

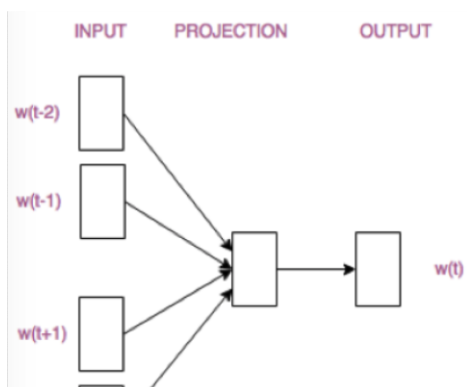
教程来自Hongyi Li的教程和Shusen Wang的教程以及Jay Alammar的博客

自从seq2seq模型引入到NLP领域之后，后续的模型基本上都延续了这encoder-decoder的结构。

其中 encoder 的目的是将字词进行编码，得到字词在多维向量空间representation，这个过程叫word embedding

1. Embedding的方法更迭

- One-hot encoding: 0,1对词库里的所有词进行编码。
 - 缺点是词的数量增多的时候会出现维度的爆炸；此外还有矩阵稀疏的问题。
- Word2vec, 推进了NLP领域的发展
 - CBOW



An example of CBOW Model

Corpus = { I drink coffee everyday }

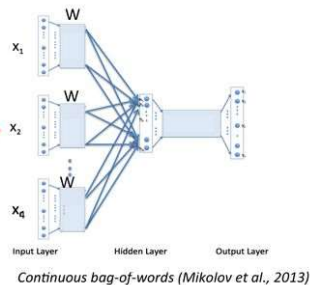
Initialize:

$$W = \begin{bmatrix} 1 & 2 & 3 & 0 \\ 1 & 2 & 1 & 2 \\ -1 & 1 & 1 & 1 \end{bmatrix}$$

Ex:

$$W_{drink} = [0, 1, 0, 0]$$

$$\begin{bmatrix} 1 & 2 & 3 & 0 \\ 1 & 2 & 1 & 2 \\ -1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \\ 1 \end{bmatrix}$$

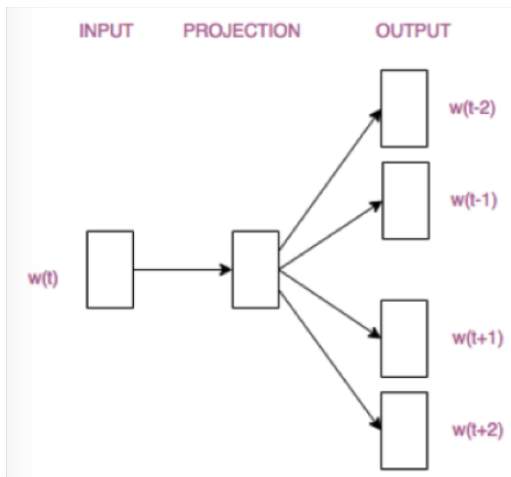


优点：向量运算有实际的意义，如King-Man+woman = queen

缺点：每个单词只有一个向量来表征，没法分辨同义词

为了减少训练的时间，有同义词之间只是微调参数，引入了霍夫曼树的方法

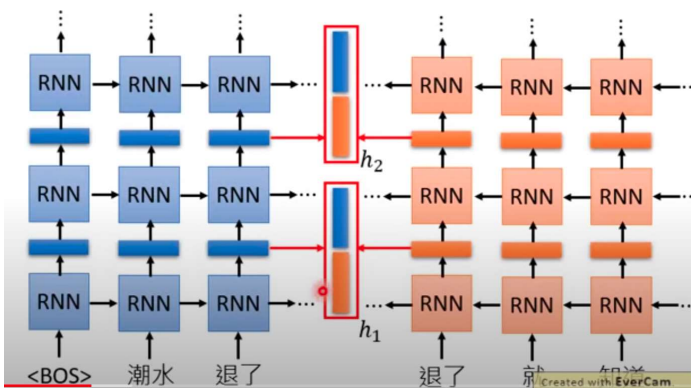
- Skip-gram



优点：解决了contextual的问题，根据上下文意思的不同，每个词的Embedding是不一样的

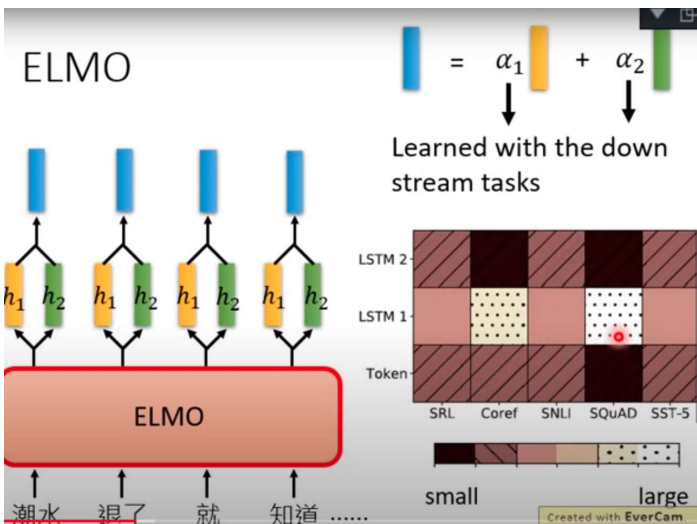
缺点：LSTM是seq2seq的，速度慢

- ELMo (Embedding from language models)



每一层的LSTM能够生成一个latent representation

不同任务的 α_1 和 α_2 是不一样的，这两个参数根据不同的任务来训练



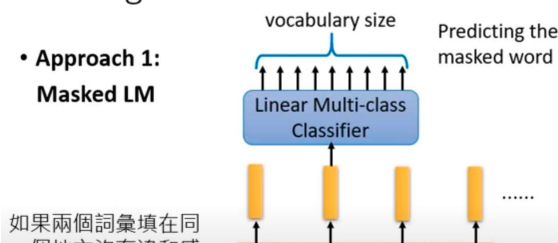
右下角的图表示的是哪一层的参数对任务的影响最大

- BERT (Bidirectional Encoder representation from transformers)

两种训练方法，一种是Masked LM，预测句子中被遮盖住的某一起训练的

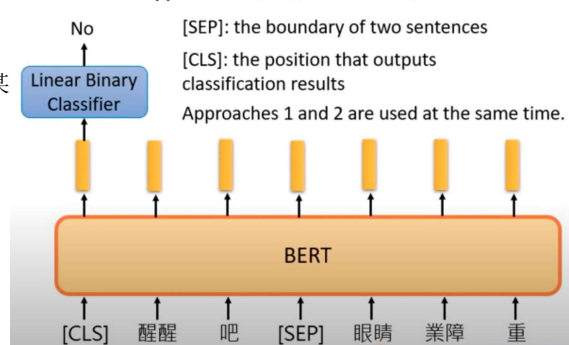
Training of BERT

- Approach 1: Masked LM



如果兩個詞彙填在同一个地方左右边和

Approach 2: Next Sentence Prediction

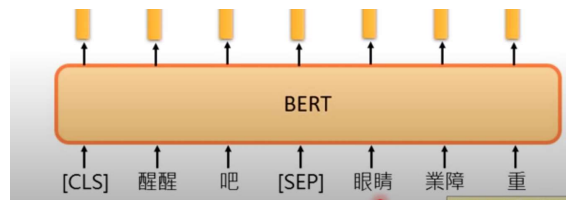
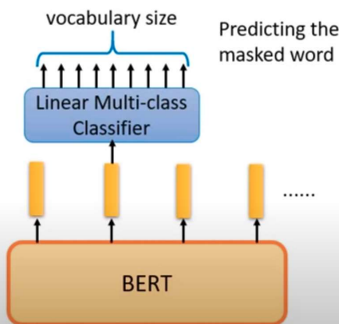


一般情况下两种都是

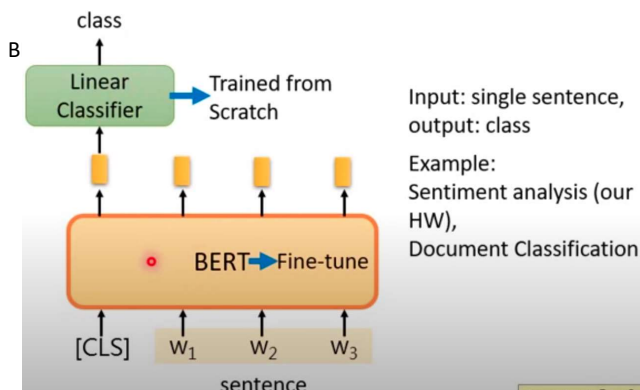
Training of BERT

Approach 1: Masked LM

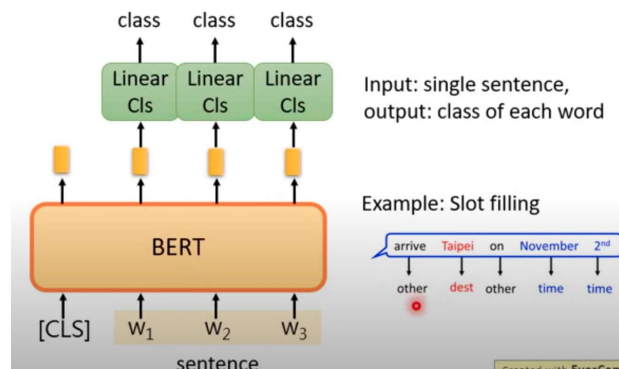
如果兩個詞彙填在同一個地方沒有違和感
那它們就有類似的embedding



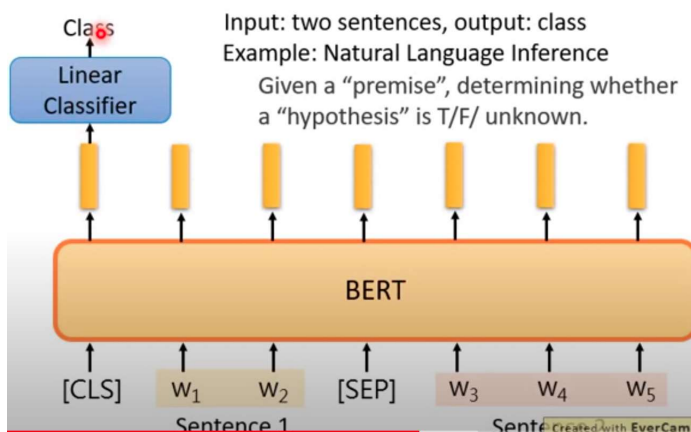
How to use BERT – Case 1



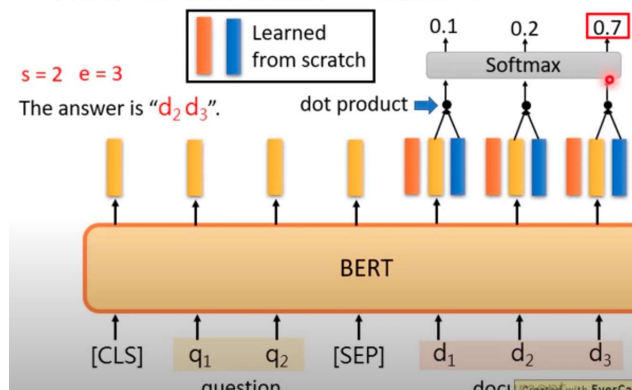
How to use BERT – Case 2



How to use BERT – Case 3



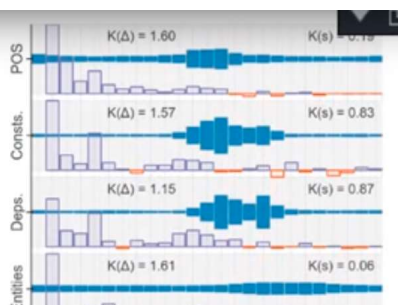
How to use BERT – Case 4



这个分层，就是把每一层encoder的hidden state提取出来
其实各个层级的hidden state加起来显示的F1是不一样的

What does BERT learn?

<https://arxiv.org/abs/1905.05950>
<https://openreview.net/pdf?id=SJzSgnRcKX>



FROM PAPER:
Dissecting Contextual Word Embeddings:
Architecture and Representation

WHAT DO YOU LEARN FROM CONTEXT,
PROBING FOR

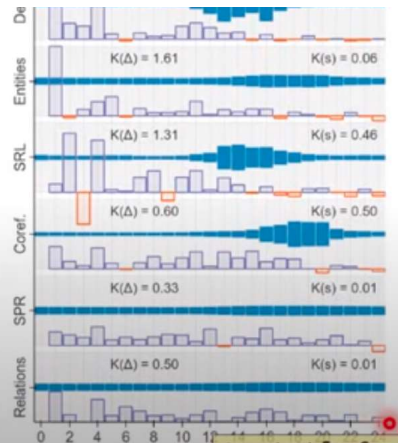
BERT Rediscovered the Classical NLP Pipeline

Evaluation as word representations

- MultiNLI dataset
- Semantic Role Labeling, Ontonotes dataset
- Constituency parsing, Penn Treebank

<https://arxiv.org/abs/1905.09330>
<https://openreview.net/pdf?id=SIzSgnRcKX>

	F1 Scores	Expected layer & center-of-gravity
	$t=0$ $t=24$	0 2 4 6 8 10 12 14 16
POS	88.5 96.7	3.39 11.68
Consts.	73.6 87.0	3.79 13.06
Deps.	85.6 95.5	5.69 13.75
Entities	90.6 96.1	4.64 13.16
SRL	81.3 91.4	6.54 13.63
Coref.	80.5 91.9	9.47 15.80
SPR	77.7 83.7	9.93 12.72
Relations	60.7 84.2	9.40 12.83



Evaluation as word representations

- MultiNLI dataset
- Semantic Role Labeling, Ontonotes dataset
- Constituency parsing, Penn Treebank
- Name entity recognition: NER

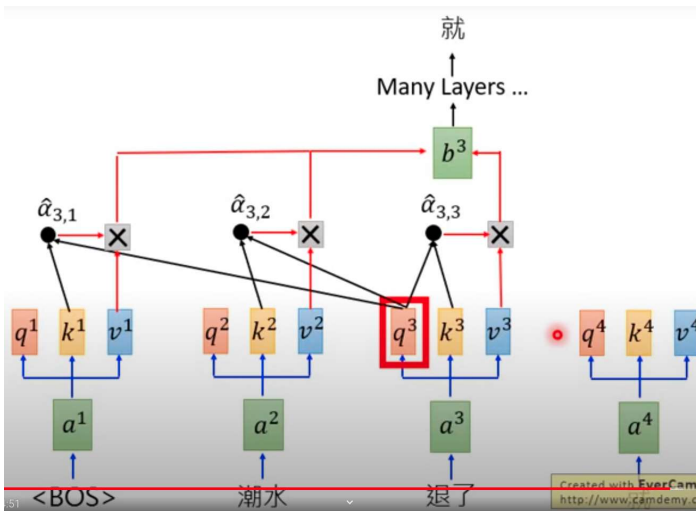
Contextual similarity

- Intra-sentence similarity
- Span representations
- Unsupervised pronominal coref

Probing contextual information

- POS: part-of-speech tagging
- Constituency parsing

- GPT (Generative Pre-training) 是Transformer的decoder



- Reading Comprehension

$d_1, d_2, \dots, d_N, "Q:", q_1, q_2, \dots, q_M, "A:"$

- Summarization $d_1, d_2, \dots, d_N, "TL;DR:"$

- Translation

English sentence 1	=	French sentence 1
English sentence 2	=	French sentence 2
English sentence 3	=	

Few-shot Learning

(no gradient descent)

1	Translate English to French:	task description
2	sea otter => loutre de mer	examples
3	peppermint => menthe poivrée	
4	plush girafe => girafe peluche	
5	cheese =>	prompt

One-shot Learning

1	Translate English to French:	task description
2	sea otter => loutre de mer	example
3	cheese =>	prompt

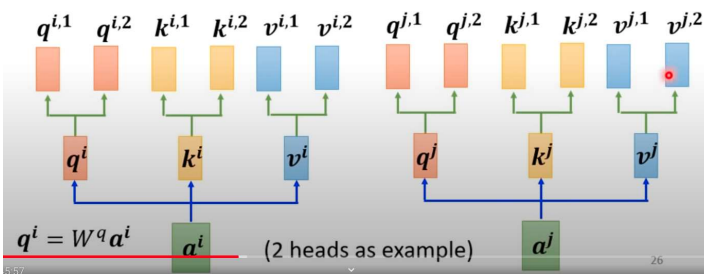
Zero-shot Learning

1	Translate English to French:	task description
2	cheese =>	prompt

Multi-head Self-attention Different types of relevance

$$q^{i,1} = W^{q,1} q^i$$

$$q^{i,2} = W^{q,2} q^i$$



2. Attention

- (NNLM) [A Neural Probabilistic Language Model](#)
- (Word2Vec) [Distributed Representations of Words and Phrases and their Compositionality](#)
- (GloVe) [GloVe: Global Vectors for Word Representation](#)
- (ELMo) [Deep contextualized word representations](#)
- (GPT) [Improving Language Understanding by Generative Pre-Training](#)
- (BERT) [BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding](#)