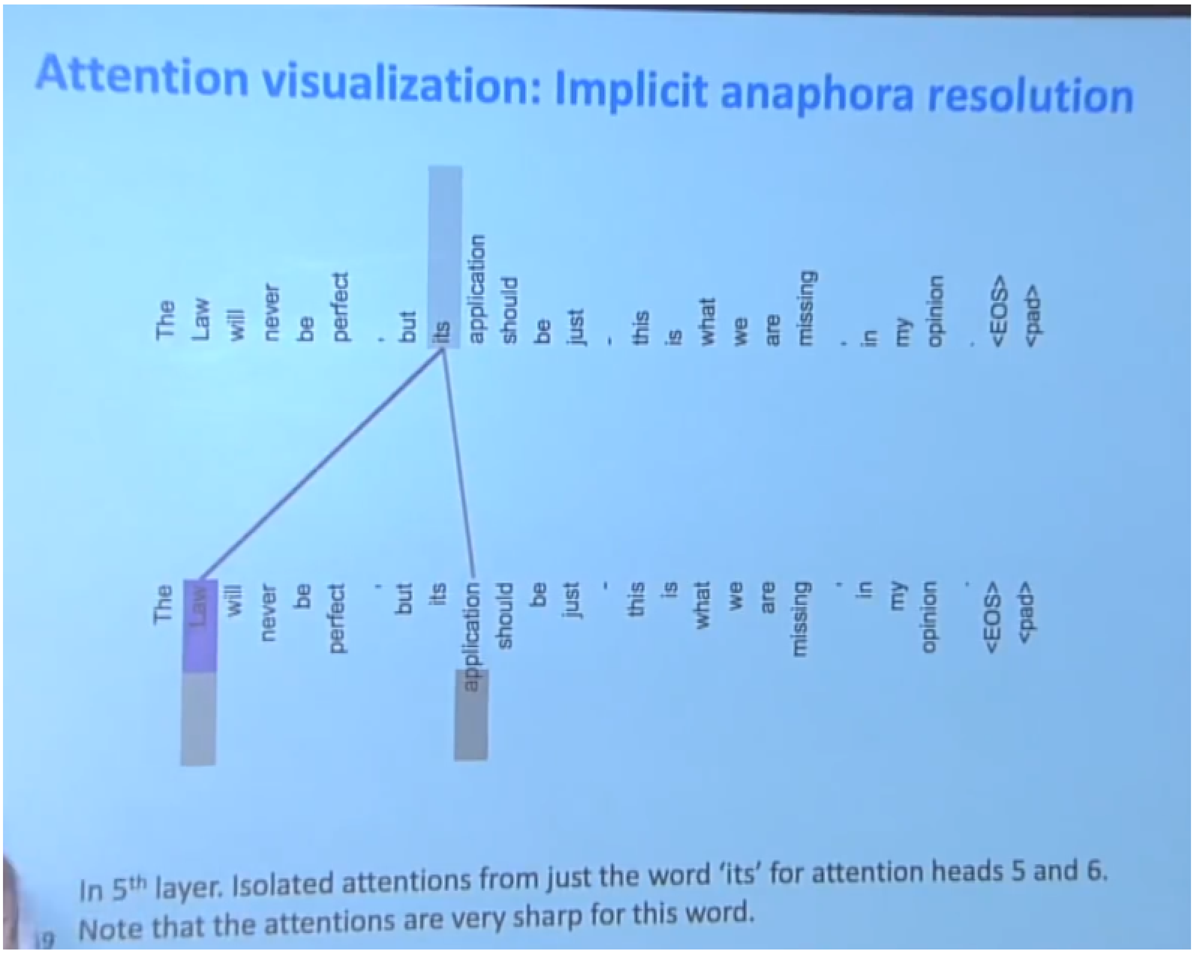
- Inputs: a query q and a set of key-value (k-v) pairs to an output
- Query, keys, values, and output are all vectors
- Output is weighted sum of values, where
- Weight of each value is computed by an inner product of query and corresponding key
- Queries and keys have same dimensionality d_k value have d_v

$$A(q, K, V) = \sum_i \frac{e^{q \cdot k_i}}{\sum_j e^{q \cdot k_j}} v_i$$

multi-head attention希望Q_i, K_i, V_i的不同i注意不同的东西。



attention可视化



BERT (Bidirectional Encoder Representations from Transformers)

Language Model也可以是双向的。seq2seq不是双向的吗？

单向是因为：

- LM会预测下一个word
- 看到两边的context是一 种作弊

解决方法