Deep Learning for NLP
Lecture 8: Encoder-Decoder Models

Dr. Mohsen Mesgar

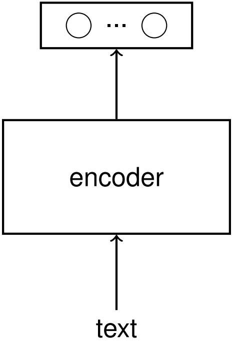
Ubiquitous Knowledge Processing Lab (UKP Lab)

This lecture

- Encoder-Decoder
- Attention
- Their applications in NLP

Encoder

A neural model to transform a text into a vector in an embedding space.

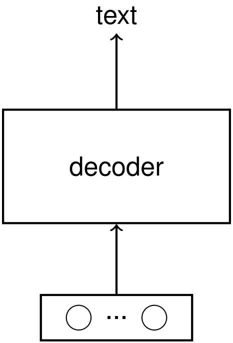


Different types of neural encoders are

- pretrained word embeddings
- MLPs, CNNs, RNNs, Transformers, ...

Decoder

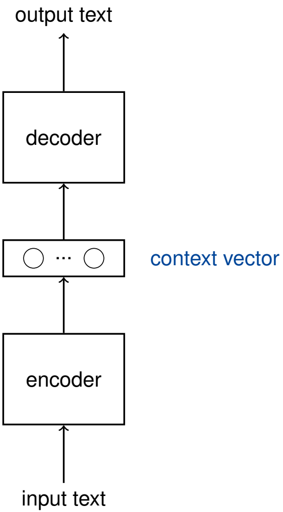
A neural model to transform a vector from an embedding space to a text.



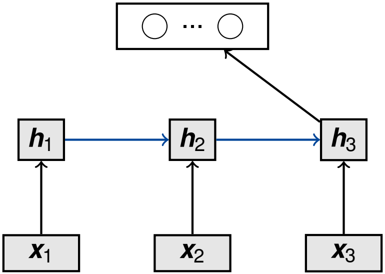
Different types of language models can be used as decoders

- RNN-based LMs
- Transformer-based LMs

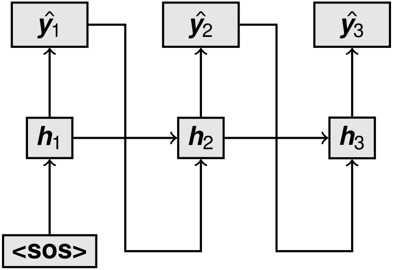
Encoder-Decoder



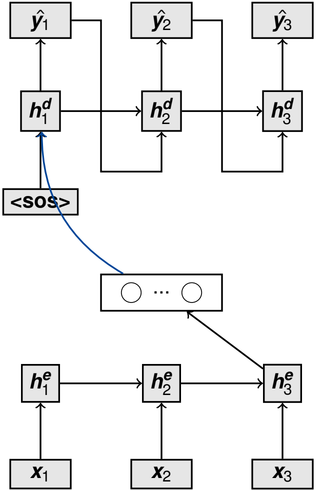
RNN Encoder



RNN Decoder



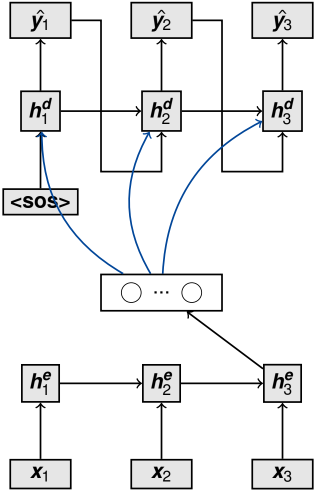
RNN-based Encoder-Decoder



RNN-based Encoder-Decoder

- The context vector is used to initialize the hidden state of the decoder.
- Its impact vanishes at the last steps of the decoder.

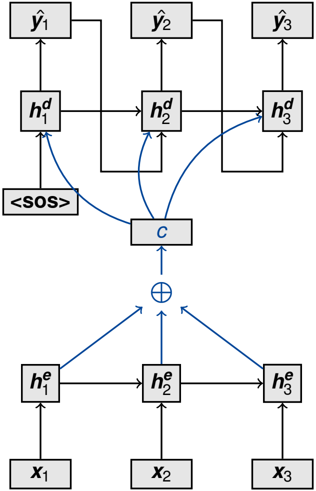
RNN-based Encoder-Decoder



RNN-based Encoder-Decoder

- The output of the encoder is known as the context vector.
- The dimensionality of the context vector is fixed.
- However, different input texts might have different length.
- So, considering the hidden state of the RNN encoder may not capture the entire input text.
- This is a problem especially for long input texts.

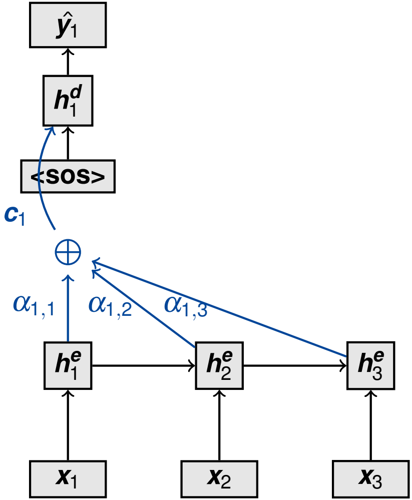
RNN-based Encoder-Decoder



RNN-based Encoder-Decoder

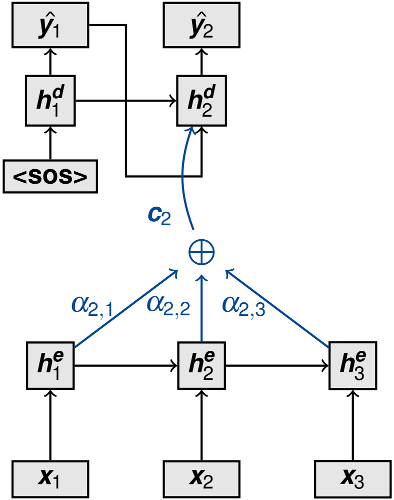
- The other problem is that context vector is unique for all decoding steps.
- The encoder treats all tokens of the input sentence equally important to produce a context vector.
- However, at any decoding step, the decoder should focus on tokens of the input sentence differently.

Attention

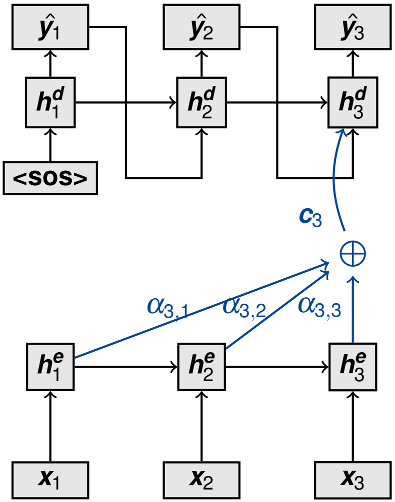


$\hat{y}_1$
h^d_1
<sos>
c_1
$\oplus$
$\alpha_{1,1}$
$\alpha_{1,2}$
$\alpha_{1,3}$
h^e_1
h^e_2
h^e_3
x_1
x_2
x_3

Attention


$\hat{y}_1$
$\hat{y}_2$
h_1^d
h_2^d
<SOS>
c_2
$\alpha_{2,1}$
$\alpha_{2,2}$
$\alpha_{2,3}$
h_1^e
h_2^e
h_3^e
x_1
x_2
x_3

Attention



$\hat{y}_1$
$\hat{y}_2$
$\hat{y}_3$
h_1^d
h_2^d
h_3^d
<SOS>
c_3
$\oplus$
$\alpha_{3,1}$
$\alpha_{3,2}$
$\alpha_{3,3}$
h_1^e
h_2^e
h_3^e
x_1
x_2
x_3

Attention

$$\boldsymbol{c_t} = \sum_{k=1}^{N} \alpha_{t,k} \boldsymbol{h_k^e}$$

$$\alpha_{t,k} = \frac{\texttt{exp}(\text{score}(\boldsymbol{h}_{t-1}^{\boldsymbol{d}}, \boldsymbol{h}_k^{\boldsymbol{e}}))}{\sum_{k'=1}^{N} \texttt{exp}(\text{score}(\boldsymbol{h}_{t-1}^{\boldsymbol{d}}, \boldsymbol{h}_{k'}^{\boldsymbol{e}}))}$$

Content-based Attention (Graves et al., 2014)

$$\text{score}(\boldsymbol{h}_t^{\boldsymbol{d}}, \boldsymbol{h}_k^{\boldsymbol{e}}) = \texttt{cosine}(\boldsymbol{h}_t^{\boldsymbol{d}}, \boldsymbol{h}_k^{\boldsymbol{e}})$$

Additive Attention (Bahdanau et al., 2015)

$$\text{score}(\boldsymbol{h}_t^{\boldsymbol{d}}, \boldsymbol{h}_k^{\boldsymbol{e}}) = \texttt{tanh}\left([\boldsymbol{h}_{\boldsymbol{t}}^{\boldsymbol{d}}; \boldsymbol{h}_{\boldsymbol{k}}^{\boldsymbol{e}}]\boldsymbol{W}^{(h)}\right)\boldsymbol{W}^{(s)}$$

Location-based Attention (Luong et al., 2015)

$$\text{score}(\boldsymbol{h}_t^{\boldsymbol{d}}, \boldsymbol{h}_k^{\boldsymbol{e}}) = \texttt{softmax}\left(\boldsymbol{h_t^d} \boldsymbol{W}^{(s)}\right)$$

Scaled Dot-Product Attention

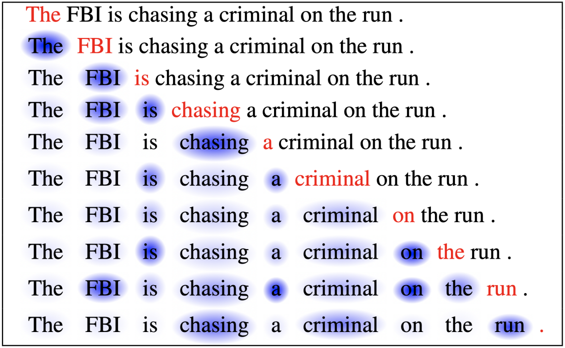
(Vaswani et al., 2017)

$$\text{score}(\boldsymbol{h}_t^{\boldsymbol{d}}, \boldsymbol{h}_k^{\boldsymbol{e}}) = \frac{\boldsymbol{h}_k^{\boldsymbol{e}} \texttt{trans}(\boldsymbol{h}_{\boldsymbol{t}}^{\boldsymbol{d}})}{\sqrt{n}}$$

- The scaling factor $\frac{1}{\sqrt{n}}$ is motivated by the concern when the input is large, the softmax function may have an extremely small gradient.
- Small gradients yields difficulties in learning.

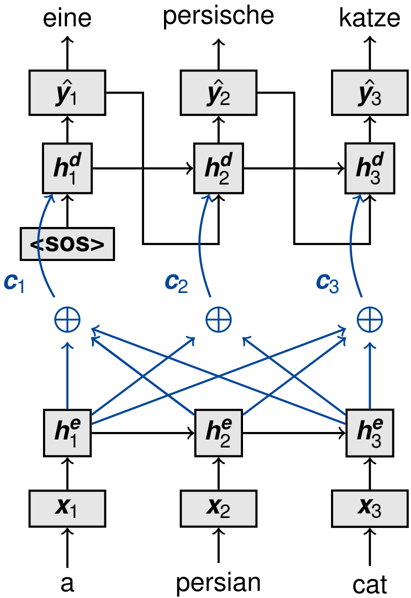
Self-Attention

- An attention mechanism to relate different tokens of an input sequence to compute a representation of the sequence itself.
- For example, the self-attention mechanism enables a model to learn the relations between a word of an input sentence and its previous words.



(Taken from Cheng et al., (2016))

Neural Machine Translation



Summarization (Rush et al. 2015)

Input $(\mathbf{x}_1, \ldots, \mathbf{x}_{18})$. First sentence of article:
russian defense minister ivanov called sunday for the creation of a joint front for combating global terrorism

Output $(\mathbf{y}_1, \ldots, \mathbf{y}_8)$. Generated headline:
russia calls for joint front against ***terrorism*** $\Leftarrow$ $g(\textit{terrorism}, \mathbf{x}, \textit{for, joint, front, against})$

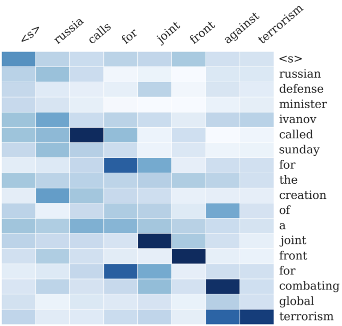
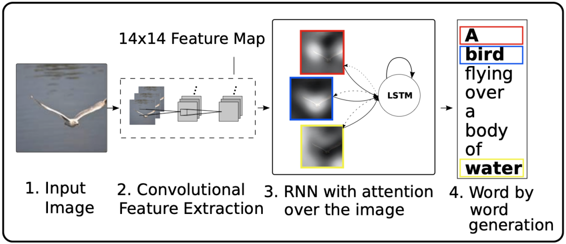


Image Caption Generation (Xu et al. 2015)



Other Applications

- lemmatizatoin: g e s p i e l t $\longrightarrow$ s p i e l e n
- Spelling correction: i _ l v o e _ u $\longrightarrow$ i _ l o v e _ y o u

Auto Encoder

- An encoder-decoder model that transforms an input sequence to itself.
- It learns the identity function F(x) = x.
- It usually add some noise to the input, then the model learns to remove the noise.
- It is used for dimensionality reduction, representation learning, and unsupervised learning.
- The encoder and decoder can be used individually to solve other tasks.

Summary

- Encoders and Decoders
- Attention mechanism
- Their applications in NLP

Thank You!