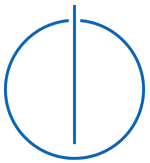


Faculty for Informatics

Technical
University
of Munich

Natural Language Processing
IN2361

PD Dr. Georg Groh

Social Computing
Research Group

Deep NLP
Part H: BERT and Contextual Embeddings

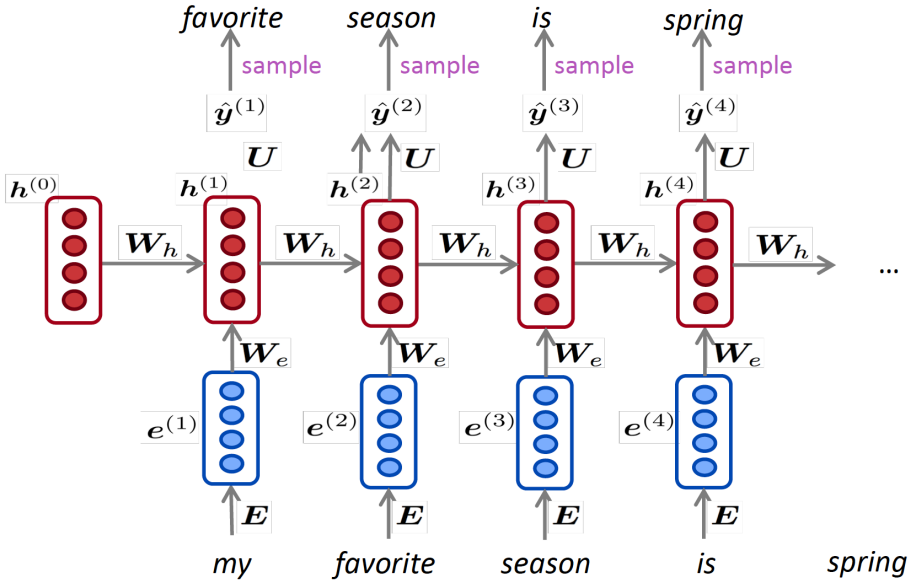
- content is based on [1] (further sources: see bibliography)
- certain elements (e.g. figures, equations or tables) were taken over or taken over in a modified form from [1]
- citations of [1] are omitted for legibility; non-[1]-sources are cited
- citations of original sources cited in [1] are omitted for legibility
- errors on these slides are fully in the responsibility of Georg Groh
- BIG thanks to Christopher Manning, Richard Socher and his colleagues at Stanford for publishing materials [1](and earlier years) of a great Deep NLP lecture

Word Vectors Up to Now → 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 ("unsupervised")

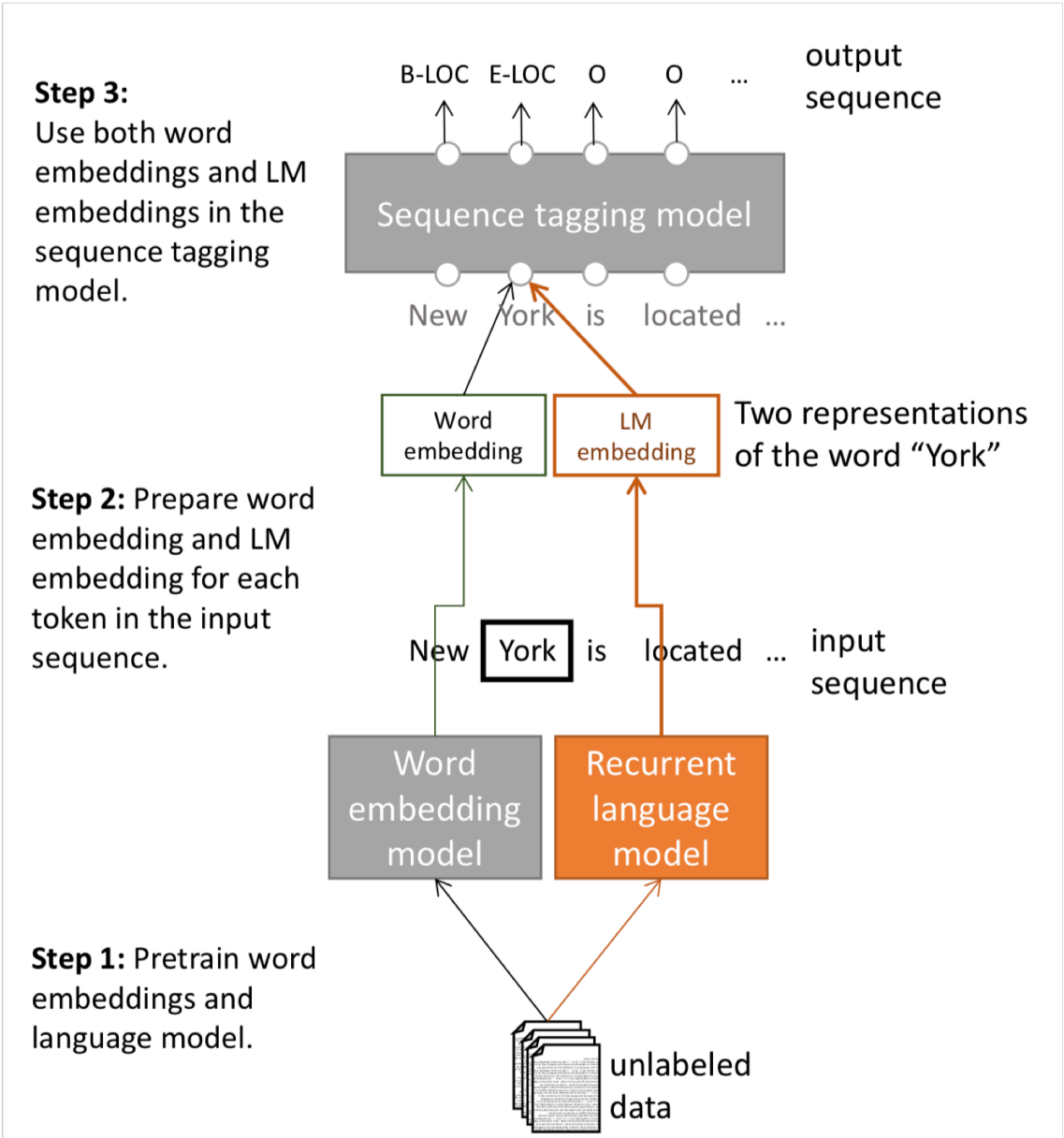
- *however*: word tokens can have different aspects (e.g. semantic (word senses), or syntactic behavior) in different contexts → make **embedding depend on context**!

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



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

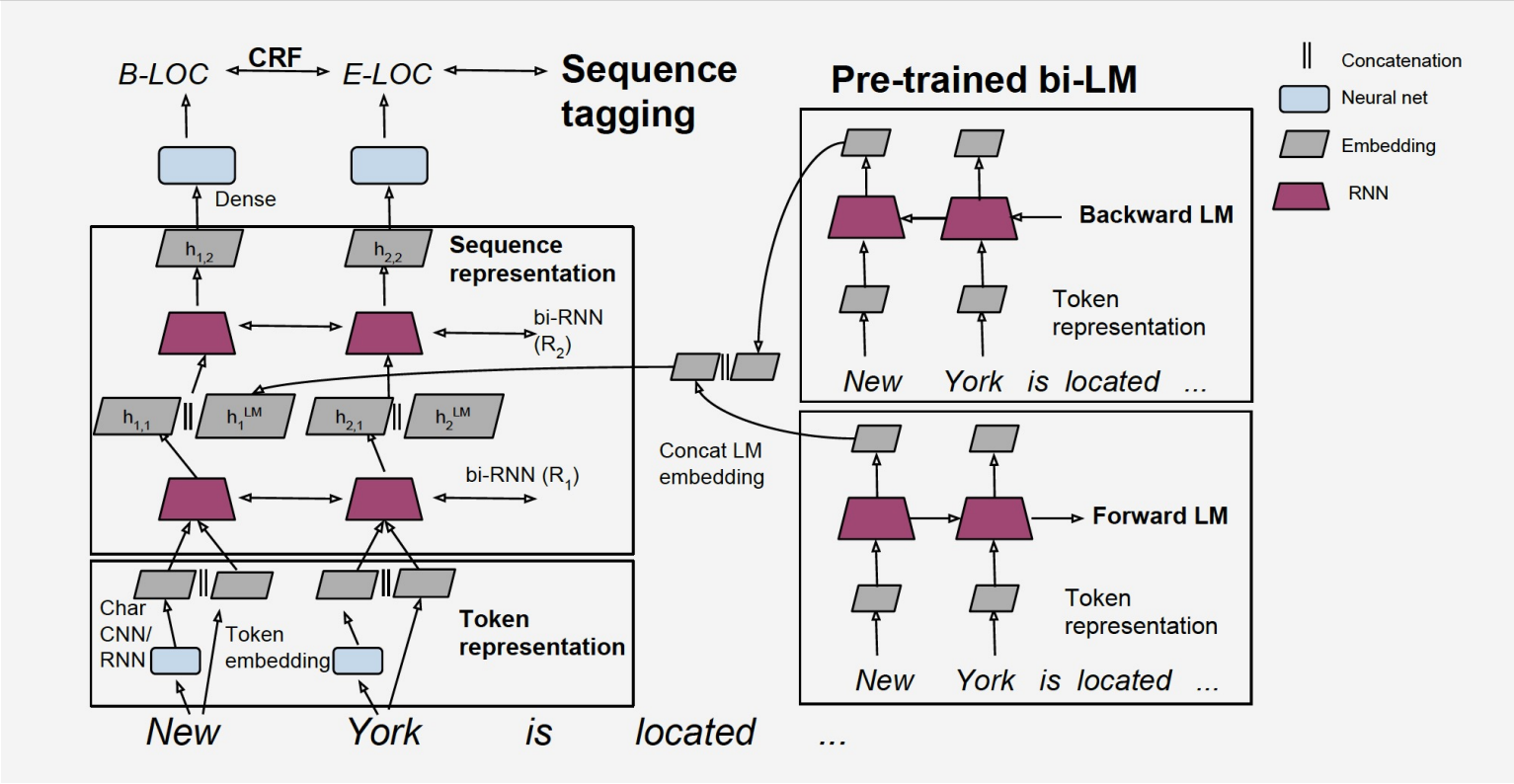
TagLM [5] : Contextual Embeddings for Sequence Tagging (NER)



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

[5]

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

17

[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)

CoVe (2017)

- other idea: CoVe: instead of ("unsupervised") language modelling, use straightforward seq2seq+attention **NMT** encoder's hidden states as contextual embeddings
- motivation: NMT encoders are designed to capture "meaning"
- better then GloVe for various downstream tasks but worse than ELMo and BERT
- future: potentially having more bi-lingual data available: maybe promising

ELMo [2] ("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 ($\mathrm{x}_k^{LM} := \text{„ } \mathrm{h}_{k,o}^{LM}\text{“}$) of token t_k via character-based CNN

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

L layers

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

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

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

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

ELMo [2]

- 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 [2]

Task	Baseline	Last Only	All layers λ=1	All layers 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

ELMo [2]

Task	Baseline	Last Only	All layers λ=1	All layers 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.

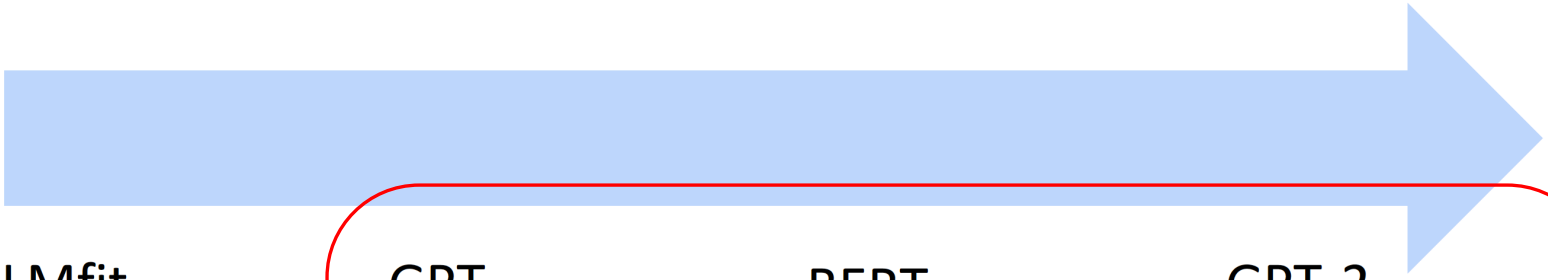
ELMo, UMLfit

- **ELMo**(2017) : 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.)

- **UMLfit** (2018): detailed care taken in terms of fine tuning multi-layer contextual embeddings (using text-classification as downstream task)

The Journey Continues (original slide from [1])

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

Transformer-Based

fast.ai

OpenAI

Google AI

OpenAI

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 [1])

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

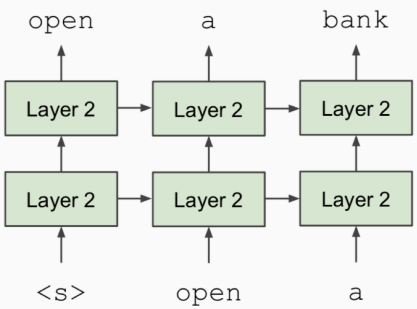
229 500 15K

BERT [2]

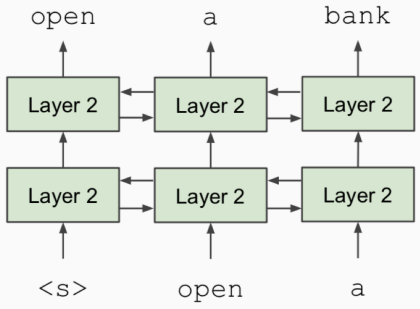
- "BERT": Bidirectional Encoder Representations from Transformers.

- observation: standard LM are always unidirectional (forward or backward) (predict $p(t_k \mid t_1, t_2, \ldots, t_{k-1})$ **or** $p(t_k \mid t_{k+1}, t_{k+2}, \ldots, t_N)$

Unidirectional context
Build representation incrementally



Bidirectional context
Words can "see themselves"



- when downstream task is sentence level: truly bi-directional models are required.
  If we use end-to-end fine tuning of essentially uni-directional pre-trained contextual embedding upstream network: possible problems

[2] Devlin at Al (2019) (the original BERT paper)

BERT [2]

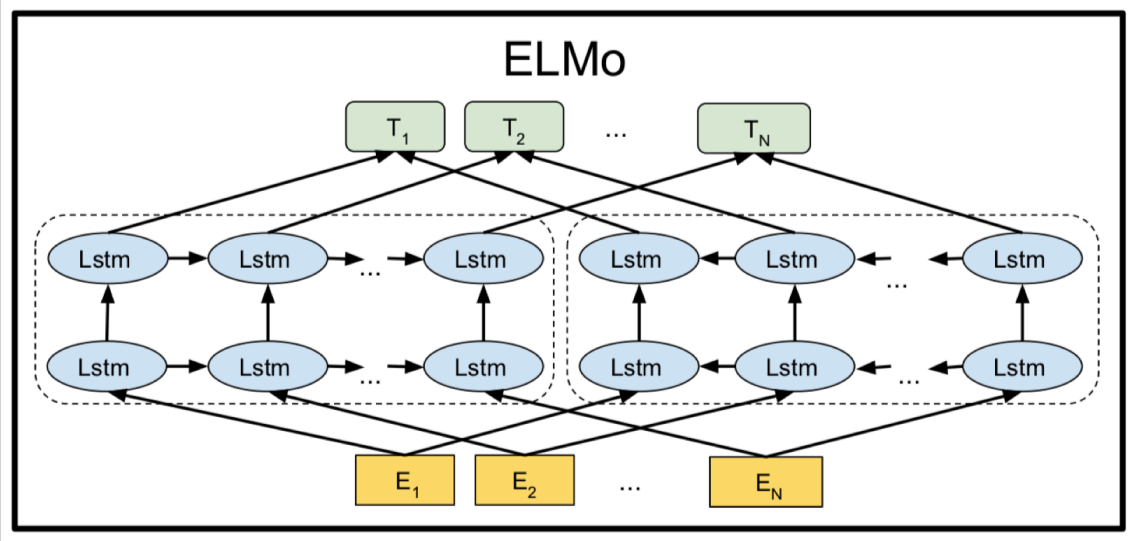
- solution: make LM “bidirectional” via masking and predicting k % of the words

store gallon
↑ ↑
the man went to the [MASK] to buy a [MASK] of milk

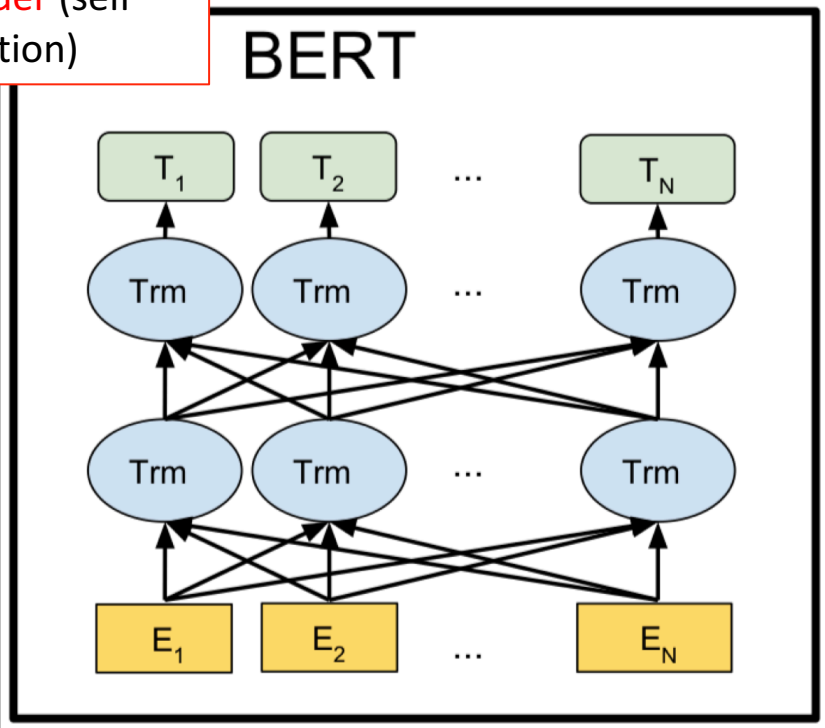
- k ≈ 15 %:
  too little masking: too expensive training (you should eventually mask and guess every word;
  too much masking: not enough context (quality suffers)

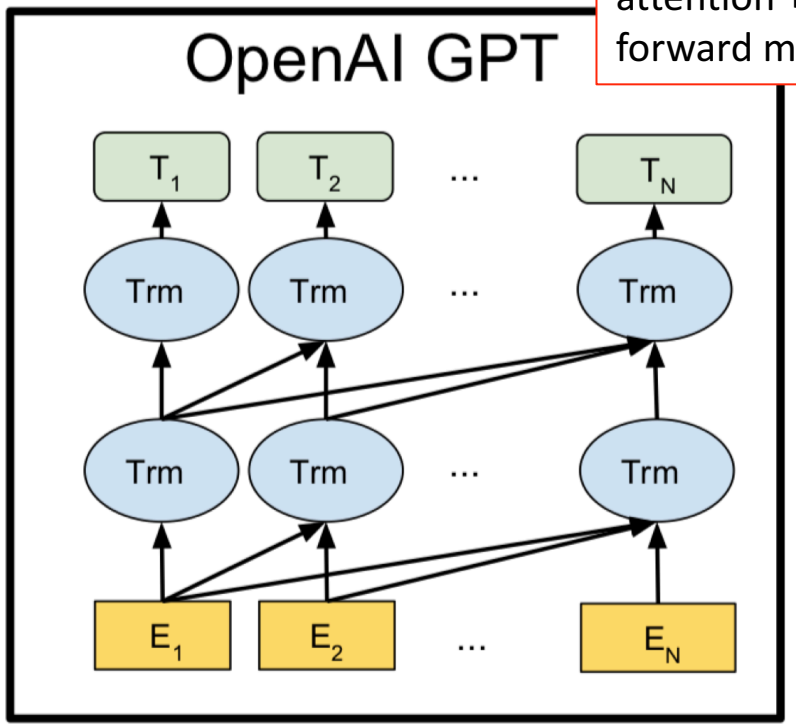
- masked „LM“ prediction: BERT pre-training **task 1**

BERT [2]



essentially a Transformer encoder (self attention)





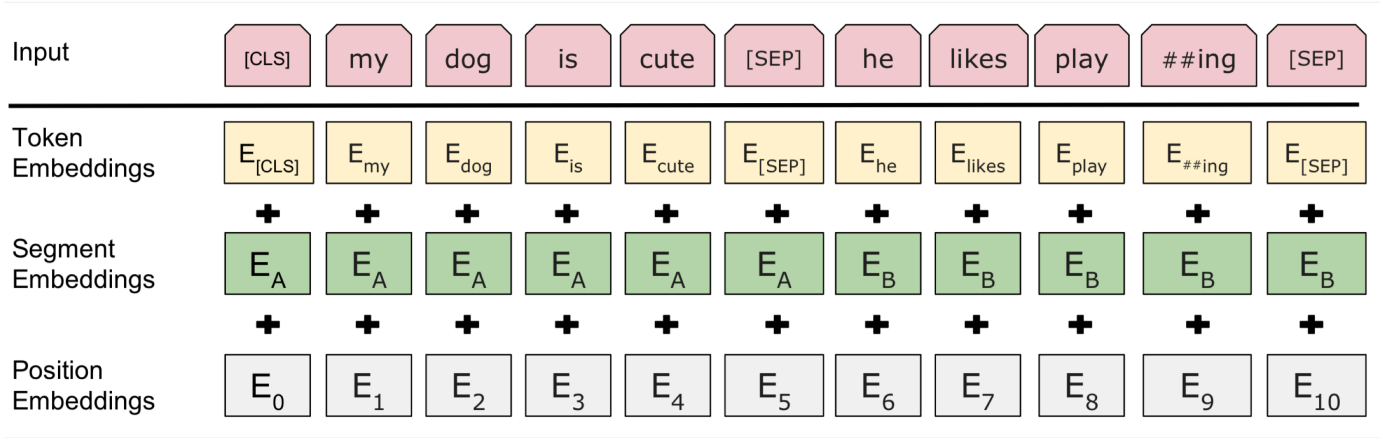
essentially a Transformer decoder (self attention + forward masking)

BERT [2]

- next sentence prediction (NSP): BERT pre-training **task 2**:
  predict whether sentence B is actual sentence that proceeds Sentence A, or a random sentence

Sentence A = The man went to the store.
Sentence B = He bought a gallon of milk.
Label = IsNextSentence

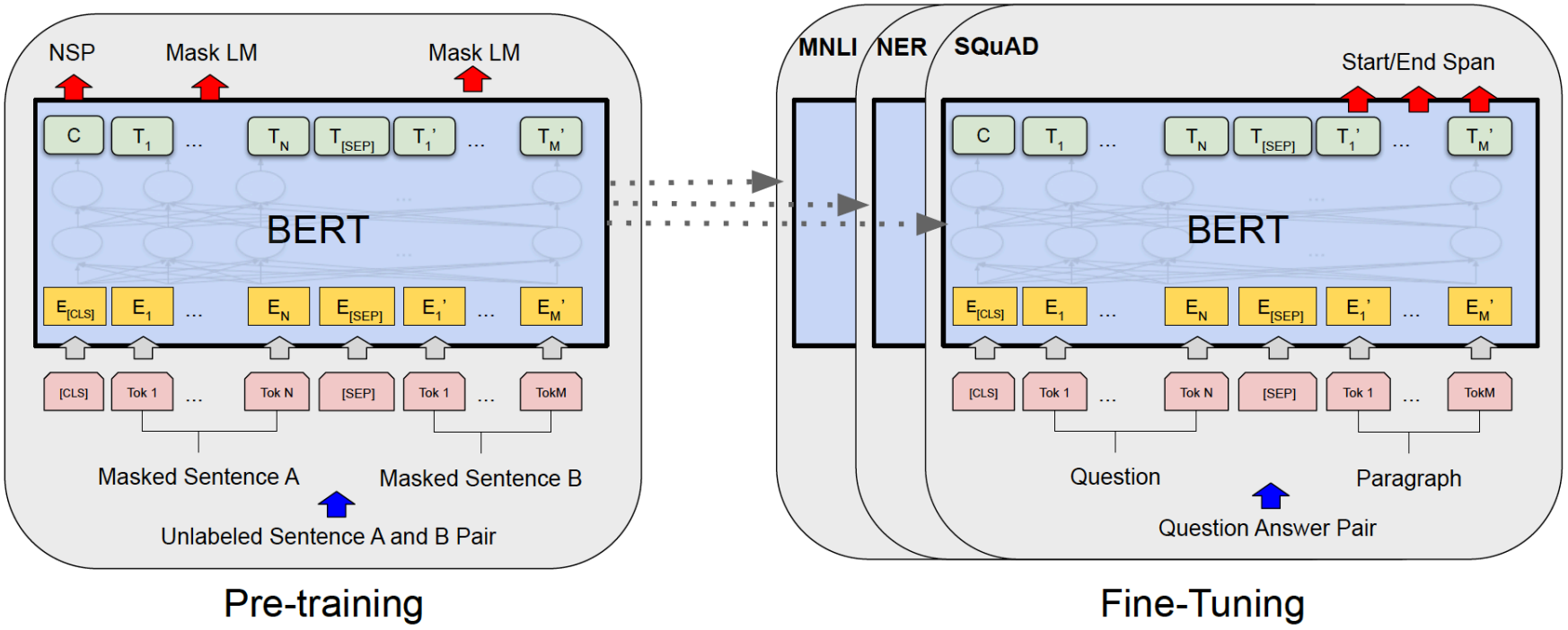
Sentence A = The man went to the store.
Sentence B = Penguins are flightless.
Label = NotNextSentence



Token embeddings are word pieces
Learned segmented embedding represents each sentence
Positional embedding is as for other Transformer architectures

BERT [2]

- 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

BERT [2] Performance on GLUE benchmark

- **MultiNLI**
- Premise: Hills and mountains are especially sanctified in Jainism.
  Hypothesis: Jainism hates nature.
  Label: Contradiction

- **CoLa**
- Sentence: The wagon rumbled down the road. Label: Acceptable
- Sentence: The car honked down the road. Label: Unacceptable

System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	**Average** -
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

BERT [2] Performance on CoNLL NER benchmark

Name	Description	Year	F1
Flair (Zalando)	Character-level language model	2018	93.09
BERT Large	Transformer bidi LM + fine tune	2018	92.8
CVT Clark	Cross-view training + multitask learn	2018	92.61
BERT Base	Transformer bidi LM + fine tune	2018	92.4
ELMo	ELMo in BiLSTM	2018	92.22
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	MEMM softmax markov model	2003	86.07

65

BERT [2] Performance on SQuAD 1.1

Rank	Model	EM	F1
	Human Performance *Stanford University* (Rajpurkar et al. '16)	82.304	91.221
1 Oct 05, 2018	BERT (ensemble) *Google AI Language* https://arxiv.org/abs/1810.04805	**87.433**	**93.160**
2 Oct 05, 2018	BERT (single model) *Google AI Language* https://arxiv.org/abs/1810.04805	85.083	91.835
2 Sep 26, 2018	nlnet (ensemble) *Microsoft Research Asia*	85.954	91.677
5 Sep 09, 2018	nlnet (single model) *Microsoft Research Asia*	83.468	90.133
3 Jul 11, 2018	QANet (ensemble) *Google Brain & CMU*	84.454	90.490

Bibliography

(1) 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

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

(3) 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)

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

(5) 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 [4].
  really strongly recommended: also read [2] and [3]

- interested student's approach:

  == standard approach