Natural Language Processing with Deep Learning

Lecture 9 — Decoder-only models and GPT

Prof. Dr. Ivan Habernal

December 17, 2025

`www.trusthlt.org`
Trustworthy Human Language Technologies Group (TrustHLT)
Ruhr University Bochum & Research Center Trustworthy Data Science and Security

RUHR
UNIVERSITÄT
BOCHUM

RUB

Motivation

We introduced BERT, a powerful transformer model for learning contextualized token embeddings

BERT can be used for

TrustHLT — Prof. Dr. Ivan Habernal   RUHR UNIVERSITÄT BOCHUM   RUB
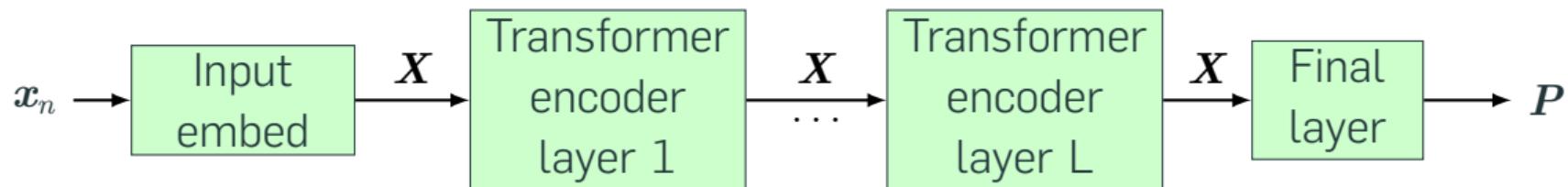
Motivation

We introduced BERT, a powerful transformer model for learning contextualized token embeddings

BERT can be used for

- text classification (one sequence, two concatenated sequences)
- sequence labeling (classify each token, e.g., NER, POS)

TrustHLT — Prof. Dr. Ivan Habernal   RUHR UNIVERSITÄT BOCHUM **RU**B

Transformer encoder (BERT)

For each input token, BERT produces contextualized word embeddings

TrustHLT — Prof. Dr. Ivan Habernal   RUB

Motivation

Although BERT is pre-trained with masked-language modeling, it is **not designed to generate text** by predicting the next token

Why?

Motivation
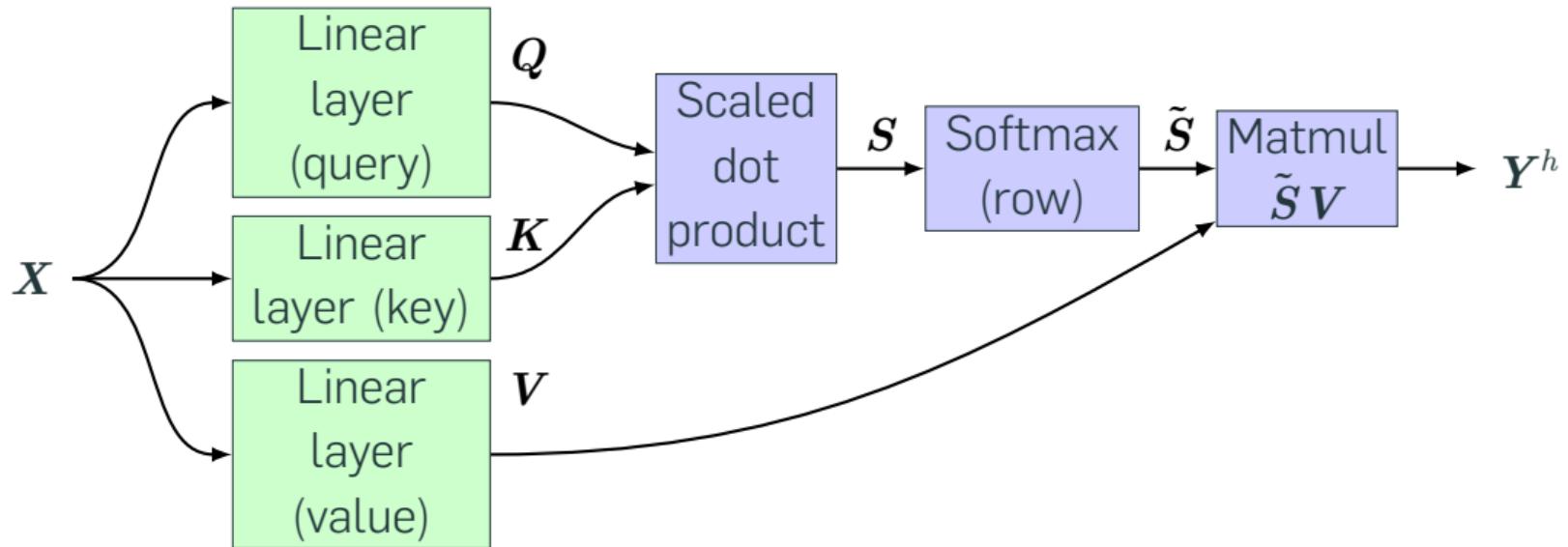
Although BERT is pre-trained with masked-language modeling, it is **not designed to generate text** by predicting the next token

Why?

We mask random tokens from the sequence and perform self-attention over past and future tokens

Can we use a transformer as a 'true' language model, aka. to conditionally generate text?

Recap: Single unmasked self-attention head (BERT)

TrustHLT — Prof. Dr. Ivan Habernal

RUHR UNIVERSITÄT BOCHUM RUB

Recap: Bidirectional / unmasked self-attention

Input: $\boldsymbol{X} \in \mathbb{R}^{\ell_x \times d_x}$, vector representations of the sequence of length ℓ_x

RUHR UNIVERSITÄT BOCHUM **RU**B

Recap: Bidirectional / unmasked self-attention

Input: $\boldsymbol{X} \in \mathbb{R}^{\ell_x \times d_x}$, vector representations of the sequence of length ℓ_x

Output: $\tilde{\boldsymbol{V}} \in \mathbb{R}^{\ell_x \times d_{\text{out}}}$, updated vector representations of tokens in $\boldsymbol{X}$

TrustHLT — Prof. Dr. Ivan Habernal

RUHR
UNIVERSITÄT
BOCHUM

RUB

Recap: Bidirectional / unmasked self-attention

Input: $\boldsymbol{X} \in \mathbb{R}^{\ell_x \times d_x}$, vector representations of the sequence of length ℓ_x

Output: $\tilde{\boldsymbol{V}} \in \mathbb{R}^{\ell_x \times d_{out}}$, updated vector representations of tokens in $\boldsymbol{X}$

Params $\boldsymbol{\mathcal{W}_{qkv}}$: $\boldsymbol{W_q}, \boldsymbol{W_k} \in \mathbb{R}^{d_x \times d_{attn}}$, $\boldsymbol{b_q}, \boldsymbol{b_k} \in \mathbb{R}^{d_{attn}}$, $\boldsymbol{W_v} \in \mathbb{R}^{d_x \times d_{out}}$, $\boldsymbol{b_v} \in \mathbb{R}^{d_{out}}$

1: **function** Attention($\boldsymbol{X}; \boldsymbol{\mathcal{W}_{qkv}}$)
2: $\quad \boldsymbol{Q} \leftarrow \boldsymbol{X}\boldsymbol{W_q} +_{\text{(rows)}} \boldsymbol{b_q}$ $\qquad\qquad\qquad\qquad\qquad \triangleright$ Query $\in \mathbb{R}^{\ell_x \times d_{attn}}$
3: $\quad \boldsymbol{K} \leftarrow \boldsymbol{X}\boldsymbol{W_k} +_{\text{(rows)}} \boldsymbol{b_k}$ $\qquad\qquad\qquad\qquad\qquad\; \triangleright$ Key $\in \mathbb{R}^{\ell_x \times d_{attn}}$
4: $\quad \boldsymbol{V} \leftarrow \boldsymbol{X}\boldsymbol{W_v} +_{\text{(rows)}} \boldsymbol{b_v}$ $\qquad\qquad\qquad\qquad\qquad \triangleright$ Value $\in \mathbb{R}^{\ell_x \times d_{out}}$
5: $\quad \boldsymbol{S} \leftarrow \frac{1}{\sqrt{d_{attn}}}(\boldsymbol{Q}\boldsymbol{K}^\top)$ $\qquad\qquad\qquad\qquad \triangleright$ Scaled score $\in \mathbb{R}^{\ell_x \times \ell_x}$
6: $\quad$ **return** $\tilde{\boldsymbol{V}} = \text{softmax}_{\text{row}}(\boldsymbol{S})\,\boldsymbol{V}$

TrustHLT — Prof. Dr. Ivan Habernal   RUHR UNIVERSITÄT BOCHUM RUB

Recap: Basic single-query attention

Input: $\boldsymbol{e} \in \mathbb{R}^{d_{\text{in}}}$, vector representation of the current token

Input: $\boldsymbol{e}_t \in \mathbb{R}^{d_{\text{in}}}$, vector representations of the context tokens $t \in [T]$

Recap: Basic single-query attention

Input: $\boldsymbol{e} \in \mathbb{R}^{d_{\text{in}}}$, vector representation of the current token

Input: $\boldsymbol{e}_t \in \mathbb{R}^{d_{\text{in}}}$, vector representations of the context tokens $t \in [T]$

Output: $\tilde{\boldsymbol{v}} \in \mathbb{R}^{d_{\text{out}}}$, vector representation of the token and context combined

Params: $\boldsymbol{W_q}, \boldsymbol{W_k} \in \mathbb{R}^{d_{\text{in}} \times d_{\text{attn}}}$, $\boldsymbol{b_q}, \boldsymbol{b_k} \in \mathbb{R}^{d_{\text{attn}}}$, $\boldsymbol{W_v} \in \mathbb{R}^{d_{\text{in}} \times d_{\text{out}}}$, $\boldsymbol{b_v} \in \mathbb{R}^{d_{\text{out}}}$

Recap: Basic single-query attention

Input: $\boldsymbol{e} \in \mathbb{R}^{d_{\text{in}}}$, vector representation of the current token

Input: $\boldsymbol{e}_t \in \mathbb{R}^{d_{\text{in}}}$, vector representations of the context tokens $t \in [T]$

Output: $\tilde{\boldsymbol{v}} \in \mathbb{R}^{d_{\text{out}}}$, vector representation of the token and context combined

Params: $\boldsymbol{W_q}, \boldsymbol{W_k} \in \mathbb{R}^{d_{\text{in}} \times d_{\text{attn}}}$, $\boldsymbol{b_q}, \boldsymbol{b_k} \in \mathbb{R}^{d_{\text{attn}}}$, $\boldsymbol{W_v} \in \mathbb{R}^{d_{\text{in}} \times d_{\text{out}}}$, $\boldsymbol{b_v} \in \mathbb{R}^{d_{\text{out}}}$

1: **function** Basic single-query attention

2: $\quad \boldsymbol{q} \leftarrow \boldsymbol{e}\,\boldsymbol{W_q} + \boldsymbol{b_q}$ $\qquad\qquad\qquad\qquad\qquad$ $\triangleright$ Query linear projection

TrustHLT — Prof. Dr. Ivan Habernal

RUHR UNIVERSITÄT BOCHUM **RUB**

Recap: Basic single-query attention

Input: $\boldsymbol{e} \in \mathbb{R}^{d_{\text{in}}}$, vector representation of the current token
Input: $\boldsymbol{e}_t \in \mathbb{R}^{d_{\text{in}}}$, vector representations of the context tokens $t \in [T]$
Output: $\tilde{\boldsymbol{v}} \in \mathbb{R}^{d_{\text{out}}}$, vector representation of the token and context combined
Params: $\boldsymbol{W_q}, \boldsymbol{W_k} \in \mathbb{R}^{d_{\text{in}} \times d_{\text{attn}}}$, $\boldsymbol{b_q}, \boldsymbol{b_k} \in \mathbb{R}^{d_{\text{attn}}}$, $\boldsymbol{W_v} \in \mathbb{R}^{d_{\text{in}} \times d_{\text{out}}}$, $\boldsymbol{b_v} \in \mathbb{R}^{d_{\text{out}}}$

1: **function** Basic single-query attention
2:      $\boldsymbol{q} \leftarrow \boldsymbol{e}\boldsymbol{W_q} + \boldsymbol{b_q}$ ▷ Query linear projection
3:      **for** $t \in [T]$ **do**
4:          $\boldsymbol{k}_t \leftarrow \boldsymbol{e_t}\boldsymbol{W_k} + \boldsymbol{b_k}$ ▷ Key linear projection

Recap: Basic single-query attention

Input: $\boldsymbol{e} \in \mathbb{R}^{d_{\text{in}}}$, vector representation of the current token

Input: $\boldsymbol{e}_t \in \mathbb{R}^{d_{\text{in}}}$, vector representations of the context tokens $t \in [T]$

Output: $\tilde{\boldsymbol{v}} \in \mathbb{R}^{d_{\text{out}}}$, vector representation of the token and context combined

Params: $\boldsymbol{W_q}, \boldsymbol{W_k} \in \mathbb{R}^{d_{\text{in}} \times d_{\text{attn}}}$, $\boldsymbol{b_q}, \boldsymbol{b_k} \in \mathbb{R}^{d_{\text{attn}}}$, $\boldsymbol{W_v} \in \mathbb{R}^{d_{\text{in}} \times d_{\text{out}}}$, $\boldsymbol{b_v} \in \mathbb{R}^{d_{\text{out}}}$

1: **function** Basic single-query attention

2:      $\boldsymbol{q} \leftarrow \boldsymbol{e}\boldsymbol{W_q} + \boldsymbol{b_q}$ ▷ Query linear projection

3:      **for** $t \in [T]$ **do**

4:          $\boldsymbol{k}_t \leftarrow \boldsymbol{e}_t \boldsymbol{W_k} + \boldsymbol{b_k}$ ▷ Key linear projection

5:          $\alpha_t = \frac{\exp(\boldsymbol{q} \cdot \boldsymbol{k}_t / \sqrt{d_{\text{attn}}})}{\sum_{u=1}^{T} \exp(\boldsymbol{q} \cdot \boldsymbol{k}_u / \sqrt{d_{\text{attn}}})}$ ▷ Softmax over scaled dot products, $\alpha_t \in (0, 1)$

Recap: Basic single-query attention

Input: $\boldsymbol{e} \in \mathbb{R}^{d_{\mathsf{in}}}$, vector representation of the current token

Input: $\boldsymbol{e}_t \in \mathbb{R}^{d_{\mathsf{in}}}$, vector representations of the context tokens $t \in [T]$

Output: $\tilde{\boldsymbol{v}} \in \mathbb{R}^{d_{\mathsf{out}}}$, vector representation of the token and context combined

Params: $\boldsymbol{W_q}, \boldsymbol{W_k} \in \mathbb{R}^{d_{\mathsf{in}} \times d_{\mathsf{attn}}}, \boldsymbol{b_q}, \boldsymbol{b_k} \in \mathbb{R}^{d_{\mathsf{attn}}}, \boldsymbol{W_v} \in \mathbb{R}^{d_{\mathsf{in}} \times d_{\mathsf{out}}}, \boldsymbol{b_v} \in \mathbb{R}^{d_{\mathsf{out}}}$

1: **function** Basic single-query attention
2:     $\boldsymbol{q} \leftarrow \boldsymbol{e}\boldsymbol{W_q} + \boldsymbol{b_q}$ ▷ Query linear projection
3:     **for** $t \in [T]$ **do**
4:         $\boldsymbol{k}_t \leftarrow \boldsymbol{e}_t \boldsymbol{W_k} + \boldsymbol{b_k}$ ▷ Key linear projection
5:         $\alpha_t = \frac{\exp(\boldsymbol{q} \cdot \boldsymbol{k}_t / \sqrt{d_{\mathsf{attn}}})}{\sum_{u=1}^{T} \exp(\boldsymbol{q} \cdot \boldsymbol{k}_u / \sqrt{d_{\mathsf{attn}}})}$ ▷ Softmax over scaled dot products, $\alpha_t \in (0,1)$
6:         $\boldsymbol{v}_t \leftarrow \boldsymbol{e}_t \boldsymbol{W_v} + \boldsymbol{b_v}$ ▷ Value linear projection
7:     **return** $\tilde{\boldsymbol{v}} = \sum_{t=1}^{T} \alpha_t \boldsymbol{v}_t$

Example: Basic single-query unmasked attention

We are at position 2, our query $\boldsymbol{q} = (11, 12)$ and keys

$$\boldsymbol{k_1} = (1, 2) \qquad \boldsymbol{k_2} = (4, 5) \qquad \boldsymbol{k_3} = (7, 8)$$

TrustHLT — Prof. Dr. Ivan Habernal

RUHR
UNIVERSITÄT
BOCHUM

RUB

Example: Basic single-query unmasked attention

We are at position 2, our query $\boldsymbol{q} = (11, 12)$ and keys

$$\boldsymbol{k_1} = (1, 2) \qquad \boldsymbol{k_2} = (4, 5) \qquad \boldsymbol{k_3} = (7, 8)$$

$$\boldsymbol{q} = \begin{pmatrix} 11 & 12 \end{pmatrix} \quad \boldsymbol{K}^\top = \begin{pmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \end{pmatrix}$$

TrustHLT — Prof. Dr. Ivan Habernal

RUHR
UNIVERSITÄT
BOCHUM   **RU**B

Example: Basic single-query unmasked attention

We are at position 2, our query $\boldsymbol{q} = (11, 12)$ and keys

$$\boldsymbol{k_1} = (1, 2) \qquad \boldsymbol{k_2} = (4, 5) \qquad \boldsymbol{k_3} = (7, 8)$$

$$\boldsymbol{q} = \begin{pmatrix} 11 & 12 \end{pmatrix} \quad \boldsymbol{K}^{\top} = \begin{pmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \end{pmatrix}$$

Dot products:

$$\boldsymbol{q} \cdot \boldsymbol{k_1} = (11, 12) \cdot (1, 2) = 11 + 24 = 35$$

TrustHLT — Prof. Dr. Ivan Habernal   RUHR UNIVERSITÄT BOCHUM   **RU**B

Example: Basic single-query unmasked attention

We are at position 2, our query $\boldsymbol{q} = (11, 12)$ and keys

$$\boldsymbol{k_1} = (1, 2) \qquad \boldsymbol{k_2} = (4, 5) \qquad \boldsymbol{k_3} = (7, 8)$$

$$\boldsymbol{q} = \begin{pmatrix} 11 & 12 \end{pmatrix} \quad \boldsymbol{K}^\top = \begin{pmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \end{pmatrix}$$

Dot products:

$$\boldsymbol{q} \cdot \boldsymbol{k_1} = (11, 12) \cdot (1, 2) = 11 + 24 = 35$$
$$\boldsymbol{q} \cdot \boldsymbol{k_2} = (11, 12) \cdot (4, 5) = 44 + 60 = 104$$

TrustHLT — Prof. Dr. Ivan Habernal   RUB

Example: Basic single-query unmasked attention

We are at position 2, our query $q = (11, 12)$ and keys

$$k_1 = (1, 2) \qquad k_2 = (4, 5) \qquad k_3 = (7, 8)$$

$$q = \begin{pmatrix} 11 & 12 \end{pmatrix} \quad K^\top = \begin{pmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \end{pmatrix}$$

Dot products:

$$q \cdot k_1 = (11, 12) \cdot (1, 2) = 11 + 24 = 35$$
$$q \cdot k_2 = (11, 12) \cdot (4, 5) = 44 + 60 = 104$$
$$q \cdot k_3 = (11, 12) \cdot (7, 8) = 77 + 96 = 173$$

RUHR UNIVERSITÄT BOCHUM **RU**B

Example: Basic single-query unmasked attention

We are at position 2, our query $\boldsymbol{q} = (11, 12)$ and keys
$\boldsymbol{k_1} = (1, 2)$ $\qquad \boldsymbol{k_2} = (4, 5)$ $\qquad \boldsymbol{k_3} = (7, 8)$

$$\boldsymbol{q} = \begin{pmatrix} 11 & 12 \end{pmatrix} \quad \boldsymbol{K}^{\top} = \begin{pmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \end{pmatrix}$$

Dot products:

$\boldsymbol{q} \cdot \boldsymbol{k_1} = (11, 12) \cdot (1, 2) = 11 + 24 = 35$
$\boldsymbol{q} \cdot \boldsymbol{k_2} = (11, 12) \cdot (4, 5) = 44 + 60 = 104$
$\boldsymbol{q} \cdot \boldsymbol{k_3} = (11, 12) \cdot (7, 8) = 77 + 96 = 173$

Raw scores $= (35, 104, 173)$, after softmax (no scaling)
$\boldsymbol{\alpha} = (0.000\ldots, 0.000\ldots, 0.999\ldots)$

Example: Basic single-query unmasked attention

From previous slide

We are at position 2, our query $q = (11, 12)$ and keys

$$k_1 = (1, 2) \qquad k_2 = (4, 5) \qquad k_3 = (7, 8)$$

Raw scores $= (35, 104, 173)$, after softmax (no scaling)

$$\alpha = (0.000\ldots, 0.000\ldots, 0.999\ldots)$$

Value at position 3 highest weight

- We are currently at position 2
- Position 3 is **in the future**

TrustHLT — Prof. Dr. Ivan Habernal   RUHR UNIVERSITÄT BOCHUM **RUB**

We want to attend only to previous tokens (4 token example)

At position 1 we should not attend to token 2, 3, and 4

At position 2 we should not attend to token 3 and 4

At position 3 we should not attend to token 4

At position 4 we can attend to all of them

$$\text{Raw associations} = \begin{pmatrix} 11 & 12 & 13 & 14 \\ 21 & 22 & 23 & 24 \\ 31 & 32 & 33 & 34 \\ 41 & 42 & 43 & 44 \end{pmatrix}$$

We want to assign zero probability (using softmax) to
"future" tokens

We want to attend only to previous tokens

$$\text{Raw associations} = \begin{pmatrix} 11 & 12 & 13 & 14 \\ 21 & 22 & 23 & 24 \\ 31 & 32 & 33 & 34 \\ 41 & 42 & 43 & 44 \end{pmatrix}$$

Assign zero probability (after softmax) to "future" tokens

TrustHLT — Prof. Dr. Ivan Habernal   RUHR UNIVERSITÄT BOCHUM **RU**B

We want to attend only to previous tokens

$$\text{Raw associations } = \begin{pmatrix} 11 & 12 & 13 & 14 \\ 21 & 22 & 23 & 24 \\ 31 & 32 & 33 & 34 \\ 41 & 42 & 43 & 44 \end{pmatrix}$$

Assign zero probability (after softmax) to "future" tokens

Replace input from $t + 1$ onwards with $-\infty$

$$\text{Raw associations "masked" } = \begin{pmatrix} 11 & -\infty & -\infty & -\infty \\ 21 & 22 & -\infty & -\infty \\ 31 & 32 & 33 & -\infty \\ 41 & 42 & 43 & 44 \end{pmatrix}$$

Assigns zero probability (using softmax) to "future" tokens

TrustHLT — Prof. Dr. Ivan Habernal   RUHR UNIVERSITÄT BOCHUM **RU**B

Uni-directional masking for self-attention

For $t_z, t_x \in [\ell_x]$

$$\text{mask}[t_x, t_z] = \begin{cases} 1 & \text{if } t_z \leq t_x \\ 0 & \text{otherwise} \end{cases}$$

Example for $\ell_x = 4$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

Creating the mask and indexing tensor by this mask very easy in pytorch

TrustHLT — Prof. Dr. Ivan Habernal

RUHR UNIVERSITÄT BOCHUM **RU**B

Left-to-right masked self-attention

Input: $\boldsymbol{X} \in \mathbb{R}^{\ell_x \times d_x}$, vector representations of the sequence of length ℓ_x
Output: $\tilde{\boldsymbol{V}} \in \mathbb{R}^{\ell_x \times d_{\text{out}}}$, updated vector representations of tokens in $\boldsymbol{X}$
Params $\boldsymbol{\mathcal{W}_{qkv}}$: $\boldsymbol{W_q}$, $\boldsymbol{W_k} \in \mathbb{R}^{d_x \times d_{\text{attn}}}$, $\boldsymbol{b_q}$, $\boldsymbol{b_k} \in \mathbb{R}^{d_{\text{attn}}}$, $\boldsymbol{W_v} \in \mathbb{R}^{d_x \times d_{\text{out}}}$, $\boldsymbol{b_v} \in \mathbb{R}^{d_{\text{out}}}$

1: **function** Attention($\boldsymbol{X}; \boldsymbol{\mathcal{W}_{qkv}}$)
2: $\quad \boldsymbol{Q} \leftarrow \boldsymbol{X}\boldsymbol{W_q} +_{\text{(rows)}} \boldsymbol{b_q}$ $\qquad\qquad\qquad\qquad\qquad$ ▷ Query $\in \mathbb{R}^{\ell_x \times d_{\text{attn}}}$
3: $\quad \boldsymbol{K} \leftarrow \boldsymbol{X}\boldsymbol{W_k} +_{\text{(rows)}} \boldsymbol{b_k}$ $\qquad\qquad\qquad\qquad\qquad$ ▷ Key $\in \mathbb{R}^{\ell_x \times d_{\text{attn}}}$
4: $\quad \boldsymbol{V} \leftarrow \boldsymbol{X}\boldsymbol{W_v} +_{\text{(rows)}} \boldsymbol{b_v}$ $\qquad\qquad\qquad\qquad\qquad$ ▷ Value $\in \mathbb{R}^{\ell_x \times d_{\text{out}}}$
5: $\quad \boldsymbol{S} \leftarrow \frac{1}{\sqrt{d_{\text{attn}}}}(\boldsymbol{Q}\boldsymbol{K}^{\top})$ $\qquad\qquad\qquad\qquad$ ▷ Scaled score $\in \mathbb{R}^{\ell_x \times \ell_x}$
6: $\quad$ **for all** $t_z, t_x \in [T]$ **do**
7: $\qquad$ **if** $\neg \,\text{mask}[t_x, t_z]$ **then** $\boldsymbol{S}[t_x, t_z] \leftarrow -\infty$ $\qquad$ ▷ Causal masking
8: $\quad$ **return** $\tilde{\boldsymbol{V}} = \text{softmax}_{\text{row}}(\boldsymbol{S})\,\boldsymbol{V}$

Single masked self-attention head (GPT)

TrustHLT — Prof. Dr. Ivan Habernal   RUHR UNIVERSITÄT BOCHUM   RUB

Left-to-right masked self-attention

$$\tilde{\boldsymbol{V}} = \mathrm{softmax}_{\mathsf{row}}(\boldsymbol{S})\,\boldsymbol{V}$$

The output $\tilde{\boldsymbol{V}}[1:t,:]$ only depends on $\boldsymbol{X}[1:t,:]$, so it can be used to "predict" $\boldsymbol{X}[t+1,:]$

TrustHLT — Prof. Dr. Ivan Habernal   RUHR UNIVERSITÄT BOCHUM RUB
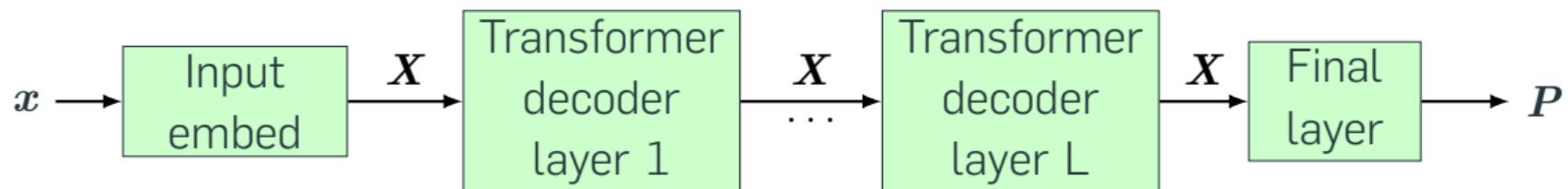
GPT (decoding-only transformer, forward pass)

1: **function** DTransformer($\boldsymbol{x}; \boldsymbol{\mathcal{W}}$)

2: . . .
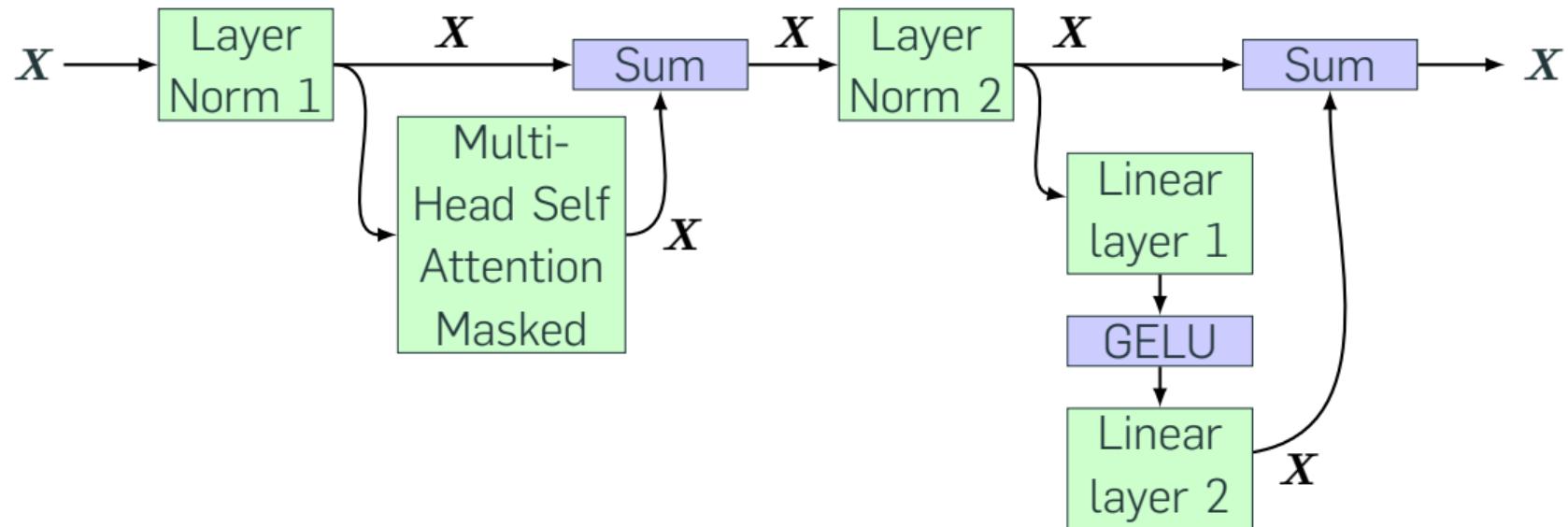
Input: — $\boldsymbol{x} \in V^*$, a sequence of token IDs, $\boldsymbol{\mathcal{W}}$ — all trainable parameters

Output: $\boldsymbol{P} \in (0, 1)^{\ell_x \times N_V}$, where each row of $\boldsymbol{P}$ is a distribution over the vocabulary conditioned on previous tokens $\hat{\boldsymbol{P}}(x[t+1]|\boldsymbol{x}[1:t])$

Transformer decoder (GPT)

$x \rightarrow$ Input embed $\xrightarrow{X}$ Transformer decoder layer 1 $\xrightarrow{X}$ ... Transformer decoder layer L $\xrightarrow{X}$ Final layer $\rightarrow P$

TrustHLT — Prof. Dr. Ivan Habernal   RUHR UNIVERSITÄT BOCHUM **RU**B

GPT (decoding-only transformer, transformer layer)

TrustHLT — Prof. Dr. Ivan Habernal

RUHR
UNIVERSITÄT
BOCHUM RUB

GPT (decoding-only transformer, forward pass)

1: **function** DTransformer($\boldsymbol{x}; \boldsymbol{\mathcal{W}}$)

2:      $\ell \leftarrow \text{length}(\boldsymbol{x})$

3:      for $t \in [\ell] : \boldsymbol{e}_t \leftarrow \boldsymbol{W_e}[x[t],:] + \boldsymbol{W_p}[t,:]$ ▷ Token emb. + positional emb.

4:      $\boldsymbol{X} \leftarrow \text{Stack row-wise}[\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots \boldsymbol{e}_\ell]$

5:      **for** $l = 1, 2, \ldots, L$ **do**

6:          $\boldsymbol{X} \leftarrow \text{LayerNormPerRow}(\boldsymbol{X}|\boldsymbol{\gamma}_l^1, \boldsymbol{\beta}_l^1)$ ▷ Normalization first

7:          $\boldsymbol{X} \leftarrow \boldsymbol{X} + \text{MHAttentionMask}(\boldsymbol{X}|\boldsymbol{\mathcal{W}}_l)$ ▷ Just added masking

8:          $\boldsymbol{X} \leftarrow \text{LayerNormPerRow}(\boldsymbol{X}|\boldsymbol{\gamma}_l^2, \boldsymbol{\beta}_l^2)$

9:          $\boldsymbol{X} \leftarrow \boldsymbol{X} + \left(\text{GELU}(\boldsymbol{X}\boldsymbol{W}_l^{\text{mlp1}} +_{(\text{row})} \boldsymbol{b}_l^{\text{mlp1}}) \boldsymbol{W}_l^{\text{mlp2}} +_{(\text{row})} \boldsymbol{b}_l^{\text{mlp2}} \right)$ ▷ MLP

10:      $\boldsymbol{X} \leftarrow \text{LayerNormPerRow}(\boldsymbol{X}|\boldsymbol{\gamma}_l, \boldsymbol{\beta}_l)$

11:      **return** $\boldsymbol{P} = \text{softmax}(\boldsymbol{X}\boldsymbol{W_u})$ ▷ Project to vocab., probabilities

GPT (decoding-only transformer, forward pass)

Differences from BERT forward-pass:

Switched the ordering of layer normalization (line 6 and 8)

No final layer projection

Attention with left-to-right masking (line 7)

Training

1 Training

Decoder-Transfomer: Training on next token prediction

1: **function** DTraining($\{\boldsymbol{x}_n\}_{n=1}^{N_{\text{data}}}$ seqs, $\boldsymbol{\theta}$ init. params, N_{epochs}, η)
2:     **for** $i \in [N_{\text{epochs}}]$ **do**
3:         **for** $n \in [N_{\text{data}}]$ **do**
4:             $\ell \leftarrow \text{length}(\boldsymbol{x}_n)$
5:             $\boldsymbol{P}_{\boldsymbol{\theta}} \leftarrow \text{DTransformer}(\boldsymbol{x}_n|\boldsymbol{\theta})$
6:             $\text{loss}_{\boldsymbol{\theta}} \leftarrow -\sum_{t=1}^{\ell-1} \log \boldsymbol{P}_{\boldsymbol{\theta}}[t, \boldsymbol{x}_n[t+1]]$
7:             $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \cdot \nabla\text{loss}_{\boldsymbol{\theta}}$
8:     **return** $\boldsymbol{\theta}$

TrustHLT — Prof. Dr. Ivan Habernal

RUHR UNIVERSITÄT BOCHUM **RUB**

Explaining line 6 (negative log likelihood)

$$\left(\text{The}\quad \text{cat}\quad \text{sat}\right) \to \boldsymbol{x}_n = \left(21\quad 11987\quad 5438\right)$$

$$\boldsymbol{P_\theta} \leftarrow \text{DTransformer}(\boldsymbol{x}_n | \boldsymbol{\theta})$$

$$\boldsymbol{P_\theta} = \begin{pmatrix} 0.001 & 0.0007 & \ldots & 0.0003 \\ 0.0013 & 0.0065 & \ldots & 0.0001 \\ 0.079 & 0.015 & \ldots & 0.0001 \end{pmatrix}$$

$\boldsymbol{P_\theta} \in (0,1)^{\ell_x \times N_V}$, where each row of $\boldsymbol{P}$ is a distribution over the vocabulary

RUHR
UNIVERSITÄT
BOCHUM   RUB

Explaining line 6 (negative log likelihood), $t = 1$

$$\boldsymbol{x}_n = (21, 11987, 5438) \; \boldsymbol{P_\theta} = \begin{pmatrix} 0.001 & \dots & 0.0041_{11987} & \dots 0.0003 \\ \vdots & & & \end{pmatrix}$$

For $t = 1$, the model should learn to predict "cat" (idx 11987)

Gold: $\boldsymbol{y} = (0, 0, \dots, 1_{11987}, \dots, 0) \in \mathbb{R}^{N_V}$

Pred: $\hat{\boldsymbol{y}} = \boldsymbol{P_\theta}[1, :] = (0.001, \dots, 0.0041_{11987}, \dots 0.0003) \in \mathbb{R}^{N_V}$

Categorical cross entropy loss (Lec. 4)

$L(\hat{\boldsymbol{y}}, \boldsymbol{y}) := -\sum_{k=1}^{K} \boldsymbol{y}_{[k]} \log\left(\hat{\boldsymbol{y}}_{[k]}\right)$
$= -1 \cdot \log(\hat{\boldsymbol{y}}[11987]) = -\log(\boldsymbol{P_\theta}[1, 11987])$
$= -\log(\boldsymbol{P_\theta}[1, \boldsymbol{x}_n[1+1]]) = -\log(\boldsymbol{P_\theta}[t, \boldsymbol{x}_n[t+1]])$

TrustHLT — Prof. Dr. Ivan Habernal   RUHR UNIVERSITÄT BOCHUM   **RU**B

Decoder model prompting

Decoding

Input: $\boldsymbol{x} \in V^*$, a prompt (sequence of token IDs)

Output: $\boldsymbol{y} \in V^*$, continuation

1: **function** DInference($\boldsymbol{x}, \boldsymbol{\theta}, \ell_{\mathsf{gen}}, \tau$)
2: $\quad \ell \leftarrow \mathsf{length}(\boldsymbol{x})$
3: $\quad$ **for** $i \in [\ell_{\mathsf{gen}}]$ **do**
4: $\quad\quad \boldsymbol{P} \leftarrow \mathsf{DTransformer}(\boldsymbol{x}|\boldsymbol{\theta})$
5: $\quad\quad \boldsymbol{p} \leftarrow \boldsymbol{P}[\ell + i - 1, :]$
6: $\quad\quad$ sample token y from $\boldsymbol{q} \propto \boldsymbol{p}^{(1/\tau)}$
7: $\quad\quad \boldsymbol{x} \leftarrow [\boldsymbol{x}, y]$
8: $\quad$ **return** $\boldsymbol{y} = \boldsymbol{x}[\ell + 1 : \ell + \ell_{\mathsf{gen}}]$

TrustHLT — Prof. Dr. Ivan Habernal

RUHR UNIVERSITÄT BOCHUM   **RU**B

Evolution of GPT

Towards GPT-1

Decoder part of the Transformer Encoder-Decoder model for MT (Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, L. Kaiser, and Polosukhin, 2017)

Dropping encoder and using only decoder that consumes input and produces output trained as a standard language model for writing Wikipedia pages as summarization task (Liu, Saleh, Pot, Goodrich, Sepassi, Ł. Kaiser, and Shazeer, 2018)

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin (2017). **"Attention Is All You Need".** In: *Advances in Neural Information Processing Systems 30.* Long Beach, CA, USA: Curran Associates, Inc., pp. 5998–6008

P. J. Liu, M. Saleh, E. Pot, B. Goodrich, R. Sepassi, Ł. Kaiser, and N. Shazeer (2018). **"Generating Wikipedia by Summarizing Long Sequences".** In: *Proceedings of the 6th International Conference on Learning Representations.* Vancouver, BC, Canada

RUHR UNIVERSITÄT BOCHUM   RUB

GPT-1

GPT-1 (Radford, Narasimhan, Salimans, and Sutskever, 2018) adapted decoder only transformer

A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever (2018). **Improving Language Understanding by Generative Pre-Training.** Technical report. OpenAI

- pre-training as LM
- fine-tuning with an extra final layer for the given task
- pre-trained on BooksCorpus (7k unique unpublished books)
- 12 decoder layers, 12 attention heads, 768 embedding size

"improving the state of the art on 9 of the 12 datasets we study"

TrustHLT — Prof. Dr. Ivan Habernal   RUHR UNIVERSITÄT BOCHUM   RUB

GPT-2

Larger GPT-1

- pre-training as LM
- pre-trained on custom web scrape (all outbounds links from Reddit with at least 3 karma points, for quality reasons), 8 million documents total
- 48 decoder layers, 1600 embedding size (1.542 billion params)

Representing inputs, prompting, etc. — next lectures

TrustHLT — Prof. Dr. Ivan Habernal   RUHR UNIVERSITÄT BOCHUM   RUB

License and credits

Credits

Ivan Habernal

Content from ACL Anthology papers licensed under CC-BY
https://www.aclweb.org/anthology