Transformer model

SOTA: BERT

- BERT achieved SOTA in SQuAD1.1
  - SQuAD (Stanford Question Answering Dataset)

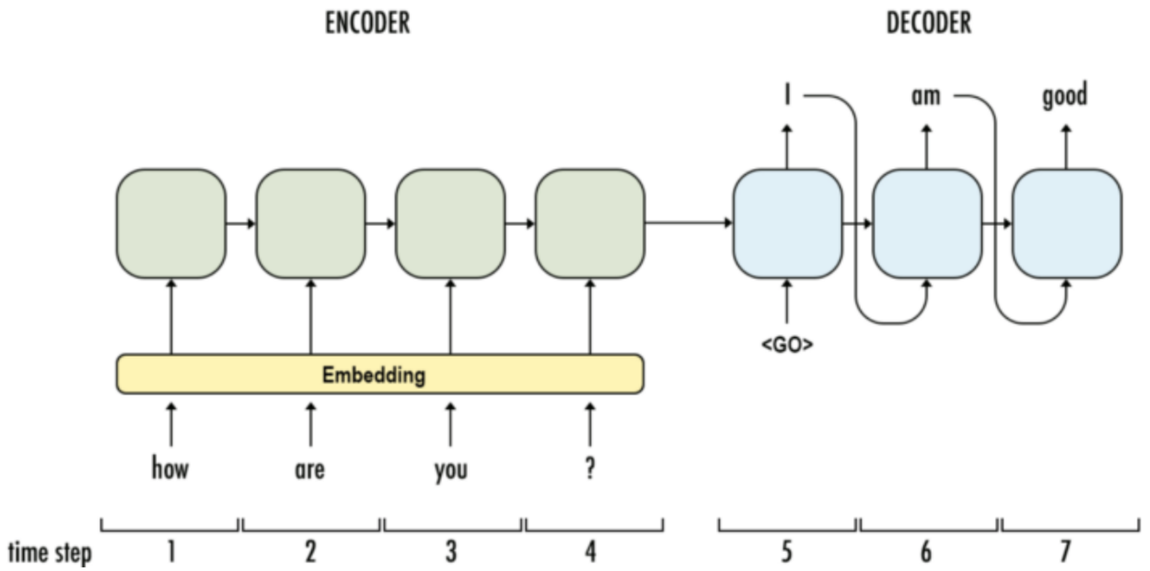
SQuAD1.1 Leaderboard

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 Sep 09, 2018	nlnet (ensemble) *Microsoft Research Asia*	85.356	91.202
3 Jul 11, 2018	QANet (ensemble) *Google Brain & CMU*	84.454	90.490

seq2seq task

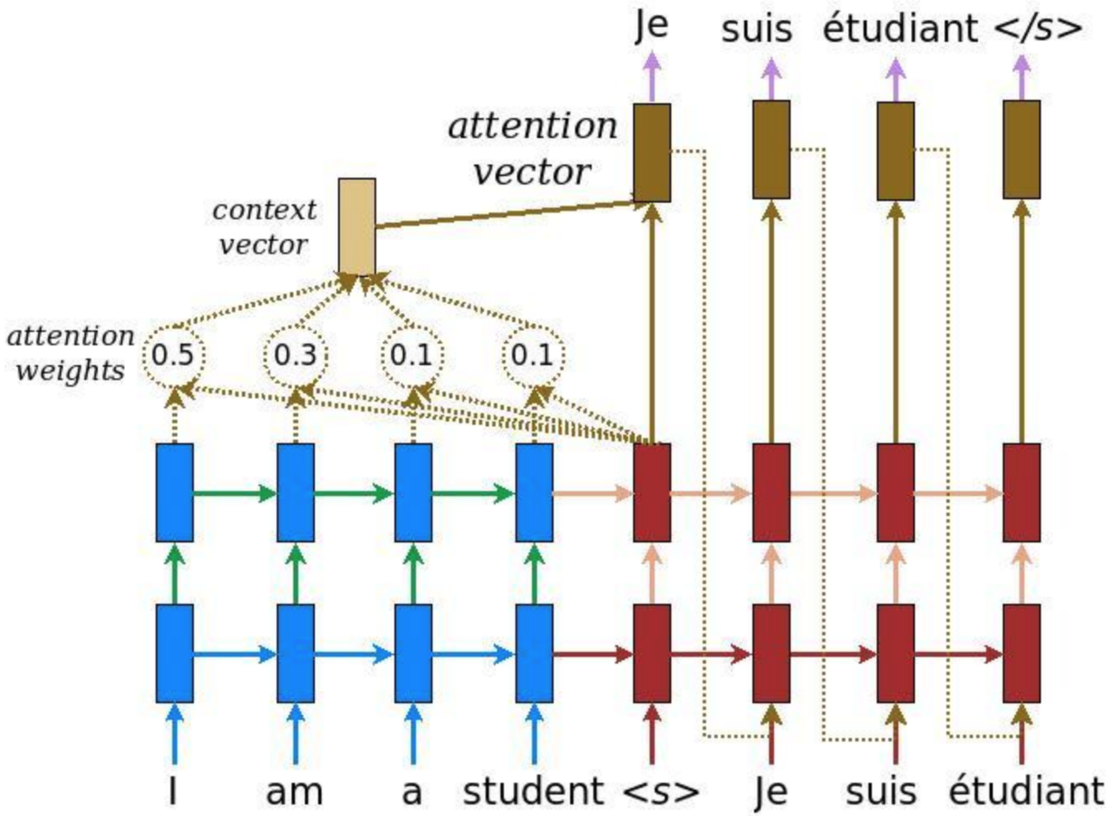
- Seq2seq model is composed of encoder and a decoder

- Encoder
  - Compiles the information it captures into a vector (context vector)
- Decoder
  - After processing the entire input sequence, the encoder sends the context to the decoder
  - Decoder produce output sequence item by item



Introduction

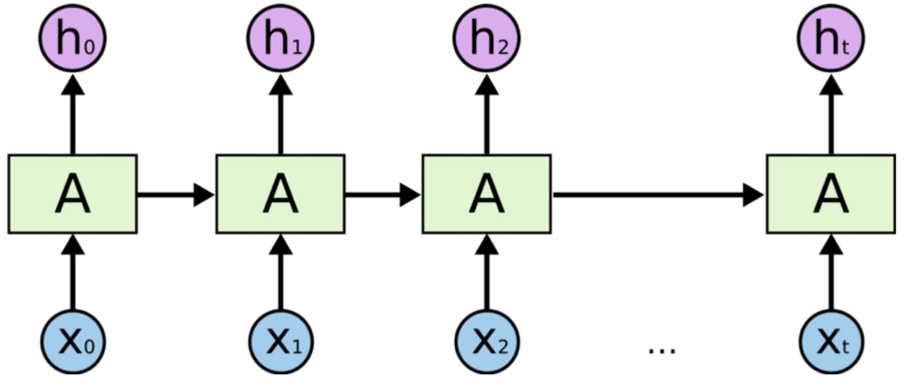
- Attention mechanism (Encoder-Decoder Attention)
  - Mostly used in conjunction with a recurrent network
  - Allow the decoder to “attend” to different parts of source sentence at each step of output

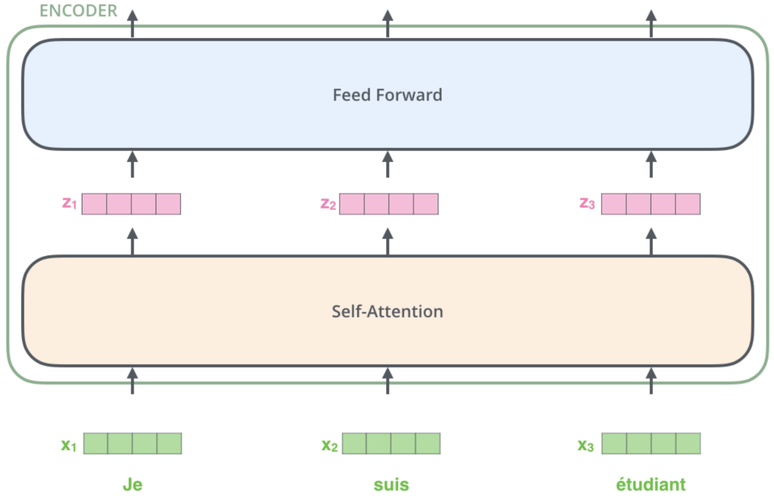


Motivation

- Limitation of RNN
  - RNN handle sequences word-by-word sequentially which is an **obstacle to parallelize**
  - **Long range dependencies** problematic

- Transformer
  - Allows for significantly more parallelization **(no RNN & CNN)**
  - Rely entirely on attention mechanism to draw global dependencies between input and output **(only self-attention)**





Model architecture

- Transformer follows encoder-decoder structure
- Encoder & Decoder composed of stack of 6 identical layers

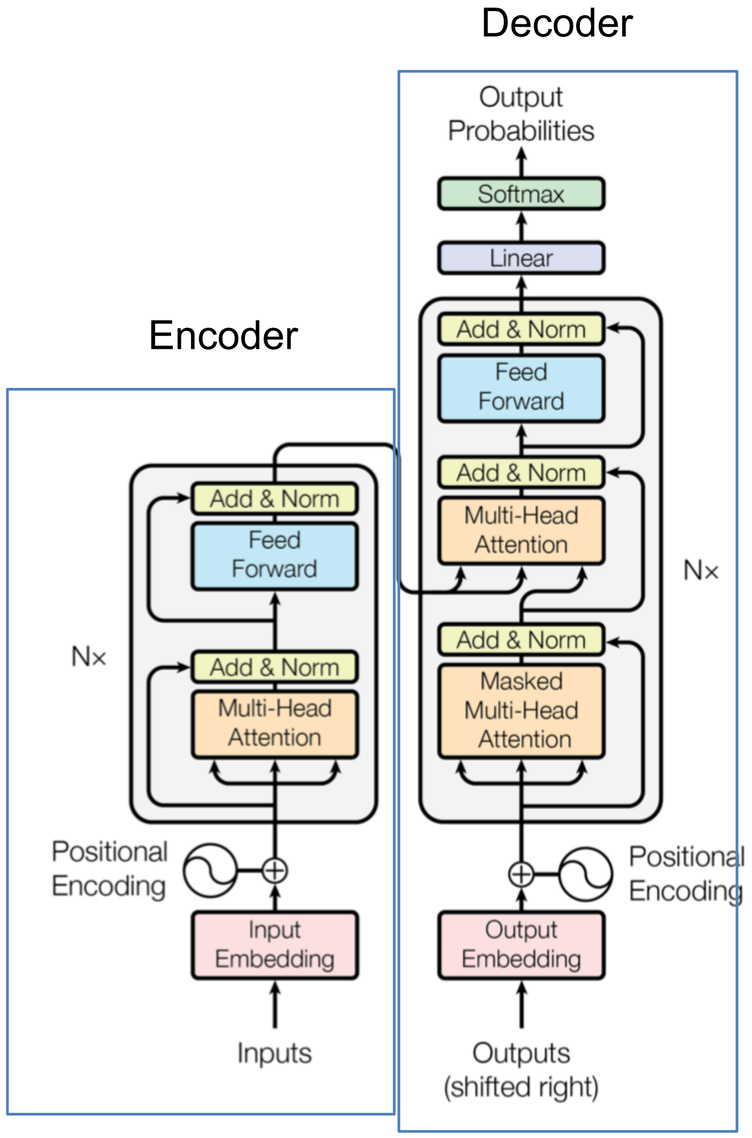


Figure 1: The Transformer - model architecture.

Encoder

- Encoder is composed of two sub-layers
  - Multi-head self-attention layer
  - Fully connected feed-forward network
- Residual connection & layer norm applied

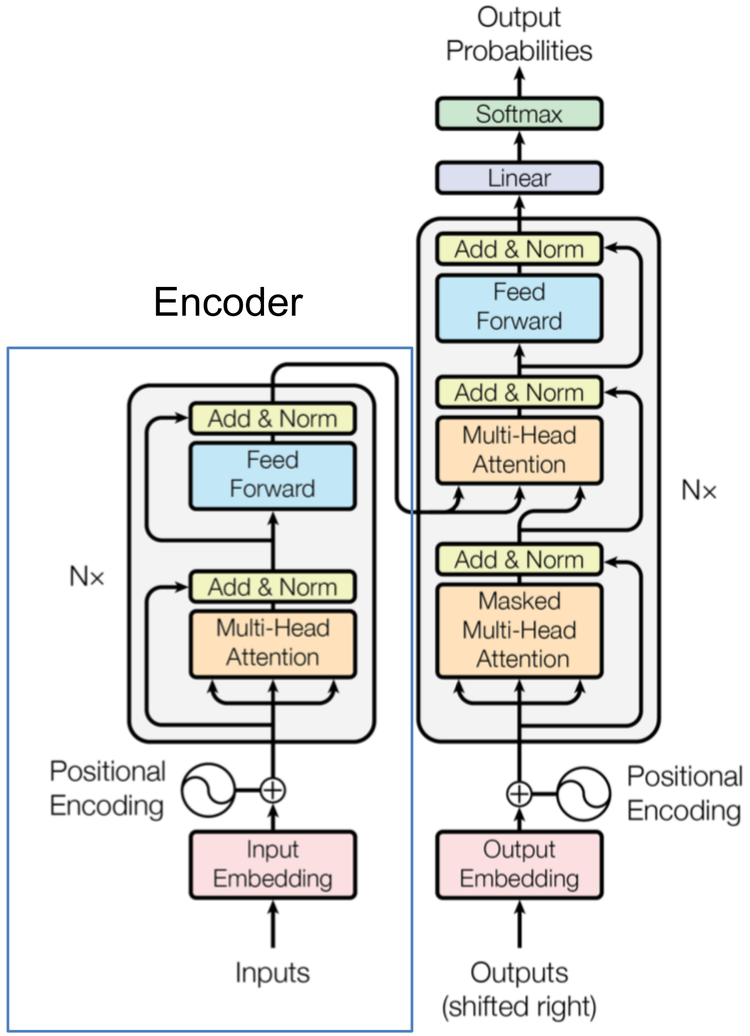
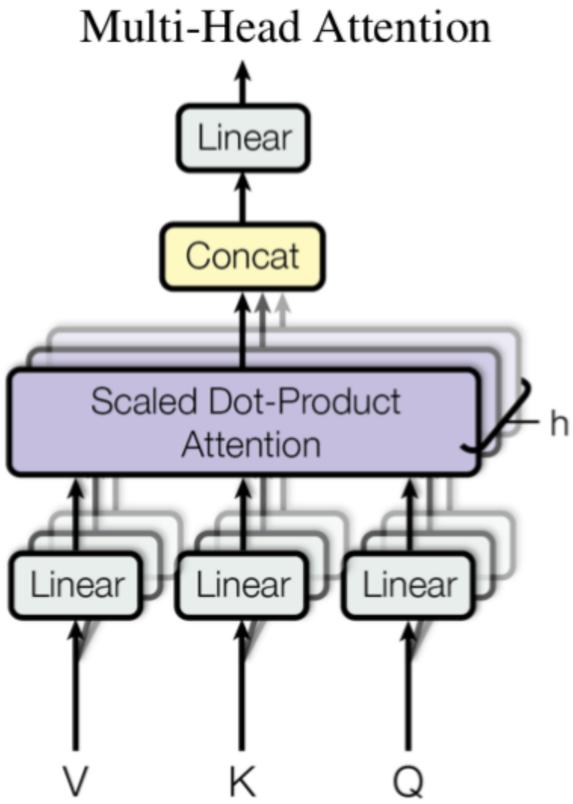


Figure 1: The Transformer - model architecture.

Multi-Head attention

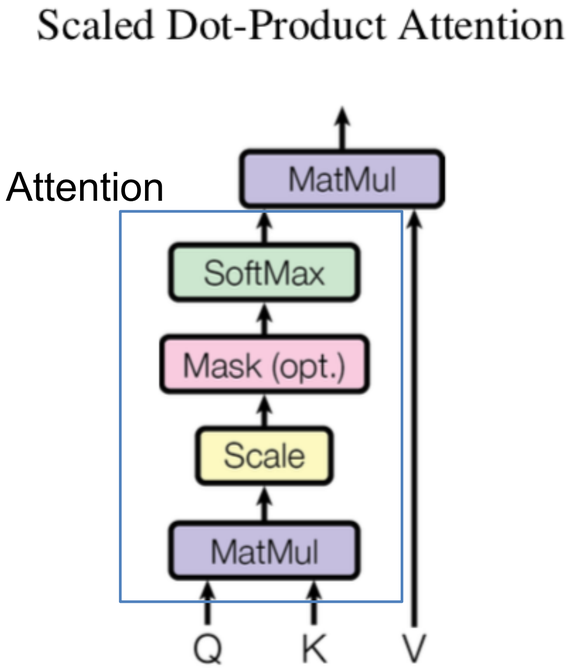
- Instead of performing single attention function
- Beneficial to linearly project the queries, keys, and values h times with different, learned linear projections

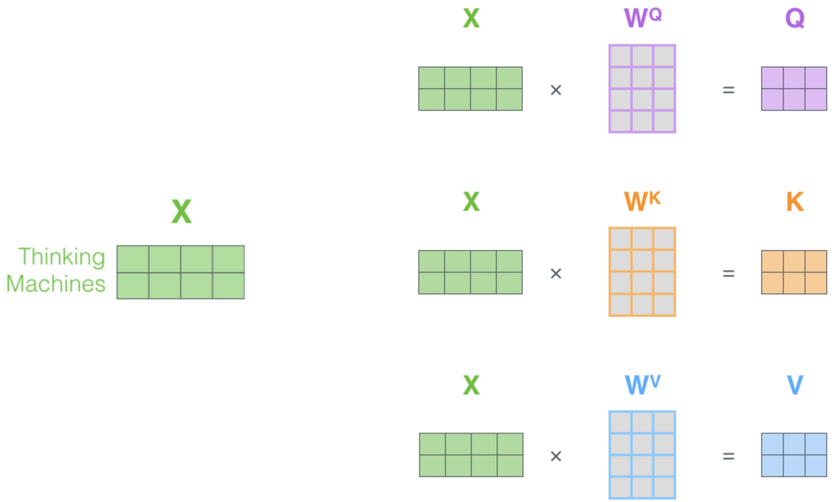
$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_\text{h})W^O$$
$$\text{where head}_\text{i} = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$



Scaled Dot-Product Attention

- Input consists of queries and keys and values
- Attention: Dot product of query with all keys
- Output: Weighted sum of the values

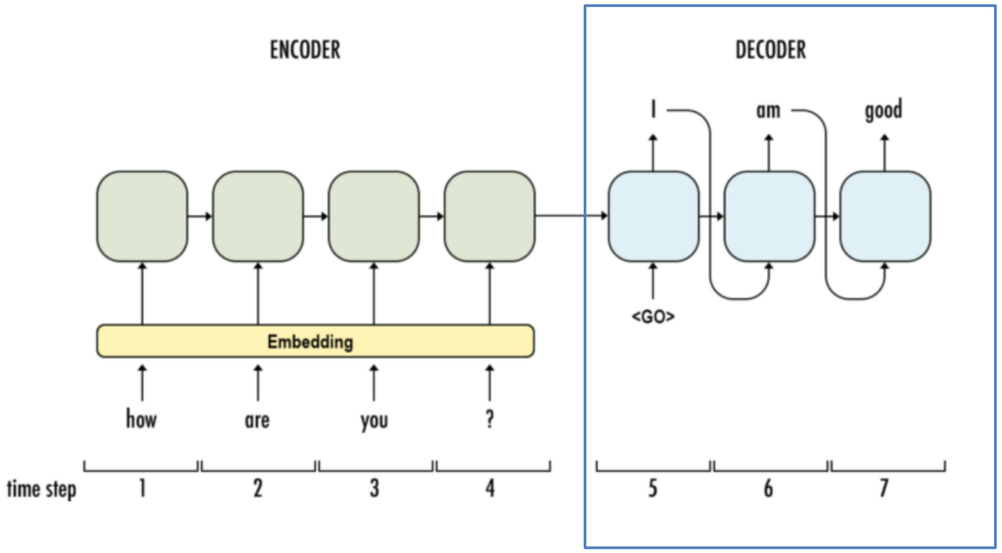




$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

Decoder

- Decoder inserts third sub-layer (encoder-decoder attention)
- Residual connection & layer norm applied
- Masked Multi-Head attention
  - Prevent positions from attending to subsequent positions



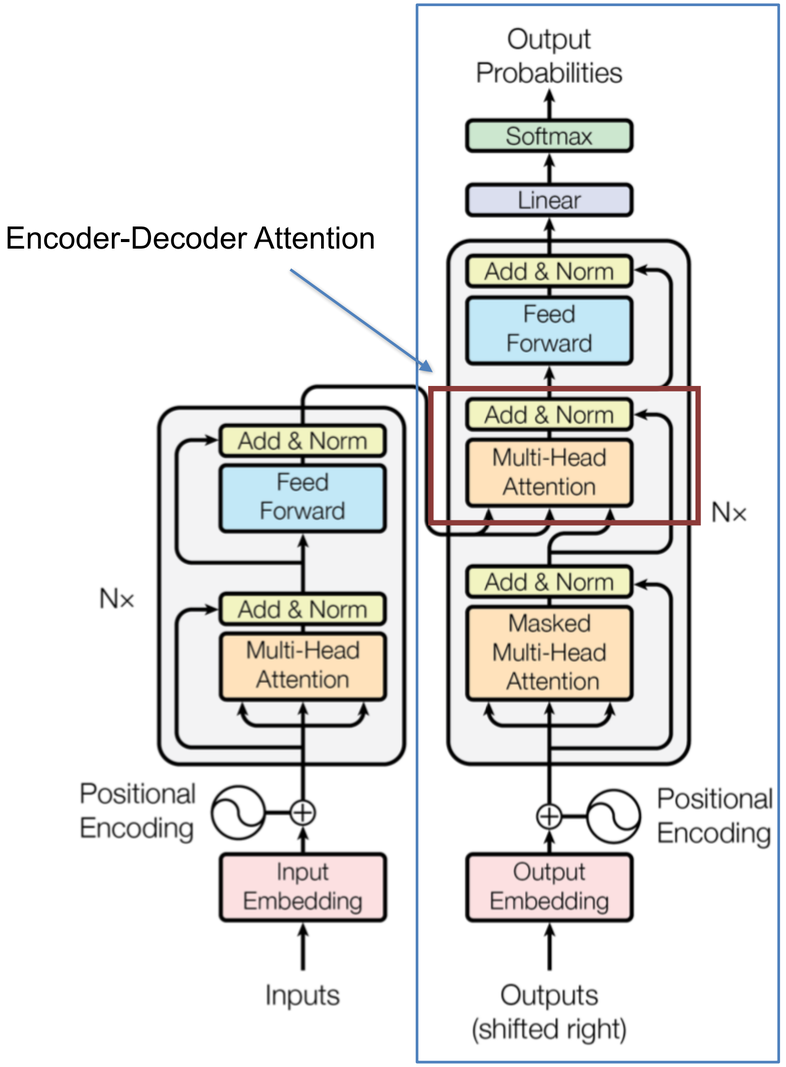
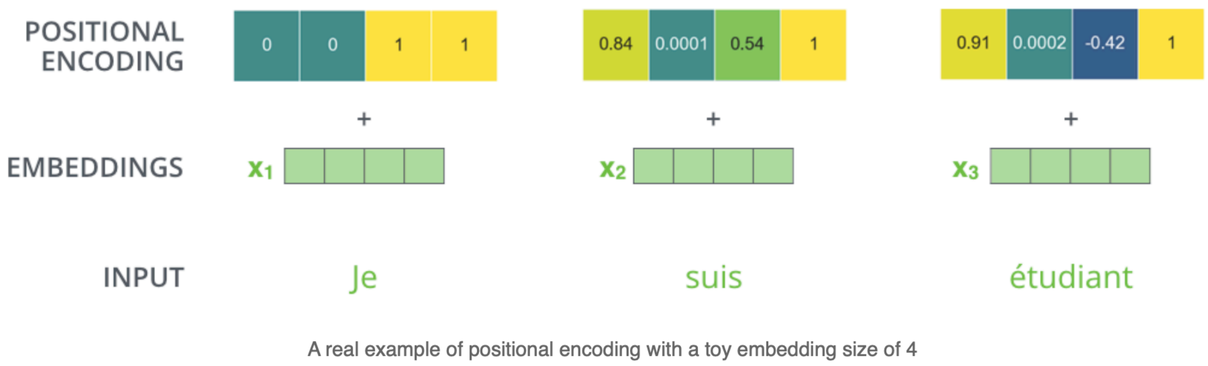


Figure 1: The Transformer - model architecture.

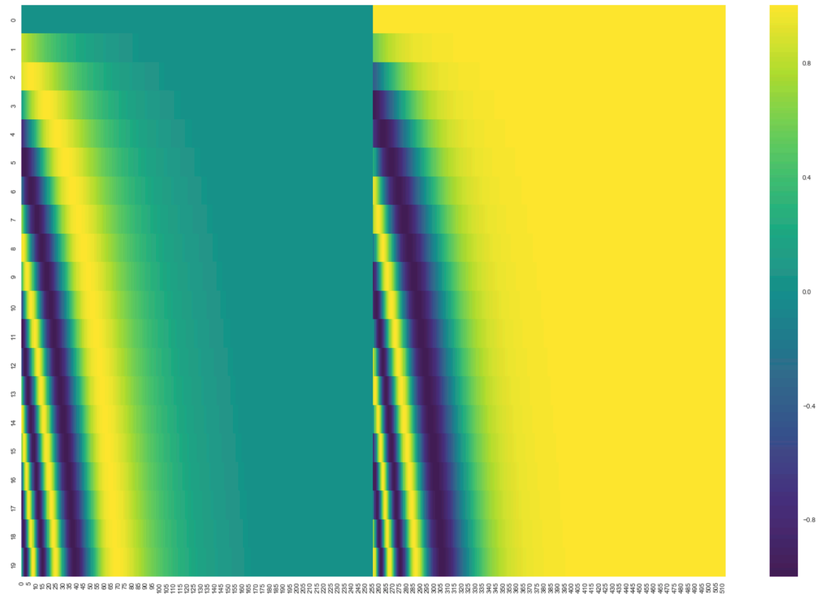
Positional Encoding

- Since the model contains no recurrence and no convolution, in order for the model to make use of the order of the sequence
- We must inject some information about the relative or absolute position of the tokens in the sequence
- This paper uses sine and cosine functions of different frequencies

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$
$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$



A real example of positional encoding with a toy embedding size of 4



Why Self-Attention

- Total computational complexity per layer
- Amount of computation that can be parallelized
- Maximum path length between long-range dependencies in the network
  - Maximum path length: maximum (any combination of positions in the sequences)

Table 1: Maximum path lengths, per-layer complexity and minimum number of sequential operations for different layer types. n is the sequence length, d is the representation dimension, k is the kernel size of convolutions and r the size of the neighborhood in restricted 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)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

Result

- Machine Translation

Table 2: The Transformer achieves better BLEU scores than previous state-of-the-art models on the English-to-German and English-to-French newstest2014 tests at a fraction of the training cost.

Model	BLEU		Training Cost (FLOPs)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [15]	23.75			
Deep-Att + PosUnk [32]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [31]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [8]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [26]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [32]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [31]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [8]	26.36	**41.29**	$7.7 \cdot 10^{19}$	$1.2 \cdot 10^{21}$
Transformer (base model)	27.3	38.1	**$3.3 \cdot 10^{18}$**	
Transformer (big)	**28.4**	**41.0**	$2.3 \cdot 10^{19}$	

Variants of Transformer

- **Directional Self attention Network (DiSAN)**
- **Universal Transformer (google brain, deepmind)**
- BERT (Google AI Language)

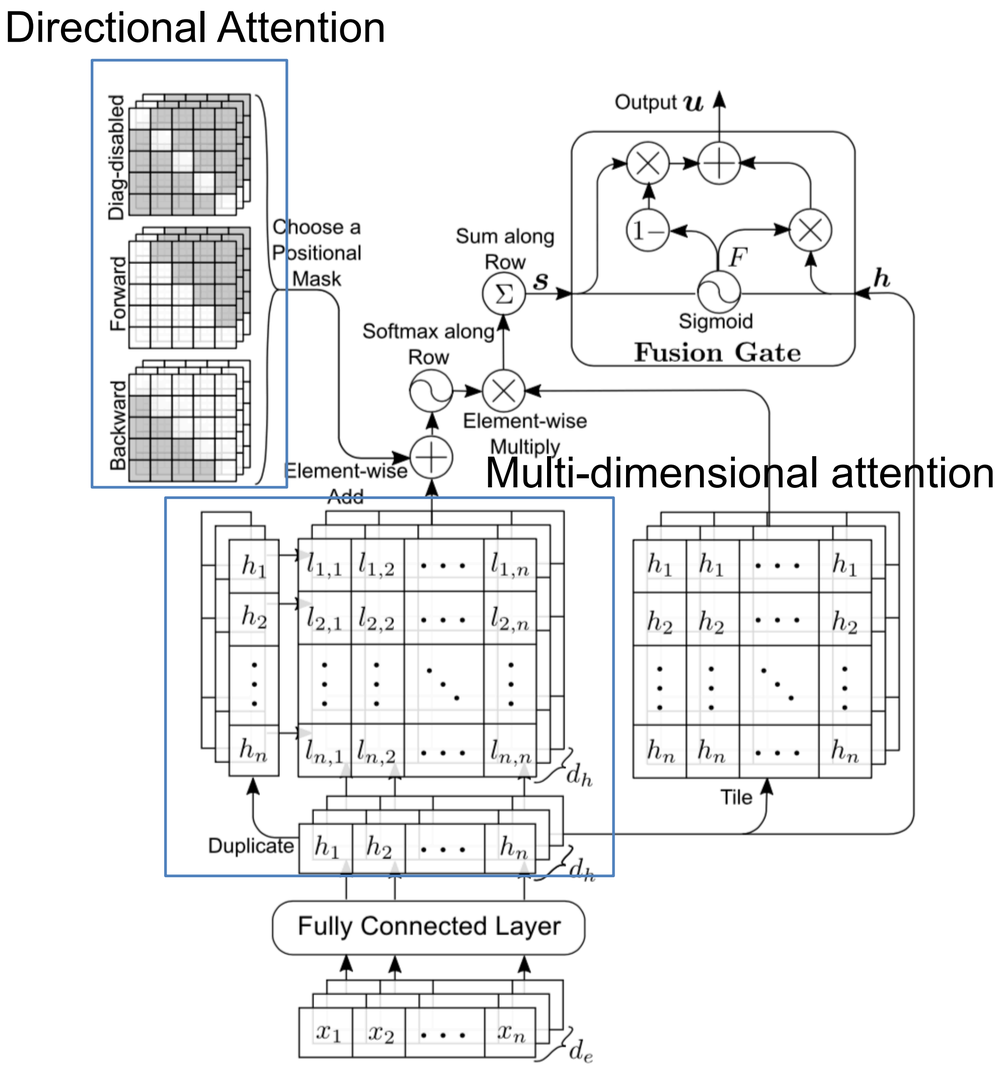
Directional Self-Attention Network (DiSAN)

Motivation

- Limitation of Transformer
  - Disadvantage of most attention mechanisms is that temporal order information is lost
    - Although, positional encoding is applied, how to model order information is an open problem
  - Only for NMT task

DiSAN

- Input
  - token embeddings x
- Fully connected layer
- Attention
  - Multi-dimensional attention
  - Directional attention
- Fusion gate to combine the output and input of the attention block



Attention mechanism

- Multiplicative Attention (**Attention is all you need**)
  – Inner product or cosine similarity

$$f(x_i, q) = \left\langle W^{(1)} x_i, W^{(2)} q \right\rangle .$$

- Additive Attention **(paper focus)**

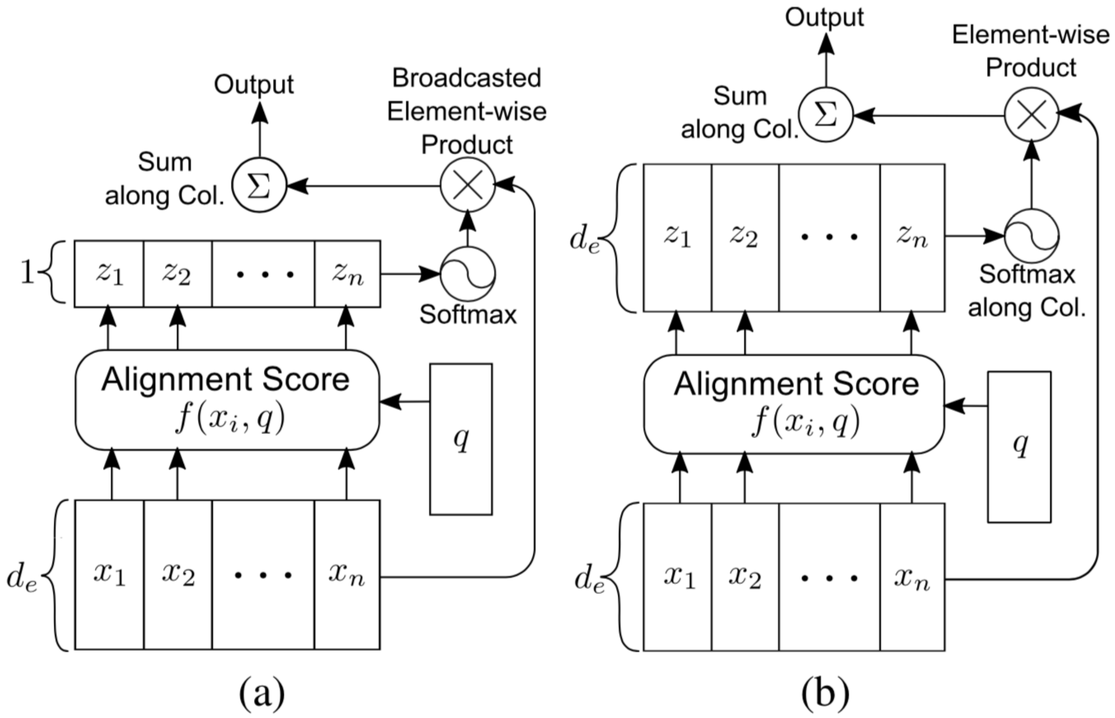
$$f(x_i, q) = w^T \sigma(W^{(1)} x_i + W^{(2)} q),$$

Multi-dimensional Attention

- Natural extension of attention at the feature-level
- Instead of a single scalar score
- Computes a feature-wise score vector by replacing weight vector w to Matrix W

(a) $f(x_i, q) = w^T \sigma(W^{(1)} x_i + W^{(2)} q),$

(b) $f(x_i, q) = W^T \sigma\left(W^{(1)} x_i + W^{(2)} q\right),$



Directional Self-Attention

- Apply positional mask so the attention between two elements can be asymmetric
- Disable attention by setting the element to $-\infty$
- As softmax is applied, the attention leads to zero

$$f(h_i, h_j) = c \cdot \tanh\left([W^{(1)}h_i + W^{(2)}h_j + b^{(1)}]/c\right) + M_{ij}\mathbf{1}.$$

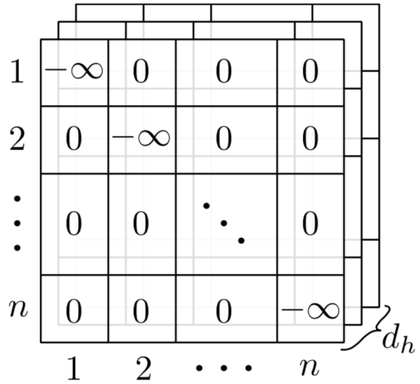
- Diagonal-disabled
  - Disable attention of each token itself

$$M_{ij}^{diag} = \begin{cases} 0, & i \neq j \\ -\infty, & i = j \end{cases}$$

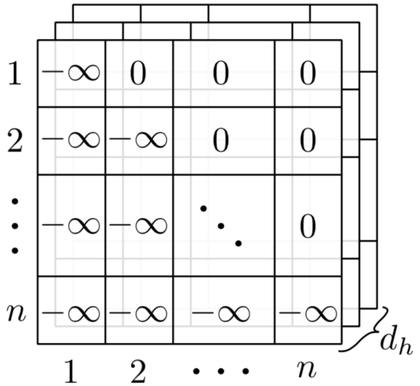
- Use mask to encode temporal order (rather than positional encoding)
  - Forward mask & Backward mask

$$M_{ij}^{fw} = \begin{cases} 0, & i < j \\ -\infty, & otherwise \end{cases}$$

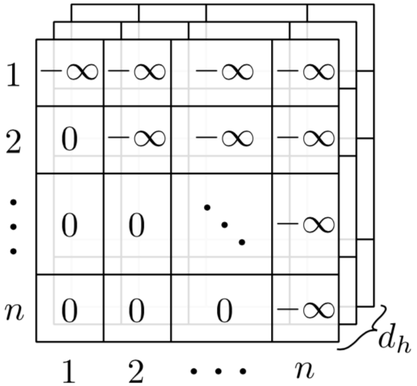
$$M_{ij}^{bw} = \begin{cases} 0, & i > j \\ -\infty, & otherwise \end{cases}$$



(a) Diag-disabled mask



(b) Forward mask



(c) Backward mask

Result

Natural Language Inference

Model Name	$\|\theta\|$	T(s)/epoch	Train Accu(%)	Test Accu(%)
Unlexicalized features (Bowman et al. 2015)			49.4	50.4
+ Unigram and bigram features (Bowman et al. 2015)			99.7	78.2
100D LSTM encoders (Bowman et al. 2015)	0.2m		84.8	77.6
300D LSTM encoders (Bowman et al. 2016)	3.0m		83.9	80.6
1024D GRU encoders (Vendrov et al. 2016)	15m		98.8	81.4
300D Tree-based CNN encoders (Mou et al. 2016)	3.5m		83.3	82.1
300D SPINN-PI encoders (Bowman et al. 2016)	3.7m		89.2	83.2
600D Bi-LSTM encoders (Liu et al. 2016)	2.0m		86.4	83.3
300D NTI-SLSTM-LSTM encoders (Munkhdalai and Yu 2017b)	4.0m		82.5	83.4
600D Bi-LSTM encoders+intra-attention (Liu et al. 2016)	2.8m		84.5	84.2
300D NSE encoders (Munkhdalai and Yu 2017a)	3.0m		86.2	84.6
Word Embedding with additive attention	0.45m	216	82.39	79.81
Word Embedding with s2t self-attention	0.54m	261	86.22	83.12
Multi-head with s2t self-attention	1.98m	345	89.58	84.17
Bi-LSTM with s2t self-attention	2.88m	2080	90.39	84.98
DiSAN without directions	2.35m	592	90.18	84.66
Directional self-attention network (DiSAN)	2.35m	587	91.08	**85.62**

Transformer

Text	Judgments	Hypothesis
A man inspects the uniform of a figure in some East Asian country.	contradiction C C C C C	The man is sleeping
An older and younger man smiling.	neutral N N E N N	Two men are smiling and laughing at the cats playing on the floor.
A black race car starts up in front of a crowd of people.	contradiction C C C C C	A man is driving down a lonely road.
A soccer game with multiple males playing.	entailment E E E E E	Some men are playing a sport.
A smiling costumed woman is holding an umbrella.	neutral N N E C N	A happy woman in a fairy costume holds an umbrella.

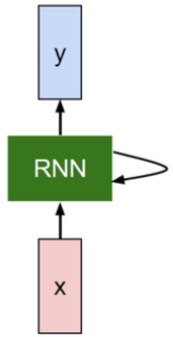
5.2 Sentiment Analysis

Model	Test Accu
MV-RNN (Socher et al. 2013)	44.4
RNTN (Socher et al. 2013)	45.7
Bi-LSTM (Li et al. 2015)	49.8
Tree-LSTM (Tai, Socher, and Manning 2015)	51.0
CNN-non-static (Kim 2014)	48.0
CNN-Tensor (Lei, Barzilay, and Jaakkola 2015)	51.2
NCSL (Teng, Vo, and Zhang 2016)	51.1
LR-Bi-LSTM (Qian, Huang, and Zhu 2017)	50.6
Word Embedding with additive attention	47.47
Word Embedding with s2t self-attention	48.87
Multi-head with s2t self-attention	49.14
Bi-LSTM with s2t self-attention	49.95
DiSAN without directions	49.41
DiSAN	**51.72**

- Natural Language Inference
  - Reason the semantic relationship between a premise sentence and a corresponding hypothesis sentence
    - Entailment, neutral, contradiction
- Sentiment analysis
  - Analyze sentiment of a sentence or paragraph
  - Negative, Positive, Neutral

Universal Transformer

- Limitation of transformer
  - Transformer does not perform well on certain tasks
    - Smaller and more structured language understanding tasks or even simple algorithmic tasks
      - e.g., copying a string (abc => abcabc)
  - Transformer does not generalize well to input lengths not encountered during training
- RNN's learning iterative or recursive (inductive bias) may be crucial for several algorithmic and language understanding tasks
  - Inductive bias: Assumption that learner uses to predict outputs given inputs that it has not encountered



<기본구조>

Universal transformer

- Built parallel structure of transformer to retain fast training speed
- But, replaced transformer's fixed stack of different transformation functions
  - single, parallel-in-time recurrent transformation function

- RNN processes a sequence symbol-by-symbol
- Universal Transformer processes all symbols at the same time but then refine its interpretation of every symbol in parallel over a variable number of recurrence mechanism

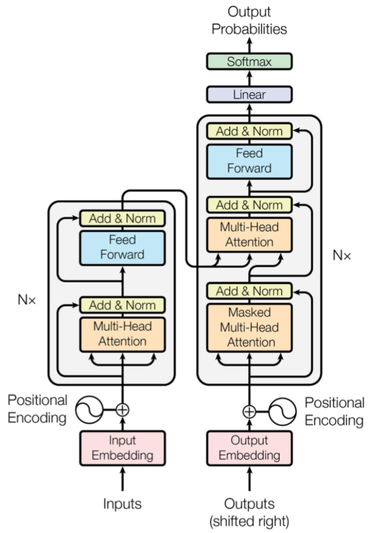
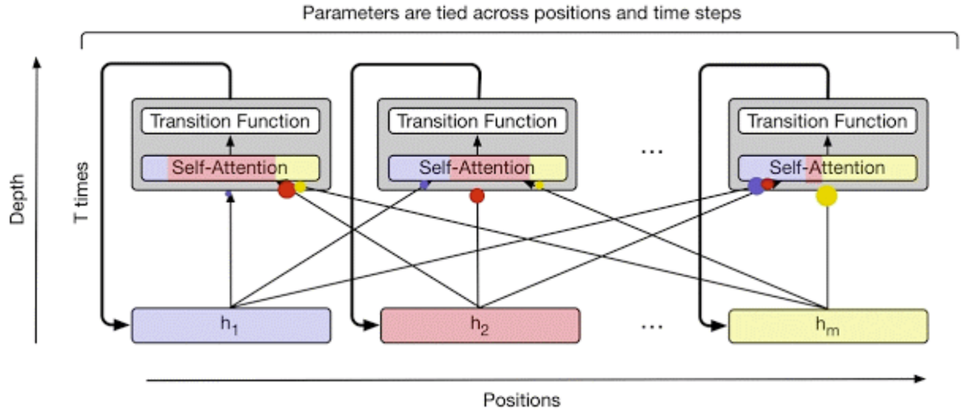


Figure 1: The Transformer - model architecture.



Universal transformer

- Applied **adaptive computation mechanism (**https://arxiv.org/abs/1603.08983**)**
  - **Allocate more processing steps** to symbols that are **more ambiguous** or require more computation

    - More context is required to infer the most likely meaning of the word “bank” compared to the less ambiguous meaning of “I” or “river”
  - Standard transformer: same amount of computation is applied unconditionally to each word

- Universal transformer’s adaptive mechanism allows the model to spend increased computation only on the more ambiguous words and fewer steps on less ambiguous words

Result

Model	BLEU
Universal Transformer *small*	26.8
Transformer *base* [31]	28.0
Weighted Transformer *base* [1]	28.4
Universal Transformer *base*	**28.9**

Table 7: Machine translation results on the WMT14 En-De translation task trained on 8xP100 GPUs in comparable training setups. All *base* results have the same number of parameters.

Model	Copy		Reverse		Addition	
	char-acc	seq-acc	char-acc	seq-acc	char-acc	seq-acc
LSTM	0.45	0.09	0.66	0.11	0.08	0.0
Transformer	0.53	0.03	0.13	0.06	0.07	0.0
Universal Transformer	0.91	0.35	0.96	0.46	0.34	0.02
Neural GPU*	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**

Table 4: Accuracy (higher better) on the algorithmic tasks, trained on decimal strings of length 40 and evaluated on length 400 from [17]. *Note that the Neural GPU was trained with a special curriculum to obtain the perfect result, while other models are trained without any curriculum.

Conclusion

- Transformer variants
  - BERT: Pre-training of Deep Bidirectional transformers for Language Understanding (2018 Oct, Google Language AI)
  - Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context (2019, Google Brain & AI)