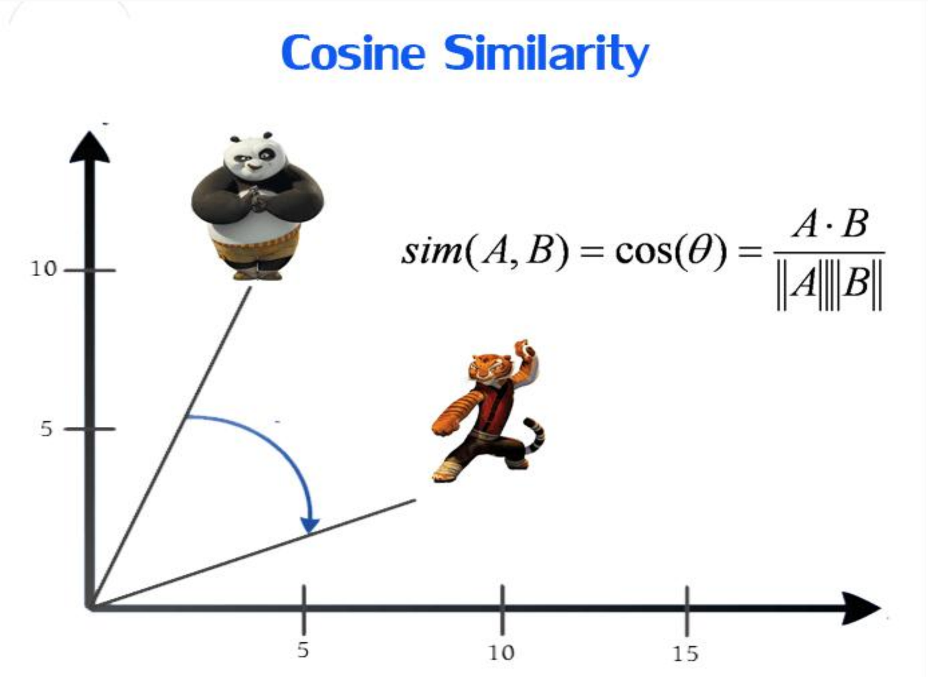
BERT: The milestones of NLP
& Some applications in Summarization

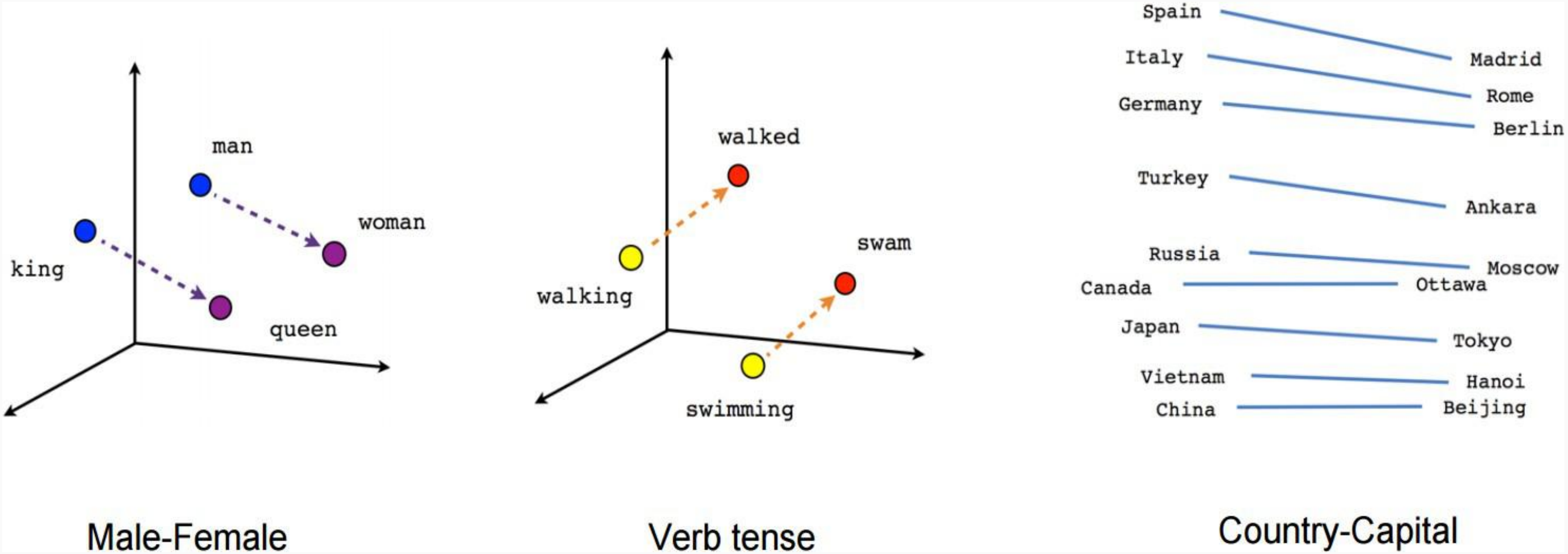
Yufeng Lv

Word Embedding

- Map the word into "semantic" space as a point

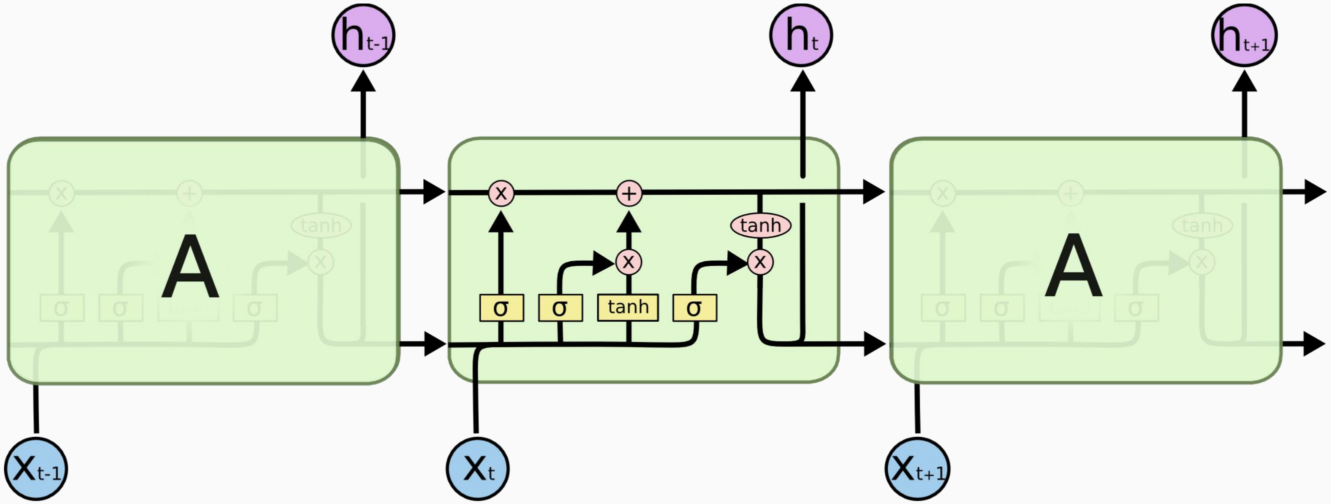


Word Embedding



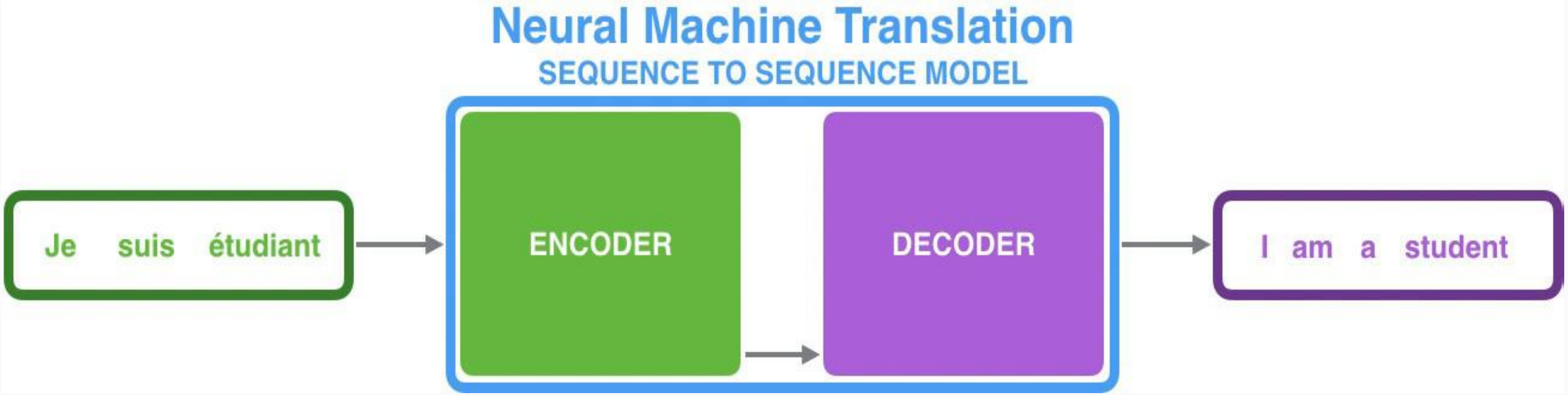
LSTM

- Semantics are context sensitive
- Avoids gradient disappearance through gate



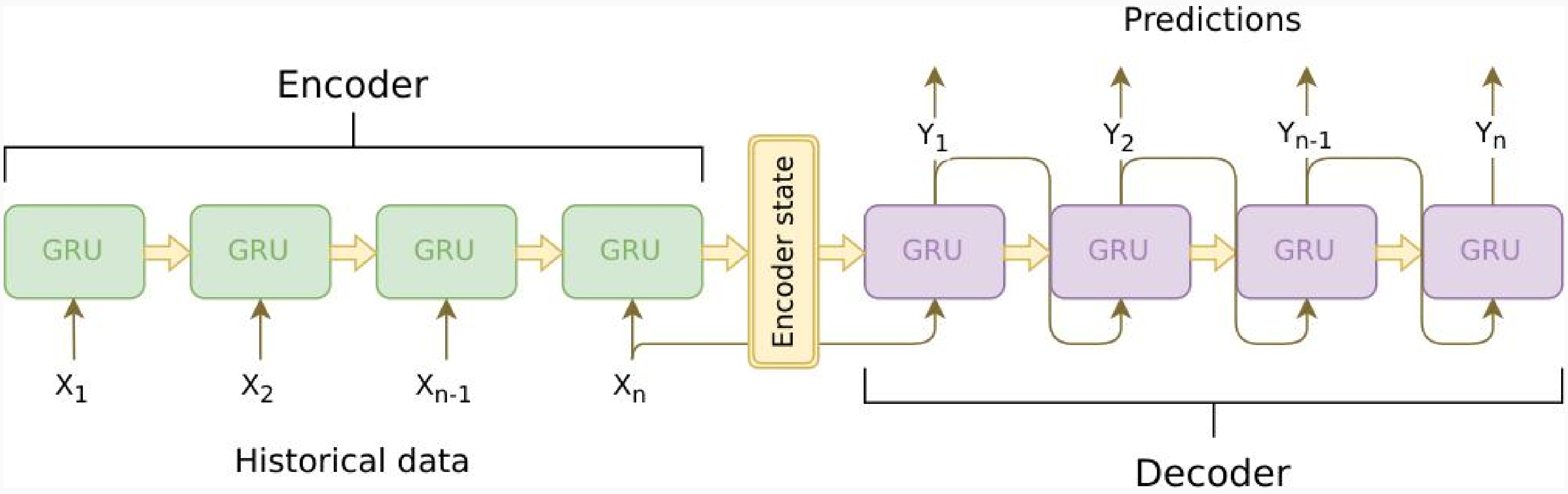
Seq2Seq

- Composed of two RNNs
- Can be used in Machine translation, summarization, Q&A and dialogue systems



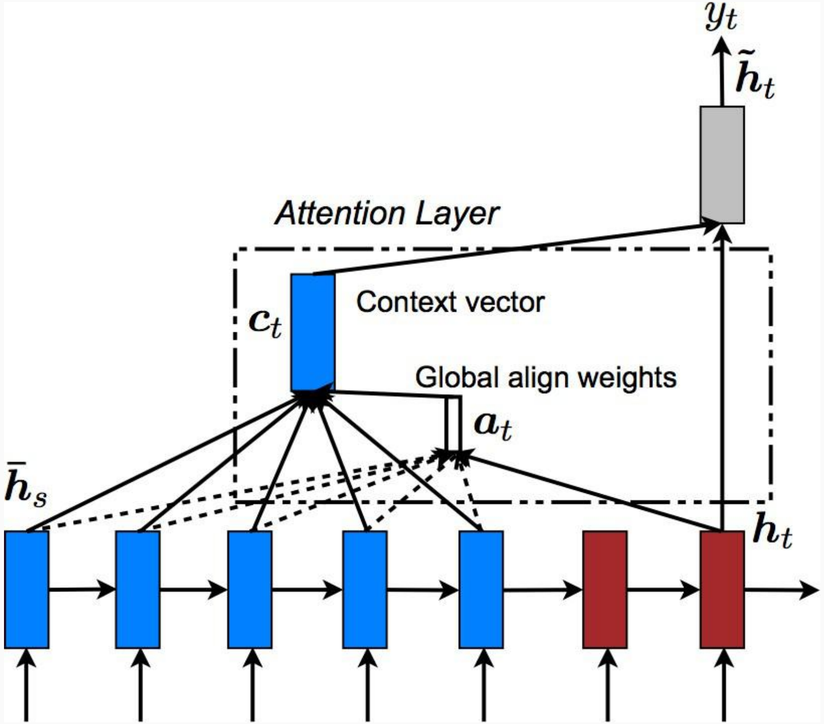
Seq2Seq

- Fixed length context vector



Attention

- Pay attention to related word



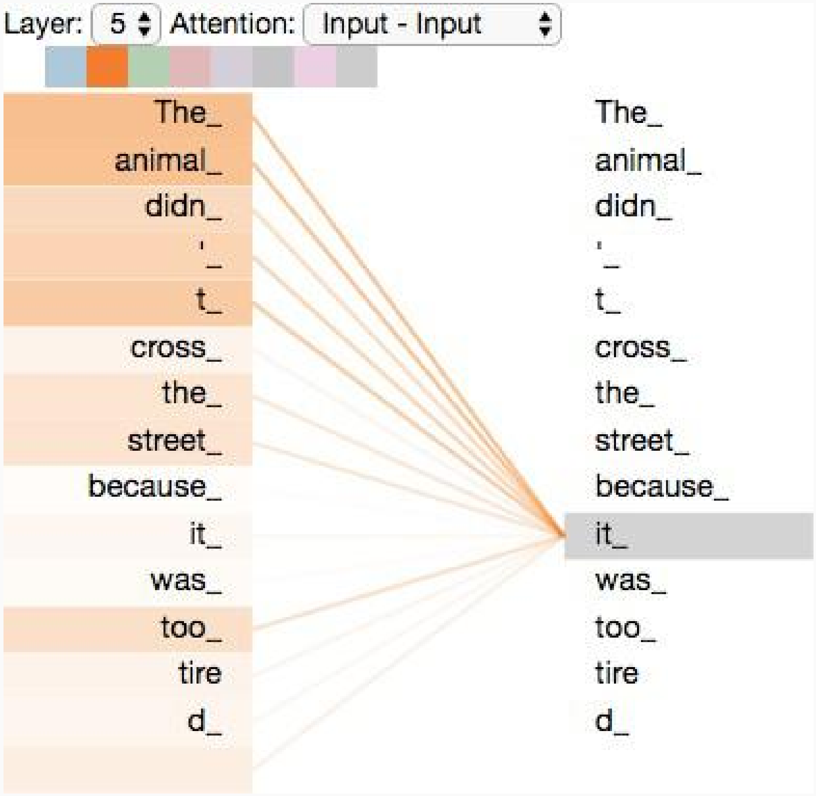
Problem

The animal didn’t cross the street because it was too tired.

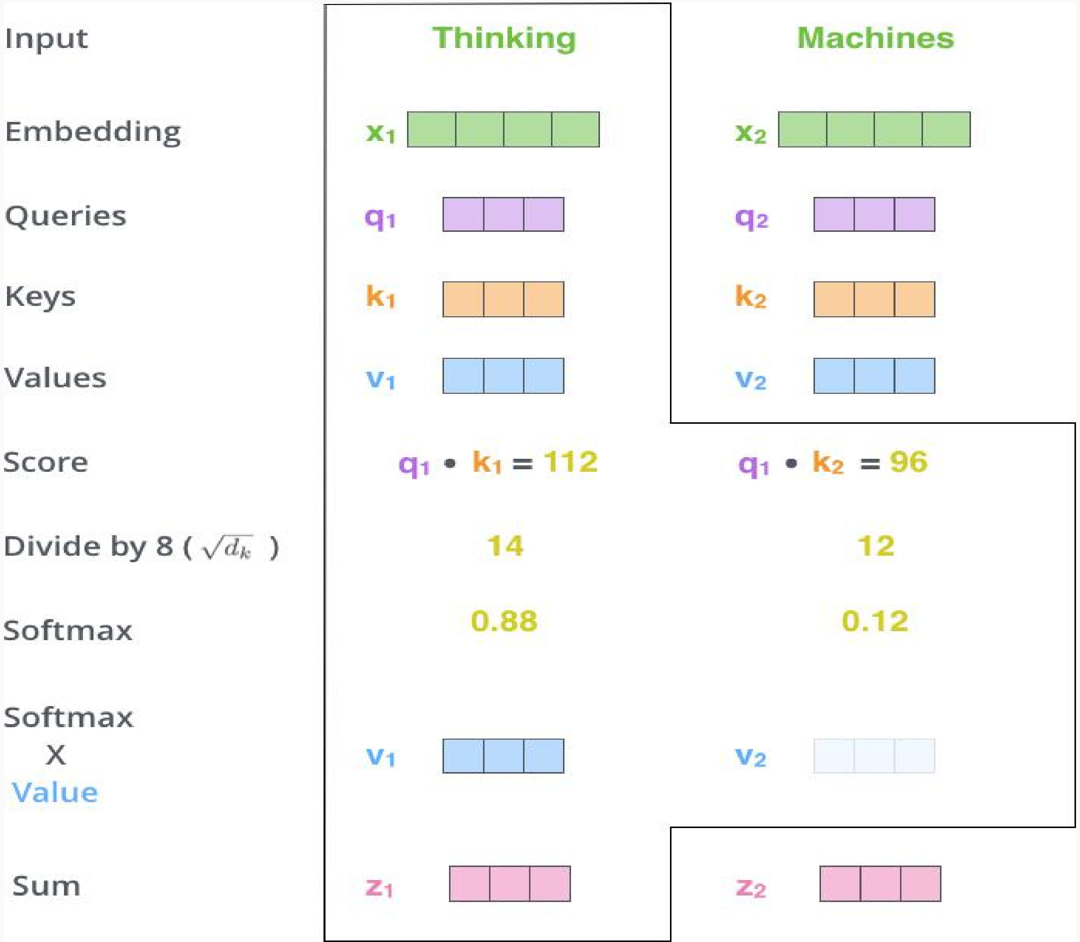
The animal didn’t cross the street because it was too narrow.

- The animal didn’t cross the street because it?
- it? was too tired.

Self-Attention

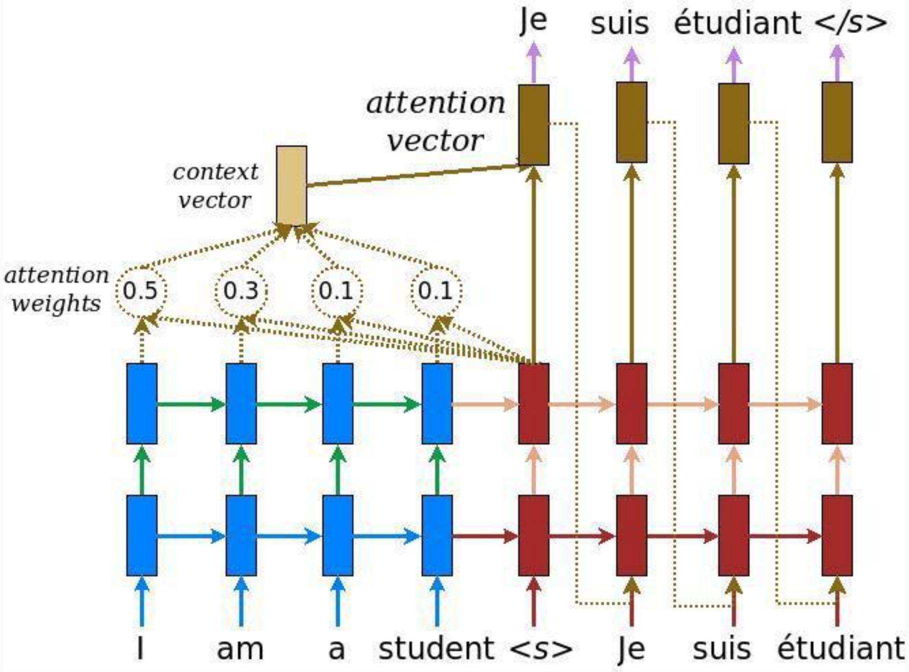


Self-Attention Calculate

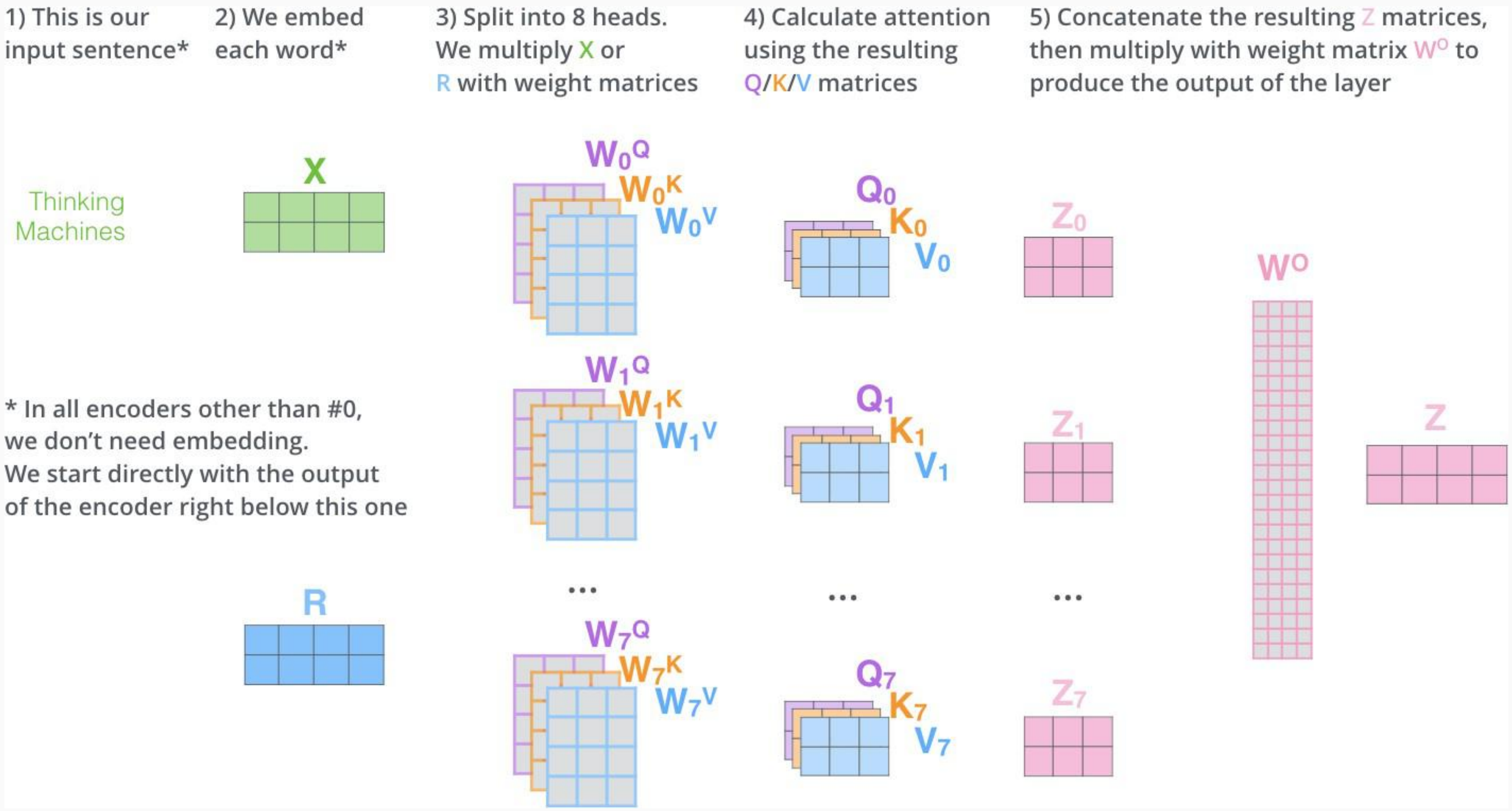


Compare with Normal Attention

- Q is decoder’s hidden state
- K is encoder’s output
- V is encoder’s output

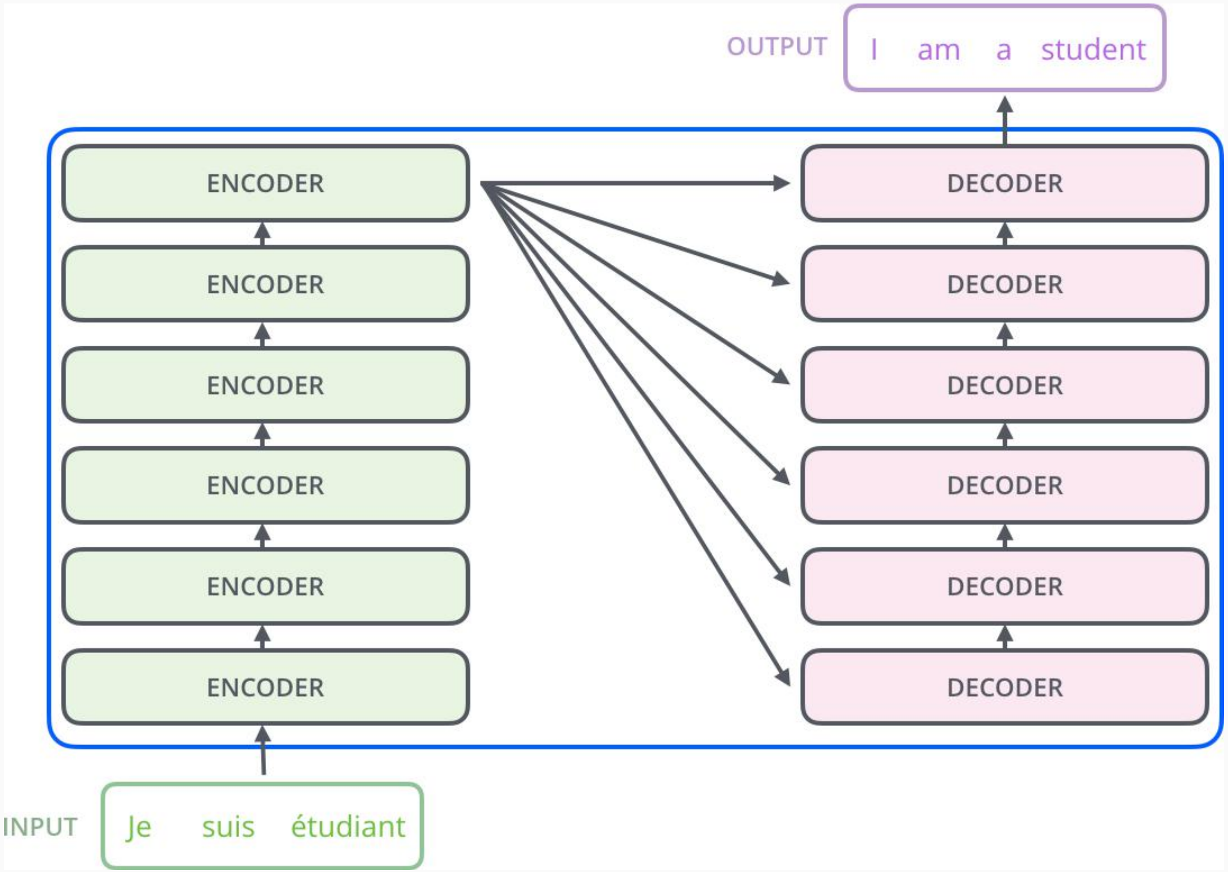


Multi-Heads



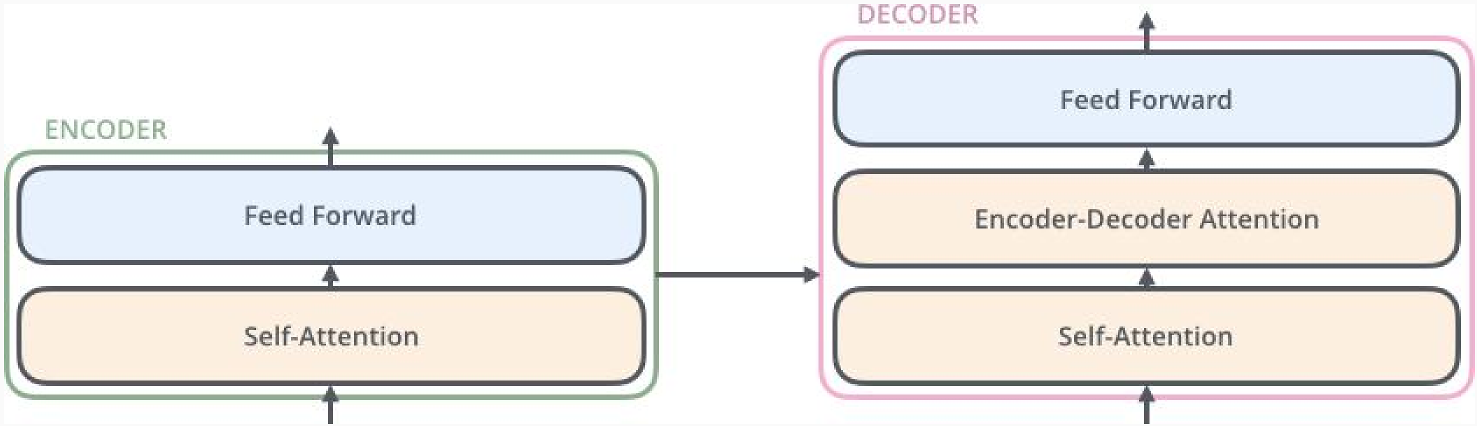
1) This is our input sentence*
2) We embed each word*
3) Split into 8 heads. We multiply X or R with weight matrices
4) Calculate attention using the resulting Q/K/V matrices
5) Concatenate the resulting Z matrices, then multiply with weight matrix W^O to produce the output of the layer
Thinking Machines
X
R
* In all encoders other than #0, we don't need embedding. We start directly with the output of the encoder right below this one
W_0^Q W_0^K W_0^V
W_1^Q W_1^K W_1^V
...
W_7^Q W_7^K W_7^V
Q_0 K_0 V_0
Q_1 K_1 V_1
...
Q_7 K_7 V_7
Z_0
Z_1
...
Z_7
W^O
Z

Transformer

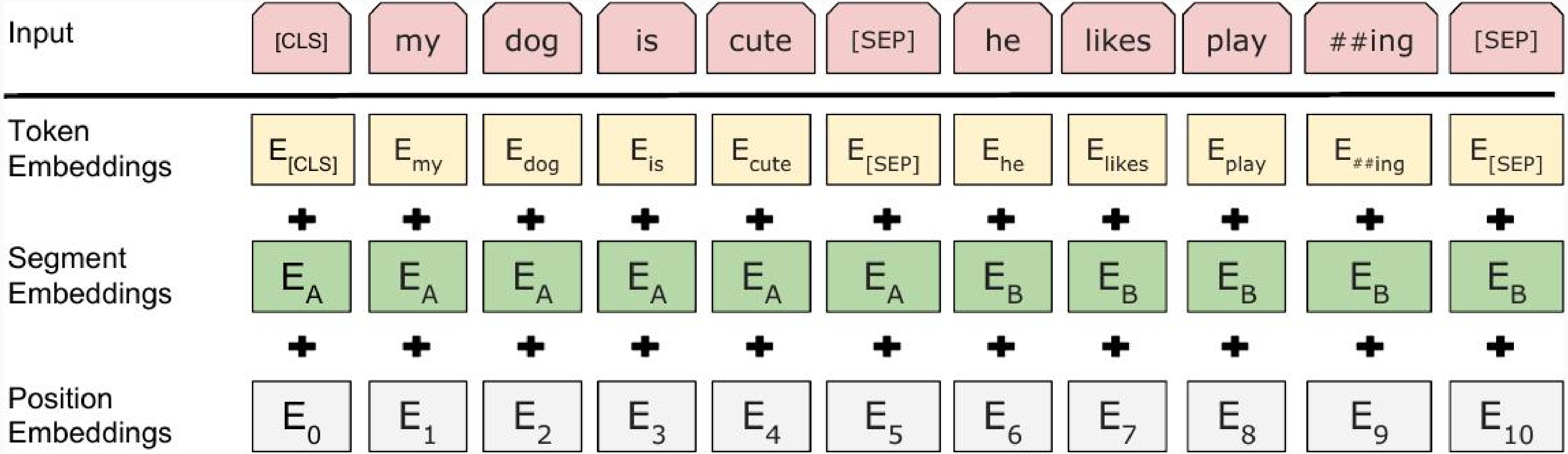


Transformer

- One layer Encoder and Decoder



Position Embedding



Contextual Word Embedding

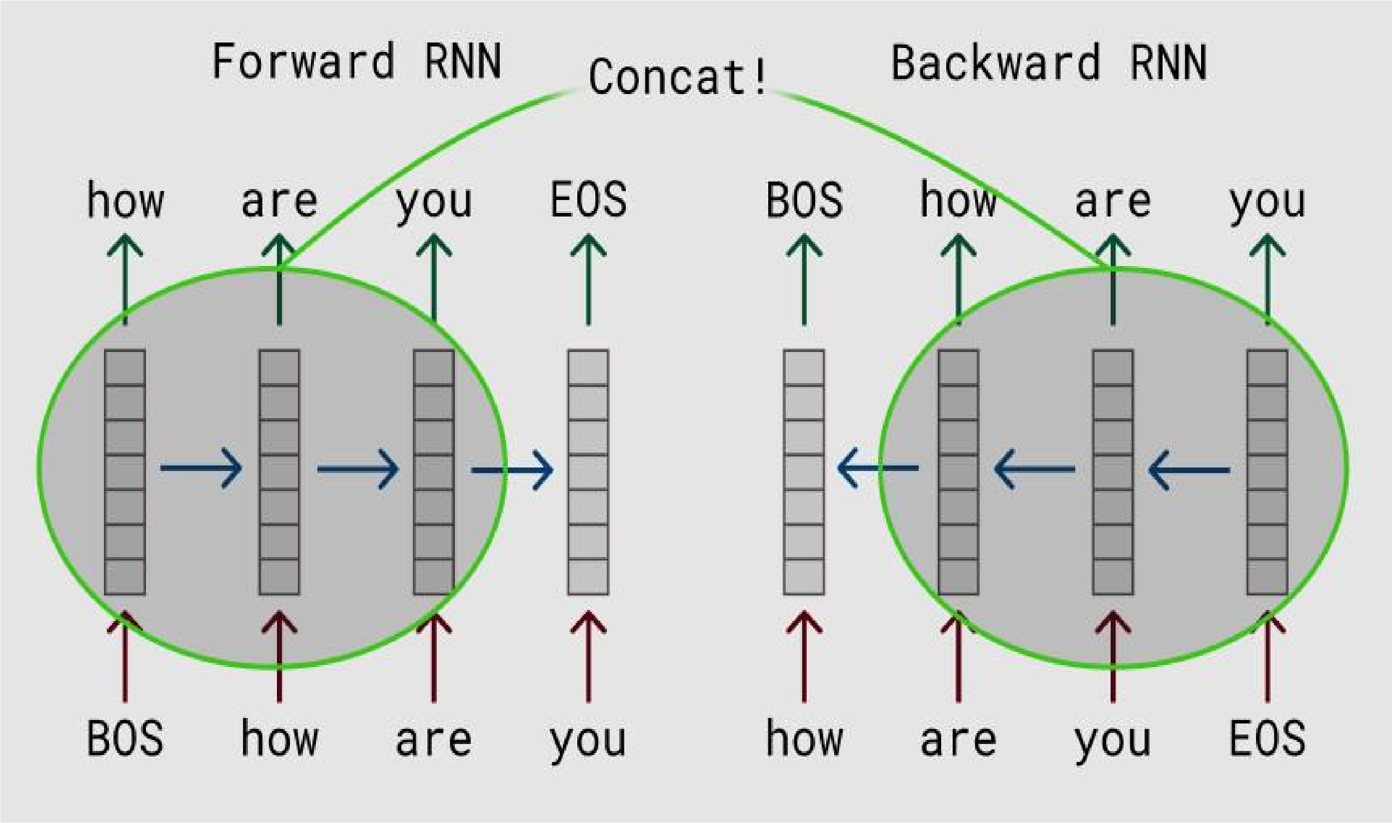
Problems

- Word Embedding without context information
- Lack of supervised data

Solutions

- Unsupervised learning
- Contextual Word Embedding

ELMo



Forward RNN
Concat!
Backward RNN
how
are
you
EOS
BOS
how
are
you
BOS
how
are
you
how
are
you
EOS

ELMo’s Problem

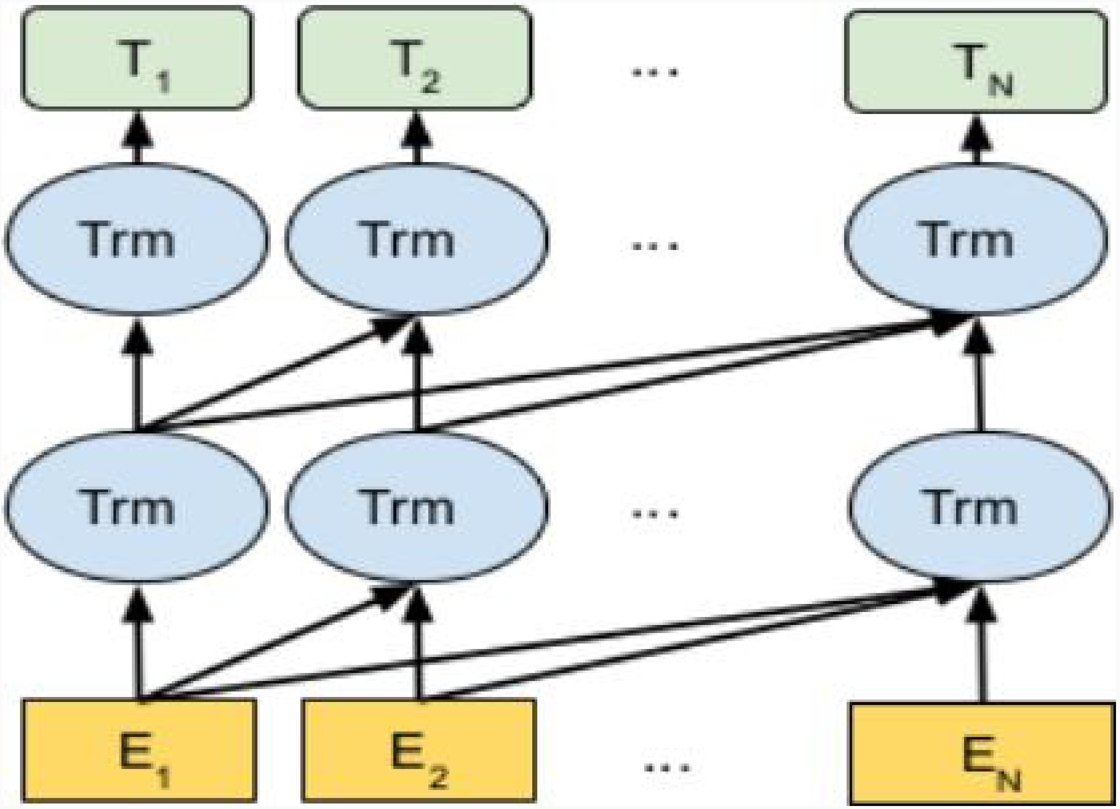
Problems

- Not suitable for a specific task

Solutions

- Fine-tuning depends on the task
- Use Transformer replace RNN/LSTM

OpenAI GPT



GPT's Problem

Problems

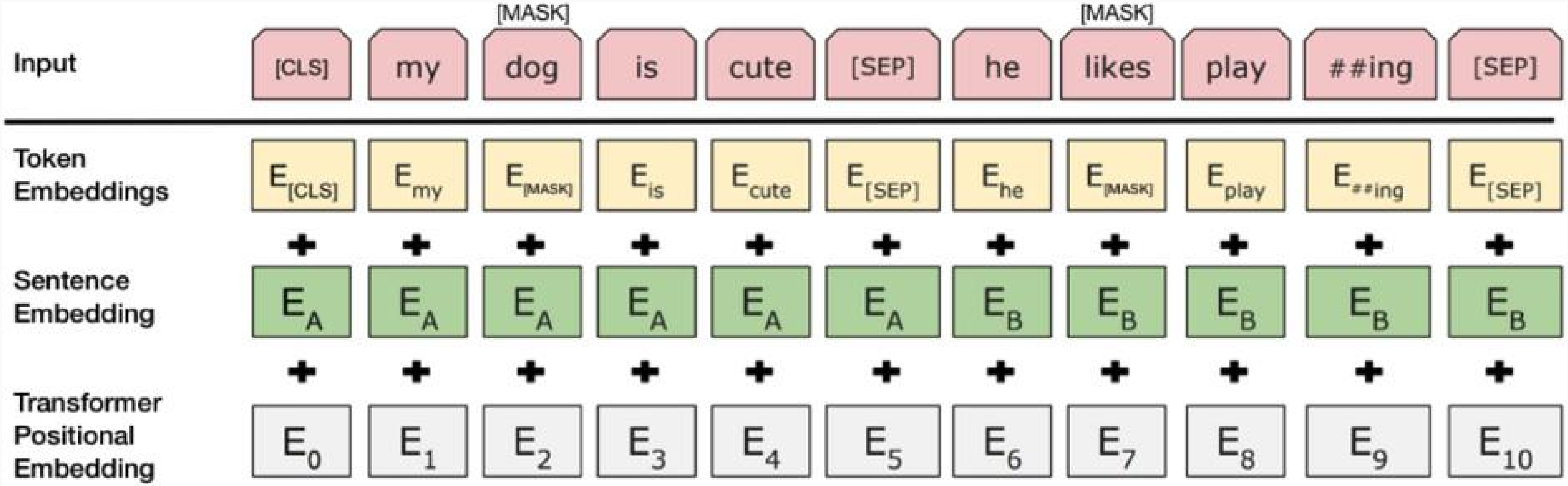
- Unidirectional
- Pre-training and Fine-tuning not matched

Solutions

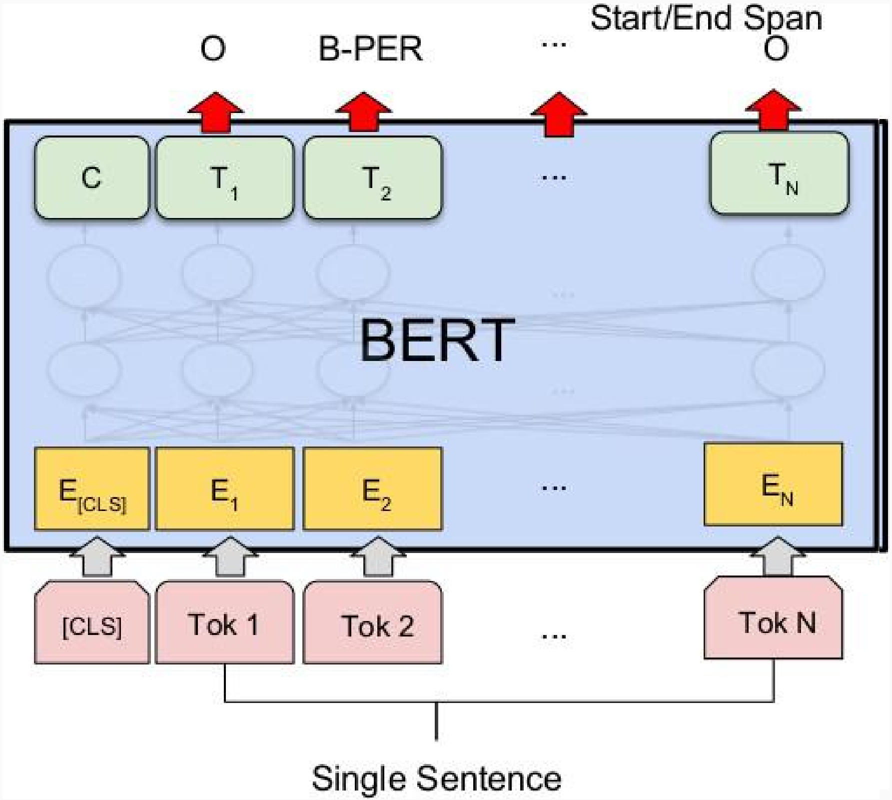
- Masked LM
- NSP Multi-task Learning

Masked LM

- Random mask 15% words, and use BERT to predict



Fine-Tuning



Start/End Span
O
B-PER
...
O
C
T1
T2
...
TN
BERT
E[CLS]
E1
E2
...
EN
[CLS]
Tok 1
Tok 2
...
Tok N
Single Sentence

Simple Fine-Tuning for Summarization

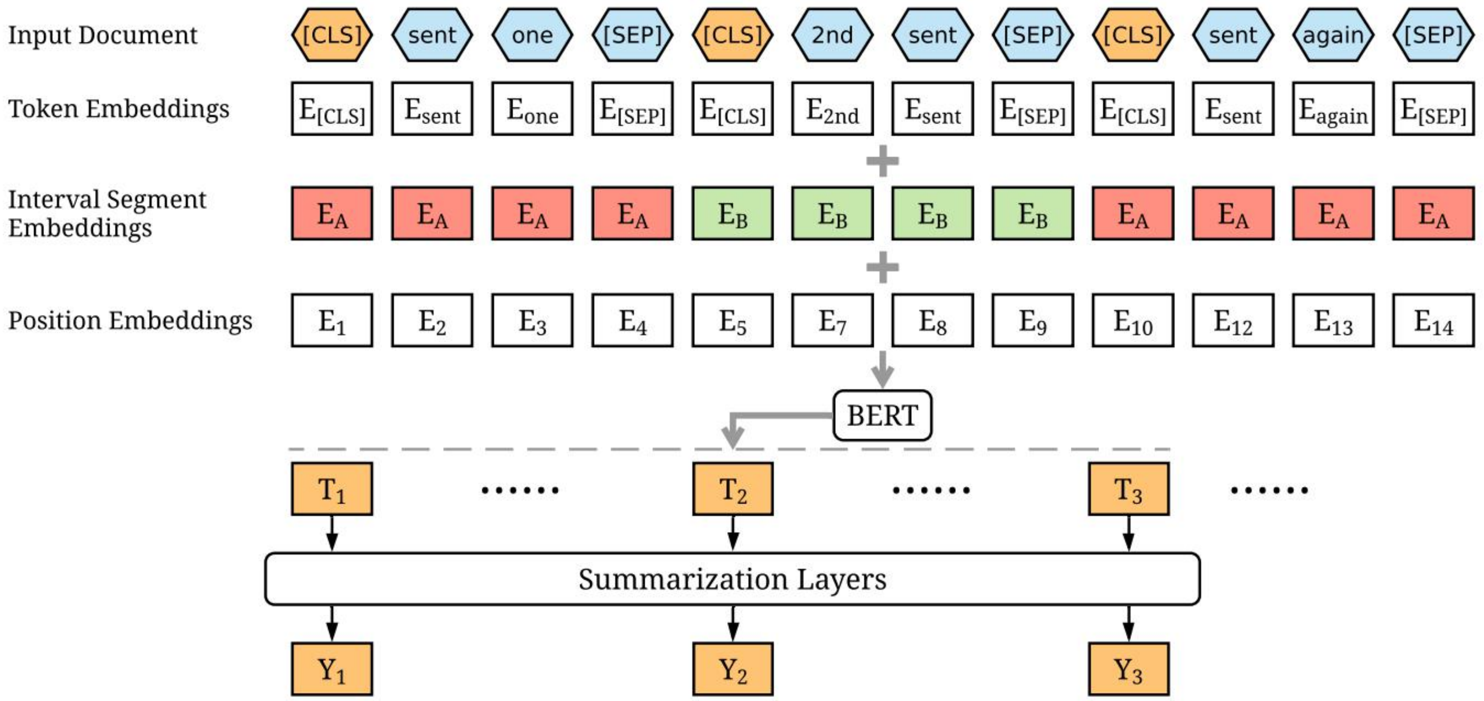
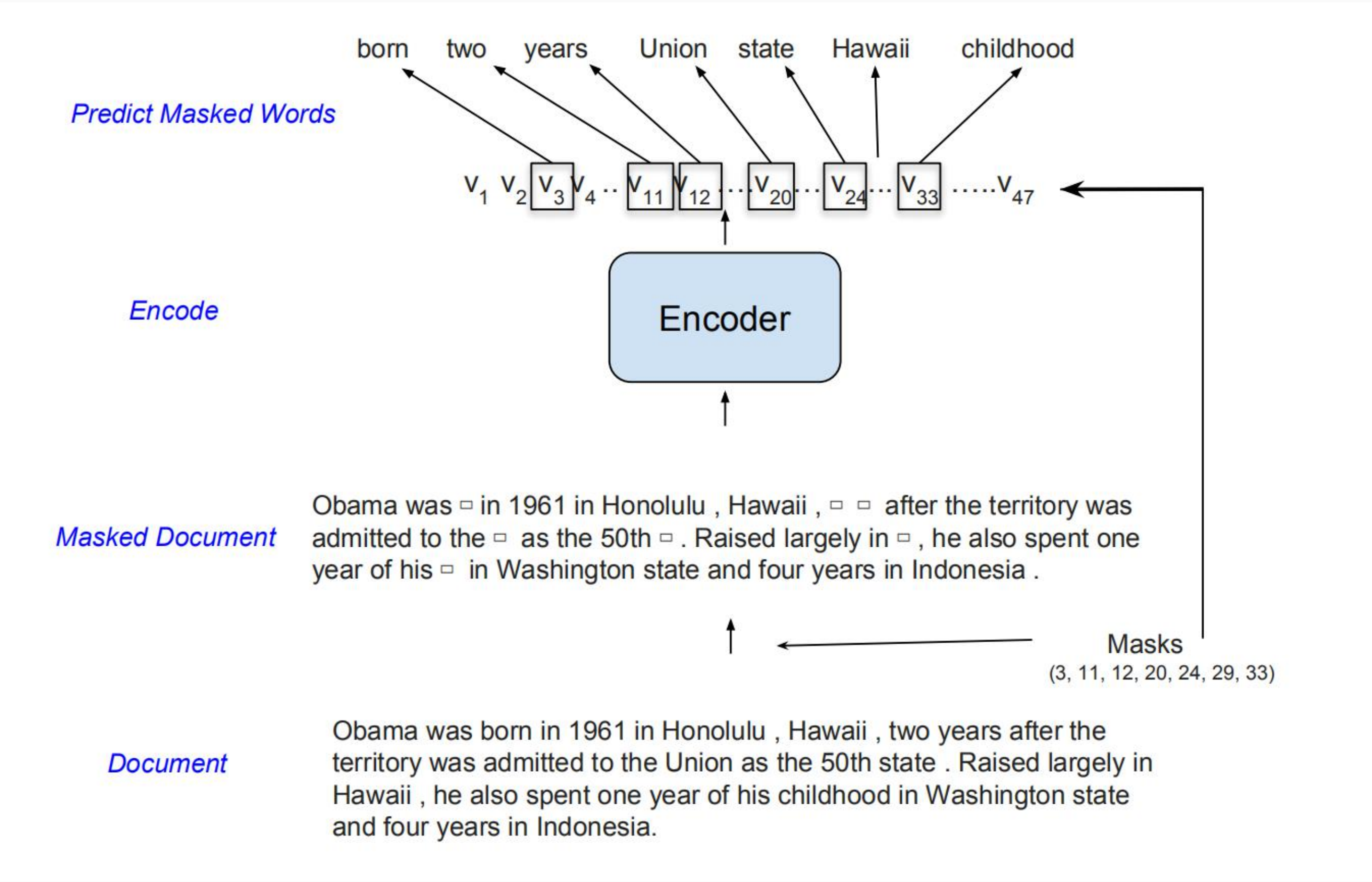
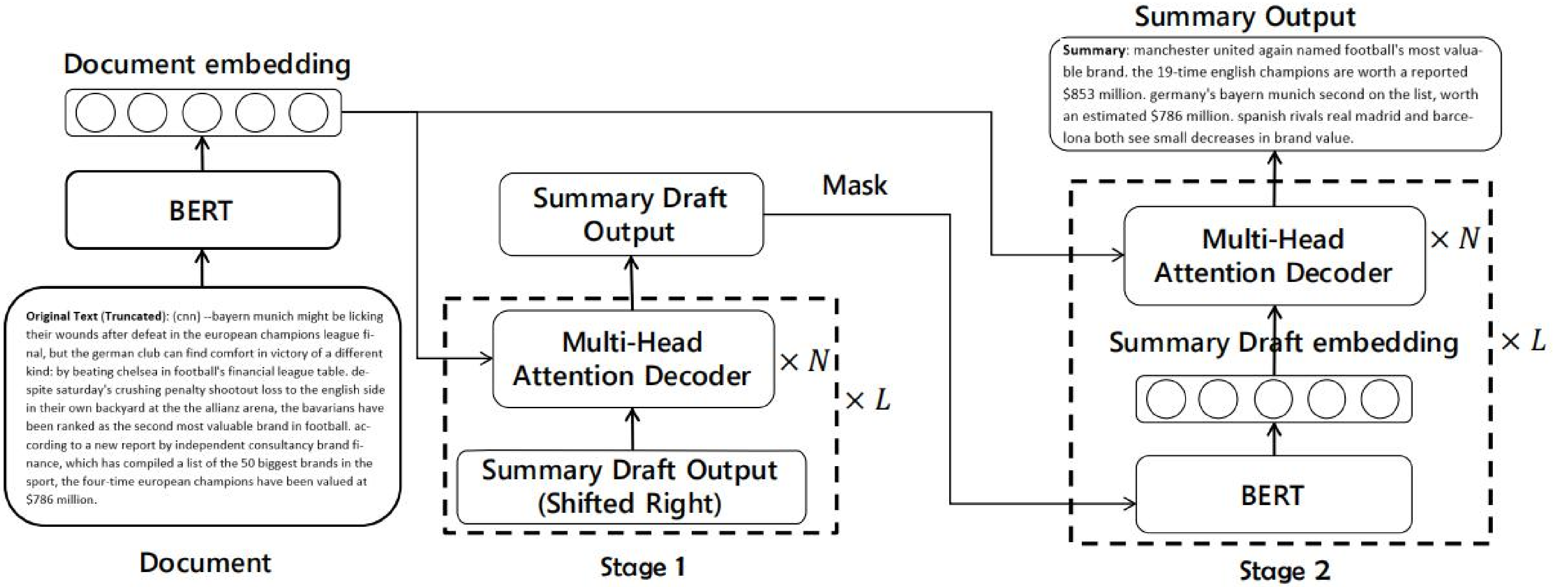


Figure 1: The overview architecture of the BERTSUM model.

Hierarchical Document Representations



Two-stage refined method



References

1. Vaswani A, Shazeer N, Parmar N, et al. Attention Is All You Need. arXiv:170603762 [cs]. June 2017. http://arxiv.org/abs/1706.03762. Accessed April 23, 2019.
2. Devlin J, Chang M-W, Lee K, Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:181004805 [cs]. October 2018. http://arxiv.org/abs/1810.04805. Accessed April 23, 2019.
3. Peters ME, Neumann M, Iyyer M, et al. Deep contextualized word representations. arXiv:180205365 [cs]. February 2018. http://arxiv.org/abs/1802.05365. Accessed April 23, 2019.
4. Liu Y. Fine-tune BERT for Extractive Summarization. arXiv:190310318 [cs]. March 2019. http://arxiv.org/abs/1903.10318. Accessed April 8, 2019.
5. Chang M-W, Toutanova K, Lee K, Devlin J. Language Model Pre-training for Hierarchical Document Representations. arXiv:190109128 [cs]. January 2019. http://arxiv.org/abs/1901.09128. Accessed April 8, 2019.
6. Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I. Language Models are Unsupervised Multitask Learners. :24.
8. Zhang H, Gong Y, Yan Y, et al. Pretraining-Based Natural Language Generation for Text Summarization. arXiv:190209243 [cs]. February 2019. http://arxiv.org/abs/1902.09243. Accessed April 8, 2019.