






cs224-lecture13-contextual-word-representations

 Created By	 Geunho Lee
 Last Edited	@Sep 20, 2020 10:33 AM
 Property	
 Tags	

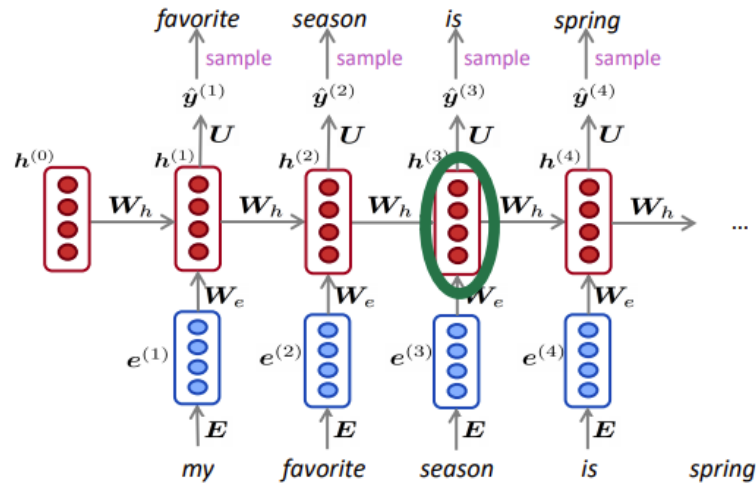
Representations of a word

기존에 배웠던 Word2vec, GloVe, fastText는 기본적으로 하나의 단어에 하나의 표상 (Representation)을 가짐

2가지 문제가 있는데

- 단어가 존재하는 곳에서의 문맥과 관계 없이 같은 표상을 가지게 됨
 - word sense disambiguation
- semantic, syntactic behavior and register와 관계 없이 하나의 표상만 가짐

- Those LSTM layers are trained to predict the next word
- But those language models are producing context-specific word representations at each position!



7

LSTM으로 이런 문제를 좀 해결해보려고 하지만, 결국에 이런 언어모델도 각 포지션에서의 context-specific word representation을 만들게 됨

Pre-ELMo and ELMo

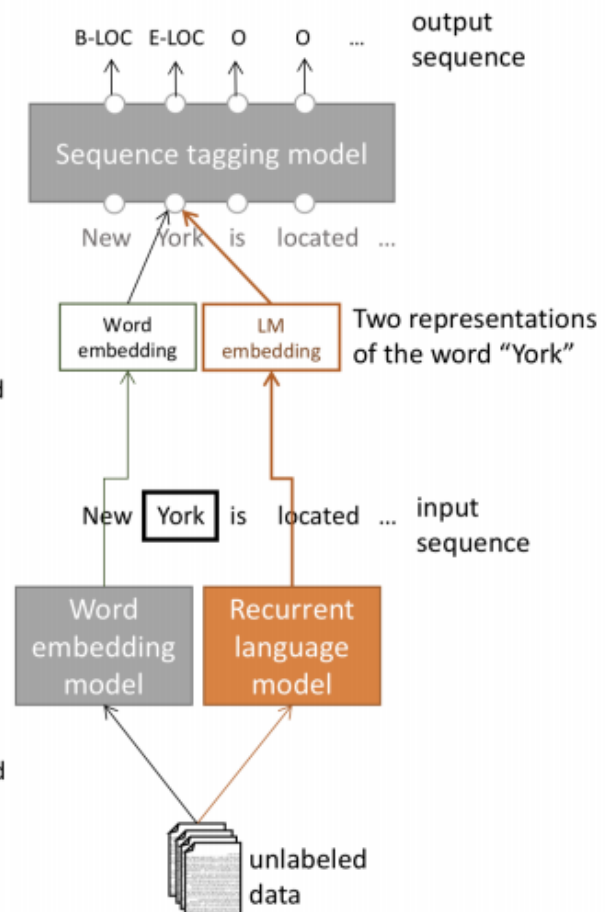
Tag LM (Peters et al. 2017)

Step 3:

Use both word embeddings and LM embeddings in the sequence tagging model.

Step 2: Prepare word embedding and LM embedding for each token in the input sequence.

Step 1: Pretrain word embeddings and language model.



Named Entity Recognition (NER)

- Find and classify names in text, for example:
 - The decision by the independent MP Andrew Wilkie to withdraw his support for the minority Labor government sounded dramatic but it should not further threaten its stability. When, after the 2010 election, Wilkie, Rob Oakeshott, Tony Windsor and the Greens agreed to support Labor, they gave just two guarantees: confidence and supply.

Person
Date
Location
Organi- zation

CoNLL 2003 Named Entity Recognition (en news testb)

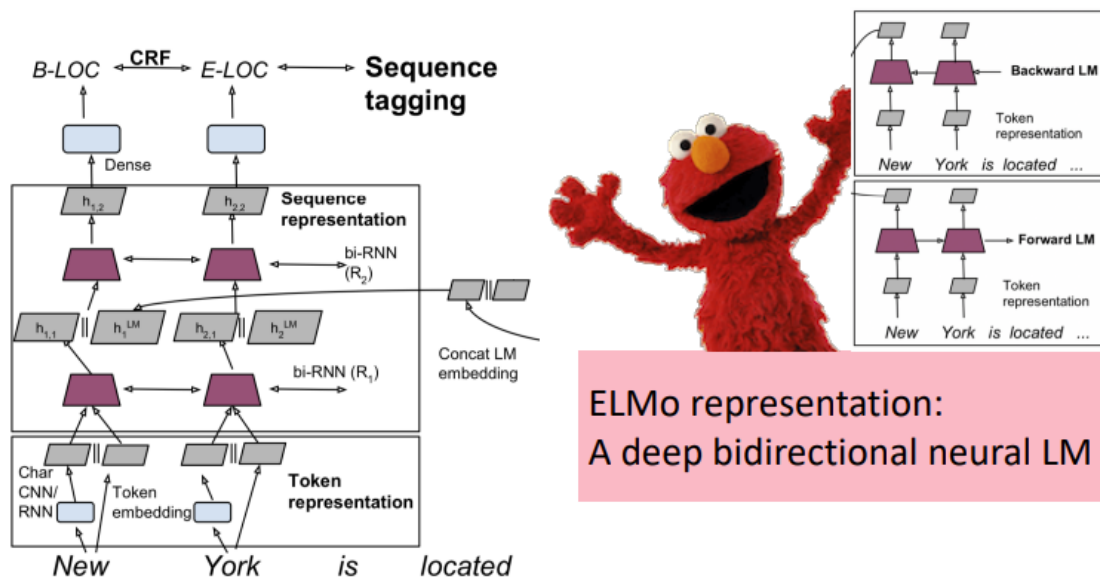
Name	Description	Year	F1
TagLM Peters	LSTM BiLM in BiLSTM tagger	2017	91.93
Ma + Hovy	BiLSTM + char CNN + CRF layer	2016	91.21
Tagger Peters	BiLSTM + char CNN + CRF layer	2017	90.87
Ratinov + Roth	Categorical CRF+Wikipeda+word cls	2009	90.80
Finkel et al.	Categorical feature CRF	2005	86.86
IBM Florian	Linear/softmax/TBL/HMM ensemble, gazettes++	2003	88.76
Stanford Klein	MEMM softmax markov model	2003	86.07

- LM trained on supervised data does not help
- forward 보다 bidirectional LM이 0.2 포인트 정도 도움 됨
- huge LM design이 작은 것보다 0.3 포인트 도움 됨
- 단순히 LM embedding보다는 BiLSTM 같이 쓰는 것이 도움 됨

ELMo: Embeddings from Language Models

- Train a bidirectional LM
- Aim at performant but not overly large LM:
 - Use 2 biLSTM layers
 - Use character CNN to build initial word representation (only)
 - 2048 char n-gram filters and 2 highway layers, 512 dim projection
 - Use 4096 dim hidden/cell LSTM states with 512 dim projections to next input
 - Use a residual connection
 - Tie parameters of token input and output (softmax) and tie these between forward and backward LMs

Breakout version of **deep contextual word vectors**



ELMo representation:
A deep bidirectional neural LM

Use learned, task-weighted
average of (2) hidden layers

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

19

- Freeze weights of ELMo for purposes of supervised model
- Concatenate ELMo weights into stack-specific model

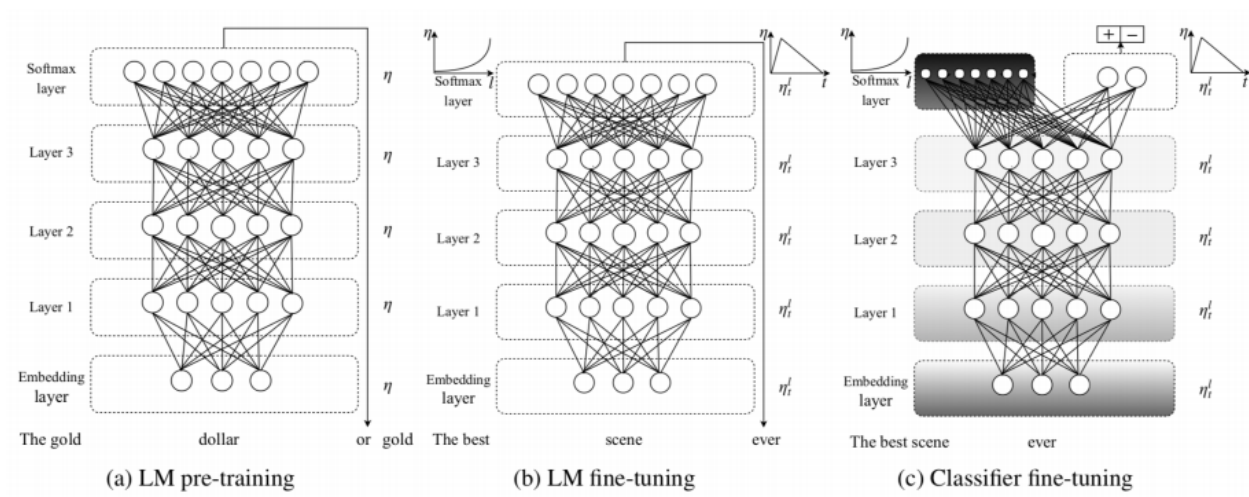
ULMfit

ULMfit

Train LM on big general domain corpus (use biLM)

Tune LM on target task data

Fine-tune as classifier on target task



25

Transformer

Let's scale it up!



ULMfit

Jan 2018

Training:

1 GPU day



GPT

June 2018

Training

240 GPU days



BERT

Oct 2018

Training

256 TPU days

~320–560

GPU days



GPT-2

Feb 2019

Training

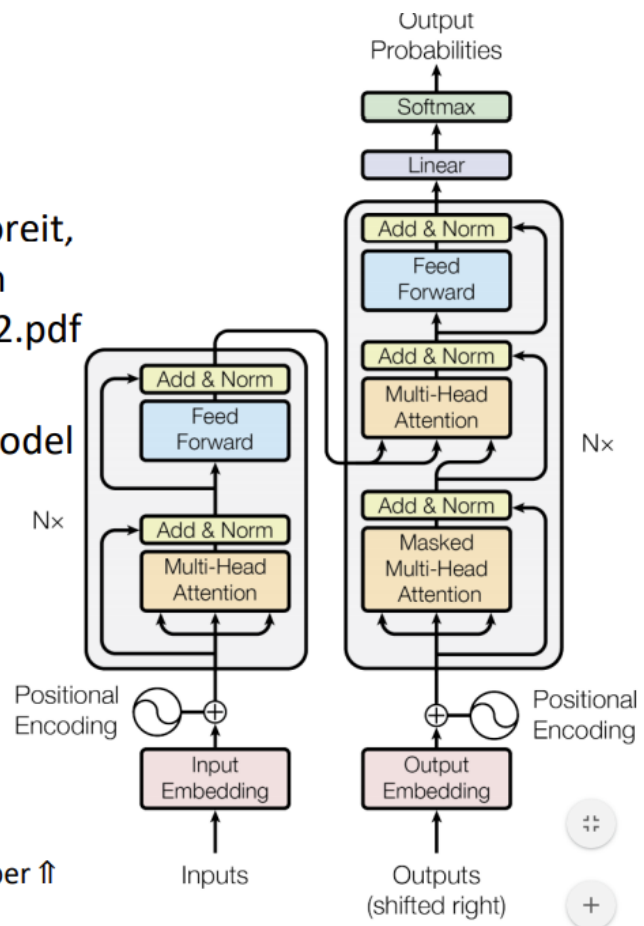
~2048 TPU v3
days according to
[a reddit thread](#)



The Transformer

Attention is all you need. 2017.
 Vaswani, 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 ↑

33

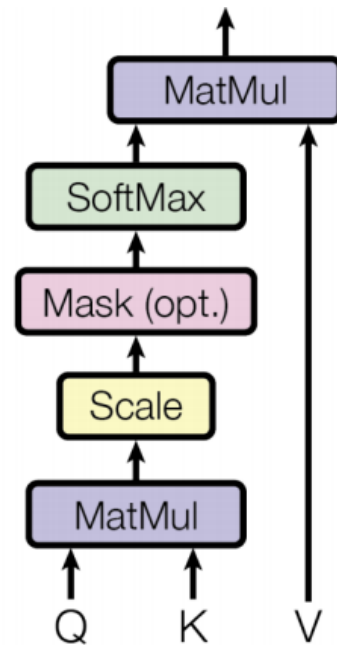
Scaled dot-product attention

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

- Solution: Scale by length of query/key vectors:

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

37



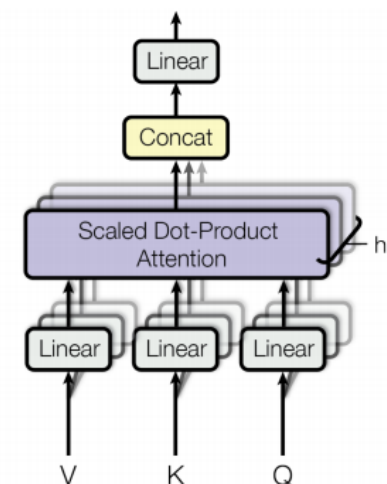
Multi-head attention

Multi-head attention

- Problem with simple self-attention:
- Only one way for words to interact with one-another
- Solution: Multi-head attention
- First map Q, K, V into h=8 many lower dimensional spaces via W matrices
- Then apply attention, then concatenate outputs and pipe through linear layer

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O$$

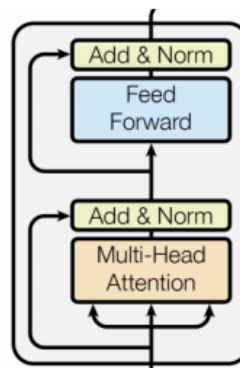
where $\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$



Complete transformer block

Each block has two “sublayers”

1. Multihead attention
2. 2-layer feed-forward NNet (with ReLU)



Each of these two steps also has:

Residual (short-circuit) connection and LayerNorm

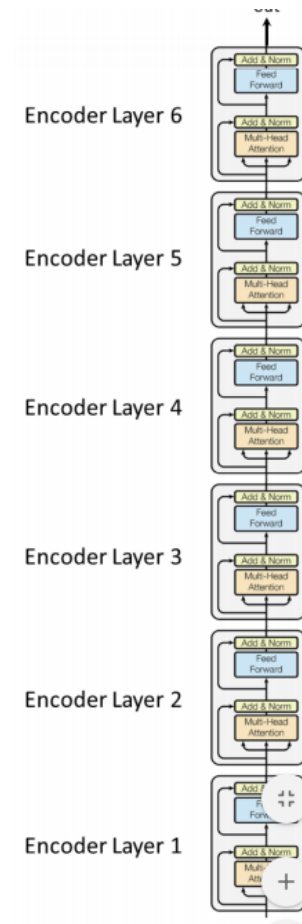
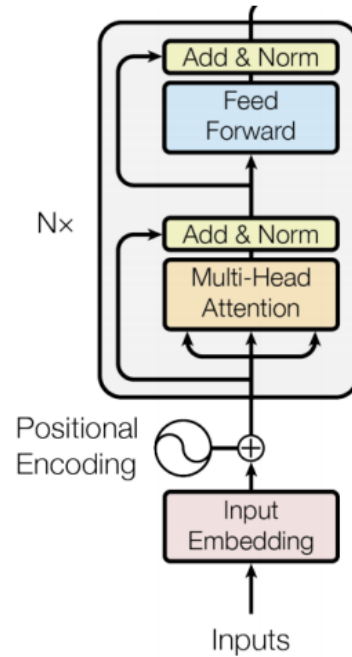
$\text{LayerNorm}(x + \text{Sublayer}(x))$

LayerNorm changes input features to have mean 0 and variance 1 per layer (and adds two more parameters)

$$\mu^l = \frac{1}{H} \sum_{i=1}^H a_i^l \quad \sigma^l = \sqrt{\frac{1}{H} \sum_{i=1}^H (a_i^l - \mu^l)^2} \quad h_i = f\left(\frac{g_i}{\sigma_i} (a_i - \mu_i) + b_i\right)$$

Complete Encoder

- Blocks are repeated 6 or more times
 - (in vertical stack)



43

Transformer Decoder

- 2 sublayer changes in decoder
- Masked decoder self-attention on previously generated outputs:



- Encoder-Decoder Attention, where queries come from previous decoder layer and keys and values come from output of encoder



44 Blocks repeated 6 times also

