Attention, Transformers and BERT

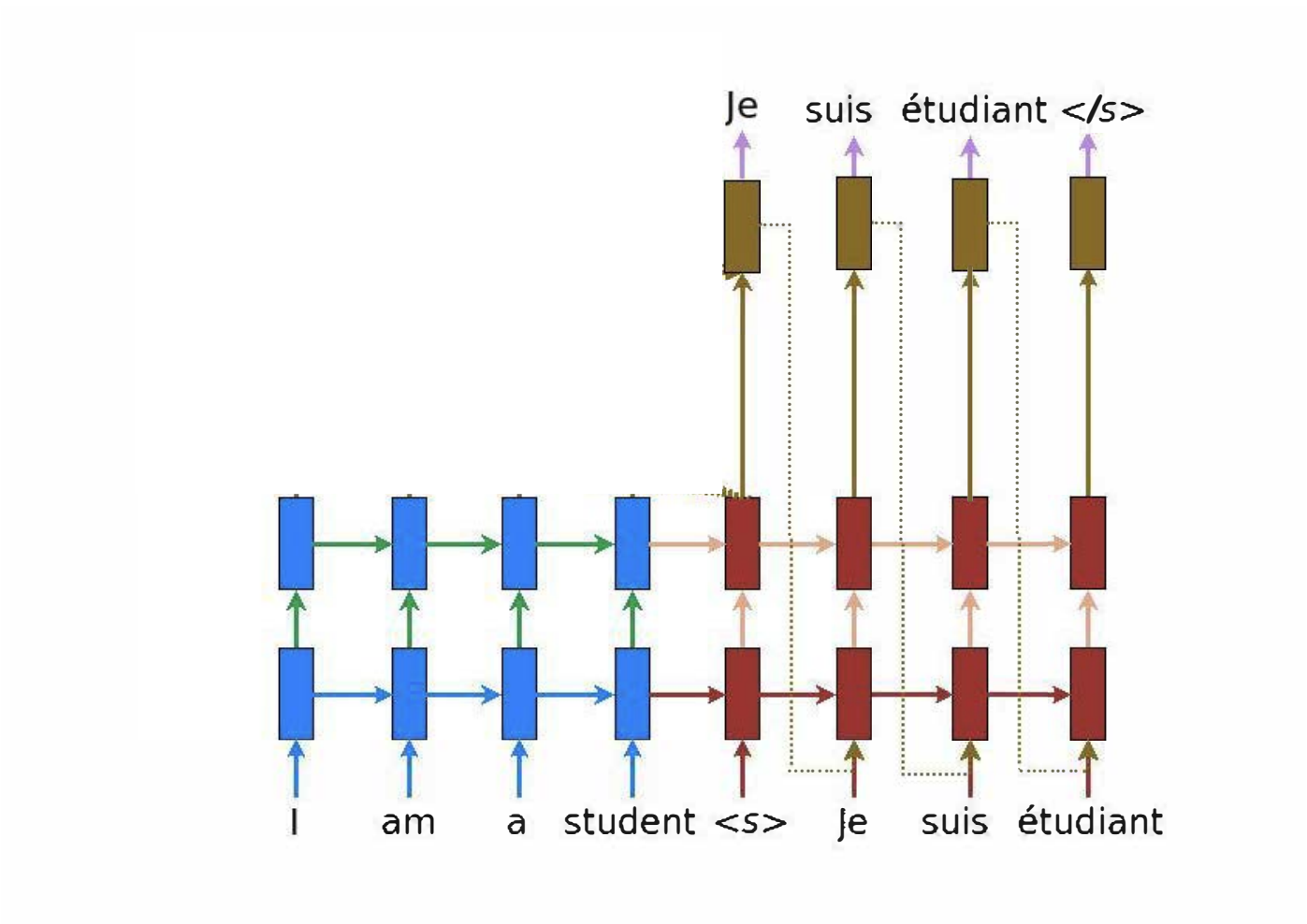
Eric Gaussier

Univ. Grenoble Alpes
UFR-IM^{2}AG
eric.gaussier@imag.fr

Material used in this course

- *Bahdanau et al., "Neural Machine Translation by Jointly Learning to Align and Translate". ICLR 2015*
- *Vaswani et al., "Attention Is All You Need". NIPS 2017*
- *Devlin et al., "BERT : Pre-training of Deep Bidirectional Transformers for Language Understanding". NAACL-HLT 2019*
- Pascal Poupart's online course : CS480/680 Lecture 19 : Attention and Transformer Networks (available on YouTube)

Modeling sequences with RNN



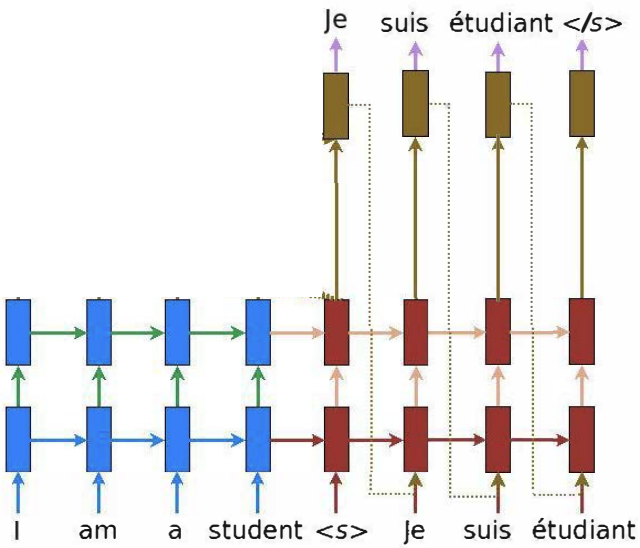
Attention in neural networks (1)

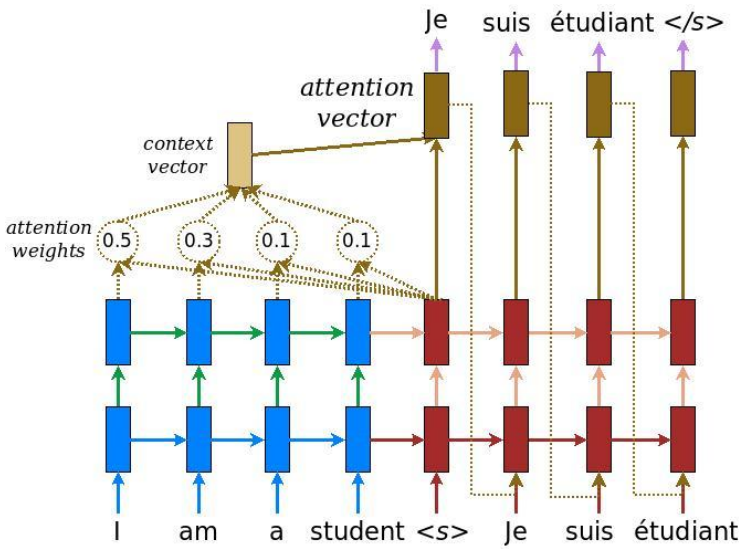
- 2014 : in Computer Vision, to highlight image parts that contribute to the output
- 2014-2015 : in Natural Language Processing (NLP), aligning words for machine translation[1]
- 2017 : transformer networks[2]

1. *Bahdanau et al. - Neural Machine Translation by Jointly Learning to Align and Translate*

2. *Vaswani et al. - Attention is All You Need*

Attention in neural networks (2)





Attention mechanism (1)

Mimics the retrieval of a value V_i for a query q based on a key K_i in a database

[picture 1]

$$attention(q, K, V) = \sum_i weight(q, K_i) \times V_i$$

Attention mechanism (2)

[picture 2]

Illustration : machine translation

- Query : hidden vector of given output word
- Keys=Values : hidden vectors of input words

Modeling sequences

Recurrent neural networks vs Transformers

1. Long range dependencies : difficult - easy
2. Gradient vanishing and exploding : yes - no
3. Number of training steps : large - small
4. Parallel computation : no - yes

Transformers

From *Vaswani et al.*

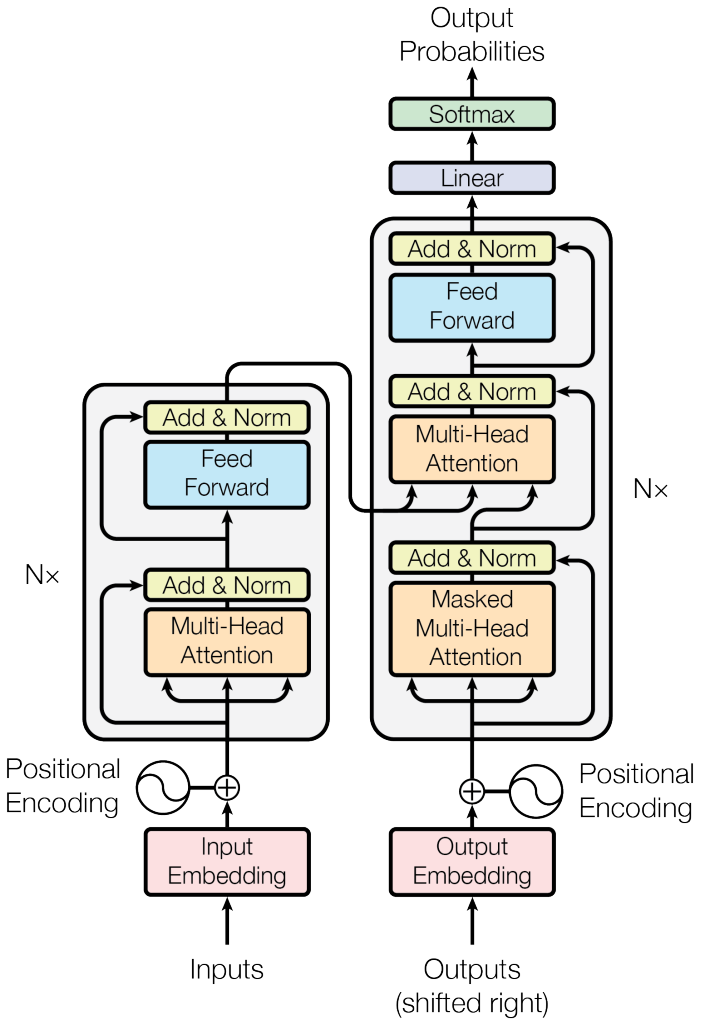


Figure 1: The Transformer - model architecture.

Multihead attention & other layers

- Compute multiple attentions per query ([picture 3])
- Masked for attention on output : an output should only depend on previous outputs

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

$$masked - attention(Q, K, V) = softmax(\frac{Q^T K + M}{\sqrt{(d_k)}})V$$

where M is a mask matrix containing only 0 and $-\infty$ values

- Normalization : $h \leftarrow \frac{h-\mu}{\sigma}$, μ/σ empirical mean/variance of h
- Positional encoding (different choices possible)

Complexity considerations

From *Vaswani et al.*

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)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

Results

From *Vaswani et al.*

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**	$2.3 \cdot 10^{19}$	

BERT architecture - from *Devlin et al.*

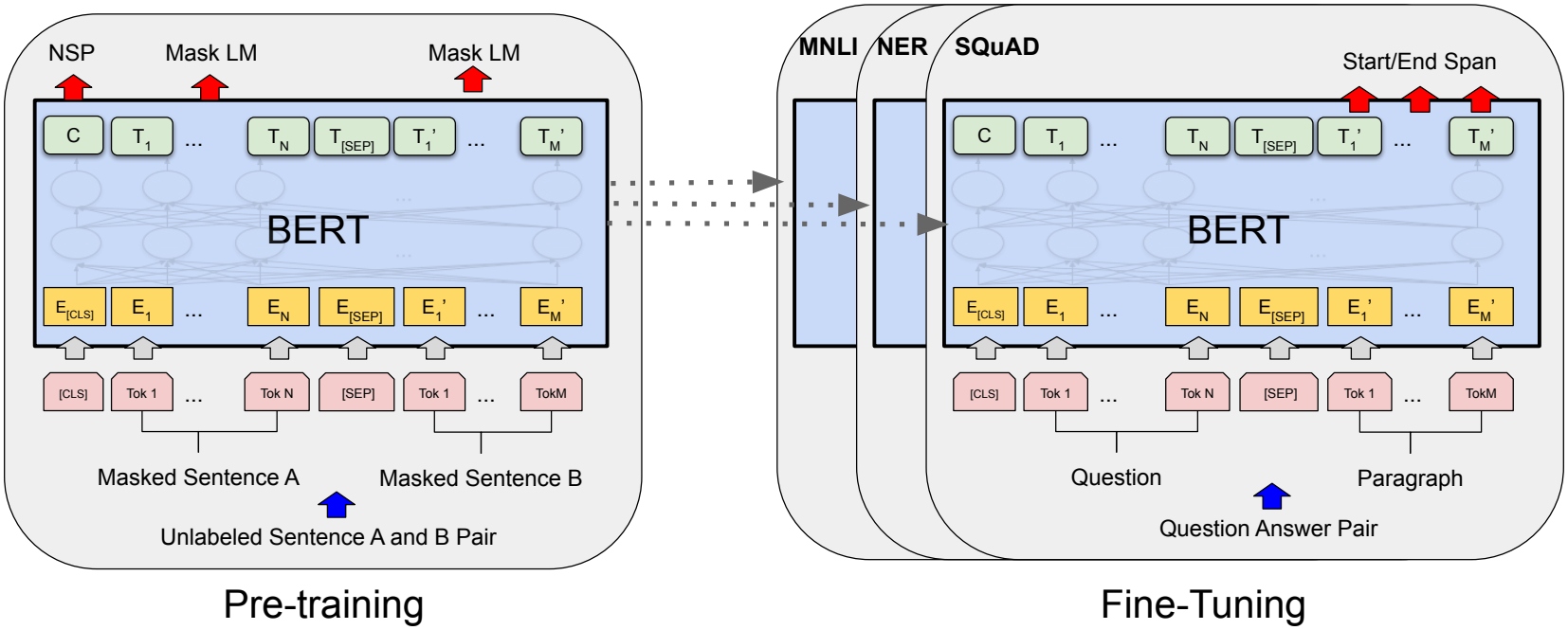


Figure 1: Overall pre-training and fine-tuning procedures for BERT. Apart from output layers, the same architectures are used in both pre-training and fine-tuning. The same pre-trained model parameters are used to initialize models for different down-stream tasks. During fine-tuning, all parameters are fine-tuned. `[CLS]` is a special symbol added in front of every input example, and `[SEP]` is a special separator token (e.g. separating questions/answers).

BERT Results

From *Devlin et al.*

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**

Table 1: GLUE Test results, scored by the evaluation server (https://gluebenchmark.com/leaderboard). The number below each task denotes the number of training examples. The "Average" column is slightly different than the official GLUE score, since we exclude the problematic WNLI set.[8] BERT and OpenAI GPT are single-model, single task. F1 scores are reported for QQP and MRPC, Spearman correlations are reported for STS-B, and accuracy scores are reported for the other tasks. We exclude entries that use BERT as one of their components.

Remarks

- State-of-the-art results at that time (still highly competitive)
- "Using extreme model sizes also leads to large improvements on very small scale tasks, provided sufficient pre-training"
- Difficulty to handle long documents

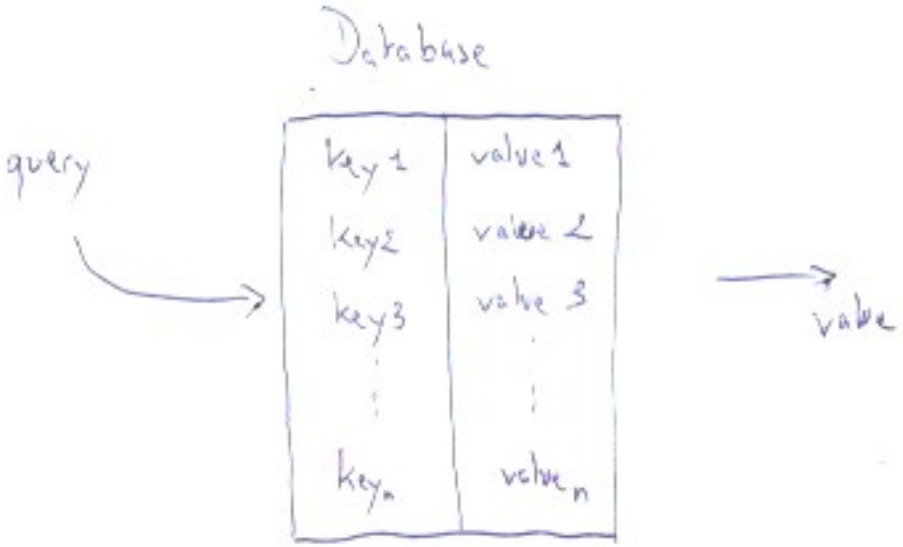
Conclusion

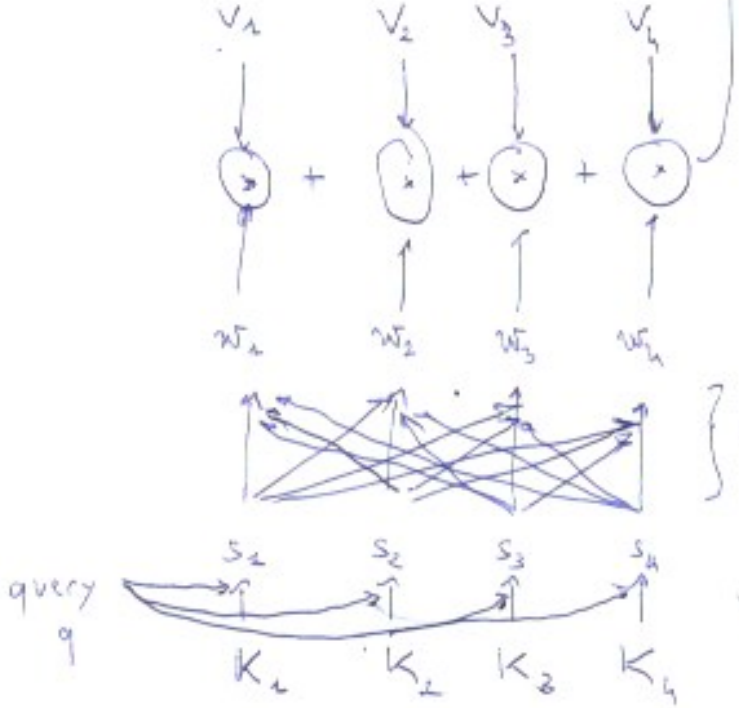
- Transformers and models *à la* BERT are predominant in NLP (any sequences ?)
- Pre-training has become an important element of all these models
- Limitations :
  - Complexity and energy costs
  - Poor(?) compositional generalization

References

- D. Bahdanau, K.H. Cho, Y. Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. ICLR 2015
- J. Devlin, M.-W. Chang, K. Lee, K. Toutanova. BERT : Pre-training of Deep Bidirectional Transformers for Language Understanding. NAACL-HLT 2019
- A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser. Attention Is All You Need. NIPS 2017

Retrieval in a database





attention value

$$= \sum_i w_i V_i$$

$$w_i = \frac{\exp(s_i)}{\sum_j \exp(s_j)}$$

$$s_i = f(q, K_i)$$

$=$

- $q^T K_i$: dot product
- $q^T K_i / \sqrt{d_i}$: scaled dot product (d_i: dimensionality of K_i)
- $q^T W K_i$: general dot product
- $w_q^T q + w_K^T K_i$: additive similarity

multi-head attention