Maybe BERT is all you need ?

Paul Peyramaure

Summary

- Speaker presentation
- Brief reminder of self-attention in Transformers
- BERT paper presentation
- An application in action recognition
- Conclusion

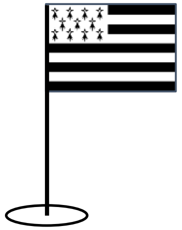
Speaker presentation

Paul Peyramaure, research engineer in VAADER team working on :

- Computer Vision, Deep Learning, fall detection, sequence classification

Working on the SilverConnect project :

- **Funded by** : the European Union, the Britanny region, the city of Rennes through APP FEDER
- **Objective** : To develop an offer of equipment and services to improve life in nursing homes (EHPAD).
- **Consortium** :



HOPPEN	IETR VAADER	famileo
Project Management Digital equipment offer	Development of a fall detection solution	Solutions to reinforce the link between residents and their families

BERT paper

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova
Google AI Language
{jacobdevlin,mingweichang,kentonl,kristout}@google.com

- [1] won the Best Long Paper Award at NACL 2019
- **BERT** stands for **B**idirectional **E**ncoder **R**epresentation of **T**ransformer
- The goal of this model is to **generate a language model designed to learn bidirectional representations** by considering both the left and right contexts
- It is a stack of Encoder blocks of Transformers

[1] BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Brief reminder of self-attention Transformers

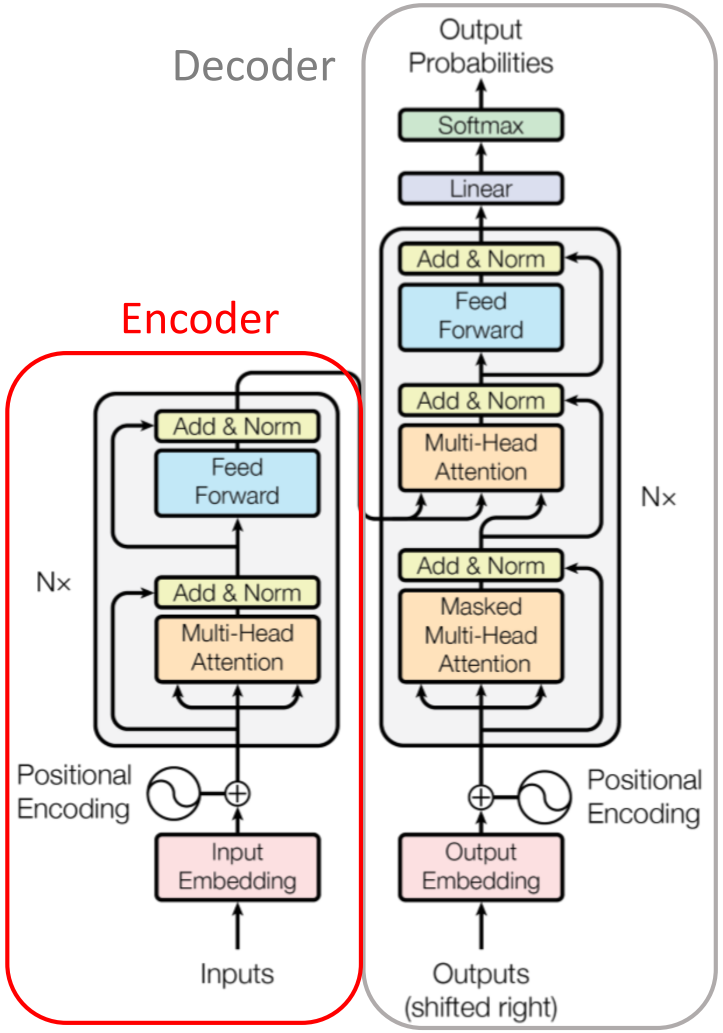
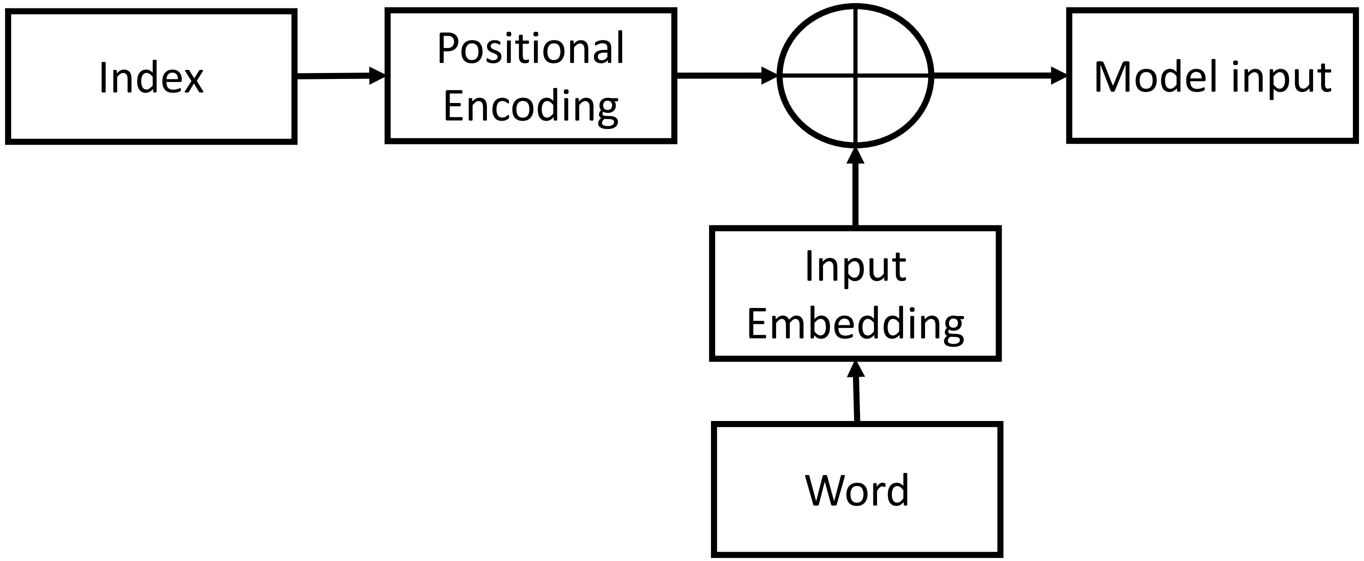


Figure 1: The Transformer - model architecture. [1]

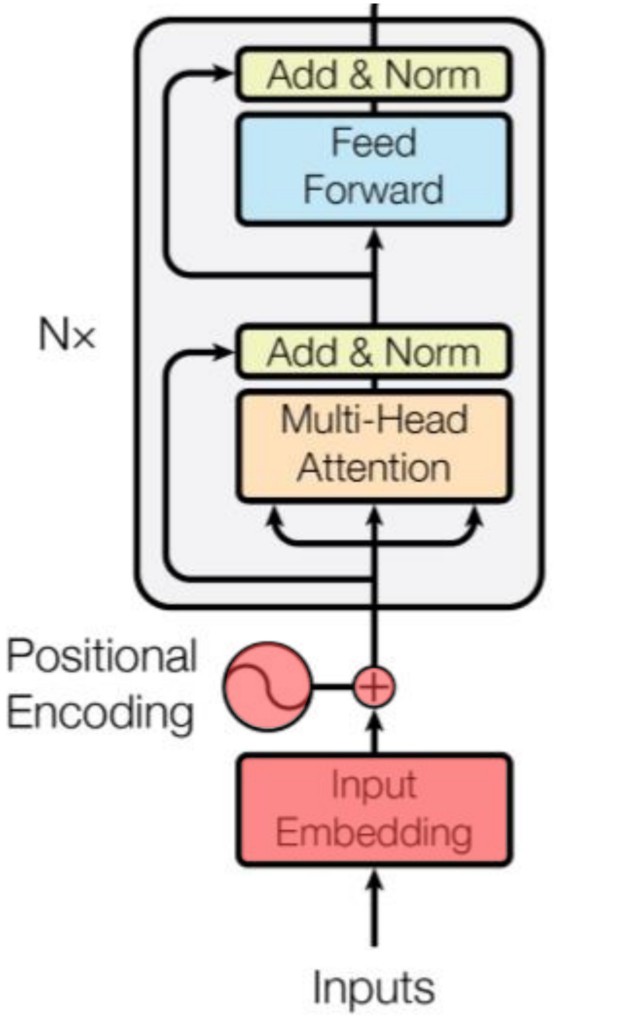
[1] Ashish Vaswani et al. 2017. Attention is all you need. In Advances in Neural Information Processing Systems, pages 6000–6010.

Brief reminder of self-attention Transformers

Input embedding and positional encoding:



Word	This	sentence	is	an	example
Index	0	1	2	3	4



Transformer encoder part

Self-attention calculation:

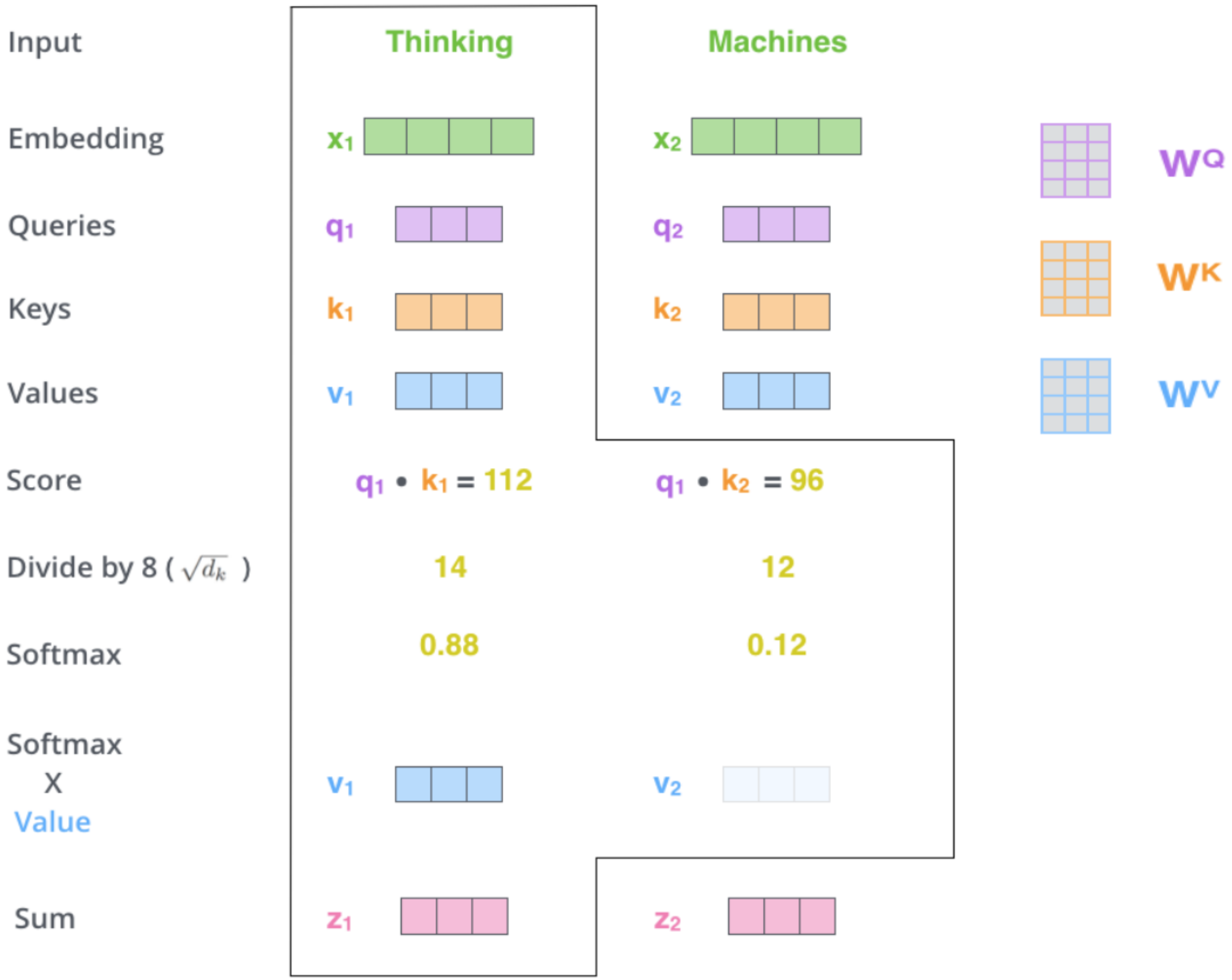
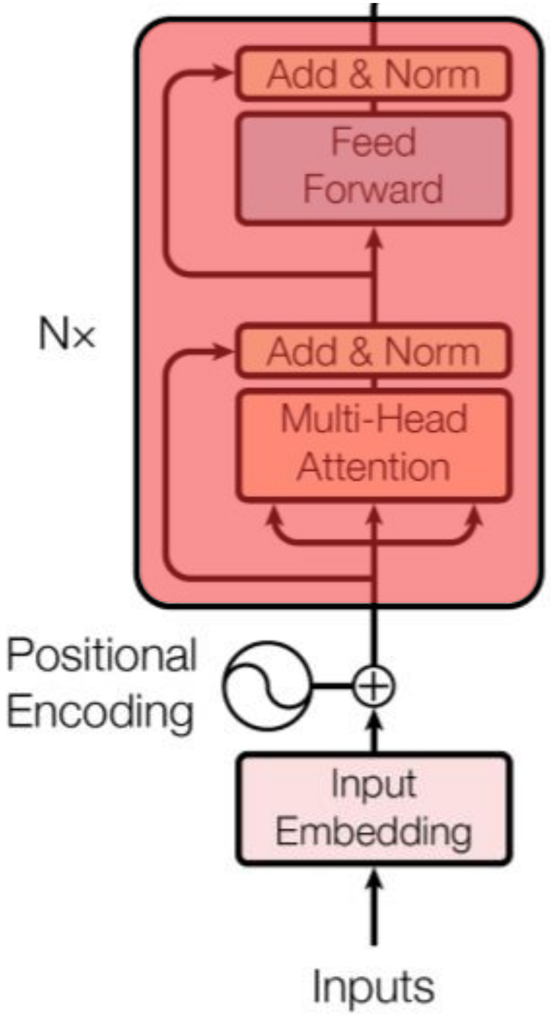


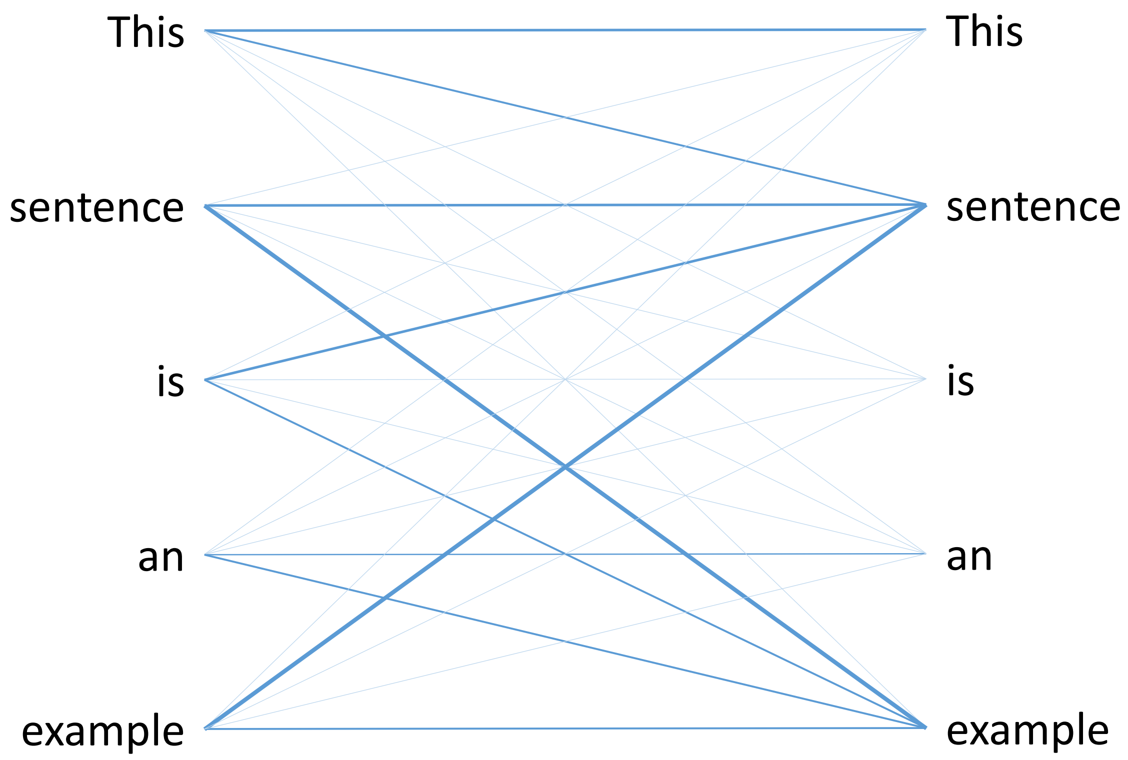
Illustration from : http://jalammar.github.io/illustrated-transformer/

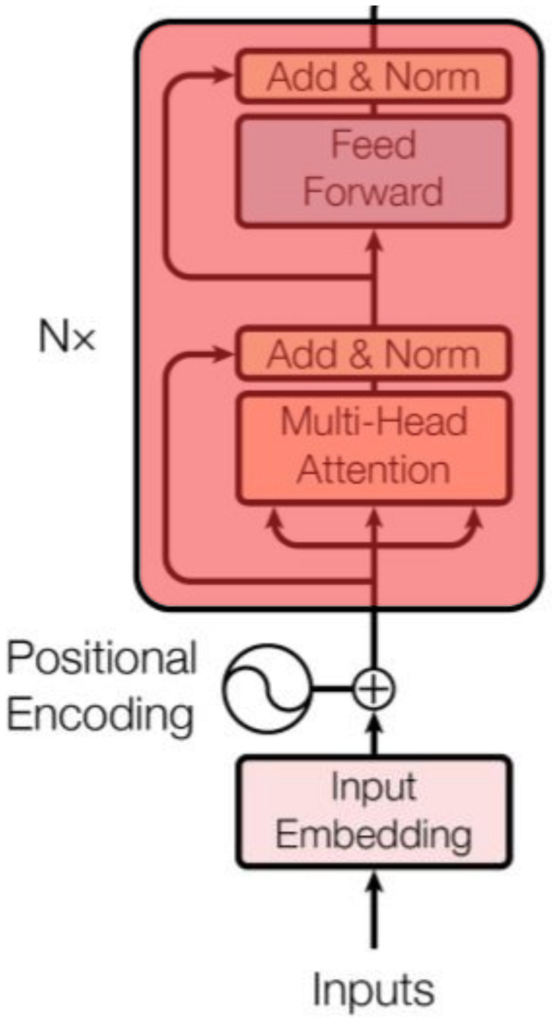


Transformer encoder part

Brief reminder of self-attention Transformers

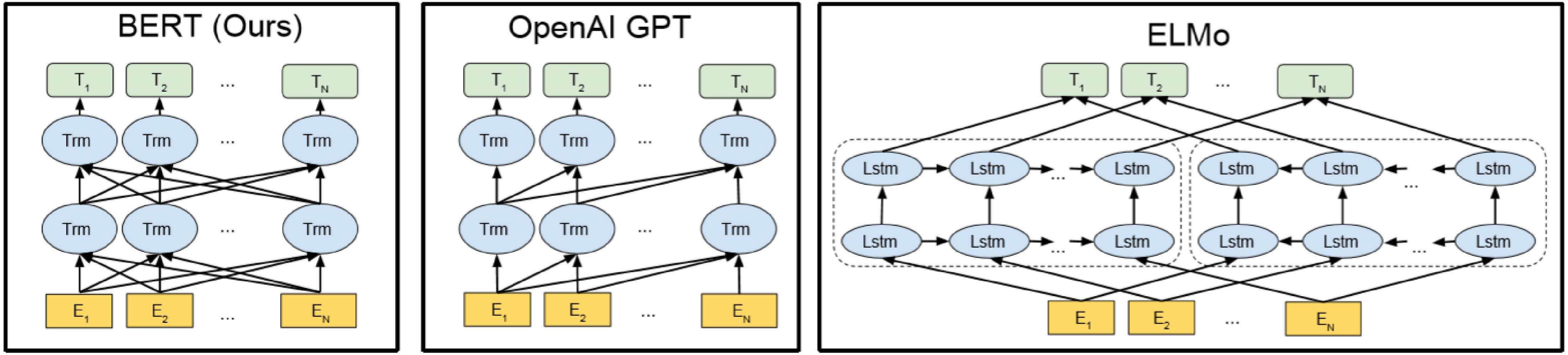
This mechanism results to something like:





Transformer encoder part

Model architectures differences



Stack of Transformer encoders

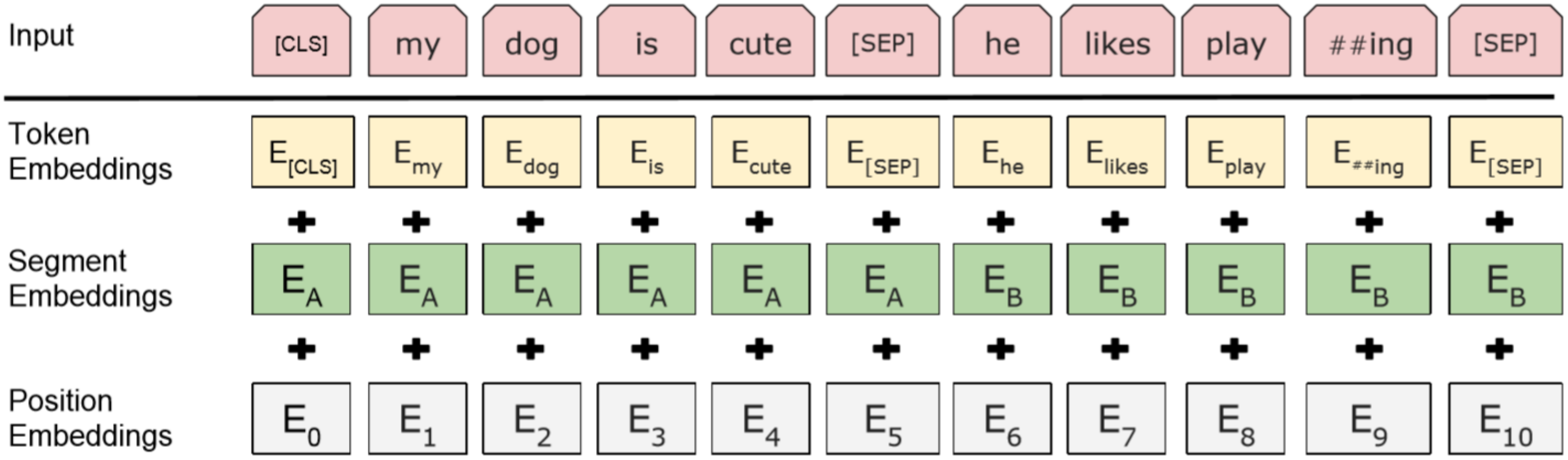
Stack of Transformer decoders

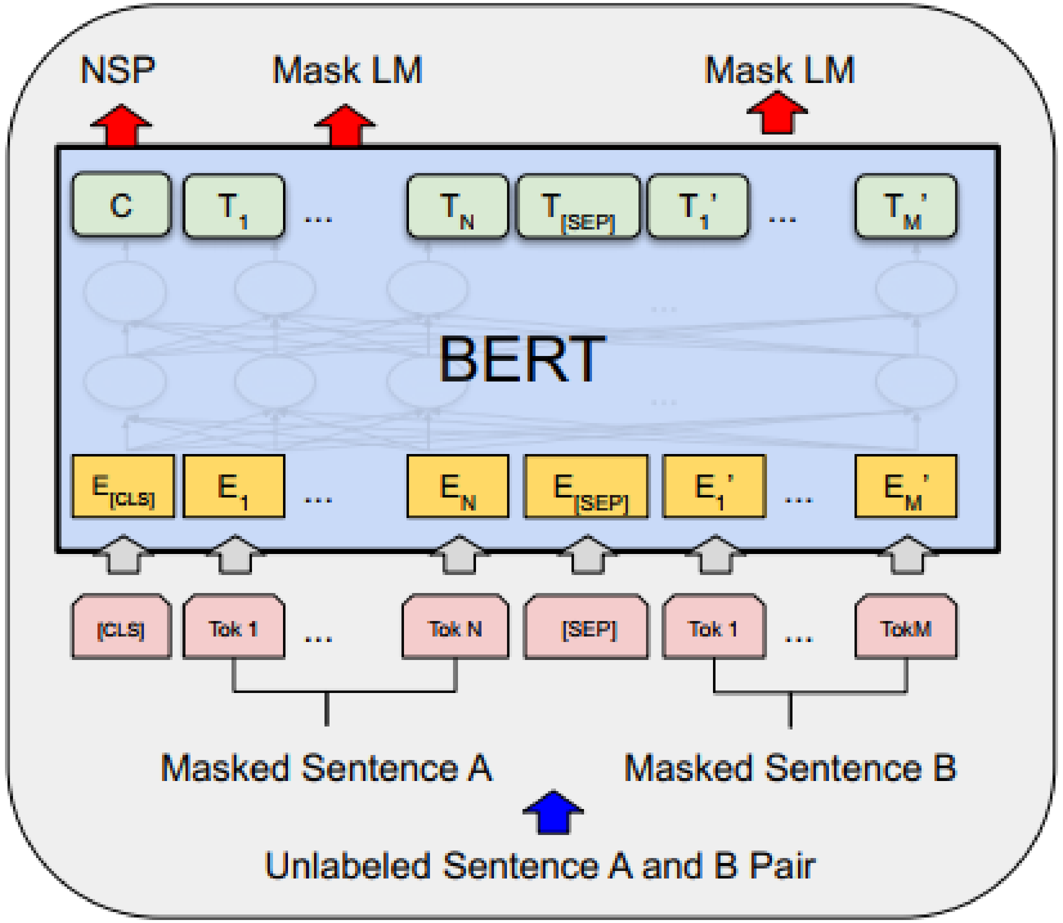
Concatenation of features extracted from left-to-right and right-to-left LSTMs

Input/Output Representations in BERT

Classification token

Token used separate
two sentences





Unsupervised Pre-training on two tasks

for Language Modeling

- *Mask Language Modeling (MLM):* Words prediction in a sentence

  Among 15% of words:

  - 80% of time: The word is replaced by the [MASK] token

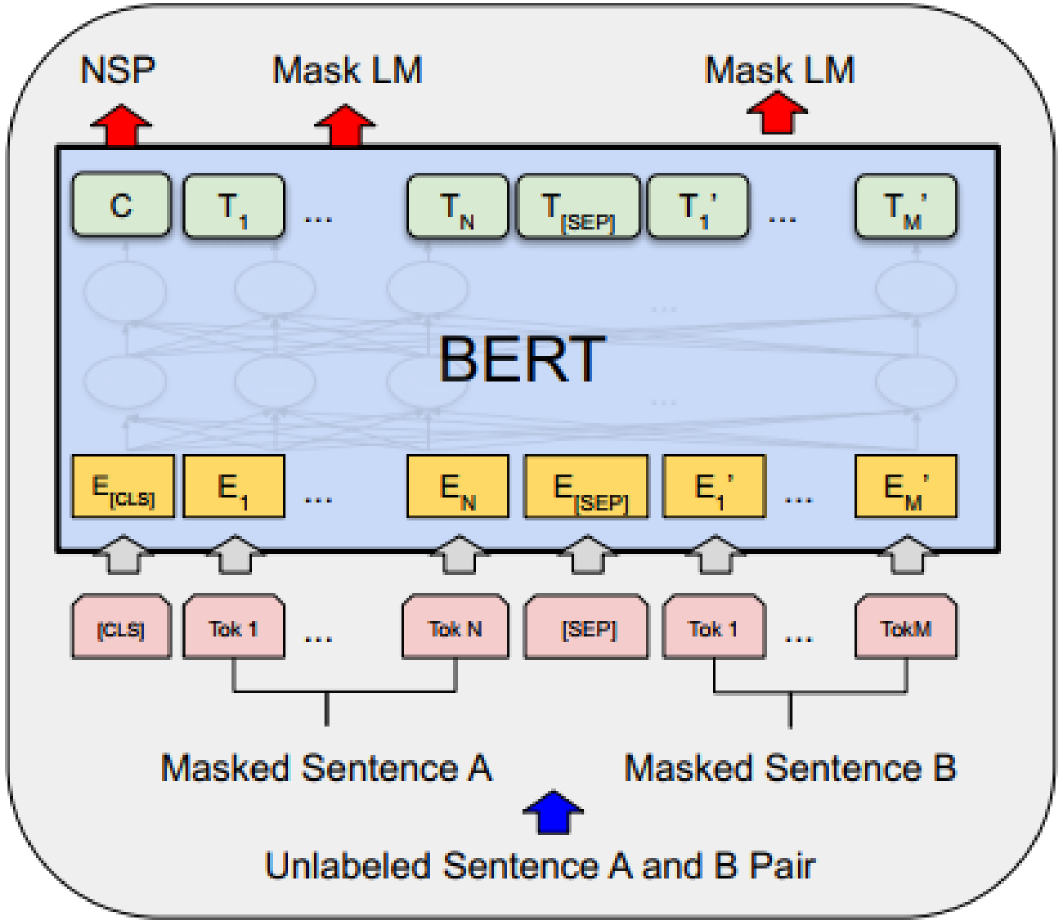
    `Ex: My dog is hairy > My dog is [MASK]`

  - 10% of time: The word is replaced by a random word

    `Ex: My dog is hairy > My dog is apple`

  - 10% of time: The original word is kept

    `Ex: My dog is hairy > My dog is hairy`



Unsupervised Pre-training on two tasks

for Language Modeling

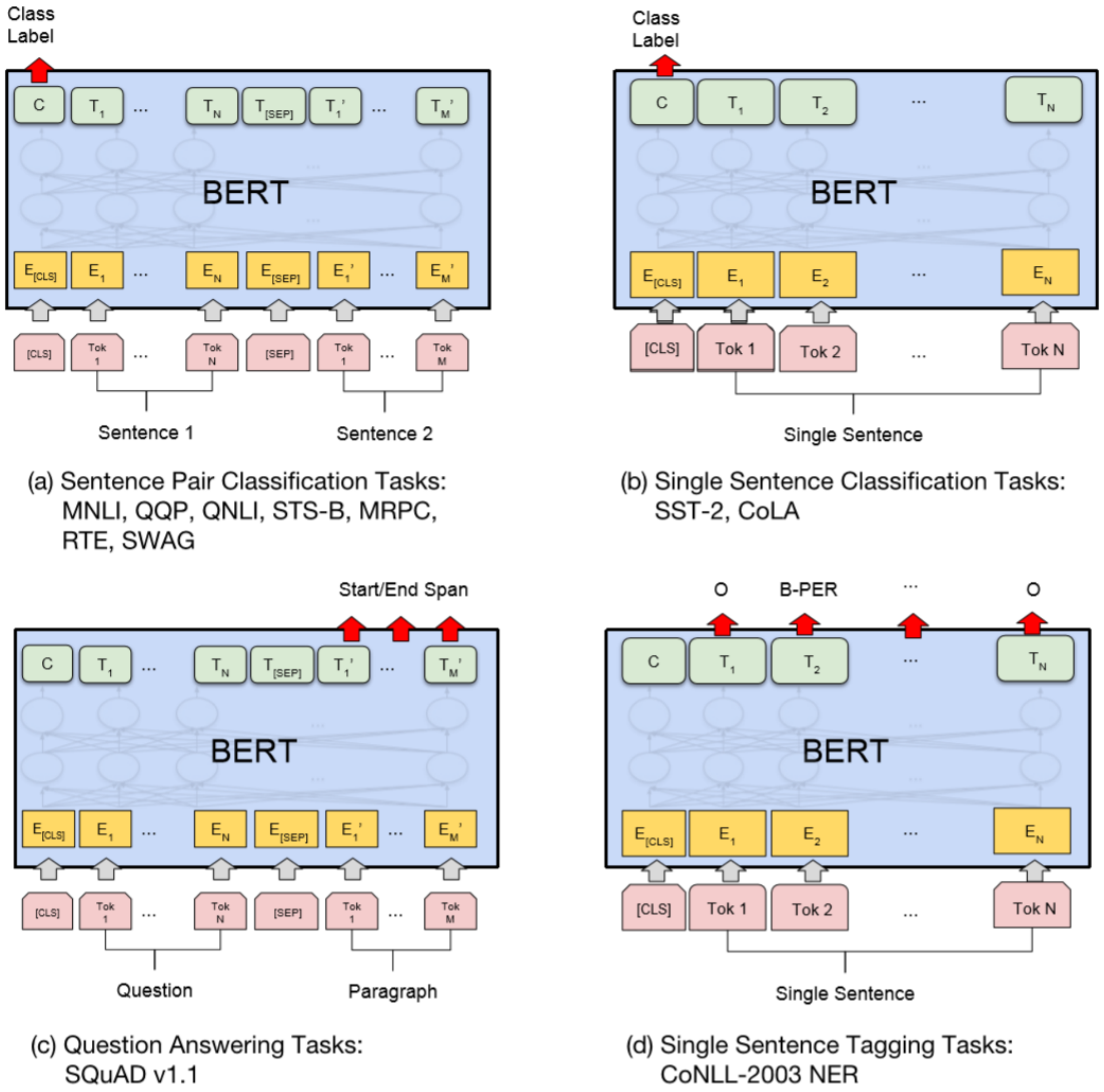
- *Next Sentence Prediction (NSP):* Predict if both sentence are continuous or not
    - 50%: next sentence -> IsNext

Input = `[CLS] the man went to [MASK] store [SEP]`
`he bought a gallon [MASK] milk [SEP]`
Label = `IsNext`

- 50%: random sentence from data -> NotNext

Input = `[CLS] the man [MASK] to the store [SEP]`
`penguin [MASK] are flight ##less birds [SEP]`
Label = `NotNext`

BERT paper



Fine-Tuning

The training method depends of the application.

- Sentences pair classification
- Single sentence classification
- Question answering
- Name Entity Recognition

The [CLS] output token is used for classification :
It aggregates information of the sequence

Experiments

Pre-training on two large datasets:

- BooksCorpus (800M words)
- English Wikipedia (2500M words)

State of the art on eleven NLP tasks in oct.2018:

System	MNLI-(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	**Average**
	392k	363k	108k	67k	8.5k	5.7k	3.5k	2.5k	-
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.

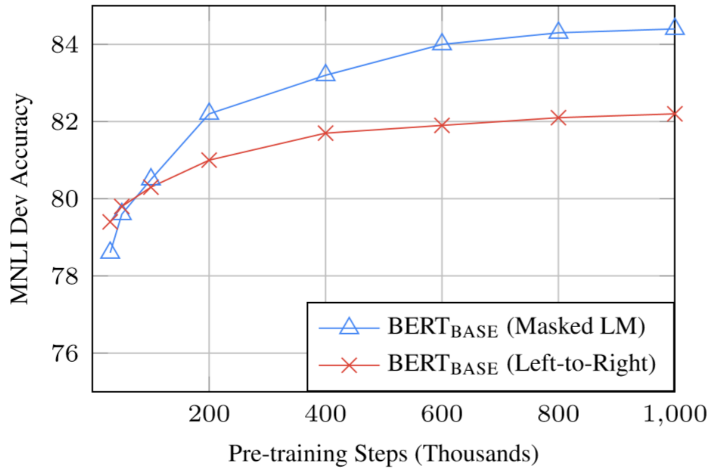


Figure 5: Ablation over number of training steps. This shows the MNLI accuracy after fine-tuning, starting from model parameters that have been pre-trained for k steps. The x-axis is the value of k.

Some other models inspired by BERT

RoBERTa [1]:

- does not use NSP in pre-training
- introduces dynamic masking to reduce overfitting risk
- 2 to 20% improvement over BERT on different tasks
- But uses 10x much more data and 4-5x more training time than BERT

DistilBERT [2]:

- Distilled (or approximate) version of BERT
- Uses haft of parameters of BERT (66M)
- Retains 97% of BERT's performance
- 4x less training time than BERT

ALBERT [3]:

- A "Lite" BERT
- Shared parameters across layers : reduced a number of parameters to store but not the number of operations..
- Replaces NSP by Sentence Order Prediction (SOP) : to model inter-sentences coherence
- Outperforms BERT and RoBERTa on many tasks

[1] Yinhan Liu et al. RoBERTa: A robustly optimized BERT pretraining approach, 2019.
[2] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter, 2019.
[3] Zhenzhong Lan et al. ALBERT: A lite BERT for self-supervised learning of language representations, 2019.

An application in action recognition

Late Temporal Modeling in 3D CNN Architectures with BERT for Action Recognition

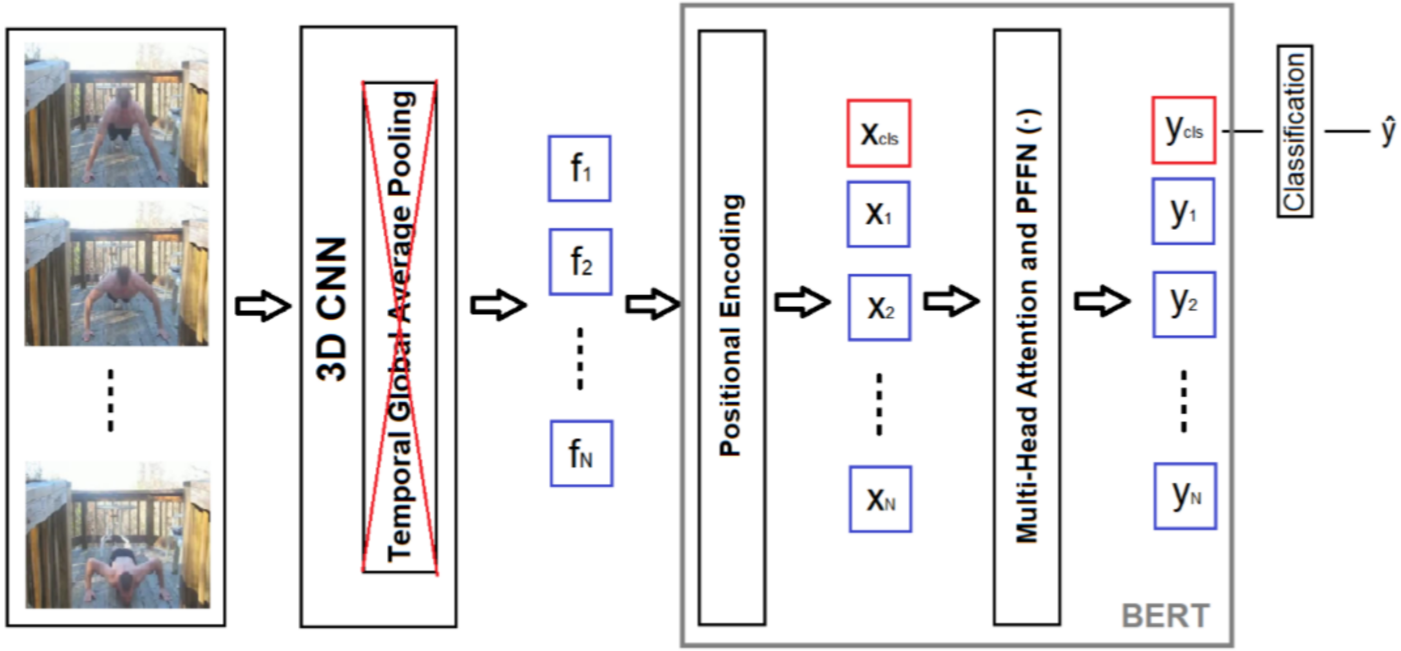


Fig. 1: BERT-based late temporal modeling

- Data for pre-trained backbone:
  - Kinetics 400 : 240k 64f. clips
  - IG65M : 65M 64f. clips
- They use one single BERT layer only fine-tuned on action recognition task
- Current state-of-the-art algorithm performing best on the datasets : UCF101 and HMDB-51

[1] M. Kalfaoglu, S. Kalkan, and A. A. Alatan. “Late Temporal Modeling in 3D CNN Architectures with BERT for Action Recognition”, Aug.2020

Conclusion

Advantages:

- BERT introduces bidirectionality for language modeling
- It has been a breakthrough in NLP in Oct. 2018

Drawbacks:

- It requires a huge amount of data to be trained and a long training time for language modeling
- As it uses transformer self-attention : complexity is quadratic with the input length

Open Question:

- The current models are getting bigger and bigger (as GPT-3 [1] with its 175B parameters…), how could these new models be realistically deployed ?

[1]Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. 2020.