DSME 6635: Artificial Intelligence for Business Research

Deep-Learning-based NLP: Attention and Transformer

Renyu (Philip) Zhang

1

Agenda

- Attention Mechanism
- Transformer: Attention is All You Need

2

2

NLP Roadmap

Bert: Pre-training of deep bidirectional transformers for language understanding
J Devlin, MW Chang, K Lee, K Toutanova - arXiv preprint arXiv …, 2018 - arxiv.org
… We introduce **BERT** and its detailed implementation in this … For finetuning, the **BERT** model is first initialized with the pre-trained … , we present **BERT** fine-tuning results on 11 **NLP** tasks. …
Save Cite Cited by 34042 Related articles All 41 versions

Attention is **all you need**
A Vaswani, N Shazeer, N Parmar… - Advances in neural …, 2017 - proceedings.neurips.cc
… the number of **attention** heads and the **attention** key and value dimensions, keeping the amount of computation constant, as described in Section 3.2.2. While single-head **attention** is 0.9 …
Save Cite Cited by 36430 Related articles All 35 versions

Probability & Vector Representations of NLP	→	Word2vec, RNN & Seq2seq	→	Transformer-based Models, BERT	→	Large Language Models (LLM)
Before 2010		2013-2017		2018-2022		2023-now

Efficient estimation of **word representations** in **vector space**
T Mikolov, K Chen, G Corrado, J Dean - arXiv preprint arXiv:1301.3781, 2013 - arxiv.org
… **vector** X = **vector**("biggest")−**vector**("big") + **vector**("small"). Then, we search in the **vector space** for the **word** … question (we discard the input question **words** during this search). When the …
Save Cite Cited by 27227 Related articles All 48 versions

[PDF] **Glove**: **Global vectors** for word representation
J Pennington, R Socher… - Proceedings of the 2014 …, 2014 - aclanthology.org
… We use our insights to construct a new model for word representation which we call **GloVe**, for **Global Vectors**, because the **global** corpus statistics are captured directly by the model. …
Save Cite Cited by 26265 Related articles All 34 versions

[PDF] Language models are unsupervised multitask learners
A Radford, J Wu, R Child, D Luan, D Amodei… - OpenAI blog, 2019 - persagen.com
Natural language processing tasks, such as question answering, machine translation, reading comprehension, and summarization, are typically approached with supervised learning on taskspecific datasets. We demonstrate that language models begin to learn these tasks without any explicit supervision when trained on a new dataset of millions of webpages called WebText. When conditioned on a document plus questions, the answers generated by the language model reach 55 F1 on the CoQA dataset-matching or exceeding
Save Cite Cited by 2834 Related articles All 16 versions

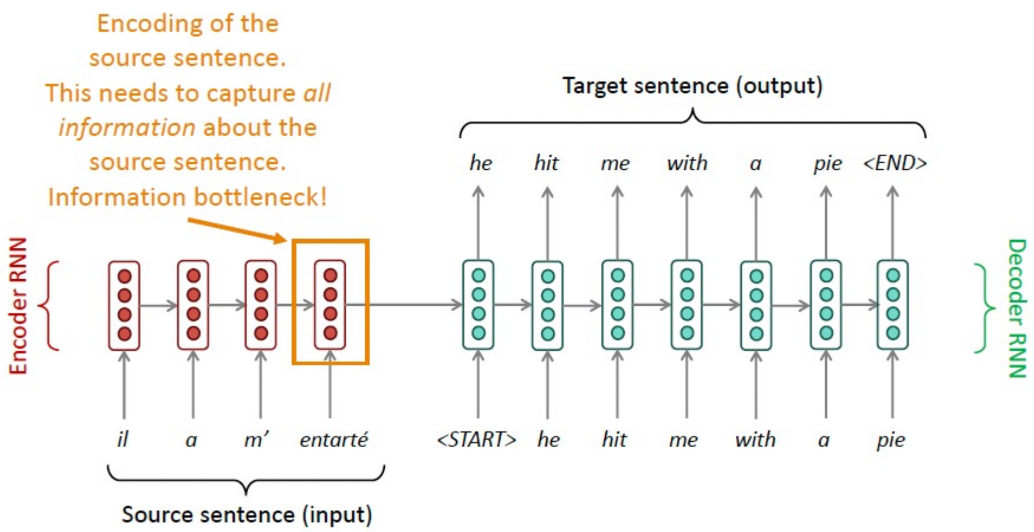
Training language models to follow instructions with human feedback
L Ouyang, J Wu, X Jiang, D Almeida… - Advances in …, 2022 - proceedings.neurips.cc
Making language models bigger does not inherently make them better at following a user's intent. For example, large language models can generate outputs that are untruthful, toxic, or simply not helpful to the user. In other words, these models are not aligned with their users. In this paper, we show an avenue for aligning language models with user intent on a wide range of tasks by fine-tuning with human feedback. Starting with a set of labeler-written prompts and prompts submitted through a language model API, we collect a dataset of …
Save Cite Cited by 4049 Related articles All 10 versions

3

3

Information Bottleneck in RNN

Reference: Stanford CS224N, Lecture 8: https://web.stanford.edu/class/cs224n/slides/cs224n-2024-lecture08-transformers.pdf



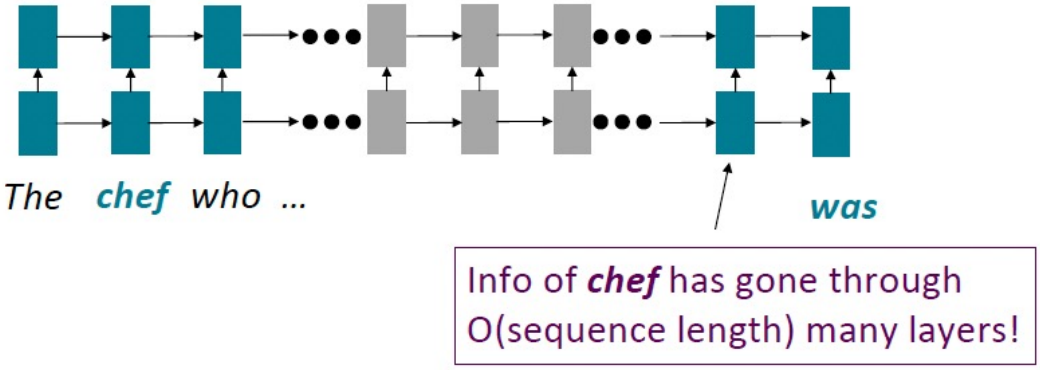
4

4

Issue with RNN: Linear Interaction Distance

Reference: Stanford CS224N, Lecture 8: https://web.stanford.edu/class/cs224n/slides/cs224n-2024-lecture08-transformers.pdf

- Human languages are intrinsically NOT linearly ordered.



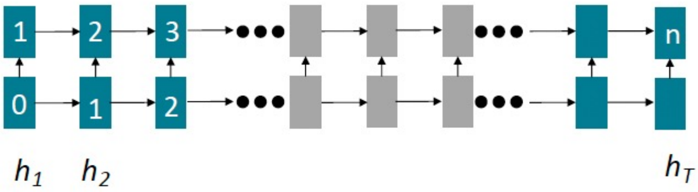
5

5

Issue with RNN: Non-parallelizability

Reference: Stanford CS224N, Lecture 8: https://web.stanford.edu/class/cs224n/slides/cs224n-2024-lecture08-transformers.pdf

- Forward and backward passes both have O(sequence length) unparallelizable operations.
- GPUs can perform independent small computations quickly in a large scale.
- Future hidden states cannot be computed (in full) before past RNN hidden states have been computed.
- Cannot scale with a very large dataset.



Numbers indicate min # of steps before a state can be computed

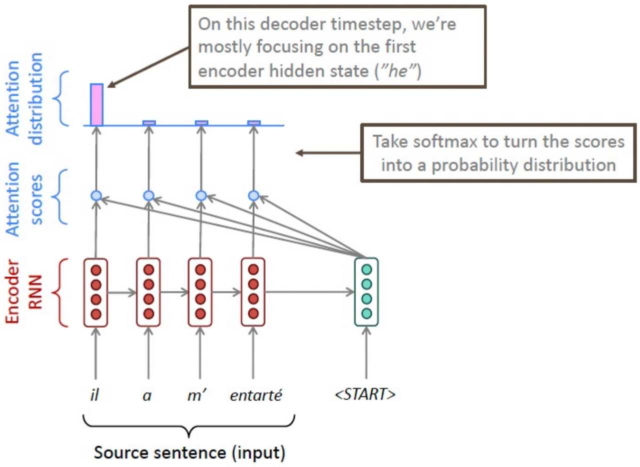
6

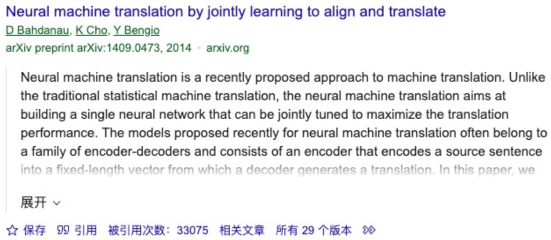
6

Attention Mechanism in Seq2Seq

Reference: Stanford CS224N, Lecture 8: https://web.stanford.edu/class/cs224n/slides/cs224n-2024-lecture08-transformers.pdf

- Idea of attention: On each step of the decoder, use direct connection to the encoder to focus on a particular part of the source sentence.




Neural machine translation by jointly learning to align and translate
D Bahdanau, K Cho, Y Bengio
arXiv preprint arXiv:1409.0473, 2014 · arxiv.org

Neural machine translation is a recently proposed approach to machine translation. Unlike the traditional statistical machine translation, the neural machine translation aims at building a single neural network that can be jointly tuned to maximize the translation performance. The models proposed recently for neural machine translation often belong to a family of encoder-decoders and consists of an encoder that encodes a source sentence into a fixed-length vector from which a decoder generates a translation. In this paper, we

展开 ∨

☆ 保存 99 引用 被引用次数：33075 相关文章 所有 29 个版本

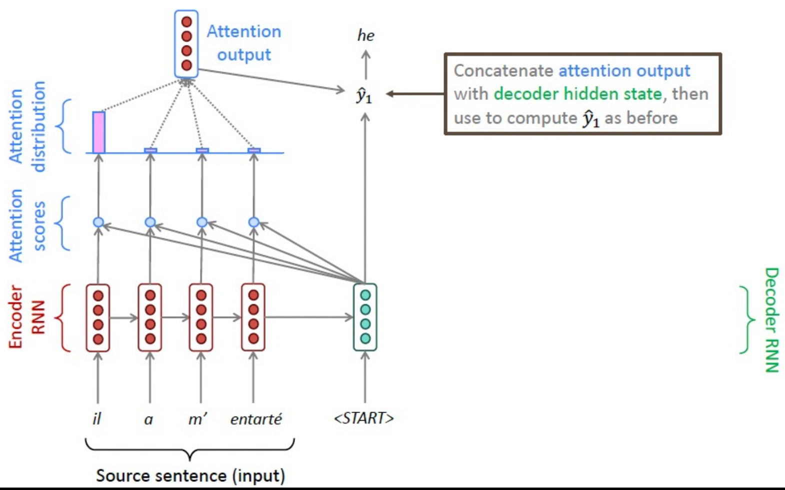
7

7

Attention Mechanism in Seq2Seq

Reference: Stanford CS224N, Lecture 8: https://web.stanford.edu/class/cs224n/slides/cs224n-2024-lecture08-transformers.pdf

- Idea of attention: On each step of the decoder, use direct connection to the encoder to focus on a particular part of the source sentence.



8

8

Attention Mechanism: Equations

Reference: Stanford CS224N, Lecture 8: https://web.stanford.edu/class/cs224n/slides/cs224n-2024-lecture08-transformers.pdf

- Idea of attention: On each step of the decoder, use direct connection to the encoder to focus on a particular part of the source sentence.
  - We have encoder hidden states $h_1, \dots, h_N \in \mathbb{R}^h$
  - On timestep *t*, we have decoder hidden state $s_t \in \mathbb{R}^h$
  - We get the attention scores e^t for this step:

$$e^t = [\boldsymbol{s}_t^T \boldsymbol{h}_1, \dots, \boldsymbol{s}_t^T \boldsymbol{h}_N] \in \mathbb{R}^N$$

  - We take softmax to get the attention distribution α^t for this step (this is a probability distribution and sums to 1)

$$\alpha^t = \mathrm{softmax}(\boldsymbol{e}^t) \in \mathbb{R}^N$$

  - We use α^t to take a weighted sum of the encoder hidden states to get the attention output $\boldsymbol{a}_t$

$$\boldsymbol{a}_t = \sum_{i=1}^{N} \alpha_i^t \boldsymbol{h}_i \in \mathbb{R}^h$$

  - Finally we concatenate the attention output $\boldsymbol{a}_t$ with the decoder hidden state s_t and proceed as in the non-attention seq2seq model

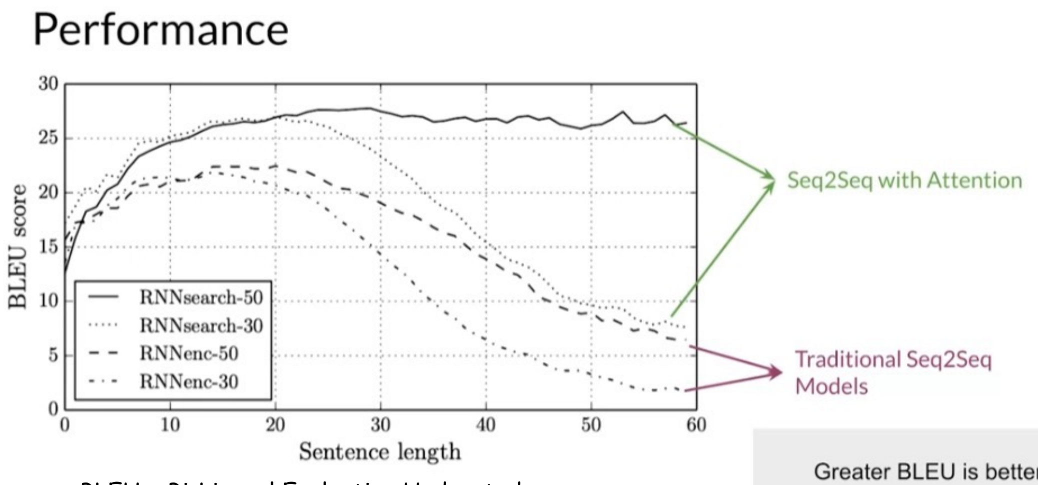
$$[\boldsymbol{a}_t; \boldsymbol{s}_t] \in \mathbb{R}^{2h}$$

9

9

Attention Performs Very Well in NMT

Reference: Stanford CS224N, Lecture 8: https://web.stanford.edu/class/cs224n/slides/cs224n-2024-lecture08-transformers.pdf



Greater BLEU is better

BLEU = Bi-Lingual Evaluation Understudy (https://en.wikipedia.org/wiki/BLEU)

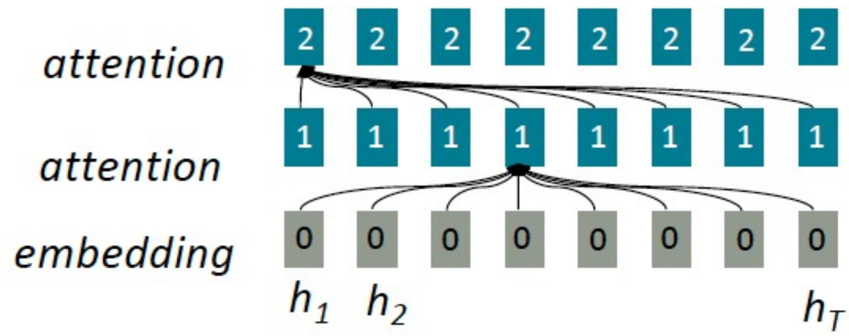
10

10

Attention Addresses RNN Issues

Reference: Stanford CS224N, Lecture 8: https://web.stanford.edu/class/cs224n/slides/cs224n-2024-lecture08-transformers.pdf

- Information Retrieval perspective: Attention treats each word's representation (i.e., hidden state) as a query to access and incorporate information from a set of values.
- Attention applied to a single sequence: Number of unparallelizable operations does not increase with sequence length. The maximum interaction distance is O(1).



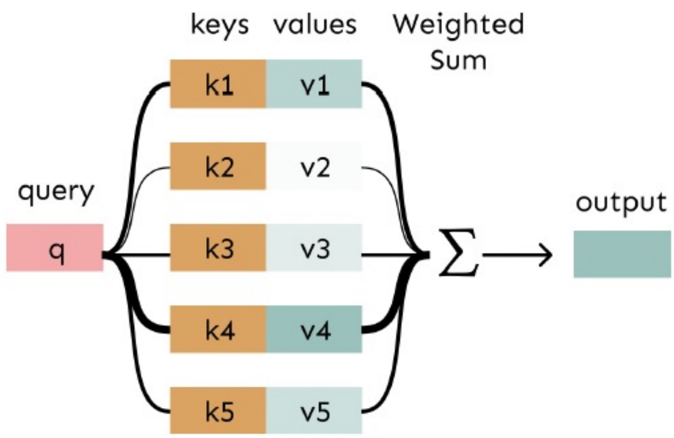
11

11

Attention as a Very General DL Technique

Reference: Stanford CS224N, Lecture 8: https://web.stanford.edu/class/cs224n/slides/cs224n-2024-lecture08-transformers.pdf

- Attention: Given a set of vector values and a vector of query, attention is a technique to compute a weighted sum of the values dependent on the query.
  - The weighted sum is a selective summary of the information contained in the values, where the query determines which values to focus on.
  - A fixed-size representation of an arbitrary set of representations (values), dependent on some other representation (query).
- In seq2seq + attention, each decoder hidden state (query) attends to all the encoder hidden states (values)..



12

12

A Family of Attention Models

Reference: Stanford CS224N, Lecture 8: https://web.stanford.edu/class/cs224n/slides/cs224n-2024-lecture08-transformers.pdf

Name	Alignment score function	Citation
Content-base attention	$\text{score}(\boldsymbol{s}_t, \boldsymbol{h}_i) = \text{cosine}[\boldsymbol{s}_t, \boldsymbol{h}_i]$	Graves2014
Additive(*)	$\text{score}(\boldsymbol{s}_t, \boldsymbol{h}_i) = \mathbf{v}_a^\top \tanh(\mathbf{W}_a[\boldsymbol{s}_t; \boldsymbol{h}_i])$	Bahdanau2015
Location-Base	$\alpha_{t,i} = \text{softmax}(\mathbf{W}_a \boldsymbol{s}_t)$ Note: This simplifies the softmax alignment to only depend on the target position.	Luong2015
General	$\text{score}(\boldsymbol{s}_t, \boldsymbol{h}_i) = \boldsymbol{s}_t^\top \mathbf{W}_a \boldsymbol{h}_i$ where $\mathbf{W}_a$ is a trainable weight matrix in the attention layer.	Luong2015
Dot-Product	$\text{score}(\boldsymbol{s}_t, \boldsymbol{h}_i) = \boldsymbol{s}_t^\top \boldsymbol{h}_i$	Luong2015
Scaled Dot-Product(^)	$\text{score}(\boldsymbol{s}_t, \boldsymbol{h}_i) = \frac{\boldsymbol{s}_t^\top \boldsymbol{h}_i}{\sqrt{n}}$ Note: very similar to the dot-product attention except for a scaling factor; where n is the dimension of the source hidden state.	Vaswani2017

13

13

Agenda

- Attention Mechanism
- Transformer: Attention is All You Need

14

14

Attention is All You Need

References: Stanford CS224N, Lecture 8: https://web.stanford.edu/class/cs224n/slides/cs224n-2024-lecture08-transformers.pdf
https://peterbloem.nl/blog/transformers

- Transformer: No RNN architecture, just attention mechanism.
- Self-attention: To generate y_t, we need to pay attention to $y_{<t}$.

$$w'_{ij} = \frac{\mathbf{q}_i{}^T \mathbf{k}_j}{\sqrt{k}}$$

Query $\mathbf{q}_i = \mathbf{W}_q \mathbf{x}_i$ Key $\mathbf{k}_i = \mathbf{W}_k \mathbf{x}_i$ Value $\mathbf{v}_i = \mathbf{W}_v \mathbf{x}_i$

$$w'_{ij} = \mathbf{q}_i{}^T \mathbf{k}_j$$
$$w_{ij} = \text{softmax}(w'_{ij})$$
$$\mathbf{y}_i = \sum_j w_{ij} \mathbf{v}_j \,.$$

Why does it work?

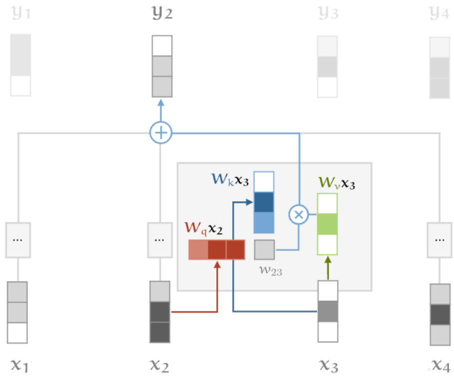


Illustration of the self-attention with key, query and value transformations.

Attention is all you need
A Vaswani, N Shazeer, N Parmar… - Advances in neural …, 2017 - proceedings.neurips.cc
… to attend to all positions in the decoder up to and including that position. We need to prevent
… We implement this inside of scaled dot-product attention by masking out (setting to −∞) …
☆ 保存 99 引用 被引用次数：109517 相关文章 所有 62 个版本 ≫

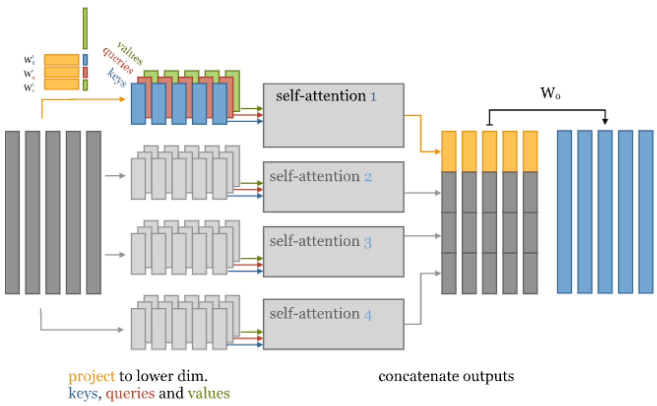
15

15

Multi-head Attention

References: Stanford CS224N, Lecture 8: https://web.stanford.edu/class/cs224n/slides/cs224n-2024-lecture08-transformers.pdf
https://peterbloem.nl/blog/transformers

- Multi-head attention is a way to speed up the training procedure.
- Instead of using a large matrix to compute all attentions, we can compute multiple attention matrices and concatenate the final vectors.
- Allows for parallel computing: Deploy attention mechanisms to multiple computing cores in parallel and sum them up at the end.
- Input dim = 256, 8 attention heads, each with 32 dimensions.



The basic idea of multi-head self-attention with 4 heads. To get our keys, queries and values, we project the input down to vector sequences of smaller dimension.

16

16

Position Encoding

References: Stanford CS224N, Lecture 8: https://web.stanford.edu/class/cs224n/slides/cs224n-2024-lecture08-transformers.pdf
https://peterbloem.nl/blog/transformers

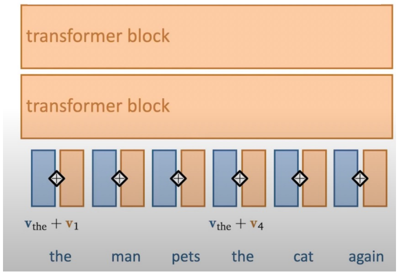
- Position embeddings: Position vectors which are learned.
- Position encoding: The function from position to vector.
- The final input of the model is the sum of word embeddings and position embeddings.

word embeddings:

$\mathbf{v}_{\text{the}}, \mathbf{v}_{\text{man}}, \mathbf{v}_{\text{pets}}, \mathbf{v}_{\text{cat}}, \mathbf{v}_{\text{again}}$

position embeddings:

$\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4, \mathbf{v}_5, \ldots$



17

17

Auto-Regression

References: Stanford CS224N, Lecture 8: https://web.stanford.edu/class/cs224n/slides/cs224n-2024-lecture08-transformers.pdf
https://peterbloem.nl/blog/transformers

- Self-supervised learning for transformers.
- To use self-attention in decoders, we need to mask the future.
- Inefficient implementation: Change the set of keys and queries to include only past words.
- Parallelizable implementation: Mask out attention to future words by setting the weight to -inf.

$$w'_{ij} = \begin{cases} q_i^T k_j, j \le i \\ -\infty, j > i \end{cases}$$

	[START]	The	chef	who
[START]		$-\infty$	$-\infty$	$-\infty$
The			$-\infty$	$-\infty$
chef				$-\infty$
who				

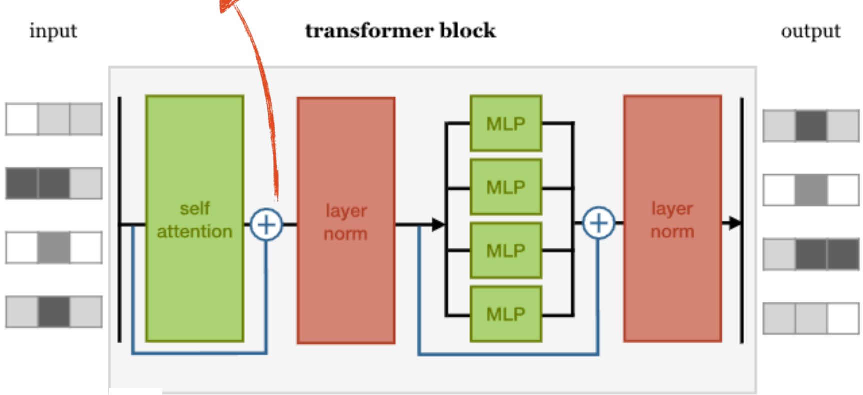
18

18

Transformer

References: Stanford CS224N, Lecture 8: https://web.stanford.edu/class/cs224n/slides/cs224n-2024-lecture08-transformers.pdf
https://peterbloem.nl/blog/transformers

- Transformer = Multi-head self-attention + MLP + position encoding + autoregression
- Need to add skip-connection and layer normalization (the order does not matter).



19

19

Layer Normalization

References: Stanford CS224N, Lecture 8: https://web.stanford.edu/class/cs224n/slides/cs224n-2024-lecture08-transformers.pdf
https://peterbloem.nl/blog/transformers

- Layer normalization: A trick to help models train faster.
- Cut down on uninformative variation in hidden values by normalizing to unit mean and standard deviation within each layer: Normalized gradients.

Layer normalization
JL Ba, JR Kiros, GE Hinton - arXiv preprint arXiv:1607.06450, 2016 - arxiv.org
… , we transpose batch **normalization** into **layer normalization** by computing the mean and variance used for **normalization** from all of the summed inputs to the neurons in a **layer** on a …
☆ Save 99 Cite Cited by 10350 Related articles All 6 versions

- Let $x \in \mathbb{R}^d$ be an individual (word) vector in the model.
- Let $\mu = \sum_{j=1}^{d} x_j$; this is the mean; $\mu \in \mathbb{R}$.
- Let $\sigma = \sqrt{\frac{1}{d}\sum_{j=1}^{d}(x_j - \mu)^2}$; this is the standard deviation; $\sigma \in \mathbb{R}$.
- Let $\gamma \in \mathbb{R}^d$ and $\beta \in \mathbb{R}^d$ be learned "gain" and "bias" parameters. (Can omit!)
- Then layer normalization computes:

$$\text{output} = \frac{x - \mu}{\sqrt{\sigma} + \epsilon} * \gamma + \beta$$

Normalize by scalar mean and variance

Modulate by learned elementwise gain and bias

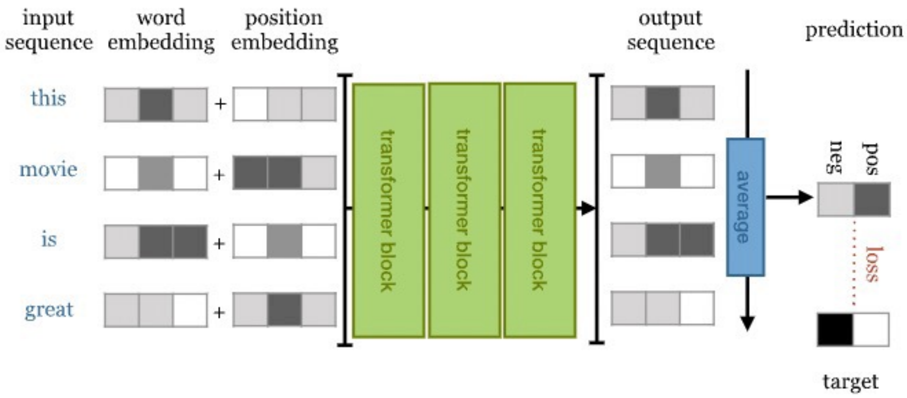
20

20

Classification Transformer

References: Stanford CS224N, Lecture 8: https://web.stanford.edu/class/cs224n/slides/cs224n-2024-lecture08-transformers.pdf
https://peterbloem.nl/blog/transformers

- Directly train a classifier on top of a transformer.



21

21

Attention is All You Need

References: Stanford CS224N, Lecture 8: https://web.stanford.edu/class/cs224n/slides/cs224n-2024-lecture08-transformers.pdf
https://peterbloem.nl/blog/transformers

- Input: Sequence in language one and Sequence in language two.
- Architecture: Encoder + Decoder
- 8 heads, 512 embedding dimensions, 2048 sentence length
- Trained on 8 GPUs for 5 days.

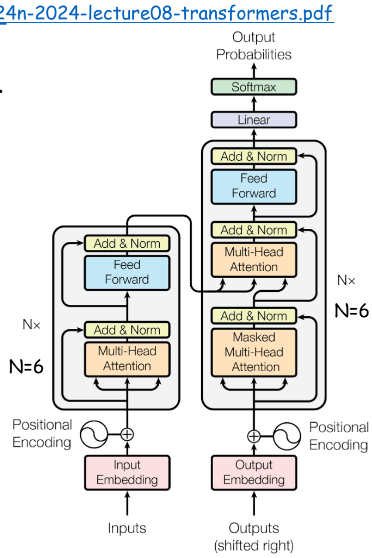


Figure 1: The Transformer - model architecture.

22

22

Attention is All You Need

References: Stanford CS224N, Lecture 8: https://web.stanford.edu/class/cs224n/slides/cs224n-2024-lecture08-transformers.pdf
https://peterbloem.nl/blog/transformers

Machine Translation

Model	BLEU EN-DE	BLEU EN-FR	Training Cost (FLOPs) EN-DE	Training Cost (FLOPs) EN-FR
ByteNet [18]	23.75			
Deep-Att + PosUnk [39]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [38]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [9]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [32]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [39]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [38]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [9]	26.36	**41.29**	$7.7 \cdot 10^{19}$	$1.2 \cdot 10^{21}$
Transformer (base model)	27.3	38.1	$\mathbf{3.3 \cdot 10^{18}}$	
Transformer (big)	**28.4**	**41.8**	$2.3 \cdot 10^{19}$	

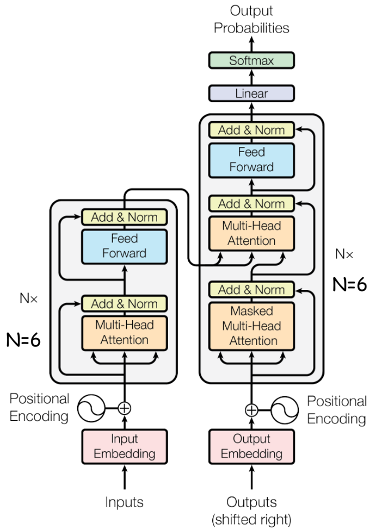


Figure 1: The Transformer - model architecture.

23

23

Attention is All You Need

References: Stanford CS224N, Lecture 8: https://web.stanford.edu/class/cs224n/slides/cs224n-2024-lecture08-transformers.pdf
https://peterbloem.nl/blog/transformers

Document Generation

Model	Test perplexity	ROUGE-L
seq2seq-attention, $L = 500$	5.04952	12.7
Transformer-ED, $L = 500$	2.46645	34.2
Transformer-D, $L = 4000$	2.22216	33.6
Transformer-DMCA, no MoE-layer, $L = 11000$	2.05159	36.2
Transformer-DMCA, MoE-128, $L = 11000$	1.92871	37.9
Transformer-DMCA, MoE-256, $L = 7500$	1.90325	38.8

The old standard

Transformers all the way down.

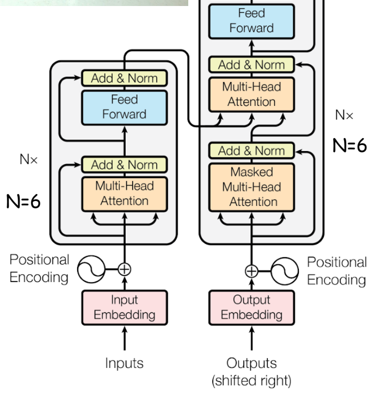


Figure 1: The Transformer - model architecture.

The parallelizability of transformer enables large-scale pre-training!

24

24

Application of Transformer: Remote Work

Remote Work across Jobs, Companies, and Space

Stephen Hansen, Peter John Lambert, Nicholas Bloom,
Steven J. Davis, Raffaella Sadun & Bledi Taska

WORKING PAPER 31007 **DOI** 10.3386/w31007 **ISSUE DATE** March 2023

The pandemic catalyzed an enduring shift to remote work. To measure and characterize this shift, we examine more than 250 million job vacancy postings across five English-speaking countries. Our measurements rely on a state-of-the-art language-processing framework that we fit, test, and refine using 30,000 human classifications. We achieve 99% accuracy in flagging job postings that advertise hybrid or fully remote work, greatly outperforming dictionary methods and also outperforming other machine-learning methods. From 2019 to early 2023, the share of postings that say new employees can work remotely one or more days per week rose more than three-fold in the U.S and by a factor of five or more in Australia, Canada, New Zealand and the U.K. These developments are highly non-uniform across and within cities, industries, occupations, and companies. Even when zooming in on employers in the same industry competing for talent in the same occupations, we find large differences in the share of job postings that explicitly offer remote work.

- Transformer is not that frequently used in business research, (probably) because of its technical barriers.
- Use DistilBERT pre-trained on 1M text chunks of job vacancy postings to measure the Work-from-homeness of the 250 M jobs (Work from Home Algorithmic Measure), achieving 99% accuracy that outperforms dictionary-based methods.
- The number of WFM jobs has risen significantly since 2019 and it differs w.r.t. different industries.

Remote work across jobs, companies, and space
S Hansen, PJ Lambert, N Bloom, SJ Davis, R Sadun… - 2023 - nber.org
The pandemic catalyzed an enduring shift to remote work. To measure and characterize this shift, we examine more than 250 million job vacancy postings across five English-speaking countries. Our measurements rely on a state-of-the-art language-processing framework that we fit, test, and refine using 30,000 human classifications. We achieve 99% accuracy in flagging job postings that advertise hybrid or fully remote work, greatly outperforming dictionary methods and also outperforming other machine-learning methods. From 2019 to …
☆ Save 99 Cite Cited by 36 Related articles All 20 versions

25

25