School of CIT -
Social Computing
Research Group

Technical University of Munich

Natural Language Processing
IN2361

Prof. Dr. Georg Groh

Chapter 11
Fine-Tuning and Masked LMs

- content is based on [1] and [2]
- certain elements (e.g. equations or tables) were taken over or taken over in a modified form from [1] and [2]
- citations of [1] and [2] or from [1] and [2] are partly omitted for legibility
- errors are fully in the responsibility of Georg Groh
- BIG thanks to Dan and James for a great book!
- BIG thanks to Chris Manning & all the guys at Stanford for excellent lecture materials!

Motivation

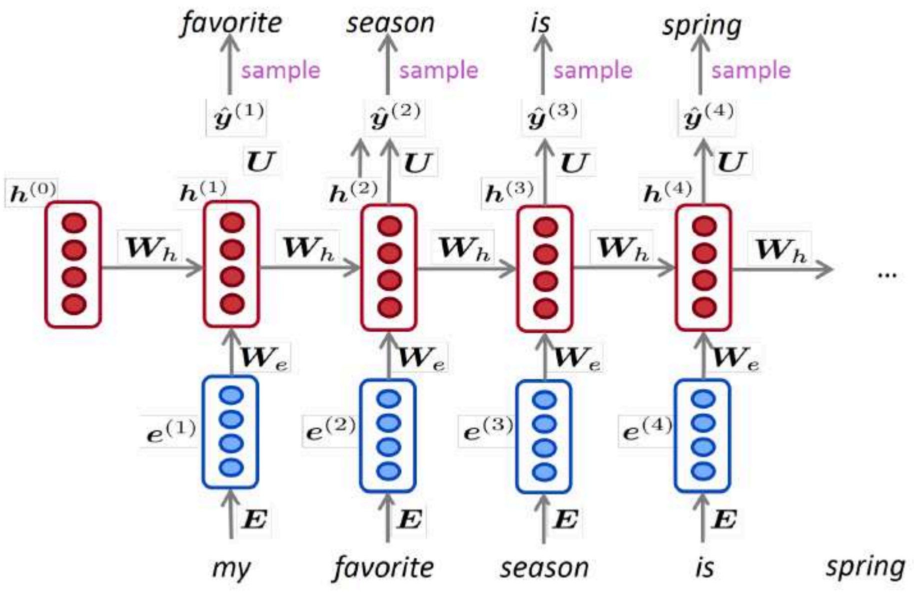
- The vocabulary size of young adult speakers of American English range from 30,000 to 100,000.
- How do children learn new words?
- Children have to learn about 7 to 10 words every single day!
- Most of this growth is not happening through direct vocabulary instruction in school.
- The bulk of this knowledge acquisition happens as a by-product of reading.
- → previously presented word learning approaches are based on distributional hypothesis.

Motivation (2) : Contextual Embeddings

- GloVe, Word2Vec etc: learn only **one** dense **embedding for each word** (more precisely for each **word type**) using distributional semantics and large amounts of text ("semi-supervised")

- *however*: word tokens can have different aspects (e.g. semantic (word senses), or syntactic behavior) in different contexts → make **embedding depend on context**!

- *initial solution idea:* use RNN style language modelling: **hidden states as context-aware embeddings** for tokens:



- → "Embedding" now (BERT etc.) : == **whole pre-trained NN** allowing to take whole context as input at test time

This Lecture’s Terminology

- Contextual-embeddings: each word *w* will be represented by a different vector each time it appears in a different context.

- Pretraining: learning some sort of representation for words or sentences by processing very large amounts of text. → pretrained language models.

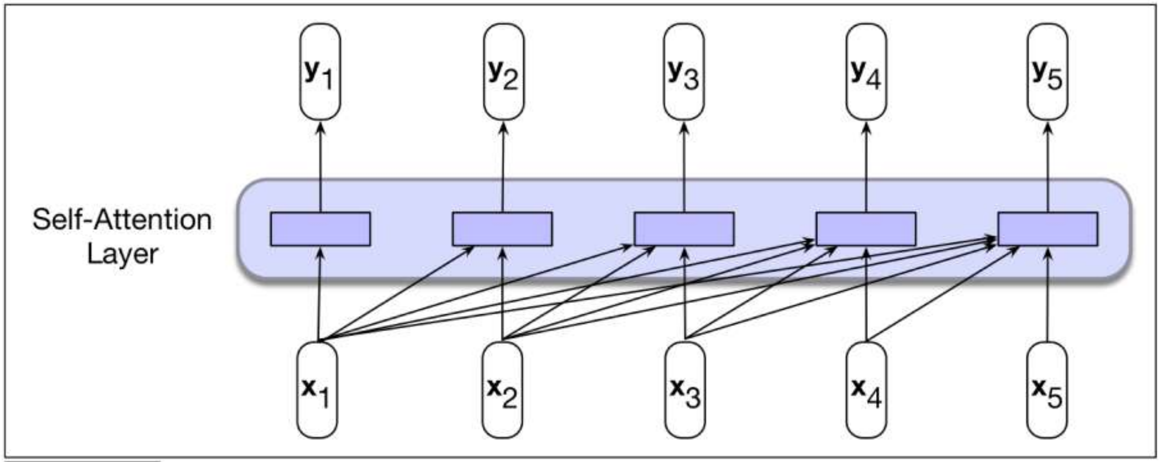
- Fine-tuning: using pretrained models + further training the model to perform some down-stream task.

- Transfer learning: acquiring knowledge from one task or domain and then applying it (transferring it) to a new task.

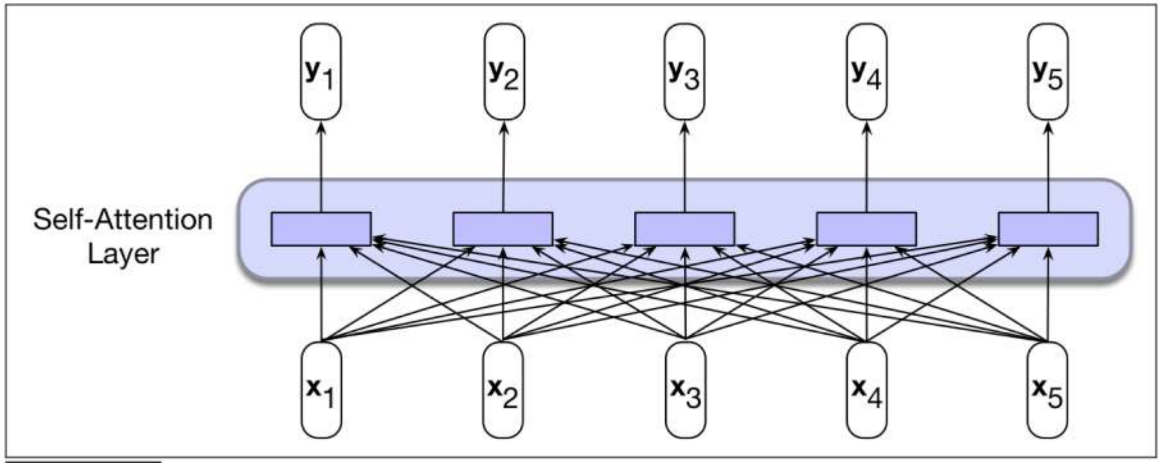
Bidirectional Transformer Encoders

- Motivation: casual (let-to-right) transformers are powerful for autoregressive generation.
- But: for e.g. sequence classification usage of information only from the left context is not enough.

Left-to-right attention



Bidirectional attention



Bidirectional Transformer Encoders

- How? → same self-attention mechanism as in casual transformers.
- as before: map sequences of input embedding vectors $(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ to sequences of output embedding vectors of the same length $(\mathbf{y}_1, \ldots, \mathbf{y}_n)$.

Recap: Self-attention:

$$
\left.\begin{aligned}
&\mathbf{q}_i = \mathbf{W^Q x}_i;\;\; \mathbf{k}_i = \mathbf{W^K x}_i;\;\; \mathbf{v}_i = \mathbf{W^V x}_i \\
&\alpha_{ij} = \frac{\exp(\text{score}_{ij})}{\sum_{k=1}^{n} \exp(\text{score}_{ik})} \\
&\text{score}_{ij} = \mathbf{q}_i \cdot \mathbf{k}_j \\
&\mathbf{y}_i = \sum_{j=i}^{n} \alpha_{ij} \mathbf{v}_j
\end{aligned}\right\}
$$

$$\mathbf{Q} = \mathbf{XW^Q};\;\; \mathbf{K} = \mathbf{XW^K};\;\; \mathbf{V} = \mathbf{XW^V}$$

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

$$
\begin{pmatrix}
\mathbf{X} \in \mathbb{R}^{N \times d_h} & \mathbf{Q} \in \mathbb{R}^{N \times d} \\
\mathbf{K} \in \mathbb{R}^{N \times d} & \mathbf{V} \in \mathbb{R}^{N \times d}
\end{pmatrix}
$$

Bidirectional Transformer Encoders

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

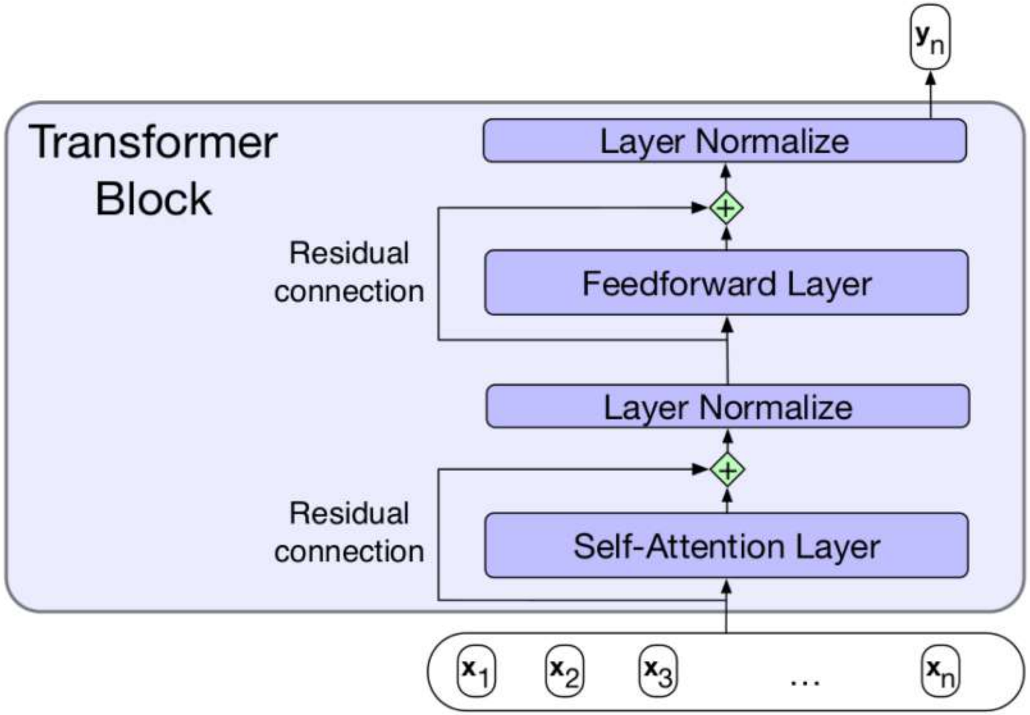
N

q1•k1	~~q1•k2~~	~~q1•k3~~	~~q1•k4~~	~~q1•k5~~
q2•k1	q2•k2	~~q2•k3~~	~~q2•k4~~	~~q2•k5~~
q3•k1	q3•k2	q3•k3	~~q3•k4~~	~~q3•k5~~
q4•k1	q4•k2	q4•k3	q4•k4	~~q4•k5~~
q5•k1	q5•k2	q5•k3	q5•k4	q5•k5

N

- In casual transformer (e.g. for language modelling / NMT decoding): upper triangular portion of $\mathbf{QK}$ matrix is masked.
- In bidirectional transformer (e.g. for NMT encoding, sequence classification) no masking
  → allow the model to contextualize each token using information from the entire input.

Bidirectional Transformer Encoders



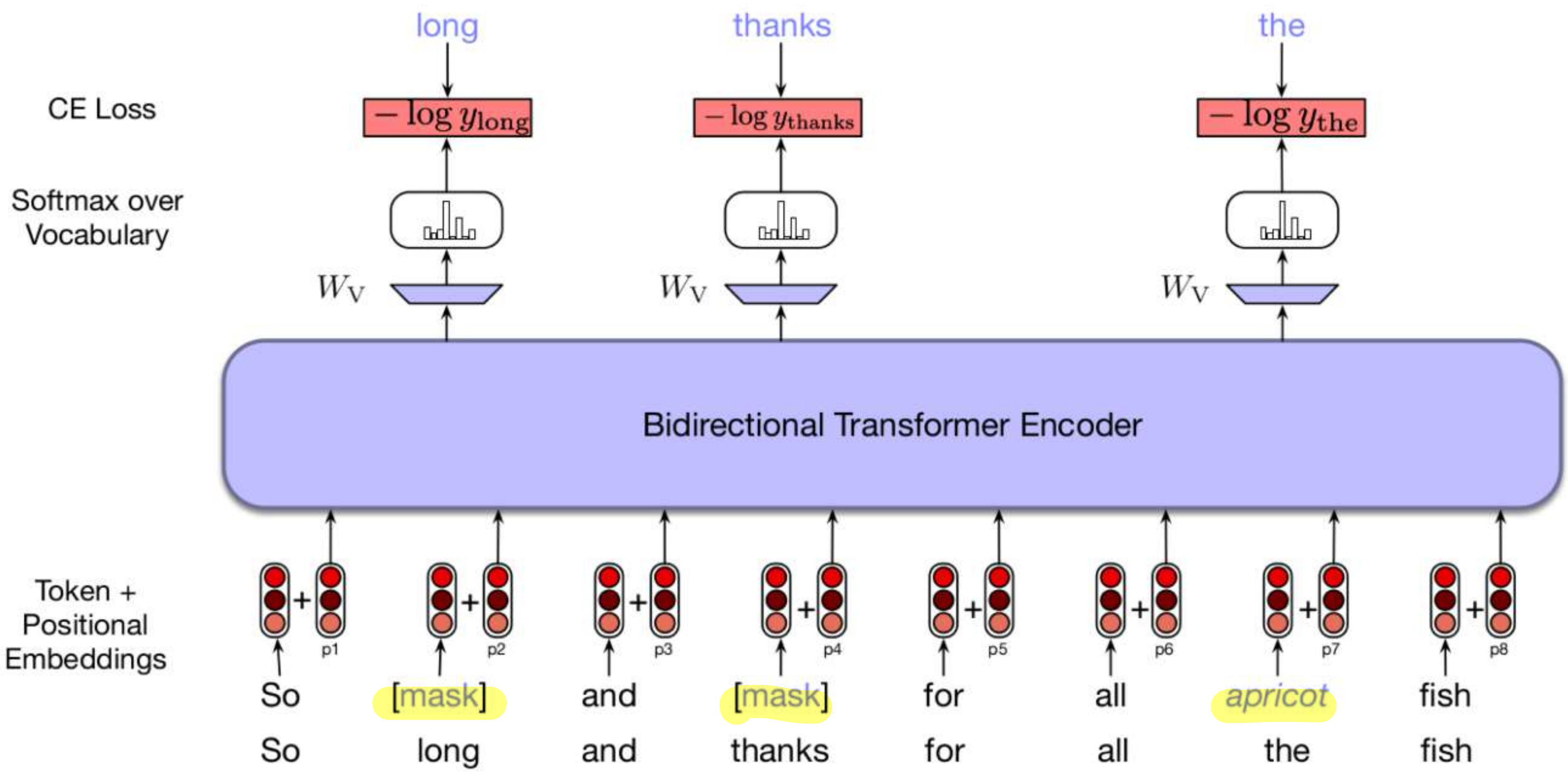
Original bidirectional encoder model BERT (Devlin et al., 2019):

- subword vocabulary: 30,000 tokens generated using the Word-Piece algorithm (Schuster and Nakajima, 2012);
- Hidden layers size: 768;
- 12 layers of transformer blocks, with 12 multi-head attention layers each.

100M parameters in the model. Fixed input of 512 tokens.

Masking Words

- How to train bidirectional transformers? Cloze task (Taylor, 1953): predict the missing elements.
- → Masked Language Modeling (MLM) (Devlin et al., 2019): the original approach to training bidirectional encoders. → mask some words



Masking Words

randomly select a token in train text and use it in one of three ways:

- replace with unique vocabulary token [MASK].
- replace with another token from the vocabulary, randomly sampled based on token unigram probabilities. (confuse the model on purpose)
- it is left unchanged.

MLM training objective: predict the original inputs for each of the masked tokens using a bidirectional encoder.

store ↑ gallon ↑

the man went to the [MASK] to buy a [MASK] of milk

In BERT:

- 15% of the input tokens are sampled for learning;
- Of these, 80% are replaced with [MASK];
- 10% are replaced with randomly selected tokens;
- the remaining 10% are left unchanged

Masking Words

randomly select a token in train text and use it in one of three ways:

- replace with unique vocabulary token [MASK].
- replace with another token from the vocabulary, randomly sampled based on token unigram probabilities. (confuse the model on purpose)
- it is left unchanged.

MLM training objective: predict the original inputs for each of the masked tokens u

k ≈ 15 %:
too little masking: too expensive training (you should eventually mask and guess every word);
too much masking: not enough context (quality suffers)

In BERT:

- 15% of the i
- Of these, 80% are replaced with [MASK];
- 10% are replaced with randomly selected tokens;
- the remaining 10% are left unchanged

Masking Spans

- Many NLP tasks (question answering, syntactic parsing, coreference, semantic role labeling) involve the identification and classification of constituents, or phrases.

- Span: a contiguous sequence of one or more words selected from a training text, prior to sub-word tokenization.

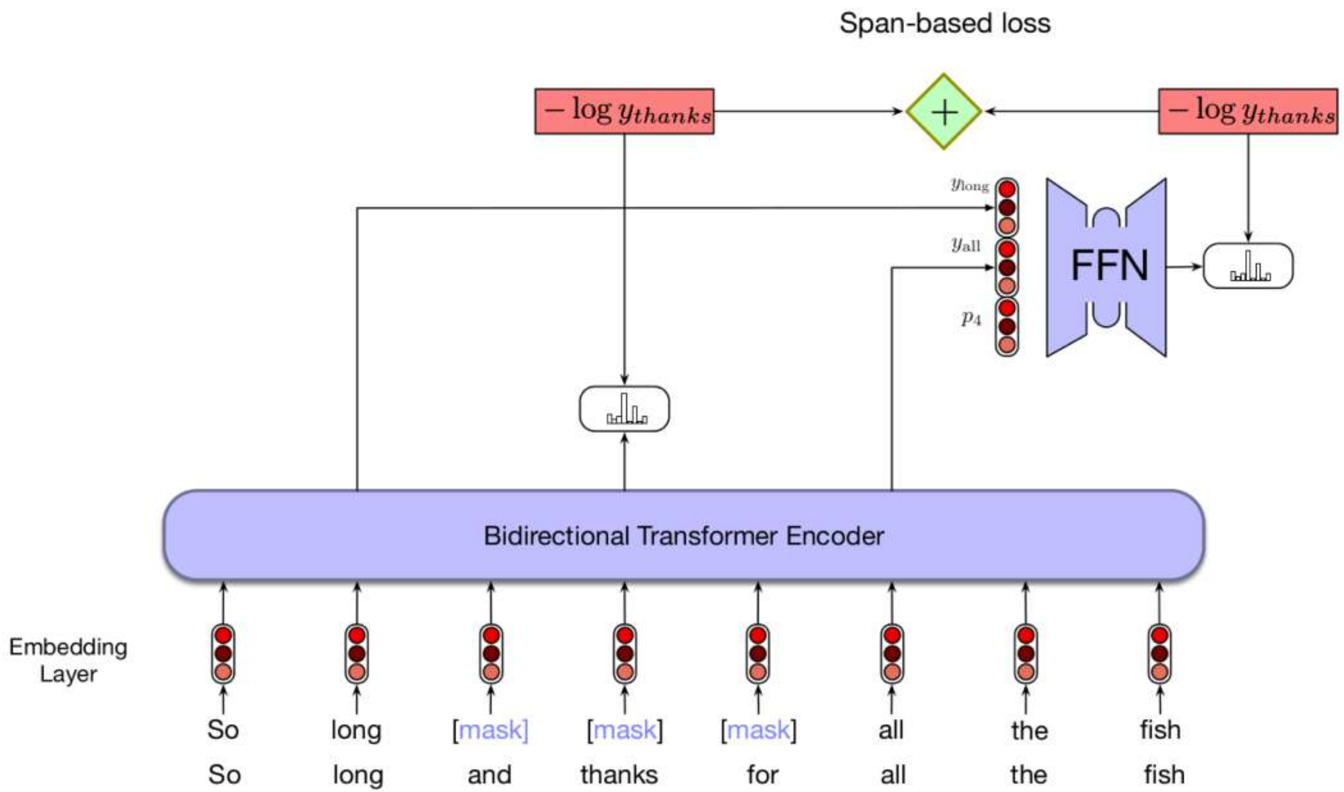
- SpanBERT (Joshi et al., 2020) originated span-masking learning technique.

Masking Spans

- The parameters for spans masking are the same as in BERT;
- Span Boundary Objective (SBO):

$$L(x) = L_{MLM}(x) + L_{SBO}(x)$$
$$L_{SBO}(x) = -logP(x|x_s, x_e, p_x)$$

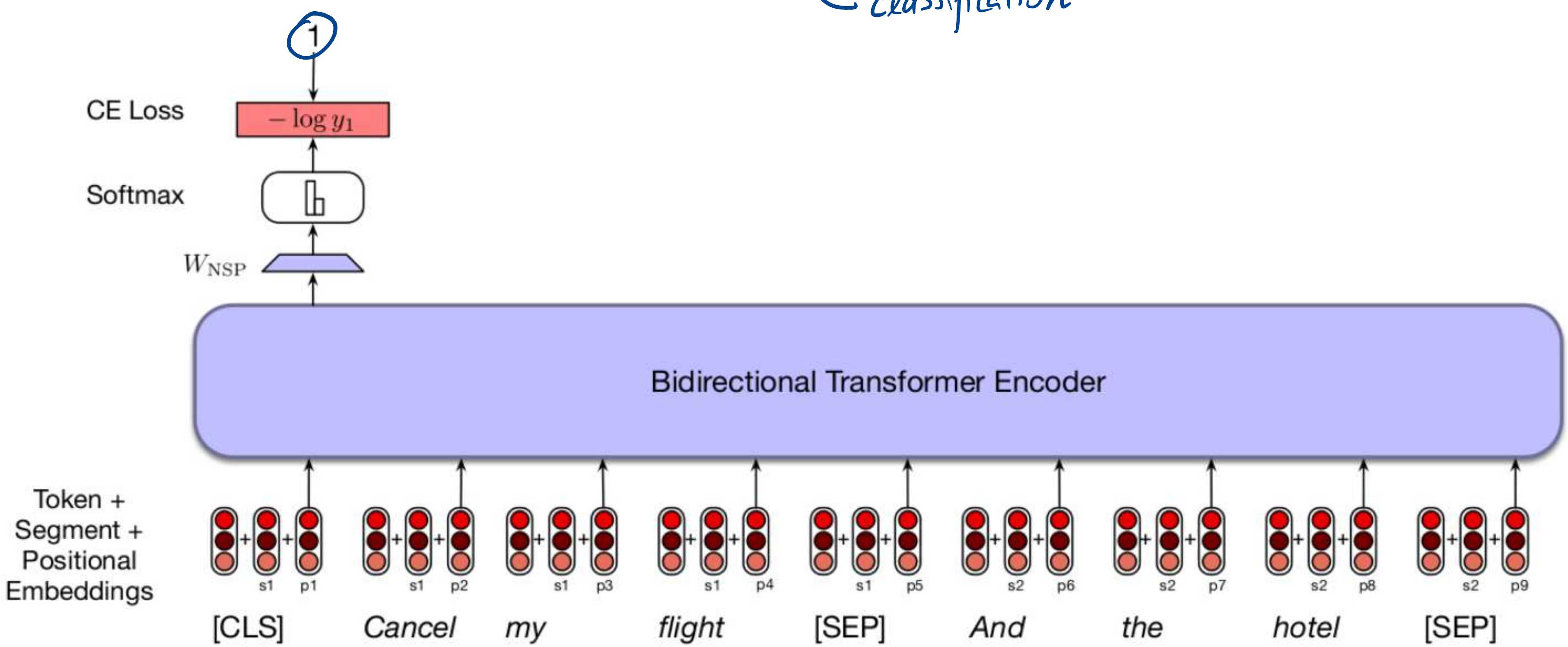
s denotes the word before the span and e denotes the word after the span, p_x: positional embedding (which word in the spam is currently predicted).



Next Sentence Prediction

- Second objective of bidirectional encoders training: Next Sentence Prediction (NSP).
- Next Sentence Prediction (NSP): given a pair of sentences, to predict whether a pair consists of a pair of adjacent sentences or a pair of unrelated sentences.

classification



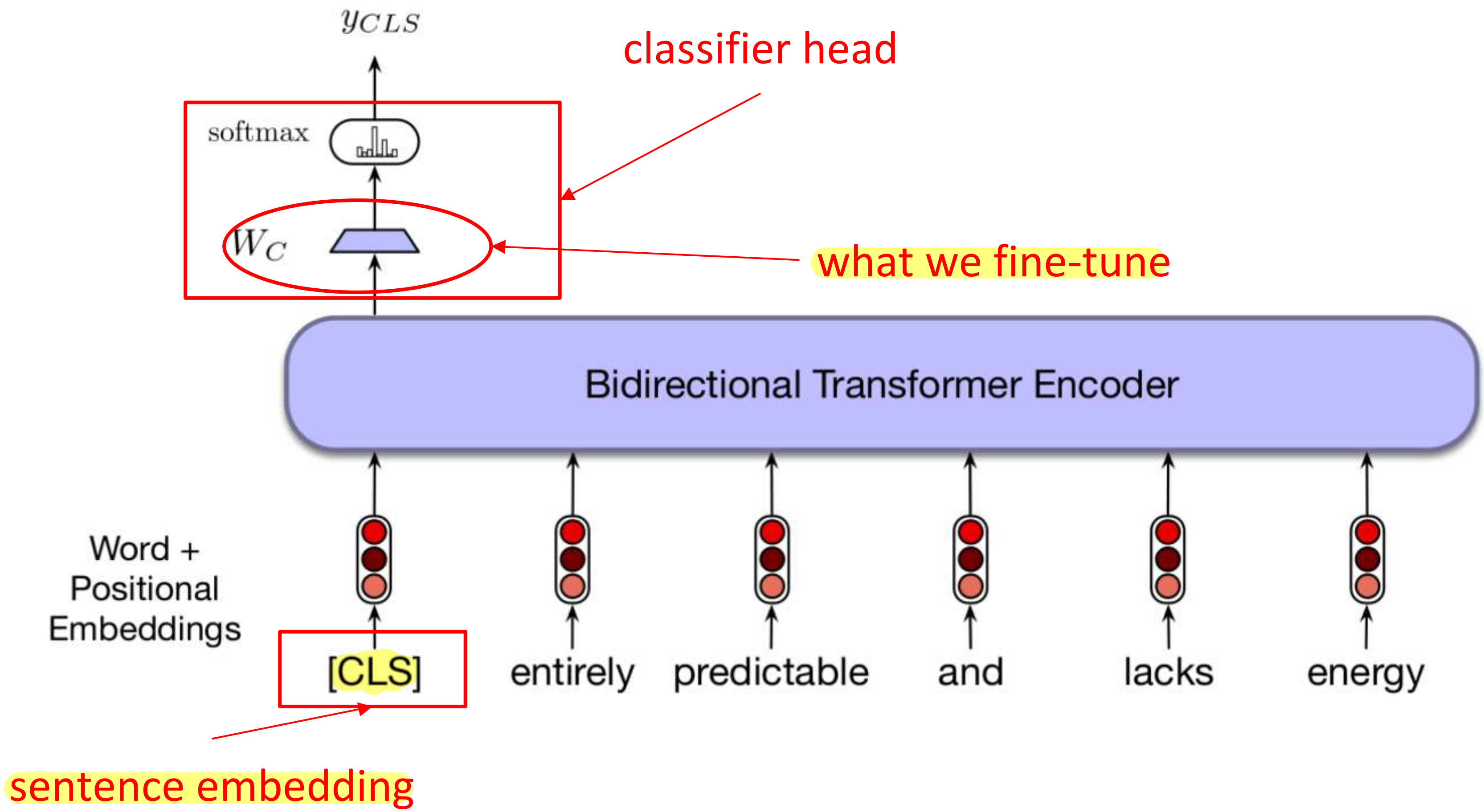
Next Sentence Prediction

- In BERT:
  - 50%: positive pairs;
  - 50%: the second sentence was randomly selected from elsewhere in the corpus;
- New tokens presented:
  - [CLS]: prepended to the input sentence pair;
  - [SEP]: placed between the sentences and after the final token of the second sentence;
- During training:
  - the output vector from the final layer ([CLS] token) represents the next sentence prediction;
  - Two-class classification: $y_i = \mathrm{softmax}(\mathbf{W}_{\mathbf{NSP}} h_i)$
    where $\mathbf{W}_{\mathbf{NSP}} \in \mathbb{R}^{2 \times d_h}$
  - NSP loss calculated as Cross-Entropy.

Transfer Learning through Fine-Tuning

- BERT, RoBERTa, ELMO – pre-trained on big corpora large language models (LLM).
- make practical use of these generalizations: interfaces from these models to downstream applications – via fine-tuning.
- fine-tuning process: train additional application-specific parameters of LLMs on labeled data for the application.
- possible downstream applications:
  - sequence classification;
  - sequence labeling;
  - sentence-pair inference;
  - span-based operations.

Sequence Classification



Pair-Wise Sequence Classification

- Natural Language Inference (NLI) or recognizing textual entailment: pair of sentences → classify the relationship between their meanings.
- MultiGenre Natural Language Inference (MultiNLI) dataset: given a premise and a hypothesis : predict their relationship – *entails*, *contradicts* or *neutral*:

  - Neutral

    a: Jon walked back to the town to the smithy.
    b: Jon traveled back to his hometown.

  - Contradicts

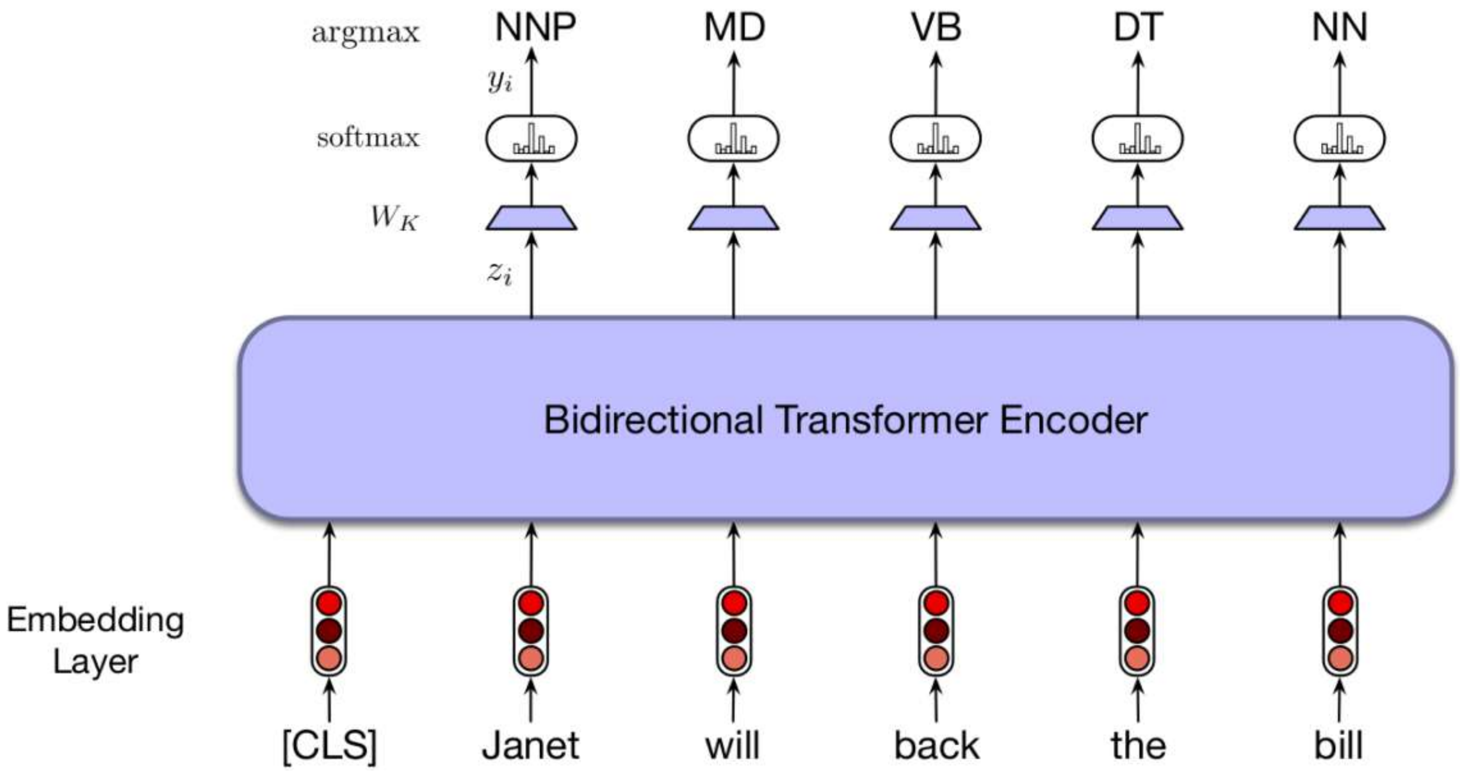
    a: Tourist Information offices can be very helpful.
    b: Tourist Information offices are never of any help.

  - Entails

    a: I'm confused.
    b: Not all of it is very clear to me.

- Fine-tuning:
  - pass the premise/hypothesis pairs through a bidirectional encoder separated with [SEP];
  - use [CLS] token as input to the classification head.

Sequence Labelling



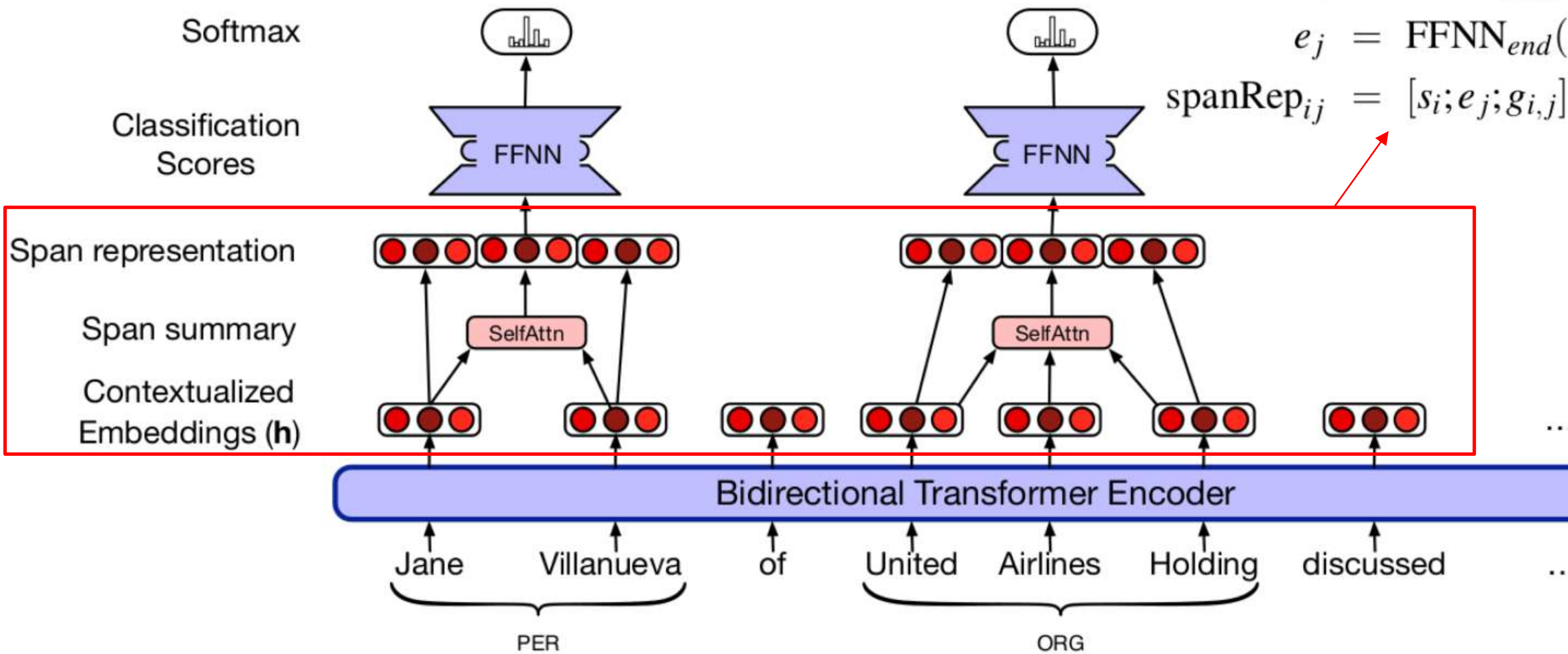
The ideal tagging: [$_{\text{LOC}}$ **Mt. Sanitas**] is in [$_{\text{LOC}}$ **Sunshine Canyon**]

but WordPiece splits:

`'Mt', '.', 'San', '##itas', 'is', 'in', 'Sunshine', 'Canyon' '.'`

Solution: assign BIO tag of the first sub-token to the whole token.

Fine-tuning for Span-Based Applications



$$s_i = \mathrm{FFNN}_{start}(h_i)$$
$$e_j = \mathrm{FFNN}_{end}(h_j)$$
$$\mathrm{spanRep}_{ij} = [s_i; e_j; g_{i,j}]$$

Main advantages over BIO-based per-word labelling:

- prone to a labeling mismatch problem;
- naturally accommodate embedded named entities (*United Airlines* and *United Airlines Holding*)

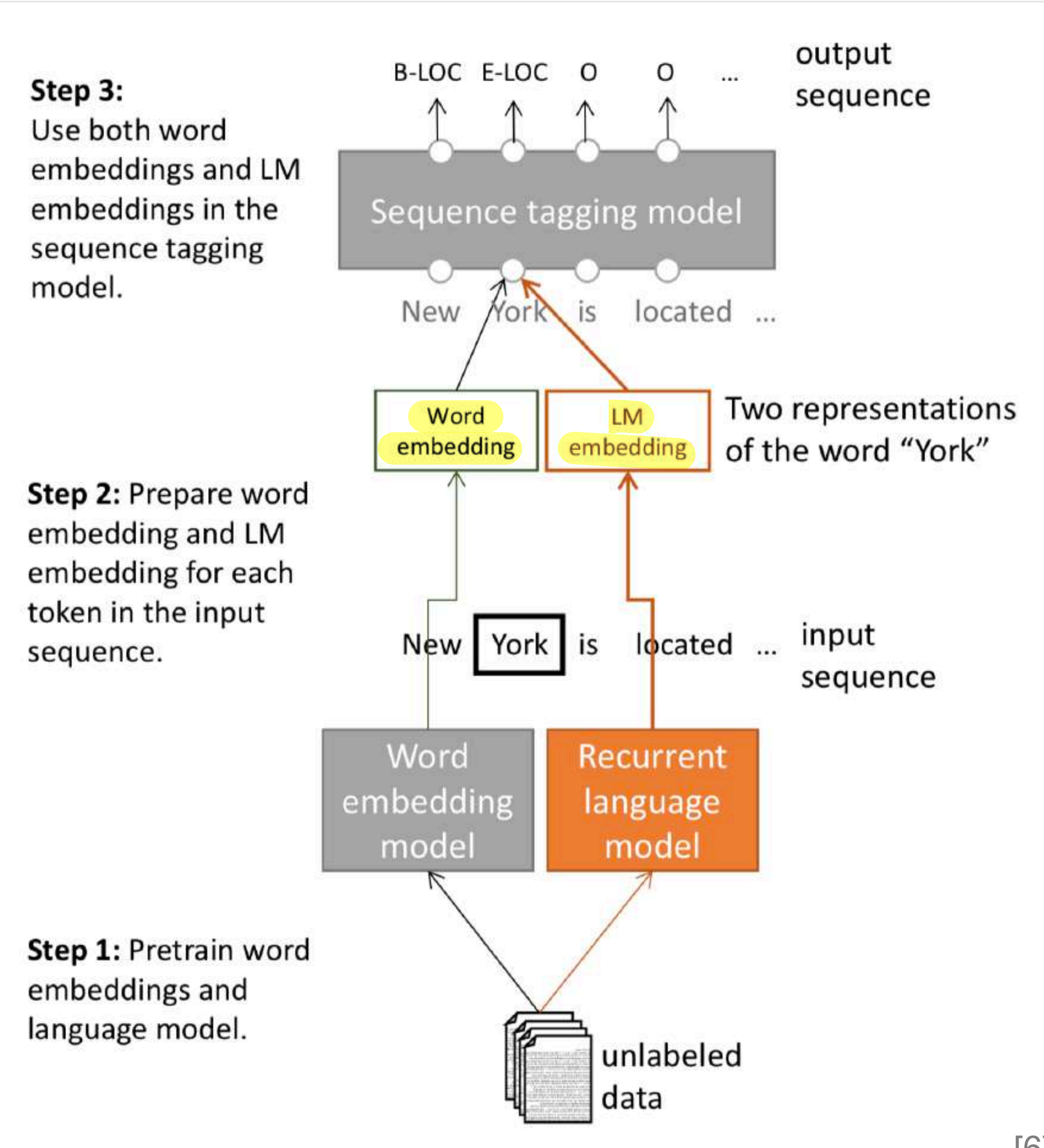
Potential Harms from Language Models

- Toxic language generation: non-toxic prompts can nonetheless lead large language models to output hate speech and abuse
- Language Models can be biased
- Generation of misinformation, phishing, radicalization, and other socially harmful activities
- Privacy issues: leak of information from training data

History and Future of Transformer-Based Architectures

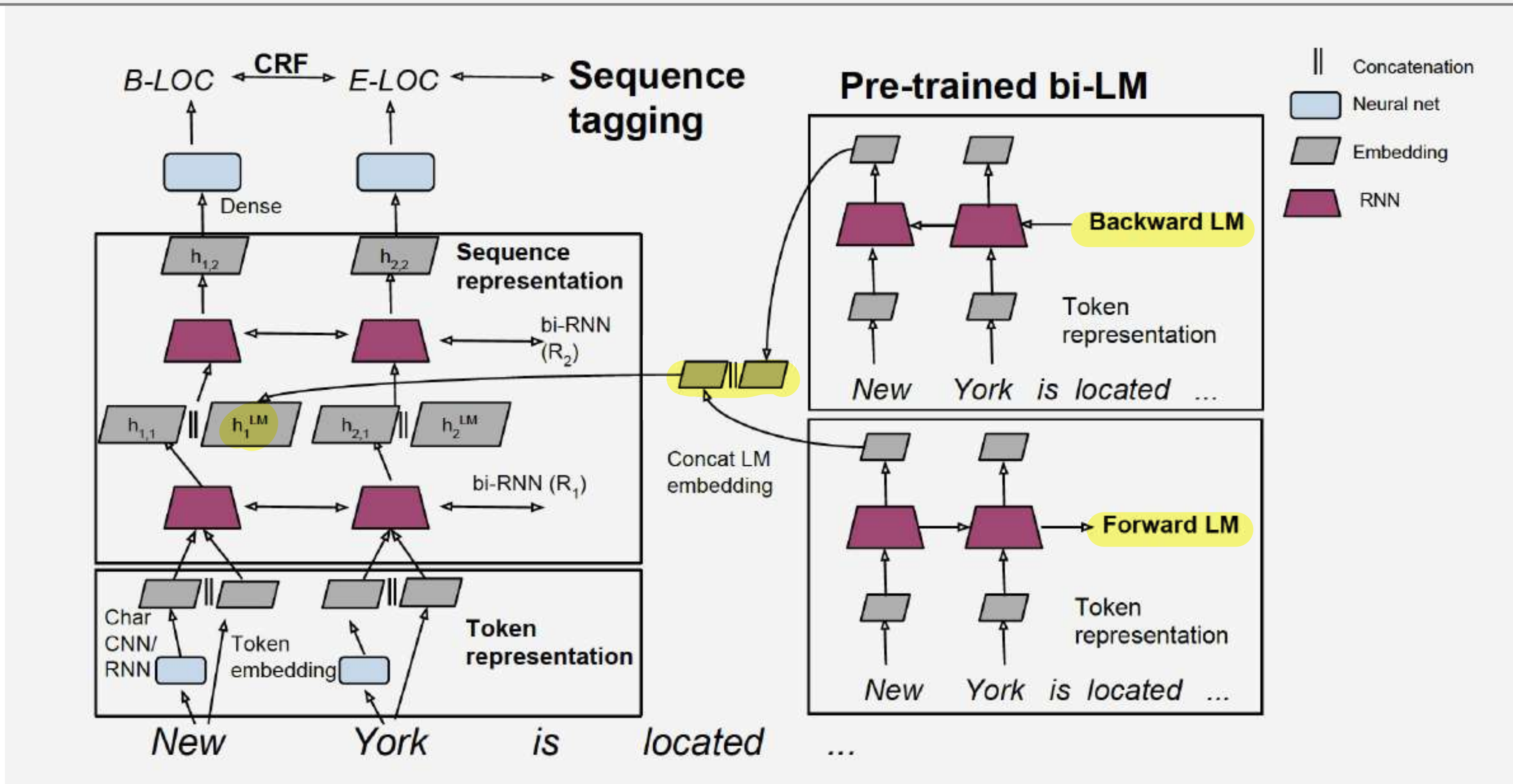
TagLM [6] : Contextual Embeddings for Sequence Tagging (NER)

- *initial solution idea for contextual embeddings:* use RNN style language modelling: **hidden states as context-aware embeddings** for tokens:



[6]

TagLM [5] : Contextual Embeddings for Sequence Tagging (NER)



[5]

Sequence tagging network: 2 bi-LSTM layers

$$\overrightarrow{\mathbf{h}}_{k,1} = \overrightarrow{R}_1(\mathbf{x}_k, \overrightarrow{\mathbf{h}}_{k-1,1}; \theta_{\overrightarrow{R}_1})$$
$$\overleftarrow{\mathbf{h}}_{k,1} = \overleftarrow{R}_1(\mathbf{x}_k, \overleftarrow{\mathbf{h}}_{k+1,1}; \theta_{\overleftarrow{R}_1})$$
$$\mathbf{h}_{k,1} = [\overrightarrow{\mathbf{h}}_{k,1}; \overleftarrow{\mathbf{h}}_{k,1}; \mathbf{h}_k^{LM}]$$

LM: LSTM forward RNN and LSTM backward RNN

$$\mathbf{h}_k^{LM} = [\overrightarrow{\mathbf{h}}_k^{LM}; \overleftarrow{\mathbf{h}}_k^{LM}]$$

character-based embedding (e.g. via a CNN θ_c) and word-based embedding (simple lookup layer θ_w, starting from some pre-trained emb. and then fine tuning θ_w) for tokens t_k

$$\mathbf{c}_k = C(t_k; \theta_c)$$
$$\mathbf{w}_k = E(t_k; \theta_w)$$
$$\mathbf{x}_k = [\mathbf{c}_k; \mathbf{w}_k]$$

TagLM [5] : Contextual Embeddings for Sequence Tagging (NER)

			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

[5]: Peters et al (2017) (Tag-LM paper, “pre-ELMo”)

TagLM [5] : Contextual Embeddings for Sequence Tagging (NER)

- LM (pre-)trained on 800 million word corpus
- from-scratch--end-to-end-trained model (downstream task: NER) without pre-training the contextual LM components → not a real benefit over simple bi-LSTM NER
- bi-directional contextual LM better than unidirectional (+0.2 F1)
- "larger, better" LM with perplexity ≈30 better than "smaller, coarser" LM (perplexity ≈ 48) (+0.3 F1)
- using just the pretrained contextual LM for the downstream task (NER) alone (i.e. without bi-LSTM NER „layer"): not very good (F1=88.17)

ELMo [3] ("Embeddings from Language Models")

- use same idea (contextual embeddings via NLM) but **go deeper**: more layers, use **output from all layers** (all "levels of abstraction")

representation of token t_k

initial (layer 0) representation (x_k^{LM} := „ $\mathrm{h}_{k,o}^{LM}$") of token t_k via character-based CNN

hidden state output of k-th „bi"-LSTM unit. $(\overrightarrow{\mathrm{h}}_{k,j}^{LM}; \overleftarrow{h}_{k,j}^{LM})$:= „ $\mathrm{h}_{k,j}^{LM}$ "

L layers

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

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

downstream-task-specific parameters;
s_j^{task}: softmax normalized mixture weights;
also: possibly layer-normalize ($\mu = 0; \sigma = 1$) the $\{\mathrm{h}_{k,j}^{LM}\}$ for each j

[3]: Peters et al (2018) (original ELMo paper)

ELMo [3]

- also: use large context: 4096 LSTM units wide layers
- initial model:
  - 2 layers, standard cross entropy loss
  - residual connections (concatenate input to layer output) for first layer
  - layer 0 character CNN: 2048 filters and two highway layers (gated variants of FF-layers to prevent vanishing gradients)

question answering: for 100k questions detect span of answer in given Wikipedia paragraph

textual entailment: 550k (hypothesis, premise) pairs → true, false?

semantic role labeling (predicate argument structure of a sentence)

co-reference resolution: cluster and relate mentions of entities

named entity resolution

5-level sentiment analysis of movie reviews

TASK	PREVIOUS SOTA		OUR BASELINE	ELMO + BASELINE
SQuAD	Liu et al. (2017)	84.4	81.1	85.8
SNLI	Chen et al. (2017)	88.6	88.0	88.7 ± 0.17
SRL	He et al. (2017)	81.7	81.4	84.6
Coref	Lee et al. (2017)	67.2	67.2	70.4
NER	Peters et al. (2017)	91.93 ± 0.19	90.15	92.22 ± 0.10
SST-5	McCann et al. (2017)	53.7	51.4	54.7 ± 0.5

ELMo [3]

Task	Baseline	Last Only	All layers λ=1	λ=0.001
SQuAD	80.8	84.7	85.0	**85.2**
SNLI	88.1	89.1	89.3	**89.5**
SRL	81.6	84.1	84.6	**84.8**

Table 2: Development set performance for SQuAD, SNLI and SRL comparing using all layers of the biLM (with different choices of regularization strength λ) to just the top layer.

adding $\lambda\|\mathbf{w}\|_2^2$ to the loss

Task	Input Only	Input & Output	Output Only
SQuAD	85.1	**85.6**	84.8
SNLI	88.9	**89.5**	88.7
SRL	**84.7**	84.3	80.9

Table 3: Development set performance for SQuAD, SNLI and SRL when including ELMo at different locations in the supervised model.

downstream

	Source	Nearest Neighbors
GloVe	play	playing, game, games, played, players, plays, player, Play, football, multiplayer
biLM	Chico Ruiz made a spectacular play on Alusik 's grounder {...}	Kieffer , the only junior in the group , was commended for his ability to hit in the clutch , as well as his all-round excellent play .
	Olivia De Havilland signed to do a Broadway play for Garson {...}	{...} they were actors who had been handed fat roles in a successful play , and had talent enough to fill the roles competently , with nice understatement .

Table 4: Nearest neighbors to "play" using GloVe and the context embeddings from a biLM.

ELMo [3]

- contributions from multiple layers to contextual embeddings:
  - lower layer contributions: better for lower-level syntax, etc. (POS tagging, syntactic dependencies, NER etc.)
  - higher layer contributions: better for higher-level semantics (sentiment, semantic role labeling, question answering, SNLI, etc.)

The Journey Continues (original slide from [2])

Let's scale it up!

ULMfit	GPT	BERT	GPT-2
Jan 2018	June 2018	Oct 2018	Feb 2019
Training:	Training	Training	Training
1 GPU day	240 GPU days	256 TPU days	~2048 TPU v3 days according to a reddit thread
		~320–560 GPU days	
fast.ai	OpenAI	Google AI	OpenAI

Transformer-Based

GPT-2 language model (cherry-picked) output

SYSTEM PROMPT (HUMAN-WRITTEN)	*In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.*
MODEL COMPLETION (MACHINE-WRITTEN, 10 TRIES)	The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science. Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved. Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow. Pérez and the others then ventured further into the valley. …

The Journey Continues (original slide from [2])

METRO NEWS... BUT NOT AS YOU KNOW IT

NEWS SPORT ENTERTAINMENT SOAPS MORE TRENDING

UK WORLD WEIRD TECH

Elon Musk's OpenAI builds artificial intelligence so powerful it must be kept locked up for the good of humanity

Jasper Hamill Friday 15 Feb 2019 10:06 am

272 SHARES

Elon Musk's scientists have announced the creation of a terrifying artificial intelligence that's so smart they refused to release it to the public.

OpenAI's GPT-2 is designed to write just like a human and is an impressive leap forward capable of penning chillingly convincing text.

It was 'trained' by analysing eight million web pages and is capable of writing large tracts based upon a 'prompt' written by a real person.

But the machine mind will not be released in its fully-fledged form because of the risk of it being used for 'malicious purposes' such as generating fake news, impersonating people online, automating the production of spam or churning out 'abusive or faked content to post on social media'.

OpenAI wrote: 'Due to our concerns about malicious applications of the technology, we are not releasing the trained model.

Elon Musk @elonmusk Follow

Replying to @georgezachary

To clarify, I've not been involved closely with OpenAI for over a year & don't have mgmt or board oversight

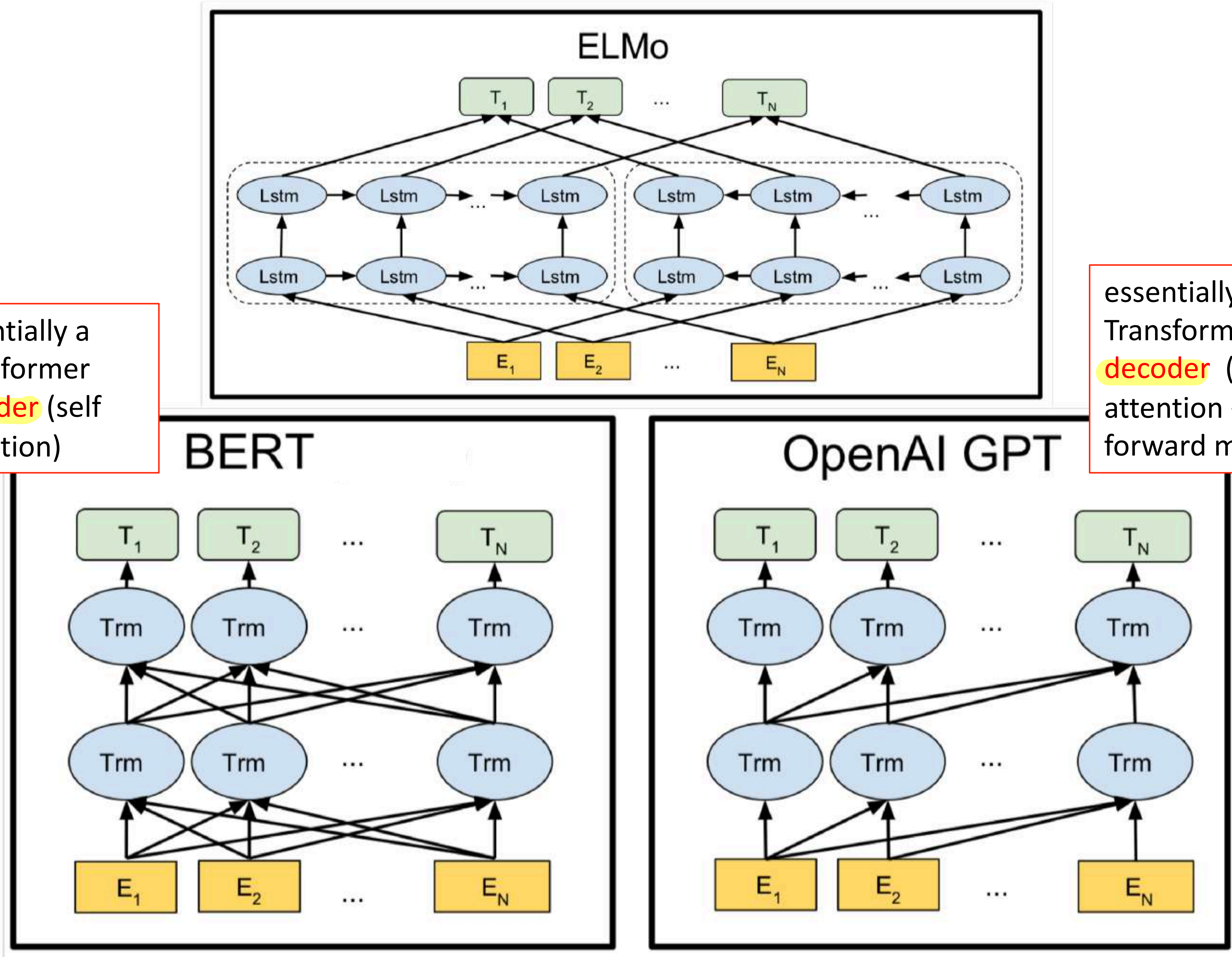
8:19 PM - 16 Feb 2019

500 Retweets 14,573 Likes

229 500 15K

BERT [4]

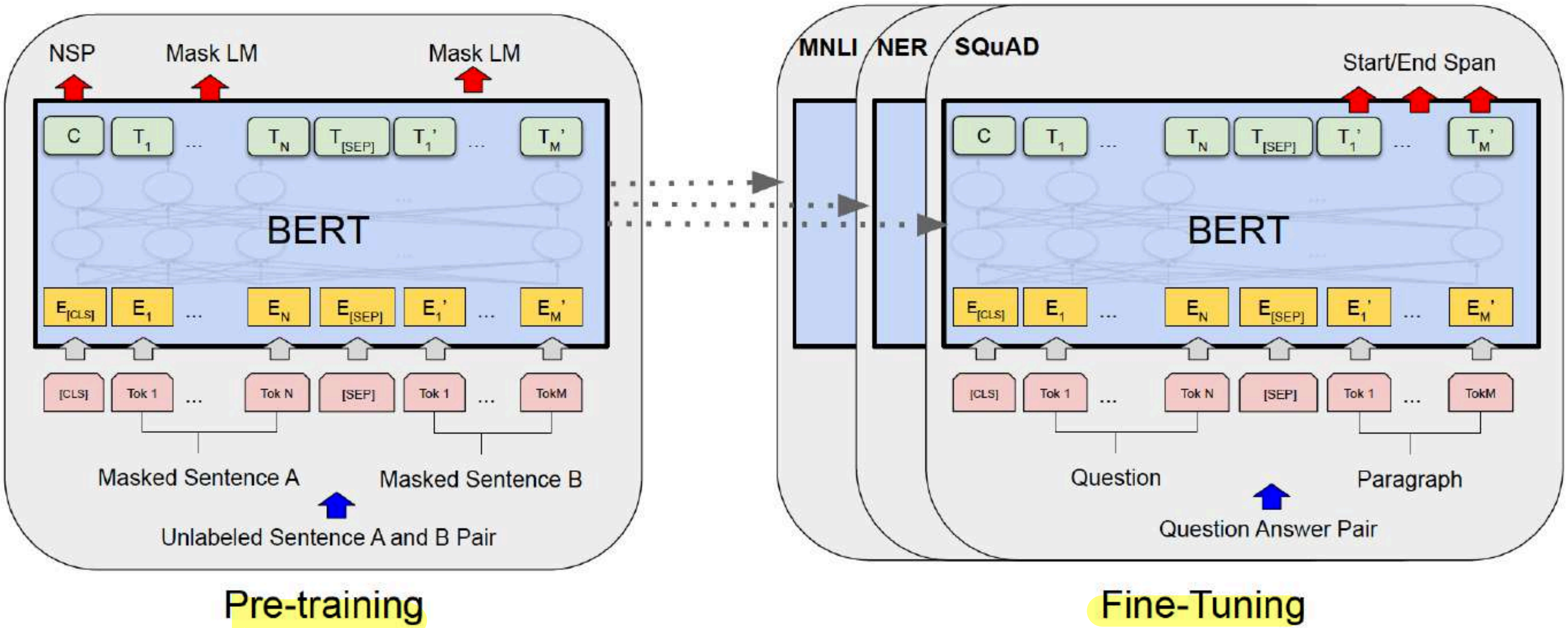
essentially a Transformer encoder (self attention)



essentially a Transformer decoder (self attention + forward masking)

BERT [4]

- As we have seen: BERT: pre-training on task 1 (mask LM) and 2 (NSP), then fine-tuning *generically* for any NLP task



- BERT-Base: 12-layers, 768-dim-hidden state vectors, 12 attention heads
- BERT-Large: 24-layer, 1024-dim-hidden state vectors, 16 attention heads

Bibliography

(1) Dan Jurafsky and James Martin: Speech and Language Processing (3rd ed. draft, version Jan, 2023); Online: https://web.stanford.edu/~jurafsky/slp3/ (URL, Oct 2023) (this slideset is especially based on chapter 11)

(2) Christopher Manning et al: "CS224n: Natural Language Processing with Deep Learning", Lecture Materials winter 2020 (slides and links to background reading) http://web.stanford.edu/class/cs224n/ (URL, Jan 2020), 2020

(3) Peters, Neumann, Iyyer, Gardner, Clark, Lee, Zettlemoyer: Deep contextualized word representations; arXiv:1802.05365v2 [cs.CL] 22 Mar 2018 (the original ELMo Paper)

(4) Devlin, Chang, Lee, Toutanova BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding; arXiv:1810.04805v2 [cs.CL] 24 May 2019 (the original BERT paper)

(5) Jay Allamar: The Illustrated BERT, ELMo, and co., Blog Post; http://jalammar.github.io/illustrated-bert/ (URL, Jan 2020)

(6) Peters, Ammar, Bhagavatula, Power: Semi-supervised sequence tagging with bidirectional language models; arXiv:1705.00108v1 [cs.CL] 29 Apr 2017 (pre-ELMo attempt)

Recommendations for Studying

- minimal approach:

  work with the slides and understand their contents! Think beyond instead of merely memorizing the contents

- standard approach:

  minimal approach + read the corresponding pages in Jurafsky [1]

- interested students

  == standard approach + read [5] plus one more self-selected Blog-post on latest developments in contextual embeddings or GPT-x