ATTENTION IS ALL YOU NEED

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.

Presenter: Omar Iken

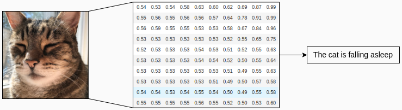
Natural Language Processing
Master 2 Data Science

March 2, 2022

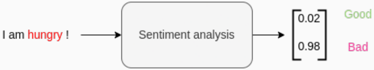
Université de Lille

Sequence models

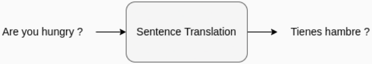
- Vector to Sequence (e.g. Image captioning)



- Sequence to Vector (e.g. Sentiments analysis)



- Sequence to Sequence (e.g. Language translation)



Outline

Background

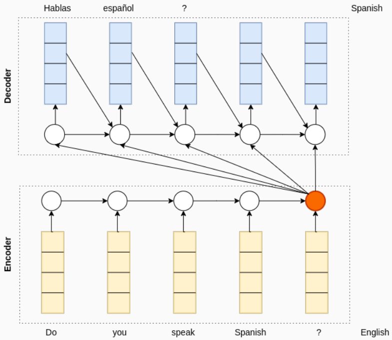


Figure 1: Encoder/Decoder

Drawbacks

- Slow to train.
- Sequential process $\implies$ Precludes parallelism.
- Long sequences $\implies$ Vanishing exploding gradients.
- Long-term dependency problem.
- Unidirectional (Most of them) $\implies$ Process text from left to right.

Introducing Parallelism

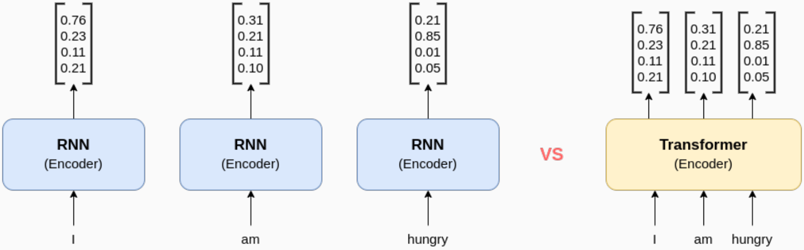


Figure 2: RNNs VS. Transformer

Transformer architecture

Overview

Main parts (Encoder / Decoder)

1. Input Embedding
2. Positional Encoding
3. Attention mechanism
4. Multi Head Attention

Applications

- NLP Tasks:
  - BERT[Devlin et al. 2018]
  - GPT-2 [Radford et al. 2019]
- Vision Tasks:
  - Image classification
  - Object Detection
  - Video Instance Segmentation
- ...

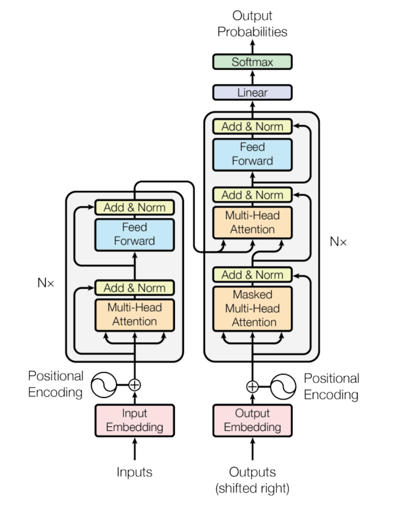


Figure 3: Transformer architecture [1].

Embedding / Encoding

Input Embedding.
Words that have the same meaning will be close in terms of Euclidean distance.

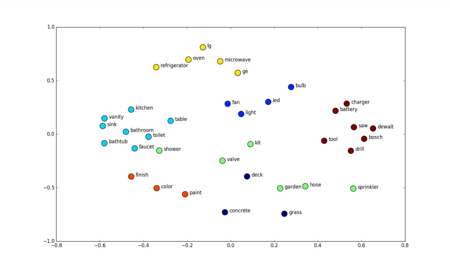


Figure 4: Word Embedding example.
[https://easyai.tech/en/ai-definition/word-embedding/]

Positional Encoding.
The position of a word plays a determining role in understanding the sequence $\Longrightarrow$ add positional information about the word within the sequence in the vector.

$$PE(pos, 2i) = sin\left(\frac{pos}{1000^{2i/d_{model}}}\right)$$
$$PE(pos, 2i+1) = cos\left(\frac{pos}{1000^{2i/d_{model}}}\right)$$

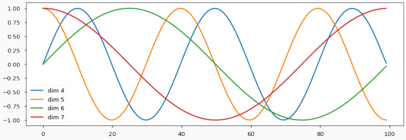


Figure 5: the wavelength w.r.t dimension.
[http://nlp.seas.harvard.edu/images/the-annotated-transformer_49_0.png]

Attention mechanism

Scaled Dot-Product Attention

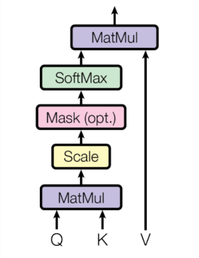


Figure 6: Scaled dot-product attention [1].

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

- Q[query]: a vector word.
- K[keys]: all other words in the sentence.
- V[value]: the vector of the word.

Multi Head Attention

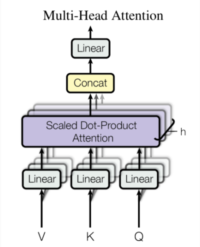


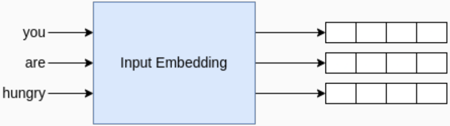
Figure 7: Multi-head attention [1].

$$MH(Q, K, V) = concat(h_1, \ldots, h_h)W^o$$
$$h_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$$

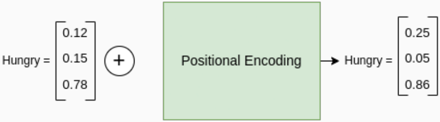
- Projection of Q, K and V in Linear Spaces.
- 8 projections of size 64 ($8 * 64 = 512$).

Encoder

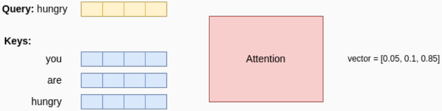
1. Input Embedding: convert a sequence of tokens to a sequence of vectors



2. Positional Encoding: add position information in each word vector.



3. Apply MultiHead attention.



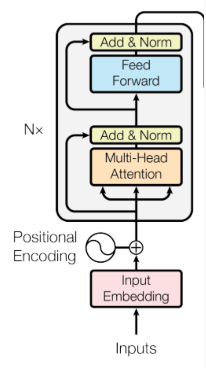
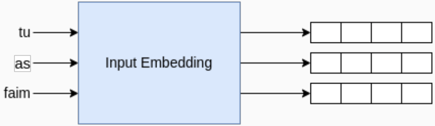


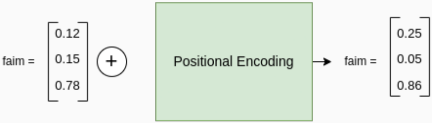
Figure 8: Encoder

Decoder

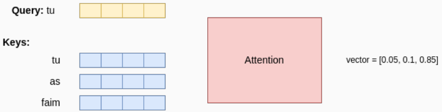
1. Word Embedding: convert a sequence of tokens to a sequence of vectors



2. Positional Encoding: add position information in each word vector.



3. Apply MultiHead attention.



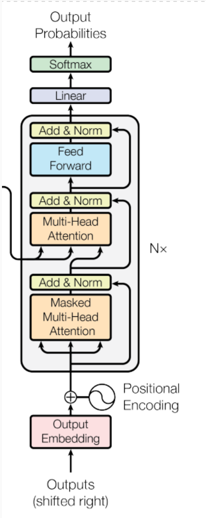
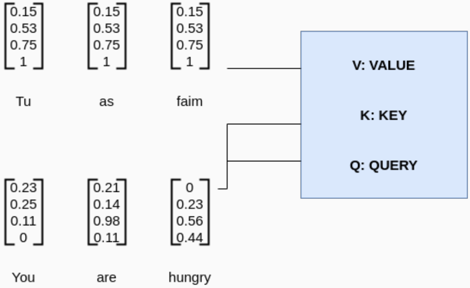


Figure 9: Decoder

Decoder, cond't

4. Feed Forward
5. Multi Head attention with encoder output

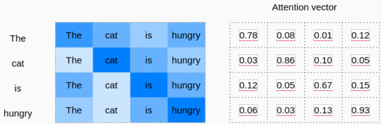


- Feed Forward (again)
- Linear + softmax

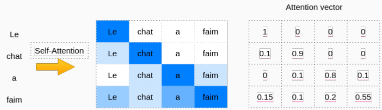
Attention Mechanism

Self-Attention

Encoder



Decoder



Why Self-Attention ?

Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(log_k(n))$
Self-Attention (restricted)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

Figure 10: Maximum path lengths, per-layer complexity and minimum number of sequential operations for different layer types [1].

Experimental Results

Results

Training Data and Batching / Hardware and Schedule

- WMT 2014 English-German dataset: 4.5 million sentence pairs
- WMT 2014 English-French dataset: 36M sentences
- 8 NVIDIA P100 GPUs
- 12 hours / 3.5 days

Experimental Results

Model	BLEU		Training Cost (FLOPs)	
	EN-DE	EN-FR	EN-DE	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**	$2.3 \cdot 10^{19}$	

Paper review

Paper Review

About the paper

- Attention Is All You Need
- Ashish Vaswani et al. 2017
- NIPS 2017

Merits of the paper

- The paper reads well and is easy to follow !
- Entirely novel architecture without recurrence or convolutions.
- New state-of-the-art results on standard WMT datasets.

Weaknesses

- Architectural details lack mathematical definitions (e.g., multi-head attention).
- Model contains lots of hyper-parameters, but not/not well discussed.

References

[1] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).