Modern NLP through practical problems

Andrej Miščič, Luka Vranješ

Agenda

00 Theoretical introduction:

- Transformer architecture
- brief overview of pretraining
- introduction to Hugging Face ecosystem

01 Practical part:

- Sentiment analysis with BERT
- Named Entity Recognition with BERT
- Abstractive summarization with BART

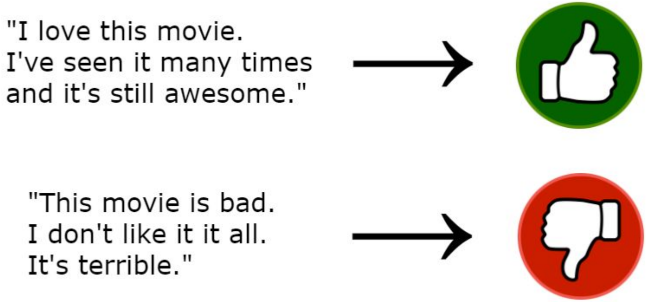
Natural Language Processing

question answering

What’s the capital of Serbia?

Beograd

text classification



named-entity recognition

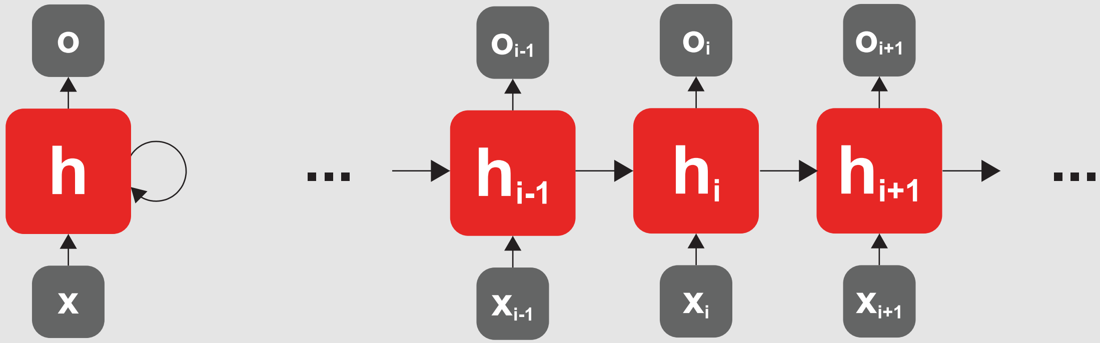
We are Andrej and Luka. We work for Valira AI.

RNNs

- extension of NNs for sequential data;

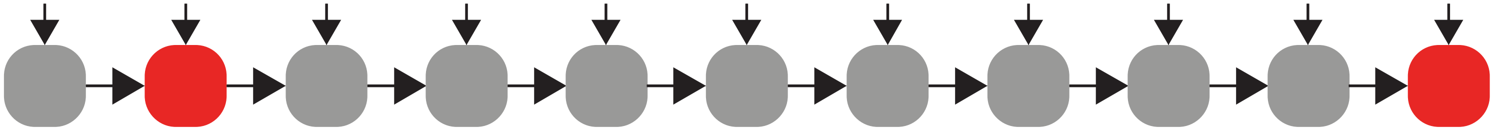
- information persists in hidden state h_i.

- improvement: LSTMs



Motivation for Transformers

- RNNs are inherently sequential which prevents parallelization;
- the problem of long-term dependencies:
    - gating somewhat mitigates this problem, however, the path length between any two dependant words is still O(n)

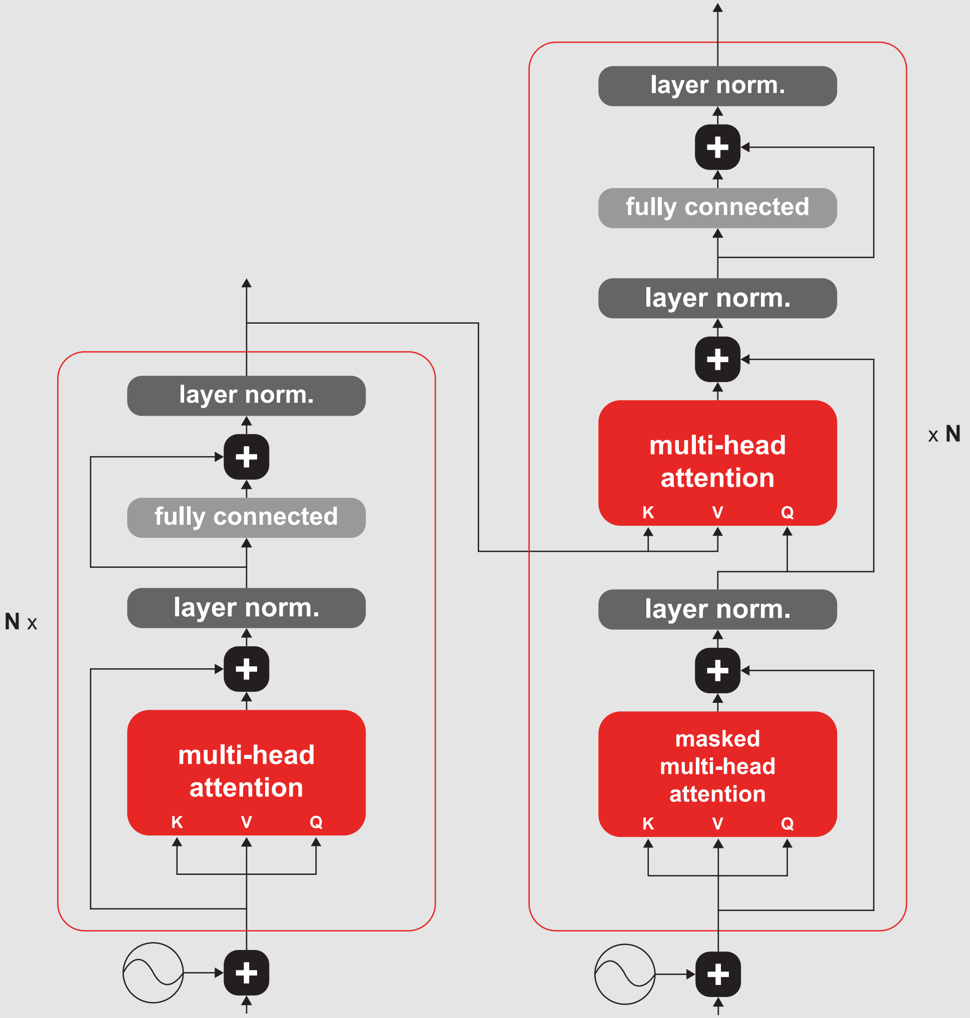


- Can we get rid of recurrence? What to replace it with?

Transformer[1]

- introduced for Neural Machine Translation;

- uses **self-attention** in place of recurrence.



[1] Vaswani et al.: Attention Is All You Need, 2017

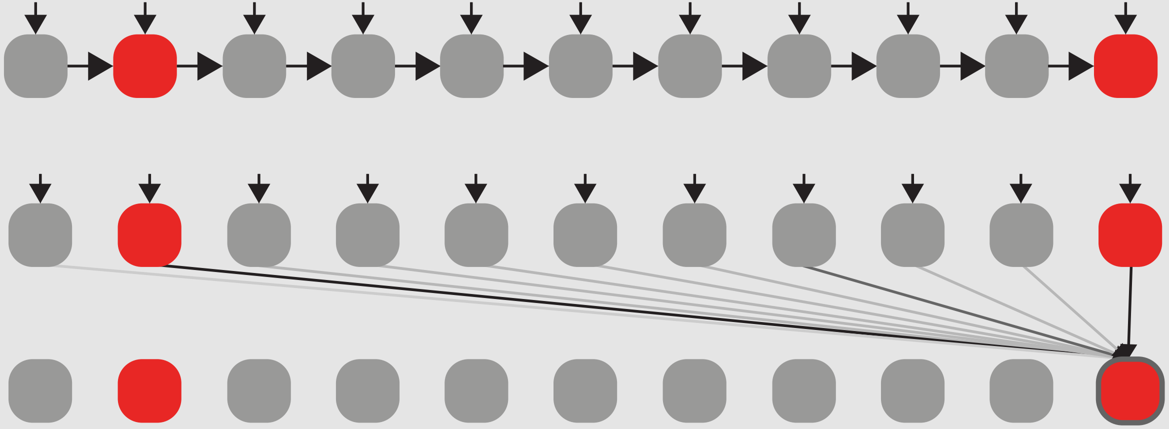
Self-attention

- RNN: path length between two words is O(n);

- in self-attention the path length is O(1).

The animal didn't cross the street because it was too wide.

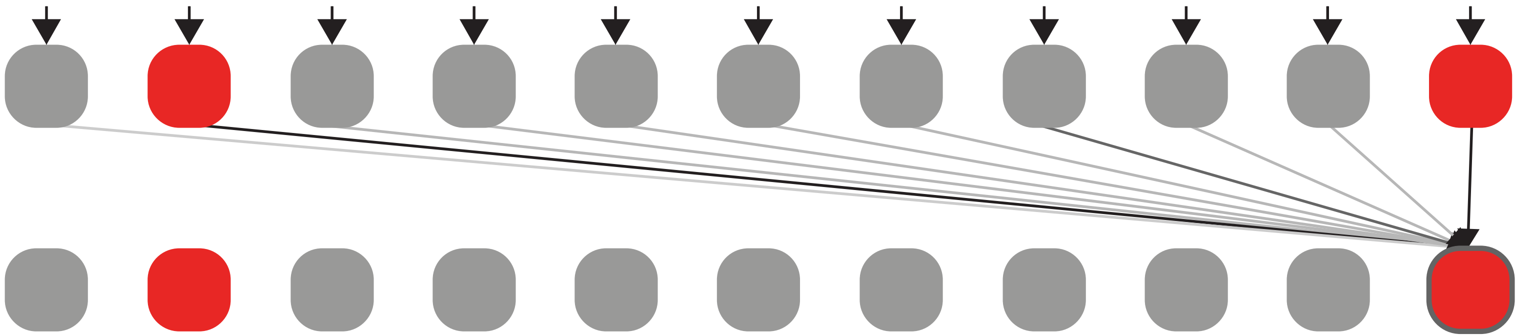
The animal didn't cross the street because it was too tired.



Example credit: Hang Dong.

Problem

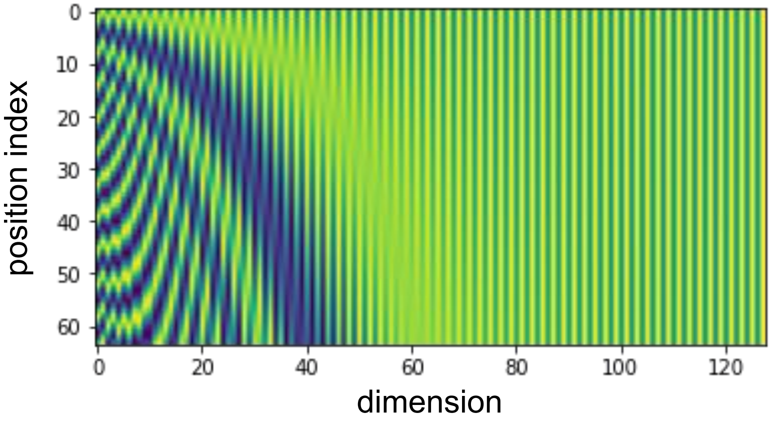
- by getting rid of recurrence we lose positional information which is important as our data is sequential;
- self-attention is permutation invariant, i.e. no matter the order of the inputs, the output will be the same.



Solution: Positional encodings

- positional encodings have the same dimension as input embeddings and are added to them before the first self-attention layer;
- they can be either:
    - LEARNED: use an embedding layer to learn a pos. embedding for each position in the sequence;
    - FIXED: set before training, used in original paper.

$$PE_{ij} = \begin{cases} \sin(i/10000^{\frac{j}{d_m}}) & \text{if j is even} \\ \cos(i/10000^{\frac{j-1}{d_m}}) & \text{if j is odd} \end{cases}$$



Pretraining

- deep learning requires lots of annotated data, which can be scarce;
- on the other hand, we have abundant unlabeled text data;

↓

- leverage this unlabeled data to pre-train word representations/models in a self-supervised manner and use them on downstream tasks.

Pretraining

- neural word embeddings, e.g. Word2Vec [1], Glove [2], (context-free);

- transfer learning (pretrain-then-finetune)
    - can we develop models that adapt to many NLP tasks with little to no modification?
    - BERT [3], T5 [4], BART [5]

[1] Mikolov et al.: Efficient Estimation of Word Representations in Vector Space, 2013.
[2] Pennington et al.: GloVe: Global Vectors for Word Representation, 2014.
[3] Devlin et al.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, 2018.
[4] Raffel et al.: Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer, 2019.
[5] Lewis et al.: BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension, 2019.

NLP problems examples

Natural Language Understanding

- text classification
- named-entity recognition
- reading comprehension
- etc.

encoder-only arch.

e.g. BERT, RoBERTa

Natural Language Generation

- machine translation
- abstractive summarization
- etc.

encoder-decoder arch.

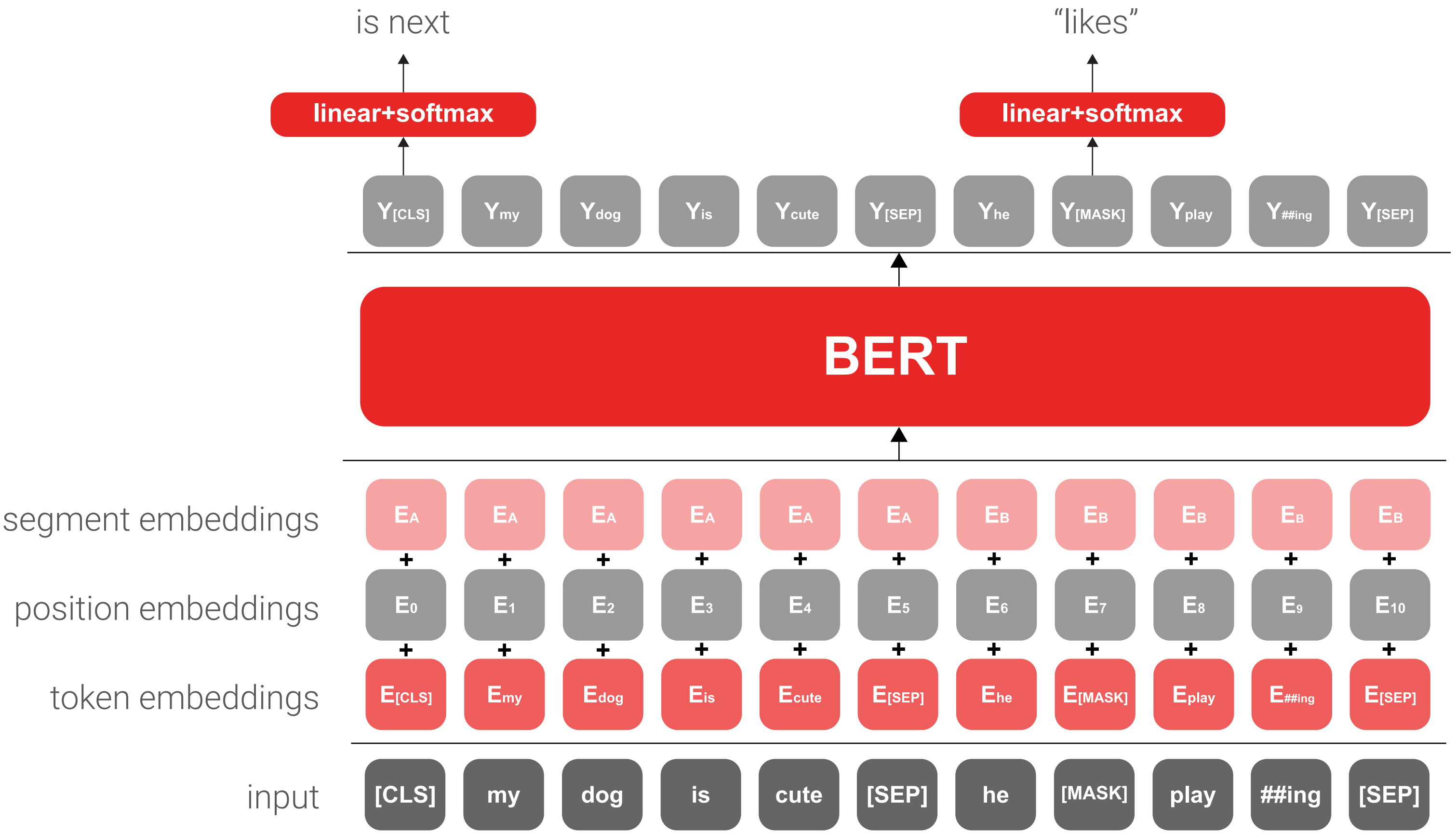
e.g. T5, BART

Pretraining objectives

- encoder-only models, i.e. BERT-like:
    - masked language modelling

- encoder-decoder models, e.g. T5:
    - span corruption, masking multiple consecutive tokens, the model generates them

Pretraining



Questions?

feel free to write to us, connect on LinkedIn or say hi in Beograd

andrej.miscic@valira.ai

luka.vranjes@valira.ai