UNIVERSITAT POLITÈCNICA DE VALÈNCIA

MIARFID Official Master's Degree in Artificial Intelligence, Pattern Recognition and Digital Imaging

Chapter 3-2. Sequence-to-Sequence Neural Networks

Neural Networks

2023/2024

Máster Universitario en Inteligencia Artificial, Reconocimiento de Formas e Imagen Digital

Departamento de Sistemas Informáticos y Computación

Index

1 Sequence-to-sequence architectures ▷ 3

2 Pre-trained models ▷ 35

3 Applications ▷ 54

4 Bibliography ▷ 69

Index

- 1 *Sequence-to-sequence architectures* ▷ 3
- 2 Pre-trained models ▷ 35
- 3 Applications ▷ 54
- 4 Bibliography ▷ 69

Sequence-to-sequence problems

1. In many problems, the length of the output sequence is different from the length of the input sequence.

2. In recurrent neural and bounded-memory networks the length of the output sequence is equal to the length of the input sequence (or less in CTC)

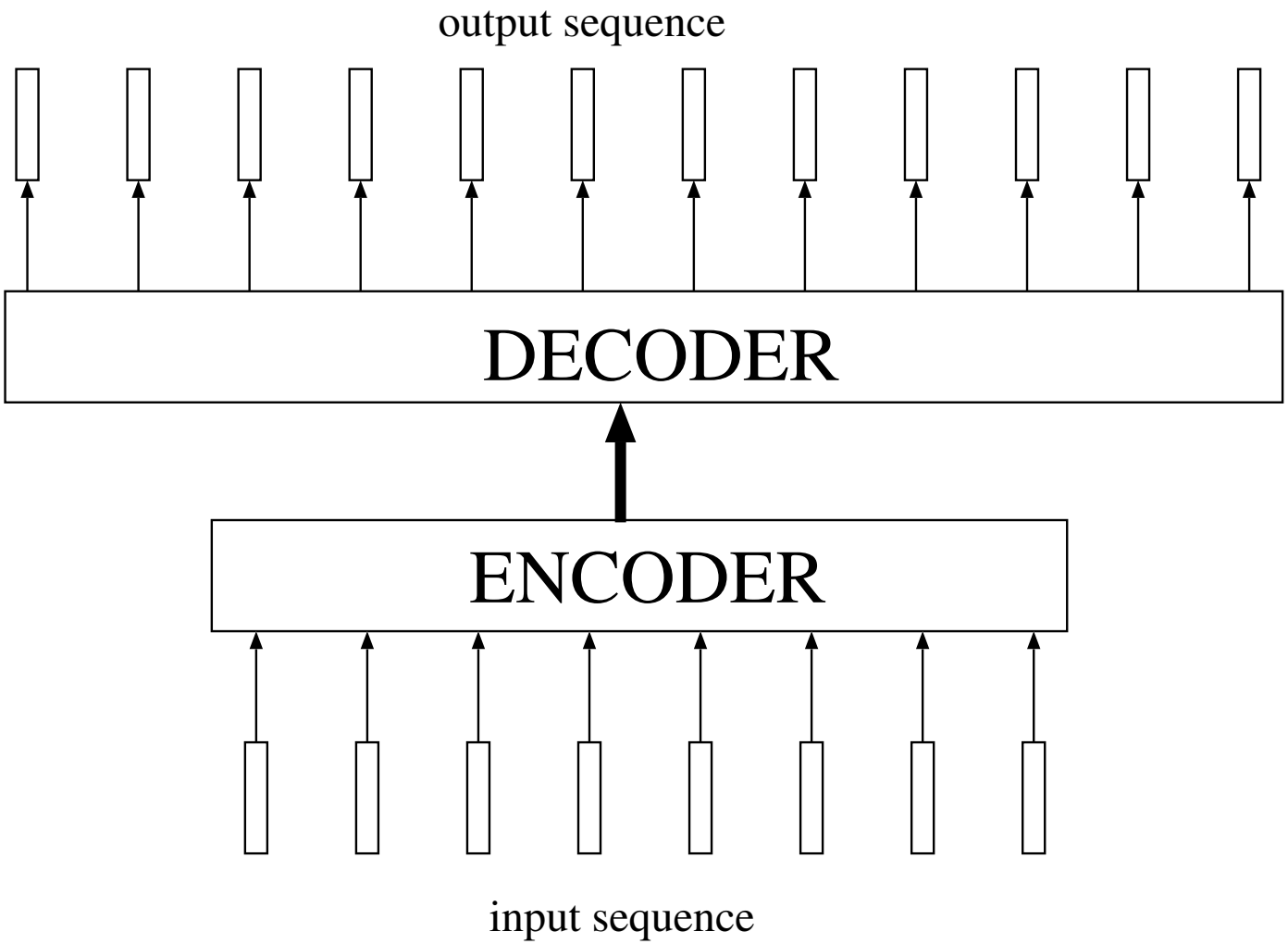
3. The encoder-decoder neural architectures constitute a successful approach to deal with sequences of different lengths.

Encoder-decoder architecture

1. Obtain a compact representation of the input sequence using an encoder neural network.

2. Generate an output sequence from the compact representation of the input sequence using a decoder neural network.

Encoder-decoder architecture



Encoder-decoder architecture

1. Project each input sequence into a sequence of continuous vectors $\rightarrow$ input embedding.

2. Obtain a contextual representation of the input sequence using an encoder neural network.

3. Generate an output sequence (typically as a-posteriori probabilistic distributions) from the contextual representation of the input sequence using a decoder neural network.

4. If the decoder is autoregresive, encode each output $\rightarrow$ output embedding.

Encoder-decoder architecture

LSTMs

Encoder-decoder architecture using LSTM (Bahdanau et al. 2015)

- The RNN-based encoder ($\mathbf{F}_e$) converts an input source token sequence $x_1, \ldots, x_J \equiv x_1^J$ into a state sequence $\mathbf{h}_1^e, \ldots, \mathbf{h}_J^e$:

$$\mathbf{h}_0^e = \mathbf{0}; \qquad \mathbf{h}_j^e = \mathbf{F}_e(\mathbf{W}_E(x_j), \mathbf{h}_{j-1}^e) = \mathbf{F}_e(\mathbf{x}_j, \mathbf{h}_{j-1}^e) \quad 1 \leq j \leq J$$

- The state sequence is converted into a vector $\mathbf{u} = \mathbf{u}(x_1^J)$:

$$\mathbf{u} = \mathbf{a}(\mathbf{h}_1^e, \ldots, \mathbf{h}_J^e) \stackrel{\triangle}{=} \mathbf{h}_J^e$$

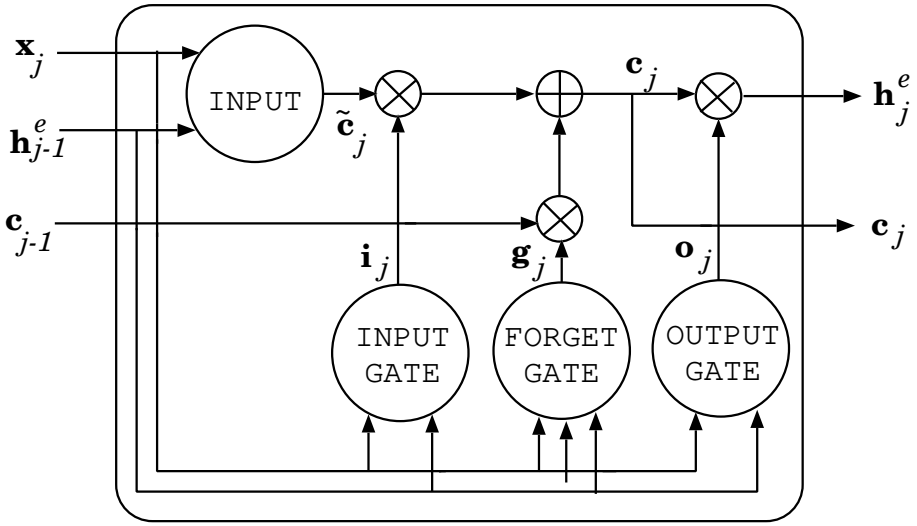
- The RNN-based decoder ($\mathbf{F}_d$) is a kind of target language model

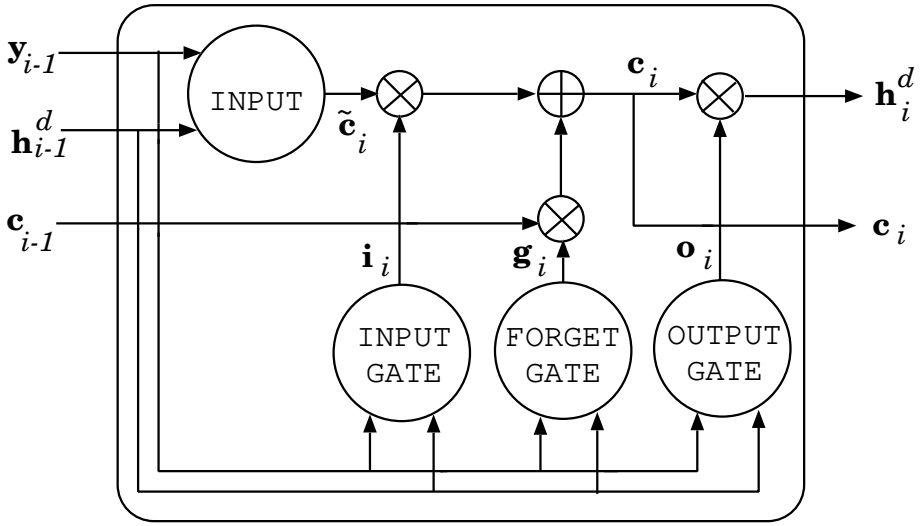
$$\mathbf{h}_0^d = \mathbf{u}\,; \qquad \mathbf{h}_i^d = \mathbf{F}_d(\mathbf{W}_E(y_{i-1}), \mathbf{h}_{i-1}^d) = \mathbf{F}_d(\mathbf{y}_{i-1}, \mathbf{h}_{i-1}^d) \quad 1 \leq i \leq I$$

- The output token sequence is obtained as:

$$p(y_1^I \mid x_1^J) = \prod_{i=1}^{I} p(y_i \mid y_1^{i-1}, u(x_1^J)) = \prod_{i=1}^{I} \mathbf{f}_{sm}(\mathbf{W}_O\ \mathbf{h}_i^d)_{i(y_i)}$$

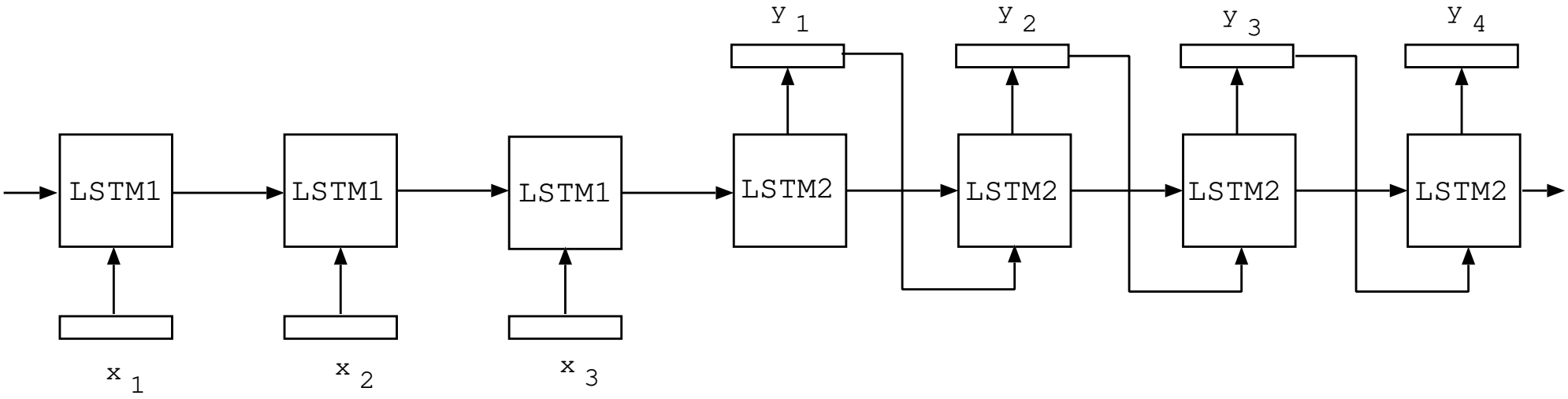
Long Short-Term Memory (LSTM)





$$\mathbf{h}^e_j = \mathbf{F}(\mathbf{x}_j, \mathbf{h}^e_{j-1}) \quad 1 \leq j \leq J \qquad \mathbf{h}^d_i = \mathbf{F}(\mathbf{y}_{i-1}, \mathbf{h}^e_{i-1}) \quad 1 \leq i \leq I \text{ with } \mathbf{h}^d_0 = \mathbf{u}$$

Encoder-decoder architecture using LSTM



Attention models

Given a sequence of encoder states $\mathbf{h}_j^e$ for $1 \leq j \leq J$ and a decoder state $\mathbf{h}_{i-1}^d$ for $1 \leq i \leq I$ an attention model for the decoder state $\mathbf{h}_i^d$ is a function:

$$\mathbf{u}_i = \mathbf{a}(\mathbf{h}_1^e, \ldots, \mathbf{h}_J^e, \mathbf{h}_{i-1}^d)$$

where $\mathbf{a}$ is:

$$\mathbf{u}_i = \sum_{j=1}^{J} f_j(\mathbf{h}_1^e, \ldots, \mathbf{h}_J^e, \mathbf{h}_{i-1}^d)\ \mathbf{W}_V \mathbf{h}_j^e$$

f_j is a softmax:

$$f_j(\mathbf{h}_1^e, \ldots, \mathbf{h}_J^e, \mathbf{h}_{i-1}^d) = \frac{\exp(\mathbf{W}_Q \mathbf{h}_{i-1}^d \cdot \mathbf{W}_K\ \mathbf{h}_j^e))}{\sum_{j'=1}^{J} \exp(\mathbf{W}_Q \mathbf{h}_{i-1}^d \cdot \mathbf{W}_K\ \mathbf{h}_{j'}^e))} \qquad 1 \leq j \leq J$$

Encoder-decoder architecture using LSTMs and attention

(Bahdanau et al. 2015)

- The RNN-based encoder ($\mathbf{F}_e$) converts an input source token sequence $x_1, \ldots, x_J \equiv x_1^J$ into a state sequence $\mathbf{h}_1^e, \ldots, \mathbf{h}_J^e$:

$$\mathbf{h}_0^e = \mathbf{0}; \qquad \mathbf{h}_j^e = \mathbf{F}_e(\mathbf{W}_E(x_j), \mathbf{h}_{j-1}^e) = \mathbf{F}_e(\mathbf{x}_j, \mathbf{h}_{j-1}^e) \quad 1 \leq j \leq J$$

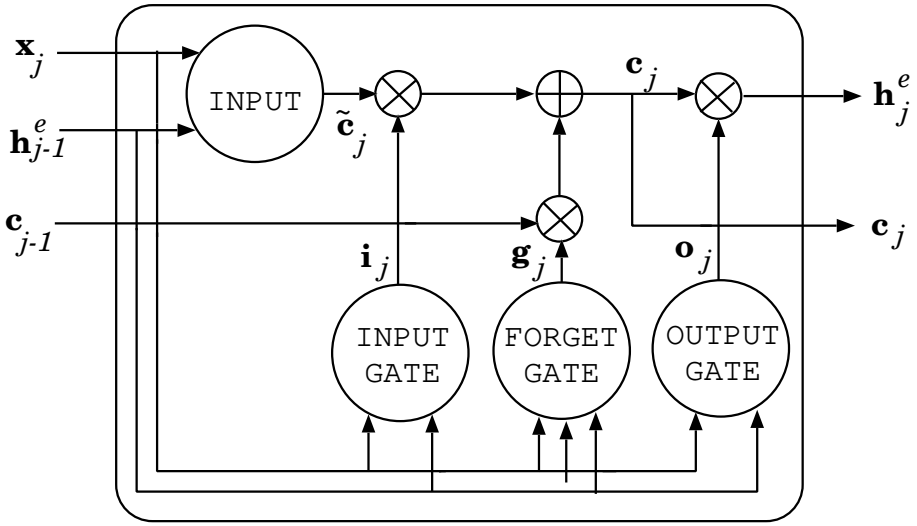
- The RNN-based decoder ($\mathbf{F}_d$) is a kind of target language model

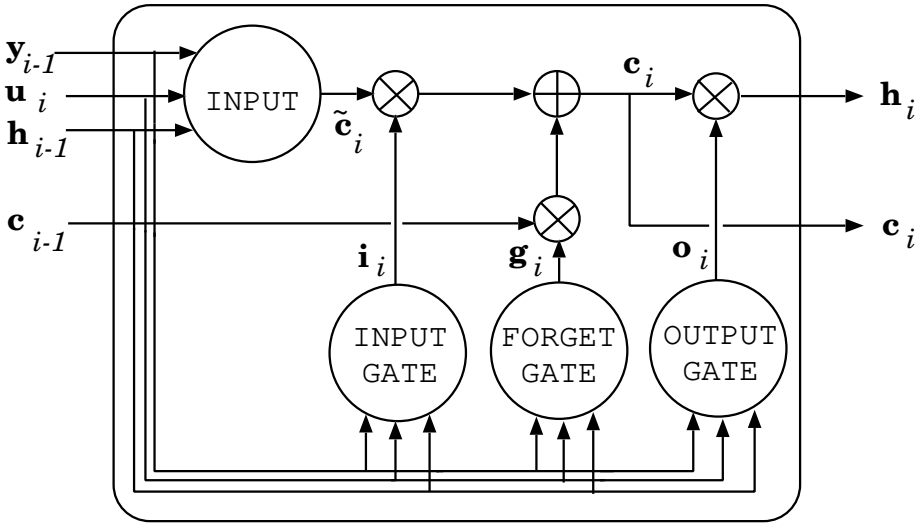
$$\begin{aligned} \mathbf{h}_0^d &= \mathbf{0} \\ \mathbf{u}_i &= \mathbf{a}(\mathbf{h}_1^e, \ldots, \mathbf{h}_J^e, \mathbf{h}_{i-1}^d) \\ \mathbf{h}_i^d &= \mathbf{F}_d(\mathbf{W}_E(y_{i-1}), \mathbf{h}_{i-1}^d \mathbf{u}_i) = \mathbf{F}_d(\mathbf{y}_{i-1}, \mathbf{h}_{i-1}^d, \mathbf{u}_i) \quad 1 \leq i \leq I \end{aligned}$$

- The output token sequence is obtained as:

$$p(y_1^I \mid x_1^J) = \prod_{i=1}^{I} p(y_i \mid y_1^{i-1}, u(x_1^J)) = \prod_{i=1}^{I} \mathbf{f}_{sm}(\mathbf{W}_O \ \mathbf{h}_i^d)_{i(y_i)}$$

Long Short-Term Memory (LSTM)

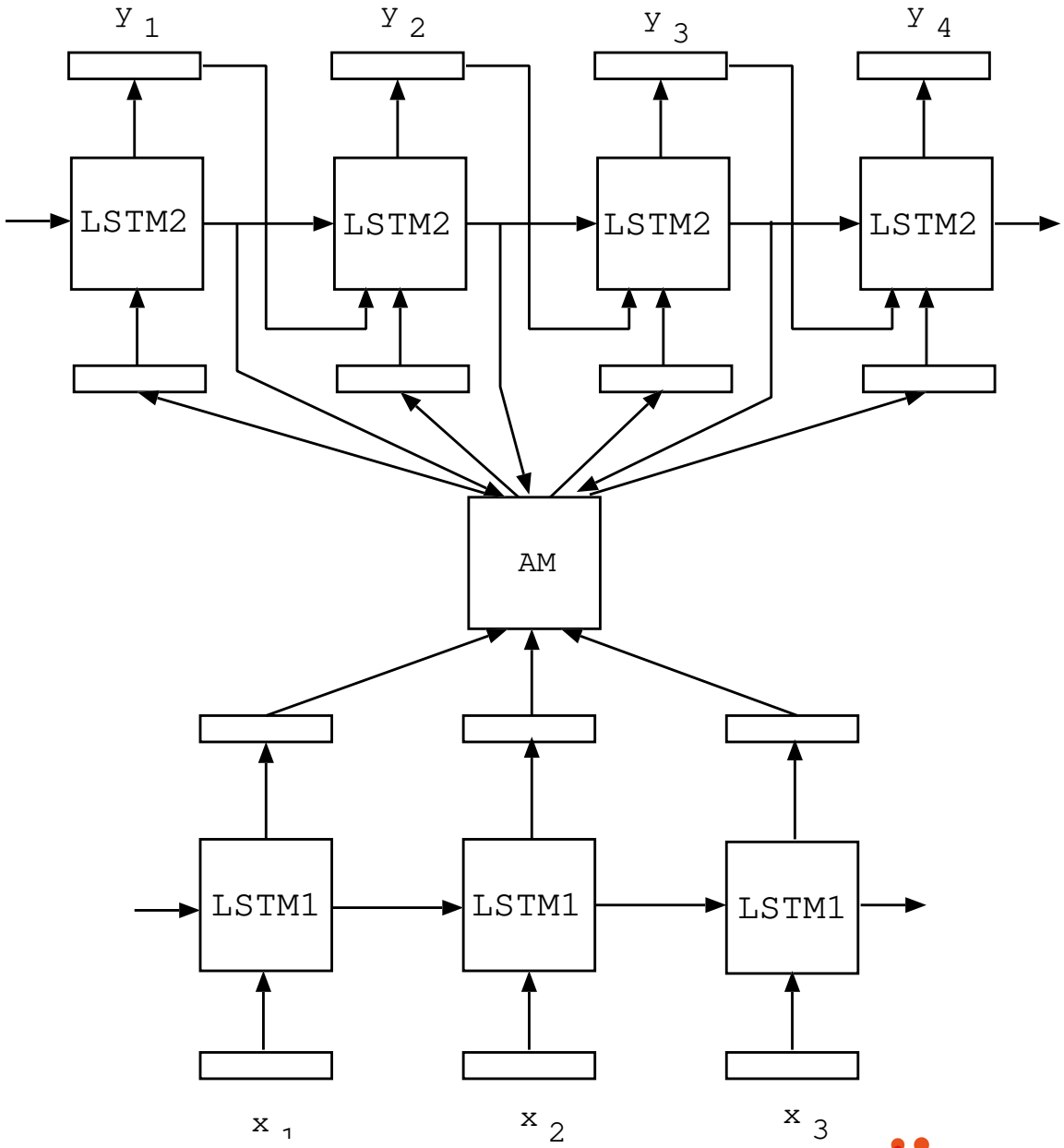




$$\mathbf{h}^e_j = \mathbf{F}(\mathbf{x}_j, \mathbf{h}^e_{j-1}) \qquad 1 \leq j \leq J$$

$$\mathbf{h}^d_i = \mathbf{F}(\mathbf{y}_{i-1}, \mathbf{h}^e_{i-1}, \mathbf{u}_i) \qquad 1 \leq i \leq I$$

Encoder-decoder + attention model architecture



Other encoder-decoder architectures

- Multilayer encoder / multilayer decoder
- Conditional recurrent units (RNN blocks with attention models in between)
- GRUs

Transformer

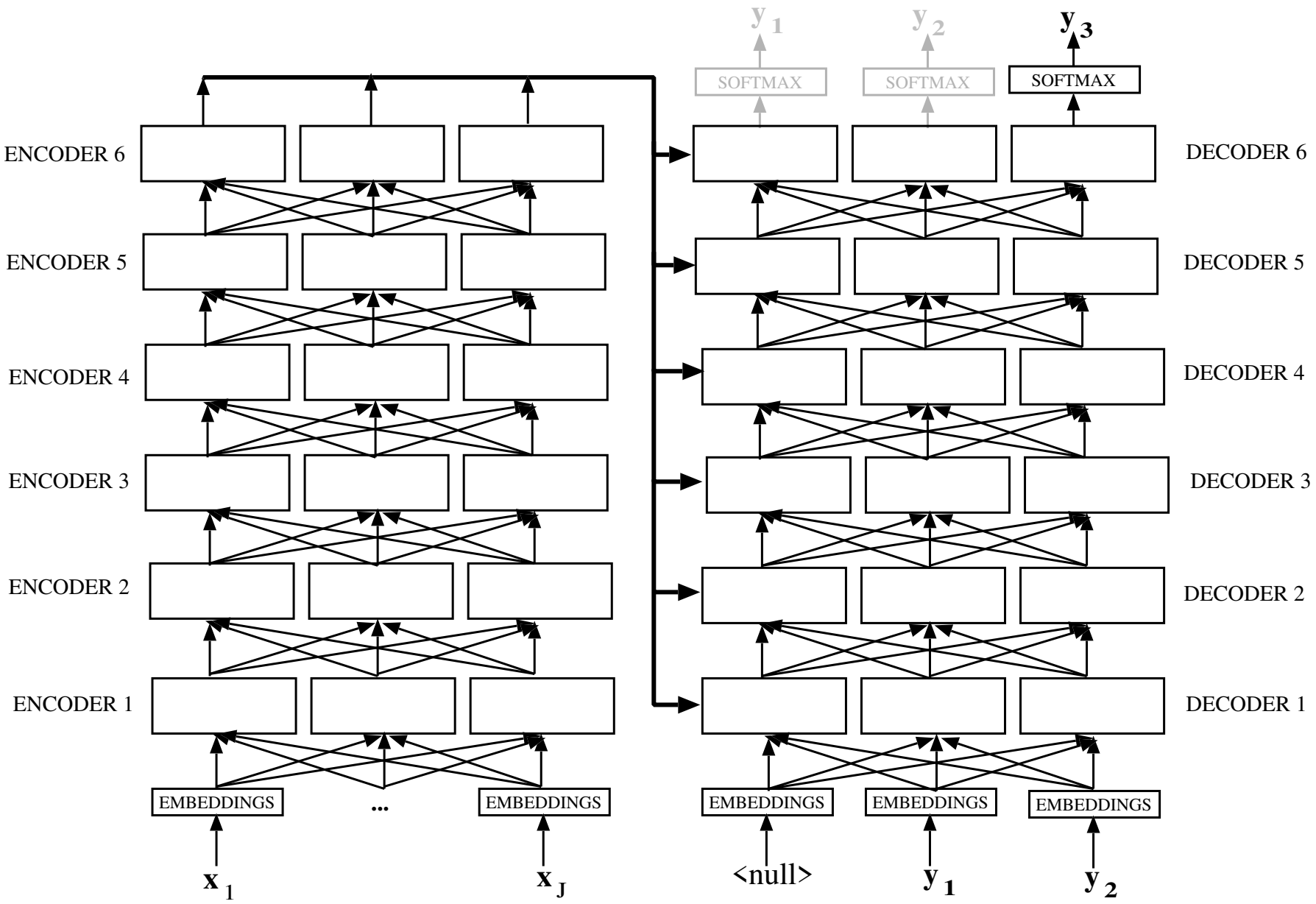
Transformer (Vaswani 2017)

- Goal: Given an input token sequence x_1^J and an output token sequence y_1^I, compute:

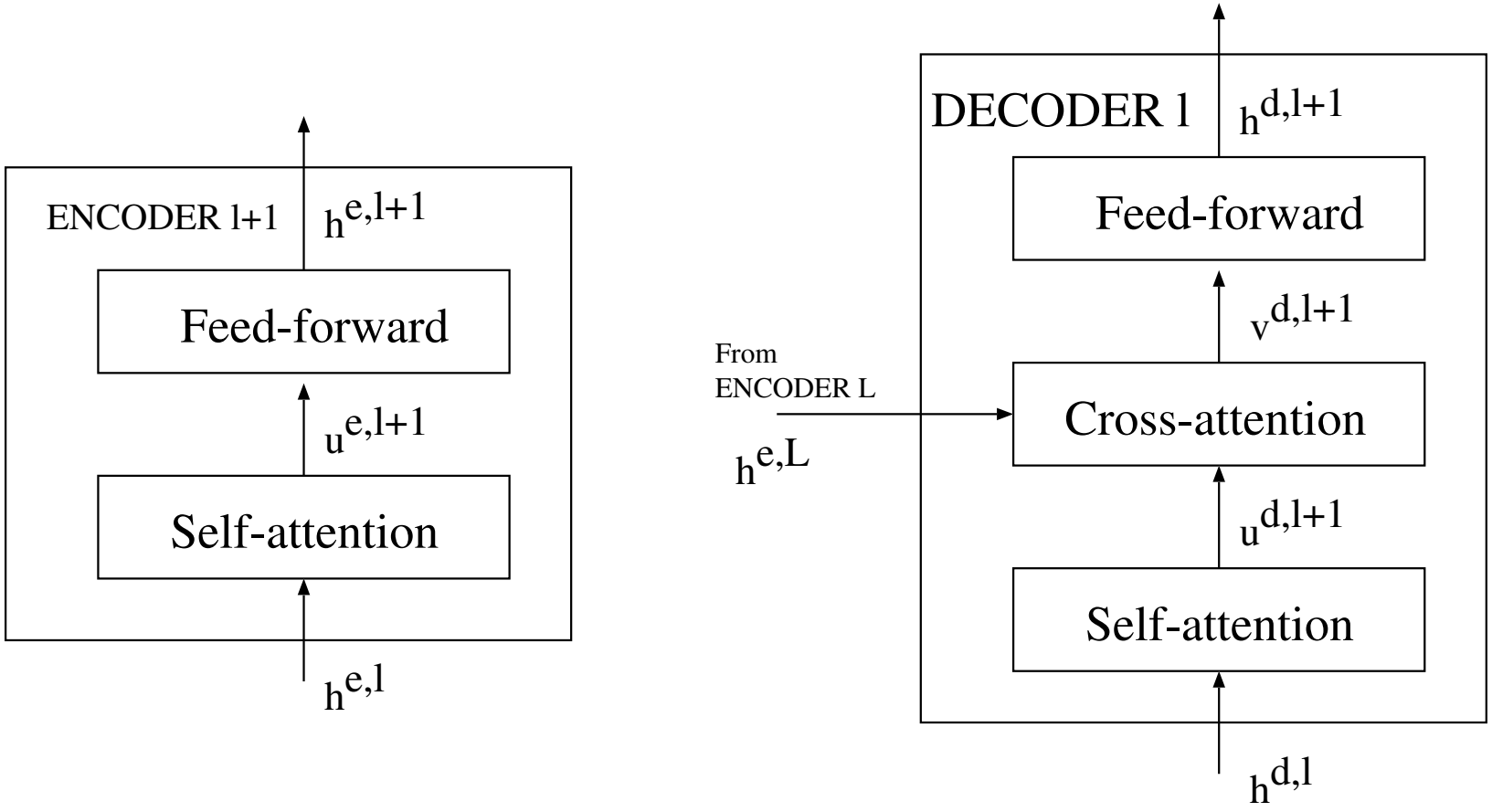
$$p(y_1^I \mid x_1^J) = \prod_{i=1}^{I} p(y_i \mid y_1^{i-1}, u(x_1^J))$$

- Feed-forward networks.
- Self-attention or intra-sentence attention: (j, j') & (i, i') in addition to the cross-attention (i, j).
- Position encoding.
- Multi-head attention.
- Faster than RNN.

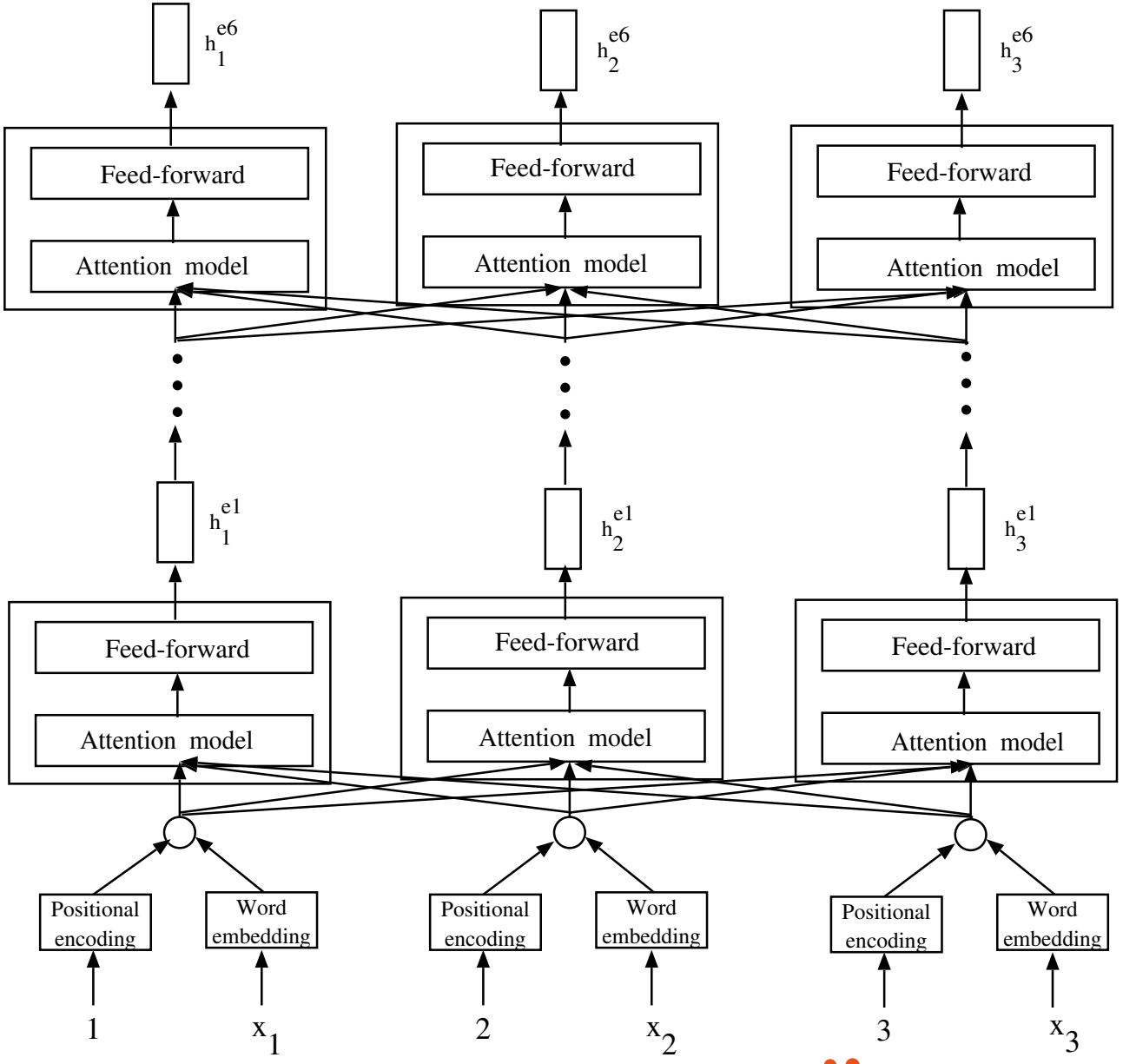
Transformer



Transformer



A simplified view of the encoder in the Transformer



Transformer

- A word x from a ordered vocabulary V_X with index $i(x)$
- Word embeddings: $\mathbf{W_E}(x) \equiv [\mathbf{W_E}]_{i(x)} = \mathbf{x} \in \mathbb{R}^{D_W}$: a row of $\mathbf{W_E}$ is the word embedding of x
- Positional embeddings: Given a sentence x_1^J the positional embedding of position j, $1 \leq j \leq J$ is $\mathbf{p}_j \in \mathbb{R}^{D_P}$:

$$\begin{aligned} p_{j,2k} &= \sin(j/10000^{2k/D_P}) \\ p_{j,2k+1} &= \cos(j/10000^{2k/D_P}) \end{aligned}$$

- Relative positional embeddings.
- Learned positional embeddings.

The encoder of Transformer

Given an discrete input sequence x_1^J,

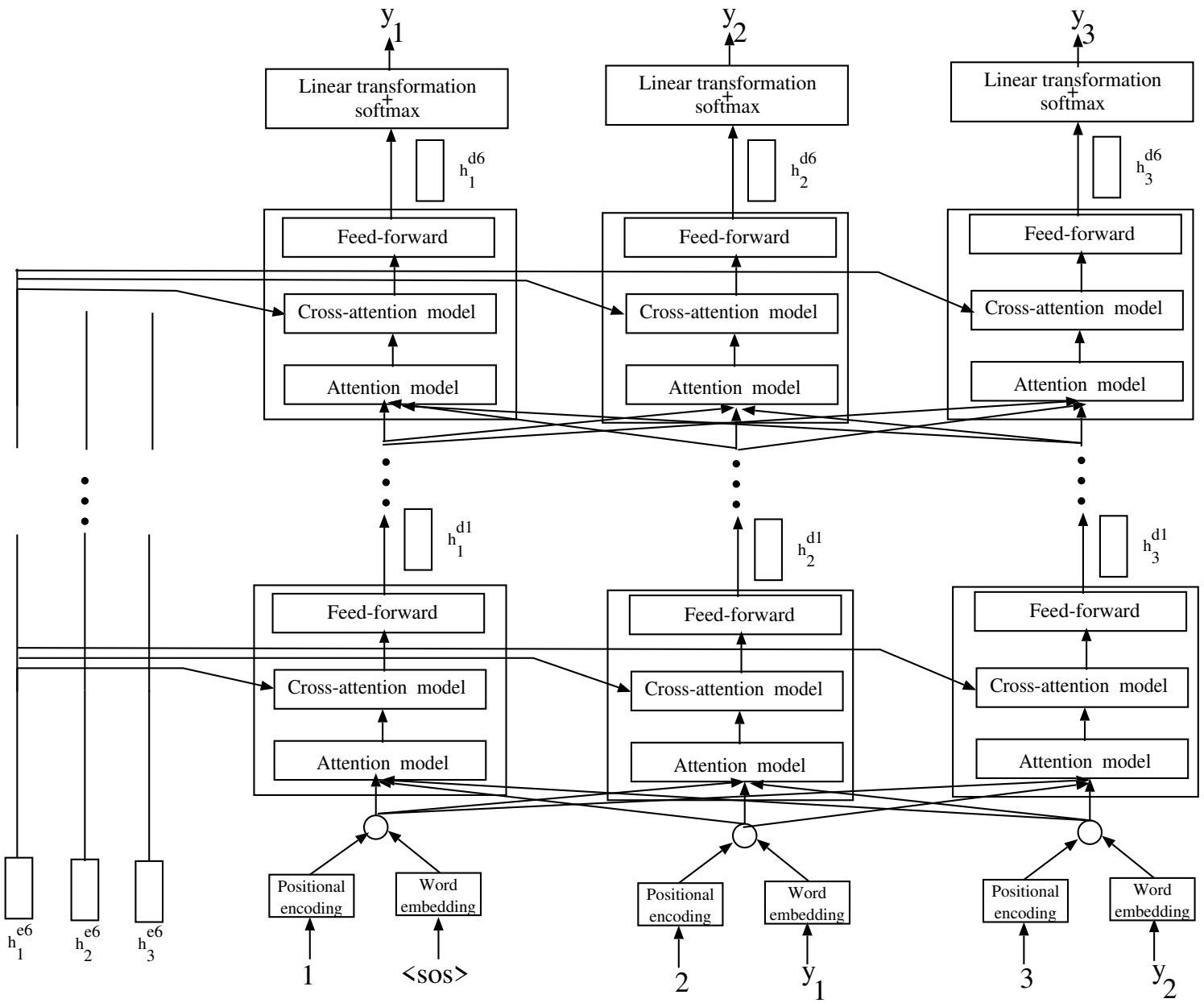
- Initialization: $\mathbf{h}_j^{e,0} = \mathbf{x}_j = \mathcal{E}(x_j) \quad 1 \leq j \leq J$
- In layer l of the encoder($1 \leq l < L$)
  - Self-attention model:

$$\mathbf{u}_j^{e,l+1} = \mathbf{a}(\mathbf{h}_1^{e,l}, \ldots, \mathbf{h}_J^{e,l}, \mathbf{h}_j^{e,l}) \quad 1 \leq i \leq J$$

  - Feed-forward network:

$$\mathbf{h}_j^{e,l+1} = \mathbf{F}_f(\mathbf{u}_j^{e,l+1}) \quad 1 \leq j \leq J$$

A simplified view of the decoder in the Transformer



The decoder of Transformer

- For $1 \leq i \leq I$
  - Initialization: $\mathbf{u}_{i-1}^{d,1} = \mathbf{y}_{i-1} = \mathcal{E}(y_{i-1})$, $(y_0 = " < sos > ")$
  - In layer l of the decoder $(1 \leq l < L)$:
    - Self-attention model:

$$\mathbf{u}_i^{d,l+1} = \mathbf{a}(\mathbf{h}_1^{d,l}, \ldots, \mathbf{h}_i^{d,l}, \mathbf{h}_i^{d,l})$$

    - Cross-attention model:

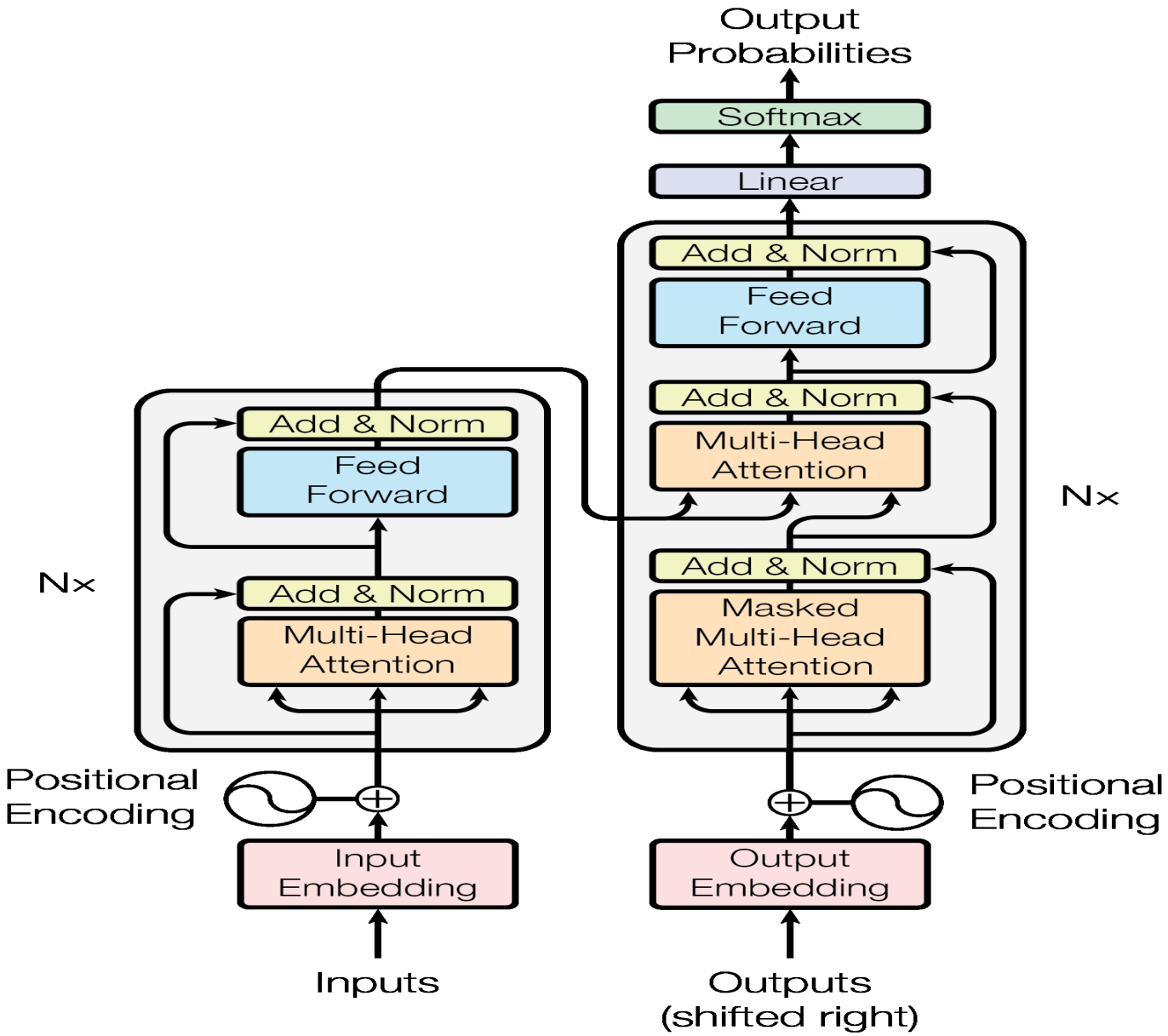
$$\mathbf{v}_i^{d,l+1} = \mathbf{a}(\mathbf{h}_1^{e,L}, \ldots, \mathbf{h}_J^{e,L}, \mathbf{u}_i^{d,l+1})$$

    - Feed-forward network:

$$\mathbf{h}_i^{d,l+1} = \mathbf{F}_f(\mathbf{v}_i^{d,l+1})$$

- Output generation: $p(y_1^I \mid x_1^J) = \prod_{i=1}^{I} p(y_i \mid y_1^{i-1}, u(x_1^J)) = \prod_{i=1}^{I} \mathbf{f}_{sm}(\mathbf{W}\ \mathbf{h}_i^L)_{i(y_i)}$

Transformer (Vaswani 2017)



Transformer

- Residual networks
- Layer normalization
- Feed-forward networks: 1 hidden layer + ReLU function
- Multi-head attention

Layer normalization (Xiong arXiv 2020)

- Given a sequence of vectors $\mathbf{z}_1, \ldots, \mathbf{z}_K$ from a layer with $\mathbf{z}_k \in \mathbb{R}^d$ for $1 \leq k \leq K$, a layer normalization $\mathbf{N}$ is $\mathbf{N}(\mathbf{z}_1, \ldots, \mathbf{z}_K) = (\bar{\mathbf{z}}_1, \ldots, \bar{\mathbf{z}}_K)$ such that:

$$\bar{z}_{k,i} = \gamma \, \frac{z_{k,i} - \mu_i}{\sigma_i} + \beta \qquad 1 \leq k \leq K, \; 1 \leq i \leq d$$

where γ and β are hyper-parameters, and

$$\mu_i = \frac{\sum_{k=1}^{K} z_{k,i}}{K} \qquad \sigma_i^2 = \frac{\sum_{k=1}^{K} (z_{k,i} - \mu_i)^2}{K} \qquad 1 \leq i \leq d$$

Residual networks and feed-forward networks (Xiong arXiv 2020)

- Given $\mathbf{z} \in \mathbb{R}^d$ and a function $\mathbf{f} : \mathbb{R}^d \rightarrow \mathbb{R}^d$, a residual function $\mathbf{R}$ is defined as:

$$\mathbf{R}(\mathbf{z}, \mathbf{f}(\mathbf{z})) = \mathbf{f}(\mathbf{z}) + \mathbf{z}$$

- Feed-forward networks, $\mathbf{F}_f$: Given a $\mathbf{z} \in \mathbb{R}^d$.

$$\mathbf{F}_f(\mathbf{z}) \;=\; \mathbf{W}_2 \; \mathbf{f}_{ReLU}(\mathbf{W}_1 \; \mathbf{z} + \mathbf{b}_1) + \mathbf{b}_2$$

where $\mathbf{W}_2$, $\mathbf{W}_1$ are the weights of the second and first layer of the feed-forward networks, and $\mathbf{b}_2$ and $\mathbf{b}_1$ the corresponding bias.

Multi-head attention in Transformer

For N heads and $1 \leq l \leq L$, $1 \leq j \leq J$ and $1 \leq i \leq I$:

- Encoder self-attention:

$$\mathbf{u}_{j,n}^{e,l+1} = \mathbf{a}_n(\mathbf{h}_1^{e,l}, \ldots, \mathbf{h}_J^{e,l}, \mathbf{h}_j^{e,l}) \quad 1 \leq j \leq J \text{ and } 1 \leq n \leq N$$

$$= \sum_{j'=1}^{J} \frac{\exp(\mathbf{W}_Q^{e,n}\mathbf{h}_j^{e,l} \cdot \mathbf{W}_K^{e,n}\mathbf{h}_{j'}^{e,l})}{\sum_{j''=1}^{J} \exp(\mathbf{W}_Q^{e,n}\mathbf{h}_j^{e,l} \cdot \mathbf{W}_K^{e,n}\mathbf{h}_{j''}^{e,l})} \mathbf{W}_V^{e,n}\mathbf{h}_{j'}^{e,l} \qquad 1 \leq j, j' \leq J$$

$\mathbf{W}_Q^{e,n}, \mathbf{W}_K^{e,n}, \mathbf{W}_V^{e,n}$ are matrix of $D_W/N \times D_W$

$$\mathbf{u}_j^{e,l+1} = \left[\mathbf{u}_{j,1}^{e,l+1}, \ldots, \mathbf{u}_{j,N}^{e,l+1}\right]; \quad \mathbf{h}_j^{e,l+1} = \mathbf{F}_f(\mathbf{u}_j^{e,l+1}) \quad 1 \leq j \leq J$$

Multi-head attention in Transformer

For N heads and $1 \leq l \leq L$, $1 \leq j \leq J$ and $1 \leq i \leq I$:

- Decoder self-attention:

$$\mathbf{u}_{i,n}^{d,l+1} = \mathbf{a}_n(\mathbf{h}_1^{d,l}, \ldots, \mathbf{h}_J^{d,l}, \mathbf{h}_i^{d,l}) \quad 1 \leq i \leq I \text{ and } 1 \leq n \leq N$$

$$= \sum_{i'=1}^{I} \frac{\exp(\mathbf{W}_Q^{d,n}\mathbf{h}_i^{d,l} \cdot \mathbf{W}_K^{d,n}\mathbf{h}_{i'}^{d,l})}{\sum_{i''=1}^{I} \exp(\mathbf{W}_Q^{d,n}\mathbf{h}_i^{d,l} \cdot \mathbf{W}_K^{d,n}\mathbf{h}_{i''}^{d,l})} \mathbf{W}_V^{d,n}\mathbf{h}_{i'}^{d,l} \qquad 1 \leq i, i' \leq I$$

$\mathbf{W}_Q^{d,n}, \mathbf{W}_K^{d,n}, \mathbf{W}_V^{d,n}$ are matrix of $D_W/N \times D_W$

$$\mathbf{u}_i^{d,l+1} = \left[\mathbf{u}_{i,1}^{d,l+1}, \ldots, \mathbf{u}_{i,N}^{d,l+1}\right]; \quad \mathbf{h}_i^{d,l+1} = \mathbf{F}_f(\mathbf{u}_i^{d,l+1}) \quad 1 \leq i \leq I$$

Multi-head attention in Transformer

For N heads and $1 \leq l \leq L$, $1 \leq j \leq J$ and $1 \leq i \leq I$:

- Decoder cross-attention:

$$\mathbf{v}_{i,n}^{d,l+1} = \mathbf{a}_n(\mathbf{h}_1^{e,L}, \ldots, \mathbf{h}_J^{e,L}, \mathbf{u}_i^{d,l+1}) \quad 1 \leq i \leq I \text{ and } 1 \leq n \leq N$$

$$= \sum_{j=1}^{J} \frac{\exp(\mathbf{W}_Q^{c,n}\mathbf{u}_i^{d,l+1} \cdot \mathbf{W}_K^{c,n}\mathbf{h}_j^{e,L})}{\sum_{j=1}^{J}\exp(\mathbf{W}_Q^{c,n}\mathbf{u}_i^{d,l+1} \cdot \mathbf{W}_K^{c,n}\mathbf{h}_j^{e,L})} \mathbf{W}_V^{c,n}\mathbf{h}_j^{e,L} \qquad 1 \leq i \leq I$$

$\mathbf{W}_Q^{c,n}, \mathbf{W}_K^{c,n}, \mathbf{W}_V^{c,n}$ are matrix of $D_W/N \times D_W$

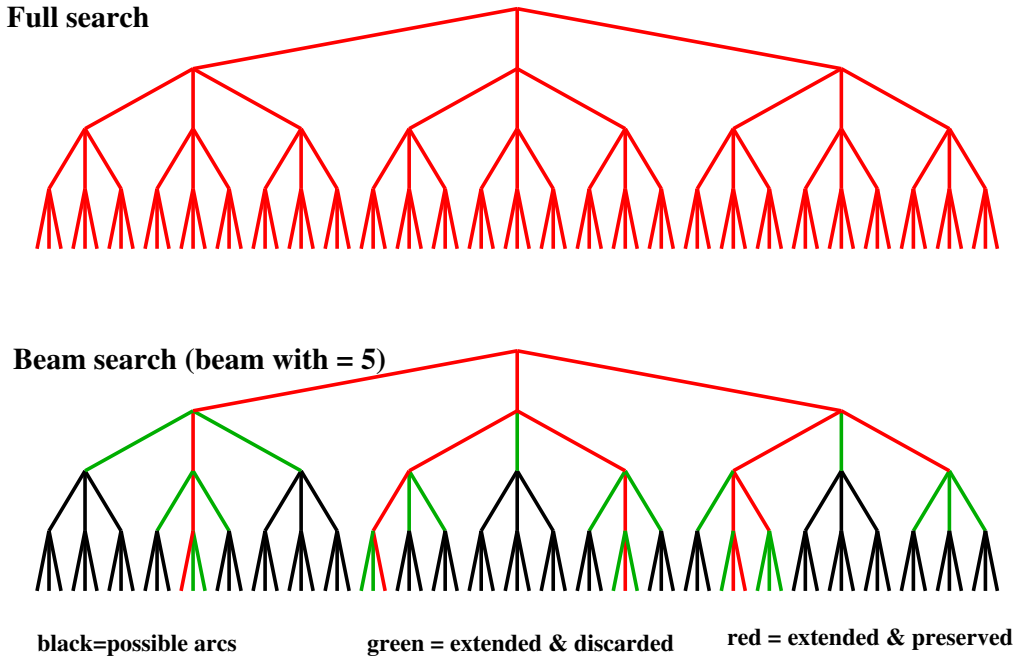
$$\mathbf{v}_i^{d,l+1} = \left[\mathbf{v}_{i,1}^{d,l+1}, \ldots, \mathbf{v}_{i,N}^{d,l+1}\right]; \quad \mathbf{h}_i^{d,l+1} = \mathbf{F}_f(\mathbf{v}_i^{d,l+1}) \quad 1 \leq i \leq I$$

Training (Koehn 2020)

- Given a training set of bilingual pairs T, maximize an objective function as the cross-entropy of the training set.
- Use of the computational graphs.
- Optimization algorithms: Based on backpropagation through time using stochastic gradient descent, Adagrad, Adadelta and Adam.
- Training issues
  - Early stopping.
  - Batch and layer normalization.
  - Data augmentation.
  - Hyperparameters
    - Dimension of source/target word embeddings.
    - Number of LSTM/GRU/Transformer in encoder/decoder.
    - Initial learning rate.
    - Patience.
    - Dropout parameter.
    - Weight decay.
    - Noisy injection (to inputs, outputs, ...)
    - ...

Inference/decoding for discrete representations (Koehn 2020)

- Smart tree search by using beam search (suboptimal search)
  - At the begining of the process there is a empty list of target hypotheses.
  - At each target position there is a bounded prefix tree of K target prefixes. For each prefix, a bounded set of extended prefixes is generated by appending the most promising words.
  - The process ends when a "eof" is generated.



Index

1 Sequence-to-sequence architectures ▷ 3

○ 2 *Pre-trained models* ▷ 35

3 Applications ▷ 54

4 Bibliography ▷ 69

Why pre-trained models?

- Training neural machine translation models from scratch is expensive.
- Large corpora are necessary.
- For many task-specific, models there are trained with low resources.

- Pre-trained models have proved to be useful in many scenarios:
  - Text classification.
  - Text generation.
  - Image generation.
  - Sentiment analysis.
  - Image description/caption.
  - Video description.
  - ...

Prompts[1]

- Definition: "A prompt is a piece of text inserted in the input examples, so that the original task can be formulated as a (masked) language modeling problem." [1]
- Discrete vs soft prompts.
  - Discrete prompts: manually-designed or automatically generated prompt.
- Frozen pre-trained models or fine-tunning models: Some models are so bigs that fine-tunning is not possible with regular resources.
- In-context learning: In addition of a piece of text, the models can see some few exaples of the task (Few-shot). This is adequate when frozen models are used,

[1] https://thegradient.pub/prompting/

Pre-trained models for text

- Bidirectional Encoder Representations from Transformers (BERT). Based on the Transformer encoder. [Devlin 2019] -Google AI Language-
- Generative Pre-trained Transformer (GPT, GPT-2, GPT-3, GPT-3.5 and GPT-4, GPT-J). Based on the Transformer decoder. [Radford 2018] -OpenAI-
- BART and mBART. Full encoder-decoder (monolingual and multilingual) Transformer [Liu 2020] -Facebook AI-
- Text-to-Text Transfer Transformer (T5. Trained with Colossal Clean Crawled Corpus (C4)). Complete Transformer. [Raffel 2020] -Google-
- PaLM (Pathways Language Model). Based on the Transformer decoder and uses complex distributed computation for accelerators. [Chowdhery 2022] -Google-
- LLaMA (Meta)
- Alpaca (Standford Univ), Falcon-40B, LaMBDA (Google)

Pre-trained models for text and images

- Contrastive Language-Image Pre-training (CLIP). Multimodal encoder text+imatges [Radford 2021] -openAI-
- DALL-E 2. CLIP based image generation from text [Ramesh 2021] -openAI-
  https://openai.com/dall-e-2/
- DALL-E Mini.
  https://github.com/borisdayma/dalle-mini
- Midjourney: text-to-image like DALL-E focused to pretty images.
  https://dallery.gallery/midjourney-guide-ai-art-explained/
  https://aicomicbooks.com/shop/zarya-of-the-dawn-download-now/
- Stable Diffusion 2; based on openCLIP. -Stability.ai-
  https://github.com/Stability-AI/stablediffusion

Pre-trained models

- Perceiver model [Hawthorne ICML 2022] -Deepmind-
- GATO model [Reed arXiv 2022] -Deepmind-
- Chinchilla. Based on the Transformer decoder and uses less compute for fine-tunning and inference. [Hoffman 2022] -DeepMind-
- Flamingo: Visual Language Model [Alaycrac 2022]. Input: Text and visual data interleaved as a prompt, and the output is a text. -DeepMind-
- Vision Transformer (ViT): BERT-like model for images. [Dosovitskiy 2020] -Google Research, Brain Team-

Pre-trained models

- ChatGPT based on GPT3.5 (OpenAI)
- Bard based on PaLM 2 (Google)
- Bing Chat based on GPT4 (Microsoft)
- LIMA based on LLaMA (Meta)
- Vicuna based on LLaMA (open source)
- GPT4ALL (Nomic AI)
- Bing Image Creator based on DALL-E (Microsoft)
- Leonardo AI.
- ImageBind (META) Multimodal.
- KosMOS-1 (Microsoft) Multimodal.

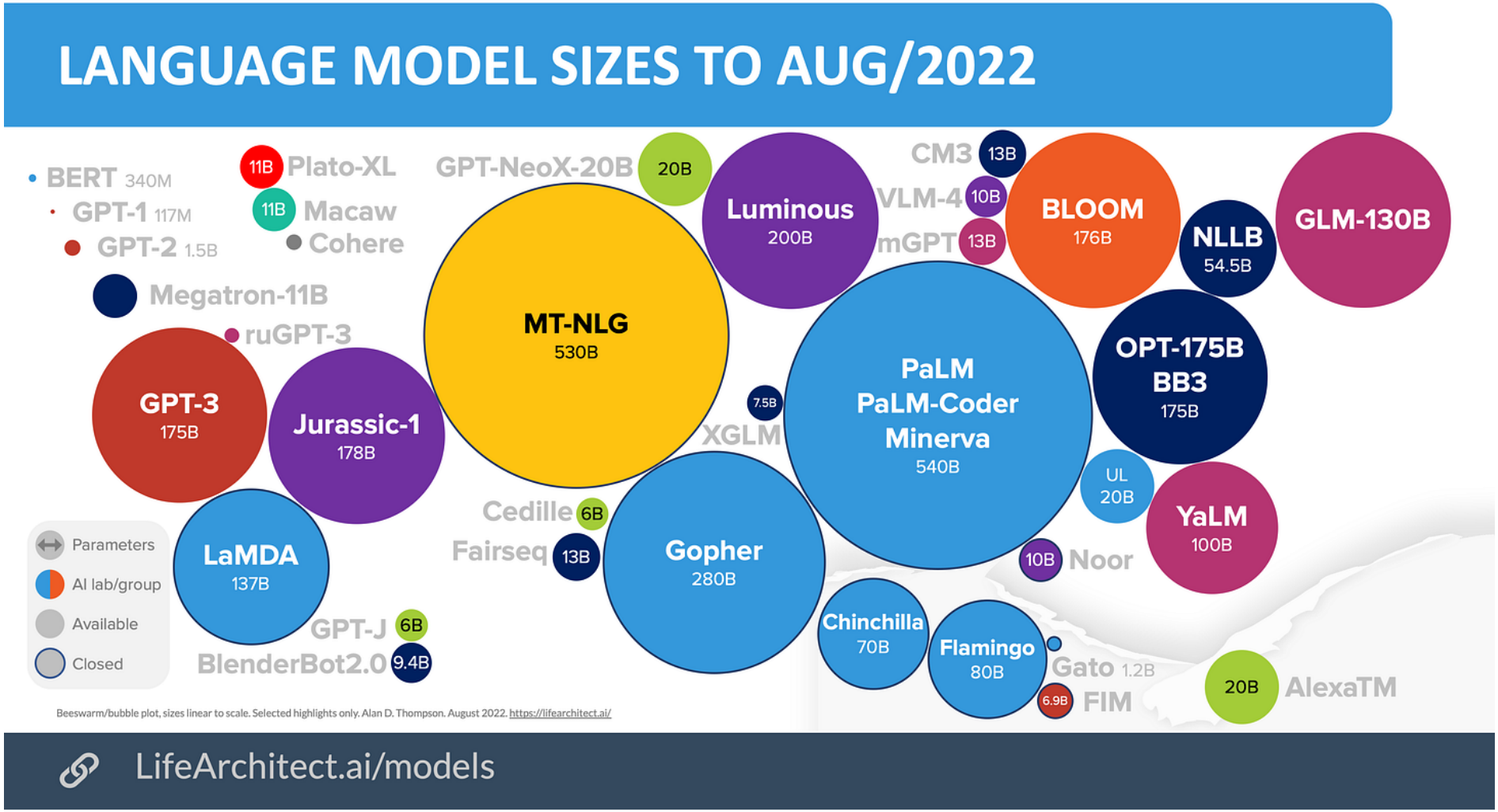
Pre-trained models

[Romero. Ultra-Large AI Models Are Over. Medium Daily Digest 2022]

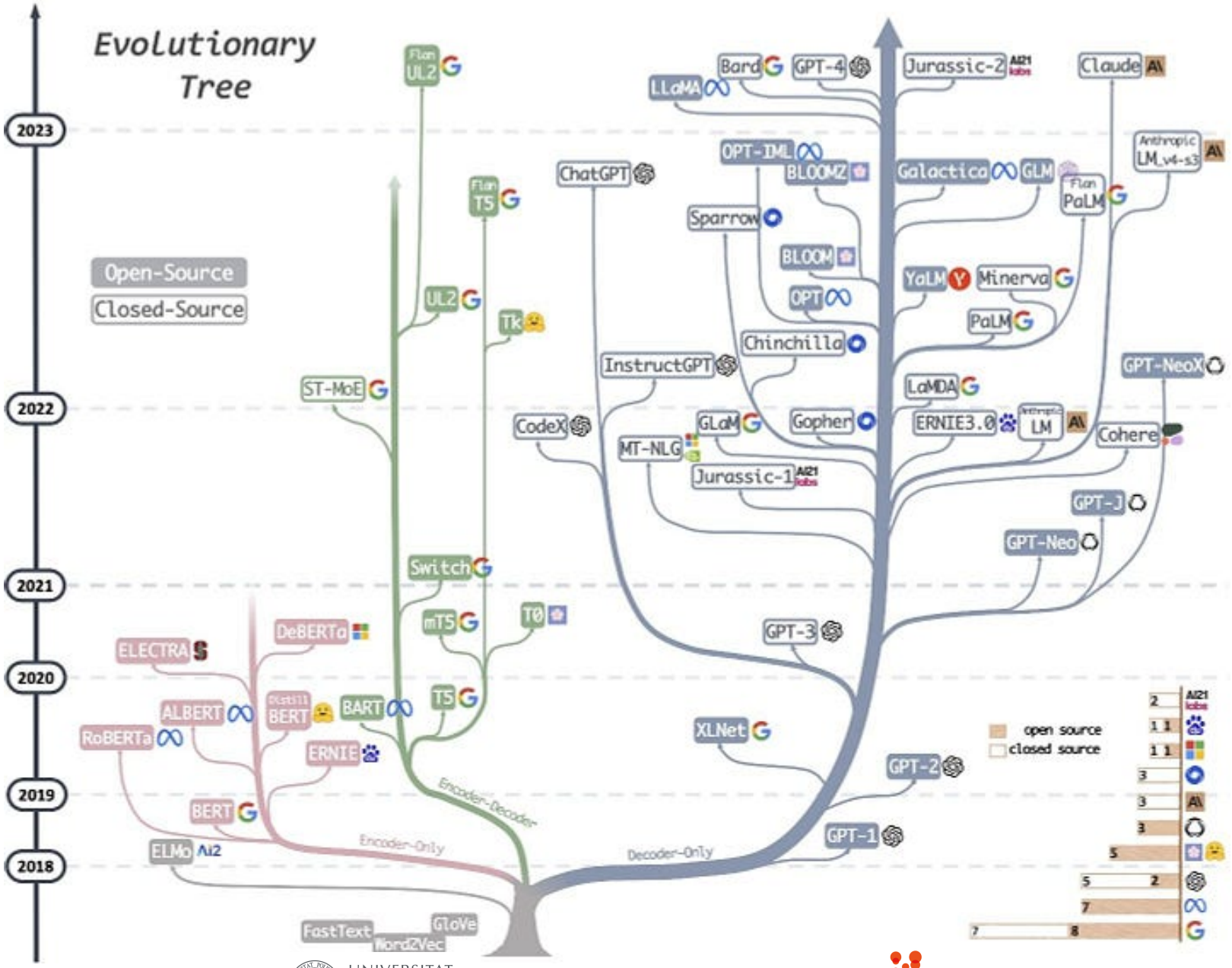
- Google: LAMBDA (137B, May 2021), PaLM (540B, Apr 2022)
- Meta: OPT (175B, May 2022), BlenderBot3 (175B,Aug 2022)
- DeepMind: Gopher (280B, Dec 2021), Chinchilla (70B, Apr 2022)
- Microsoft-Nvidia: MT-NLG (530B, Oct 2021)
- BigScience: BLOOM (176B, June 2022)
- Baidu: PCL-BAIDU Wenxin (260B, Dec 2021), ERNIE-ViLG (24B, 2022)
- Yandex: YaLM (100B, June 2022)
- Tsinghua: GLM (130B, July 2022)
- AI21 labs: Jurassic-1 (178B, Aug 2021)
- Aleph Alpha: Luminous (200B, Nov 2021)

Pre-trained models

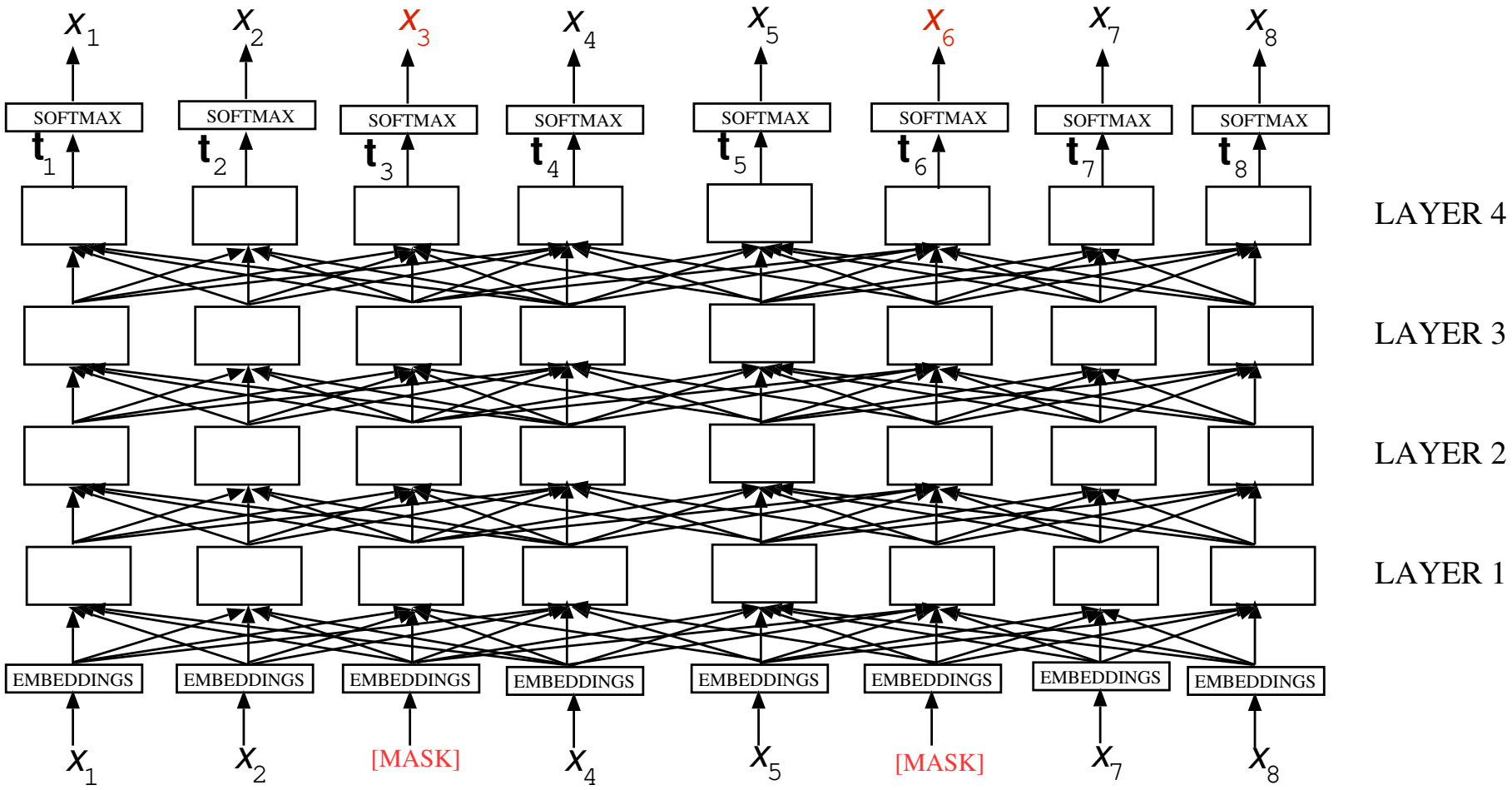
[Romero. Ultra-Large AI Models Are Over. Medium Daily Digest. 2022]



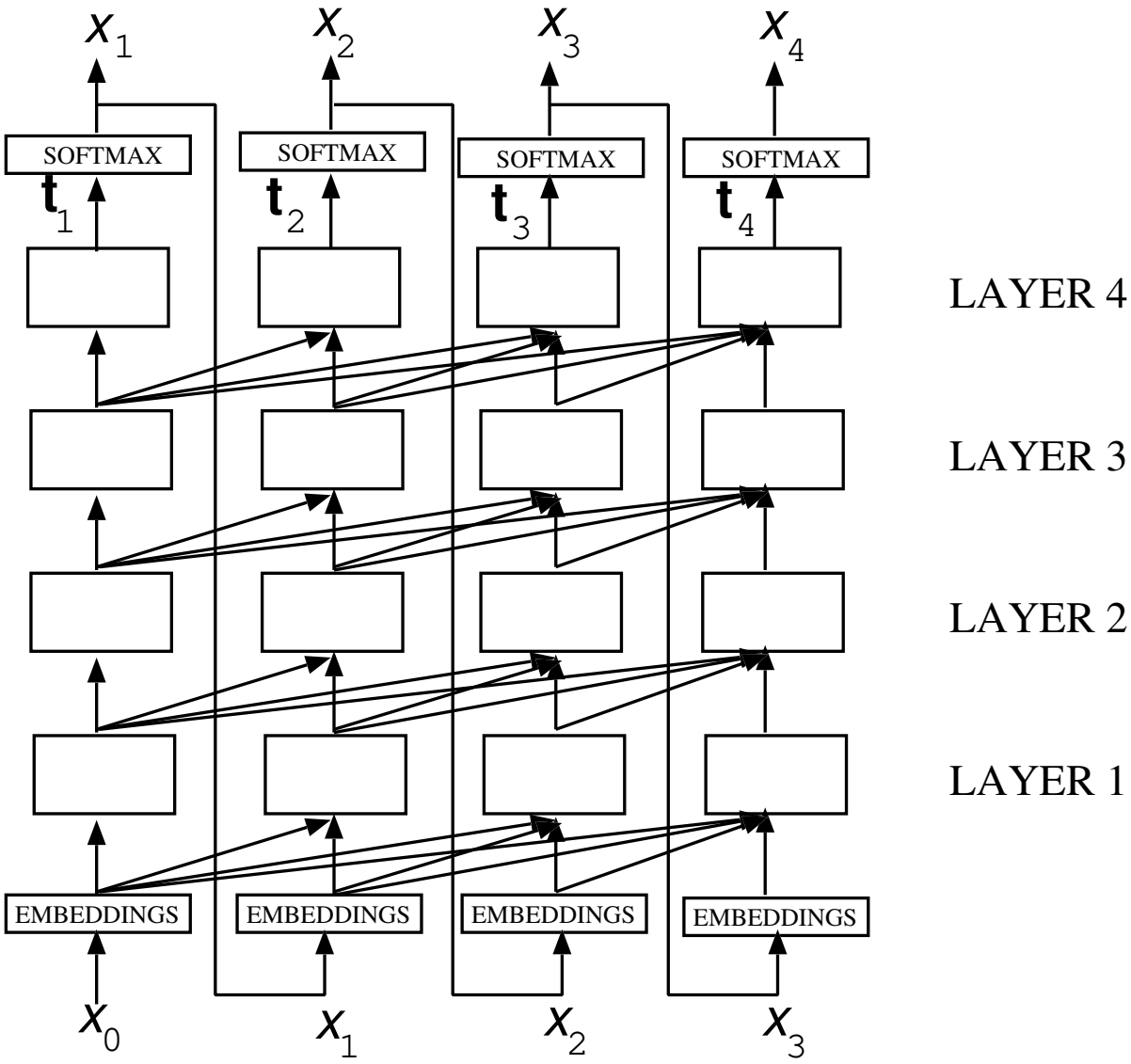
Pre-trained models [Yann LeCun 2023]



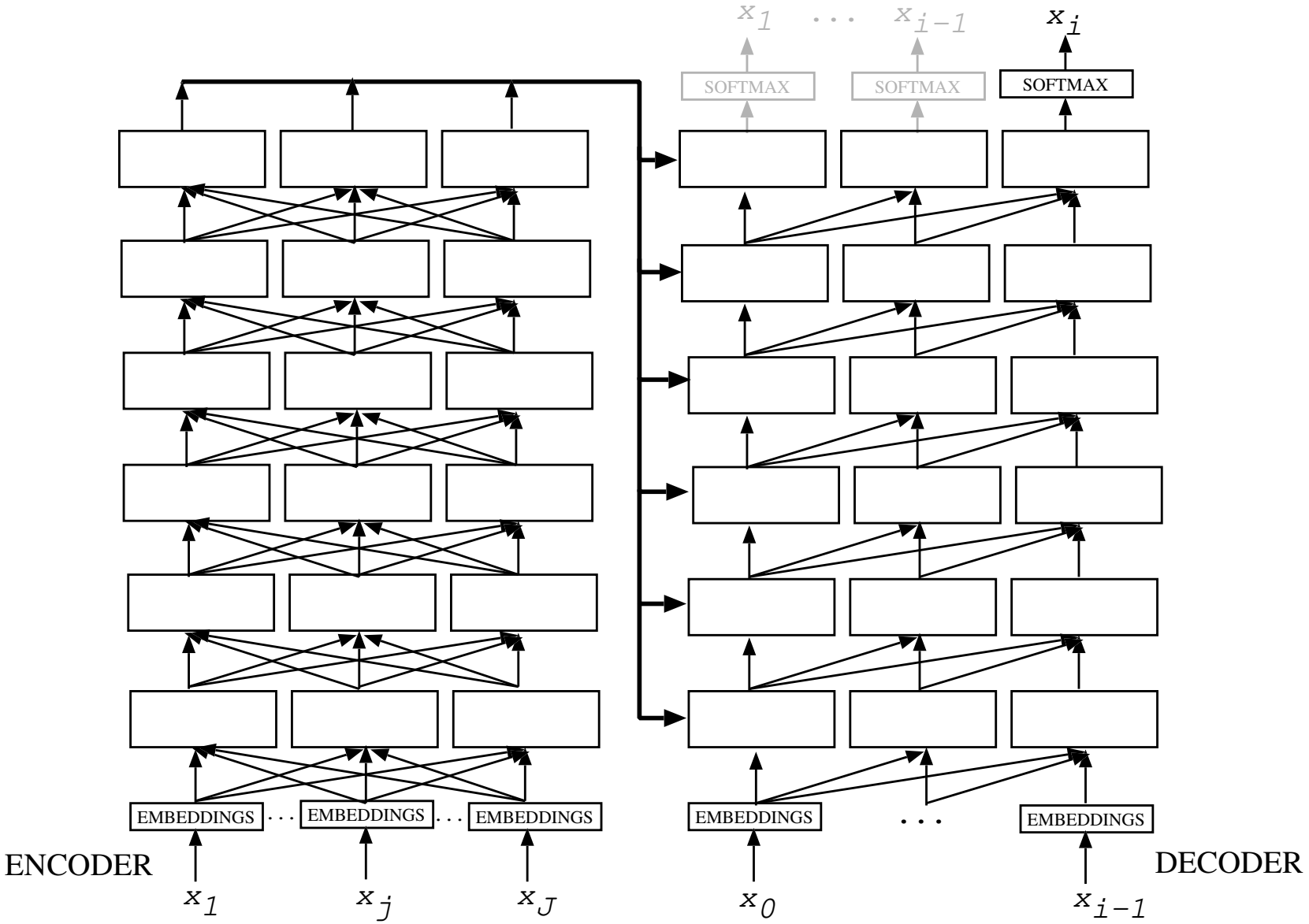
BERT model [Devlin Google 2019]



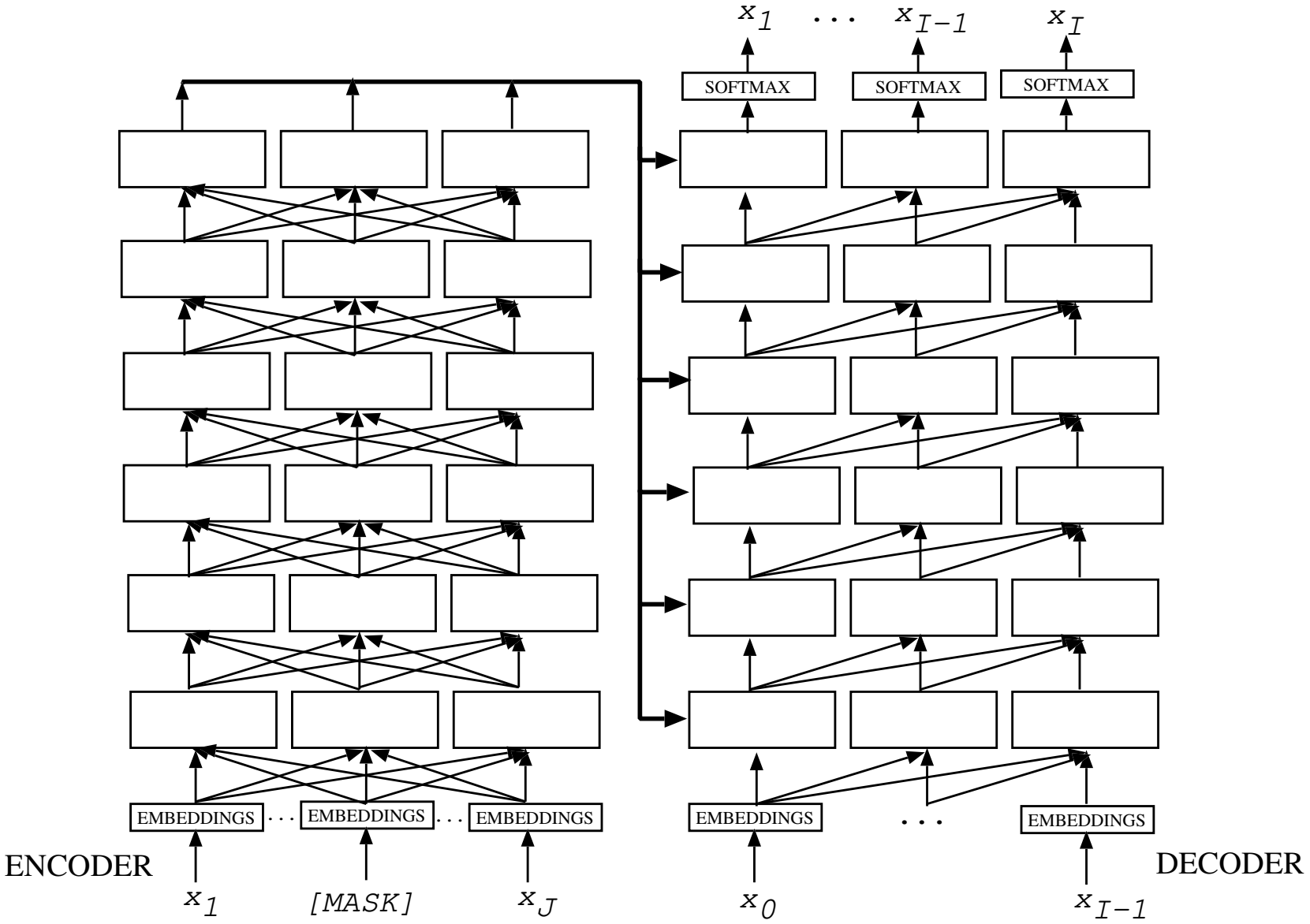
GPT model [Radford OpenAI 2018]



BART/T5 model [Lewis ACL 2020] [Raffel JMLR 2020]



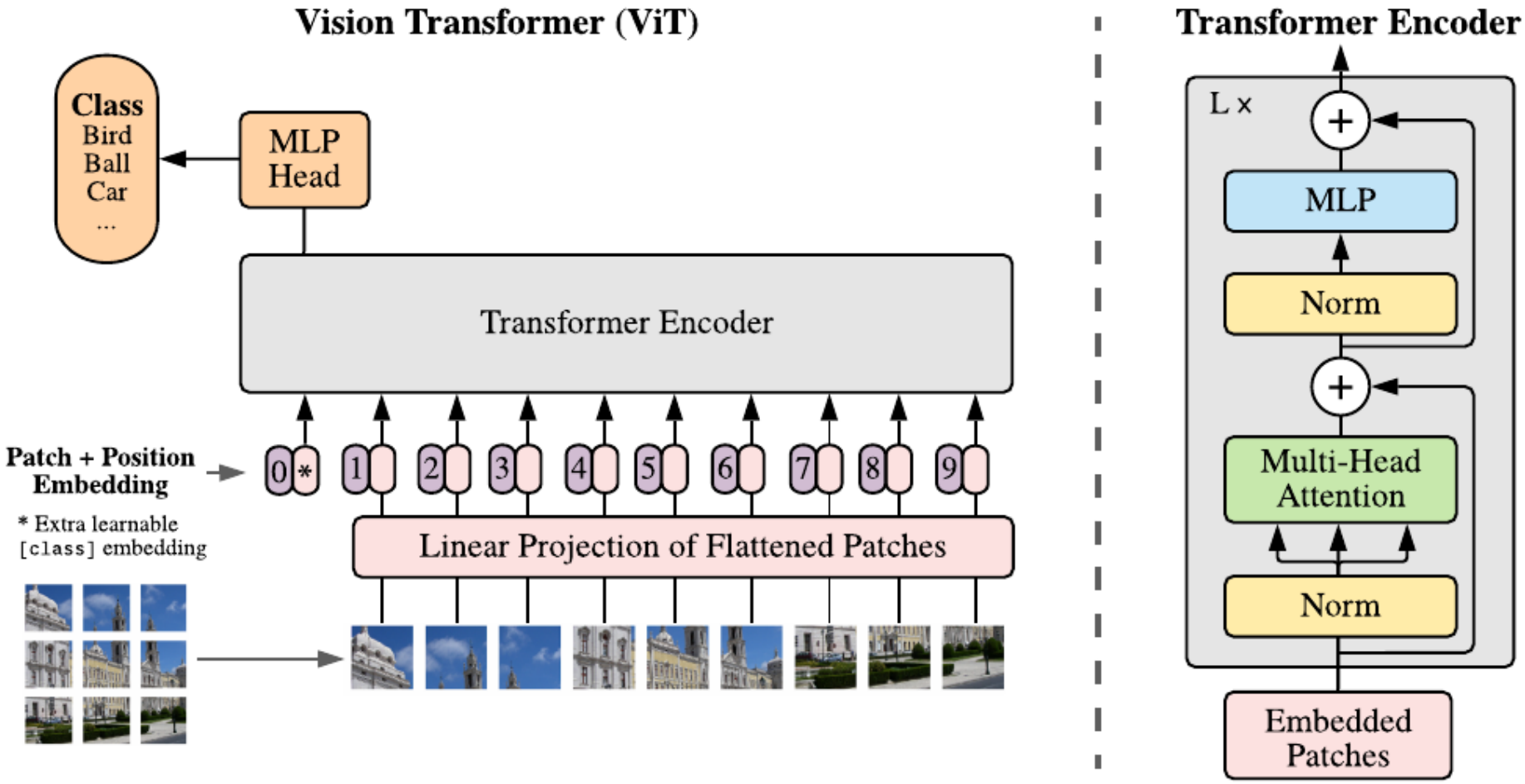
Training BART/T5 model



Training BART/T5 model

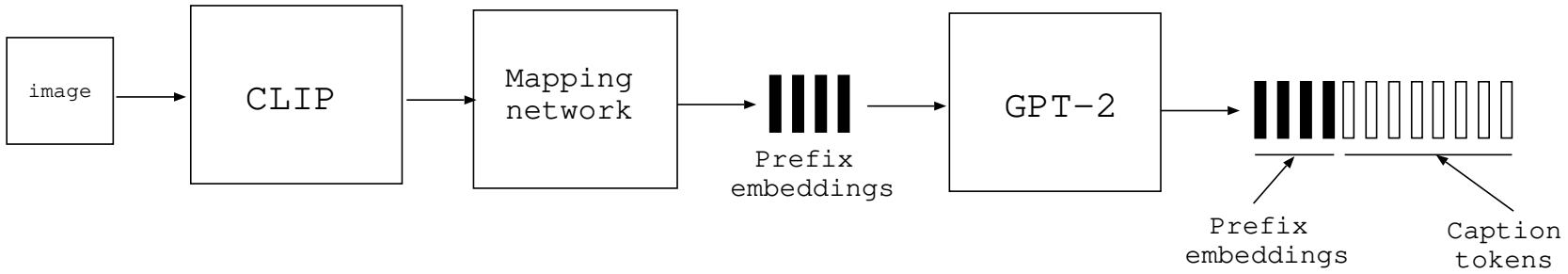
- BART (Amazon): masking input words & full input sentence.
- T5 (Google): masking input sub-sequences & input masked sub-squences only.

Vision Tranformer [Dososvitskiy 2020]



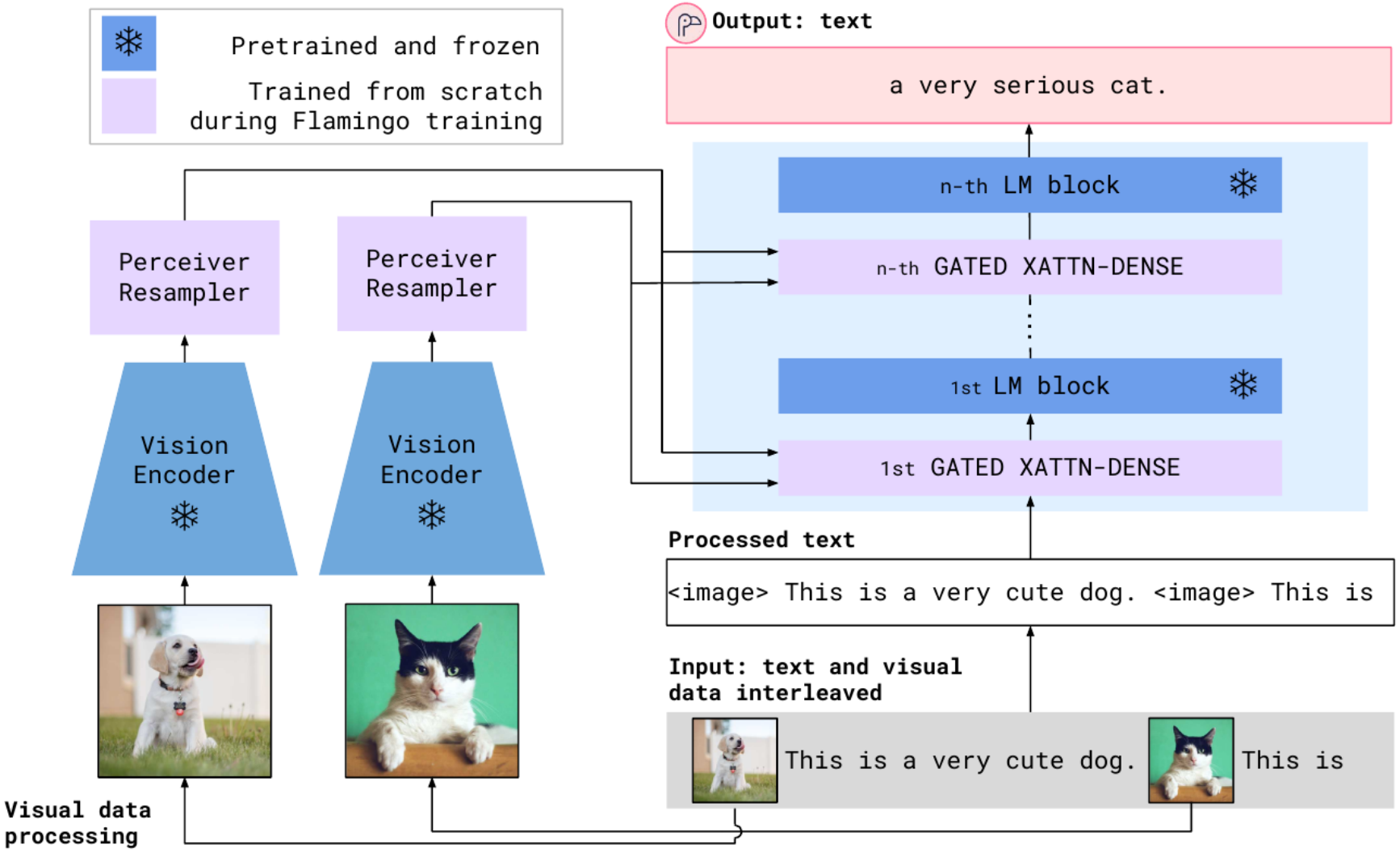
CLIP model [Radford ICML 2021]

- Image encoder: ResNet-50 (CNN) or Vision Transformer (ViT).
- Text encoder: GPT architecture.
- Training: Optimize a symmetric cross entropy loss over the similarity scores between image and test embeddings.
- Application: ClipCap for image captioning [Mokady arXiv 2021]



- Another application: DALL-2 for image generation from text. [Ramesh OpenAI 2021]

Flamingo model [Alyarac arXiv 2022]



Important issues in pre-trained models

- Fine-tunning of pre-trained models
  - Direct fine-tunning: Impossible in regular instalations in the case of very large models.
  - Indirect solutions: LoRa (reduces the number of trainable parameters by using Low-rank matrix) (Microsoft) [Hu arXiv 2021]
  - Other solutions: Adapters, Lamini, ...
- Inference with pre-trained models
  - Direct inference: a number of very big GPUs.
  - A "simple" solutions: Weight quantization.
- Fine-tunning and quantization:
  - QLoRa (Univ. Whashington)
- Prompt tuning [Liu ACL 2022]

Index

1 Sequence-to-sequence architectures ▷ 3

2 Pre-trained models ▷ 35

○ 3 *Applications* ▷ 54

4 Bibliography ▷ 69

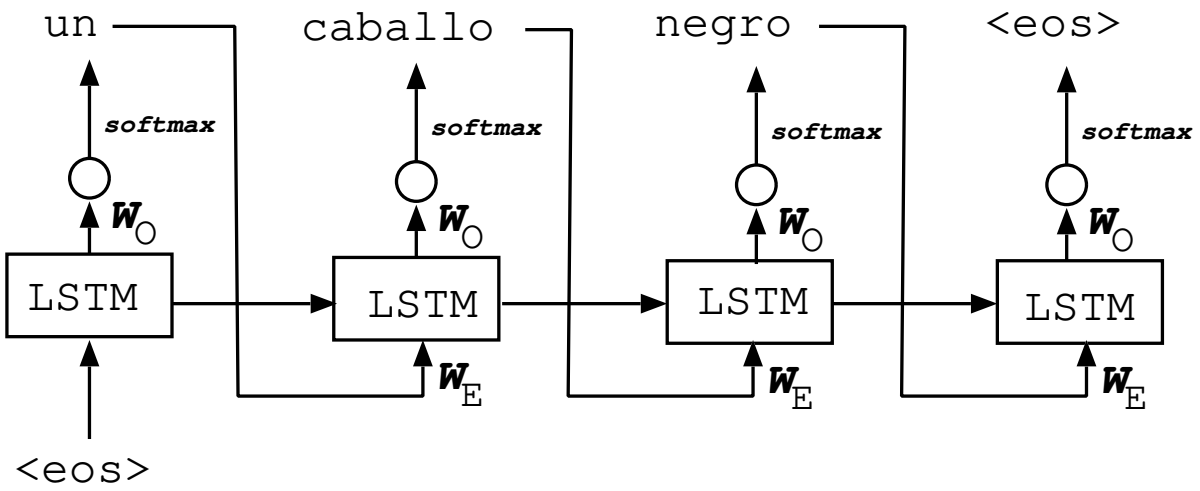
Applications of LSTM o Transformer-based models

- Language modelling (Sundermeyer, 2015).
- Image description using LSTM and CNN (Vinyals 2015)
- Multilingual multimodal image description (Elliott 2016)
- Video description using LSTM and CNN (Venugopalan 2015)
- Visual question answering (Li 2016)
- Text summarization (Sinha 2018)
- Modeling human behaviour (Katzer 2020).
- Image Generation.
- ...

A LSTM application: language modeling (Sundermeyer, 2015)

- For a word sequence $x_1, \ldots, x_N \equiv x_1^N \in \Sigma^N$

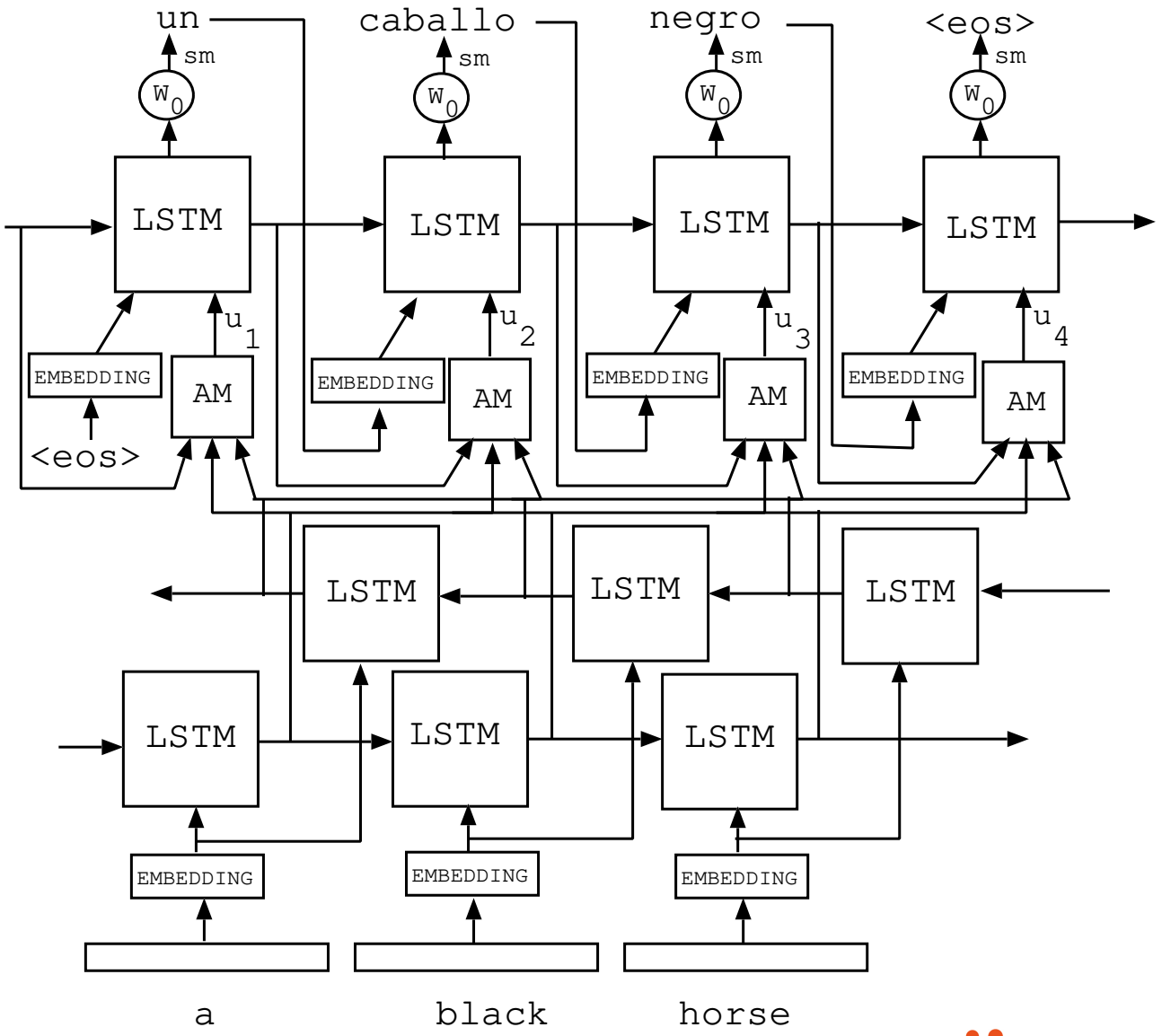
$$p(x_1^N) = \prod_{n=1}^{N} p(x_n \mid x_1^{n-1}) = \prod_{n=1}^{N} \mathbf{f}_{sm}(\mathbf{W}_O \ \mathbf{h}_n)_{i(x_n)}$$



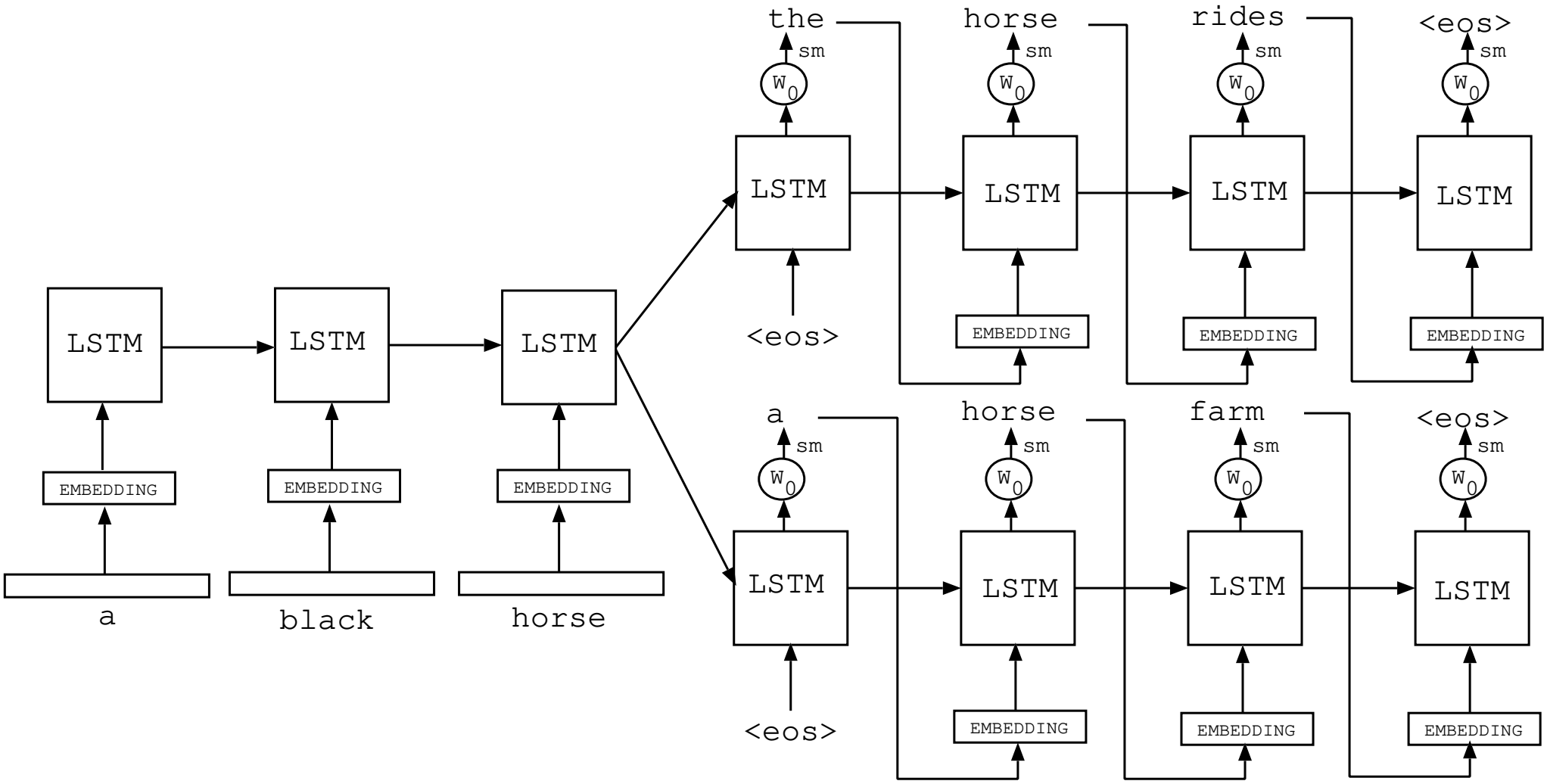
- The training is based on the optimization of the objetive function can be the cross-entropy (equivalent to maximize the likelihood or minimizing the perplexity):

$$L(\mathbf{\Theta}) = -\log \prod_{n=1}^{N} p(x_n \mid x_1^{n-1}; \mathbf{\Theta}) = -\sum_{n=1}^{N} \log \mathbf{f}_{sm}(\mathbf{W}_O \ \mathbf{h}_n)_{i(x_n)}$$

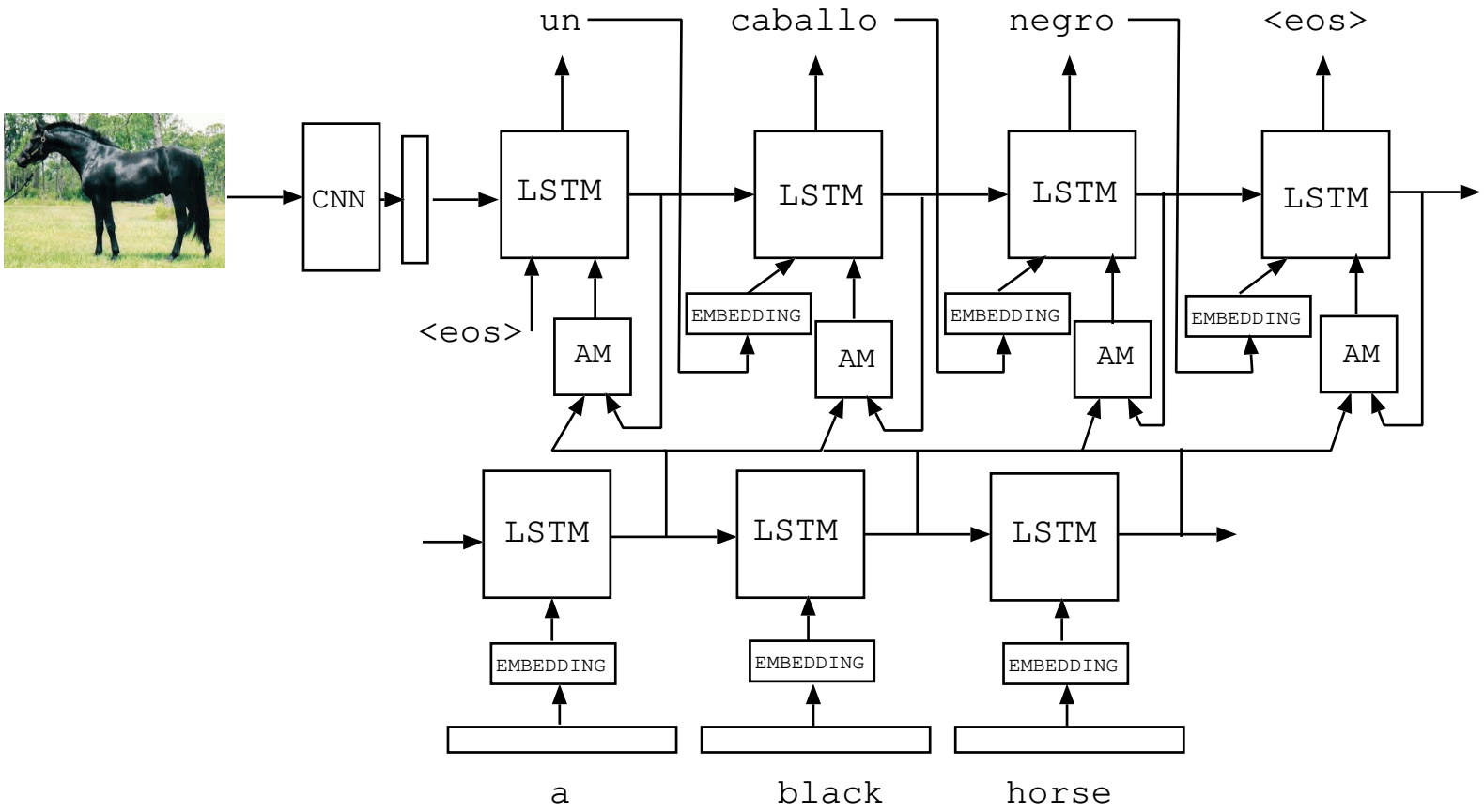
Machine Translation with bidirectional encoder-decoder with attention model (Luong 2015)



Sentence embedding with encoder-decoder (Mishra, 2019)



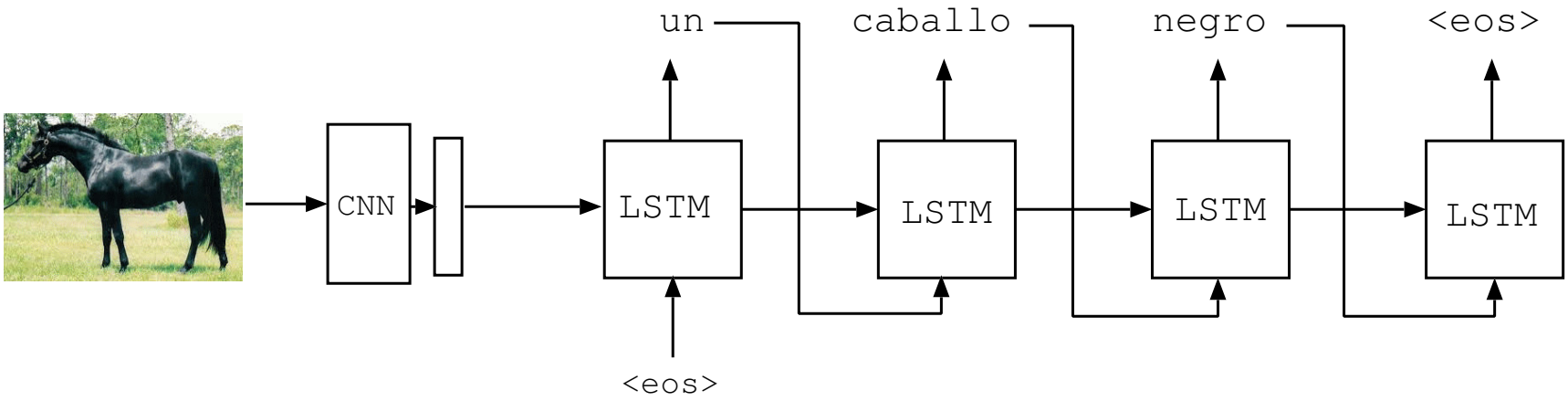
Multimodal machine translation



Given an image x and a description of the image in a source language s, search for a translation of s: $\hat{t} = \underset{t}{\text{argmax}}\, p(t \mid x, s)$

Multimodal transformers. [Yao& Wan 2020]

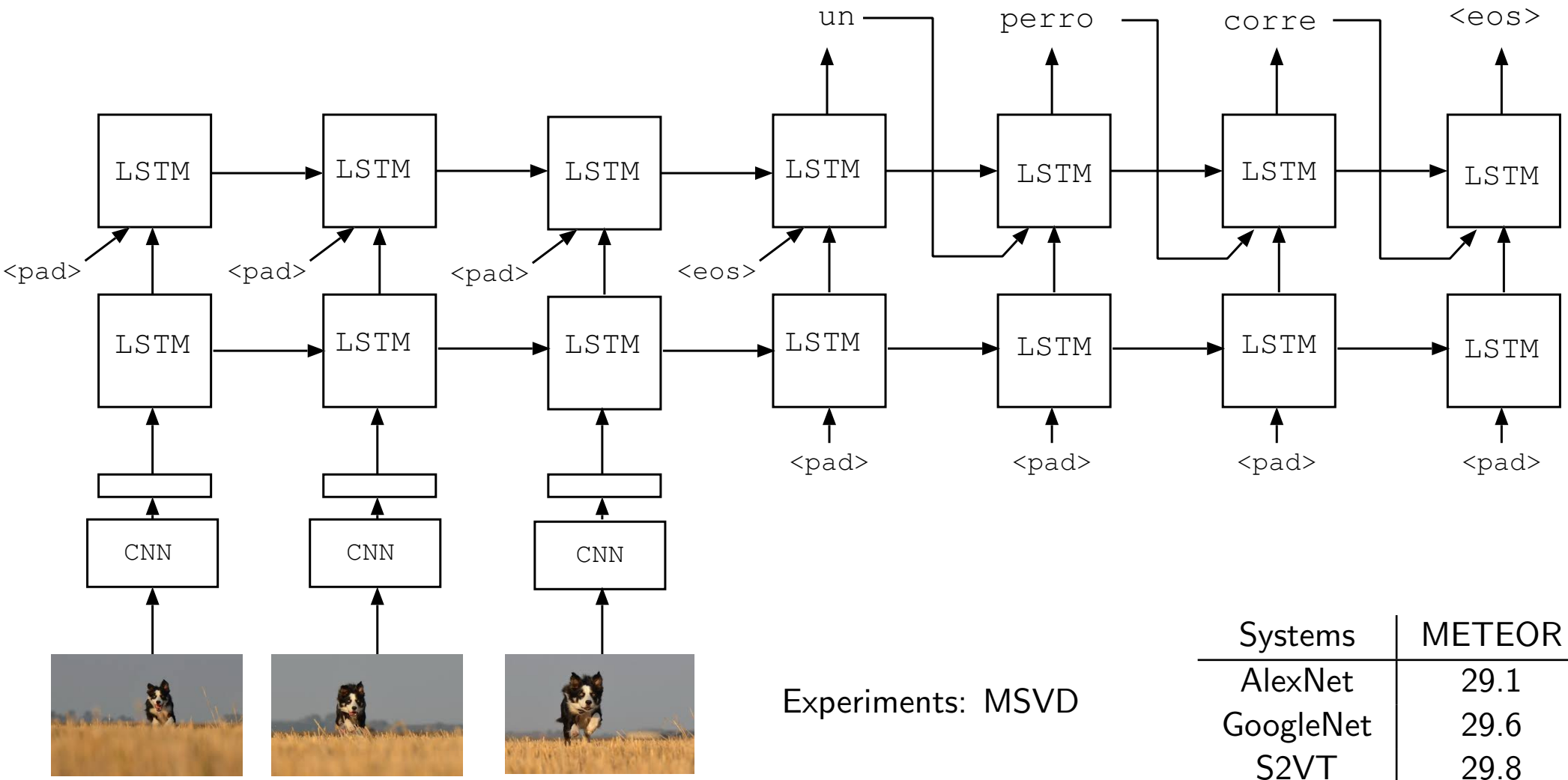
Image description using LSTM and CNN (Vinyals 2015)



Experiments (BLEU-1):

Systems	Corpus		
	PASCAL	Flickr30	Flickr5
State-of-the-art	25	56	58
NIC	59	66	63
Human	69	68	70

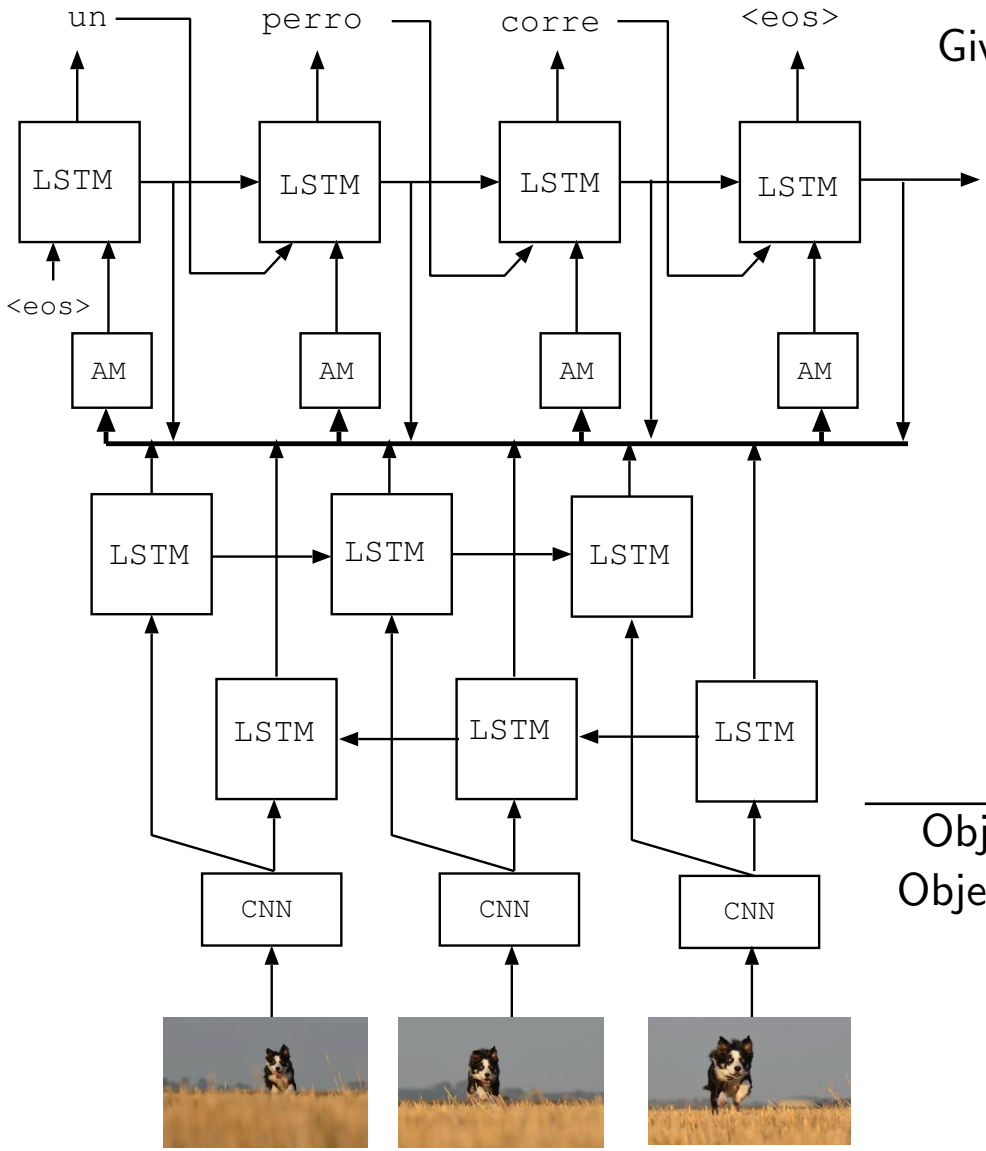
Video description using LSTM and CNN (Venugopalan 2015)



Experiments: MSVD

Systems	METEOR
AlexNet	29.1
GoogleNet	29.6
S2VT	29.8

Video description using LSTM and CNN (Bolaños 2018)



Given a video v, search for a description of the video $\hat{t}$:

$$\hat{t}_1^{\hat{I}} = \underset{I,t}{\text{argmax}}\, p(t_1^I \mid v) = \underset{I,t}{\text{argmax}} \prod_{i=1}^{I} p(t_i \mid t_1^{i-1}, v)$$

Microsoft Research Video Description Corpus

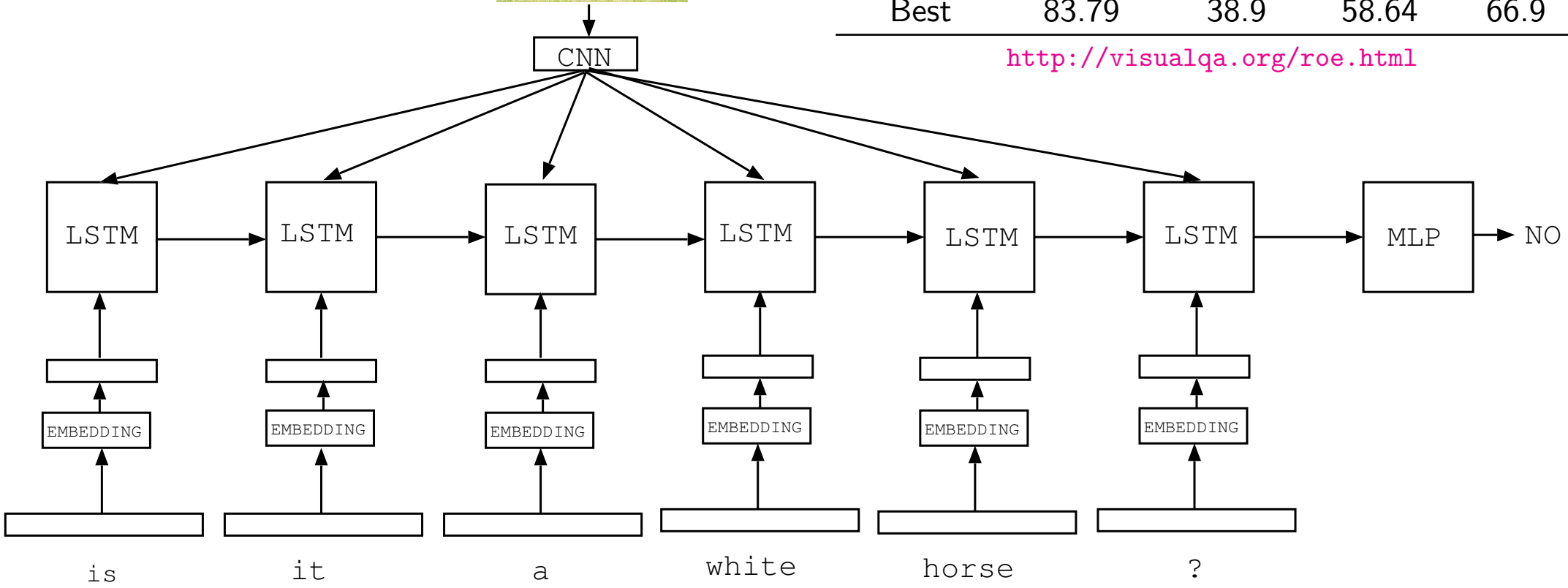
Encoder	BLEU (%)	METEOR (%)
Objects + LSTM (Yao 2015)	51.5	32.5
Objects + BLSTM (Peris 2016)	53.6	32.6

Visual question answering

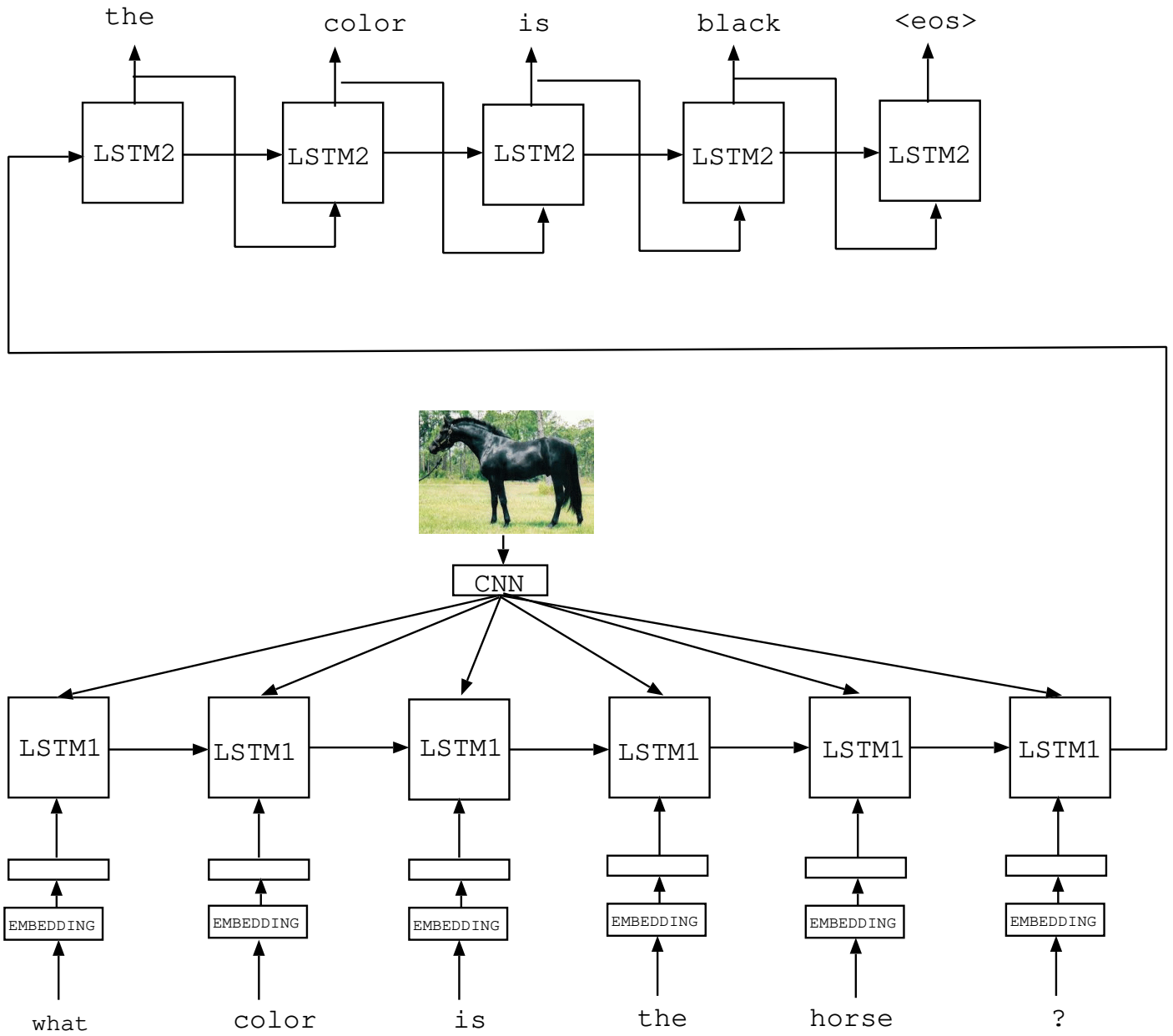
VQA Challenge 2016

System	Yes/No	Number	Other	Overall
UPV-UB	79.24	35.72	40.79	56.1
Best	83.79	38.9	58.64	66.9

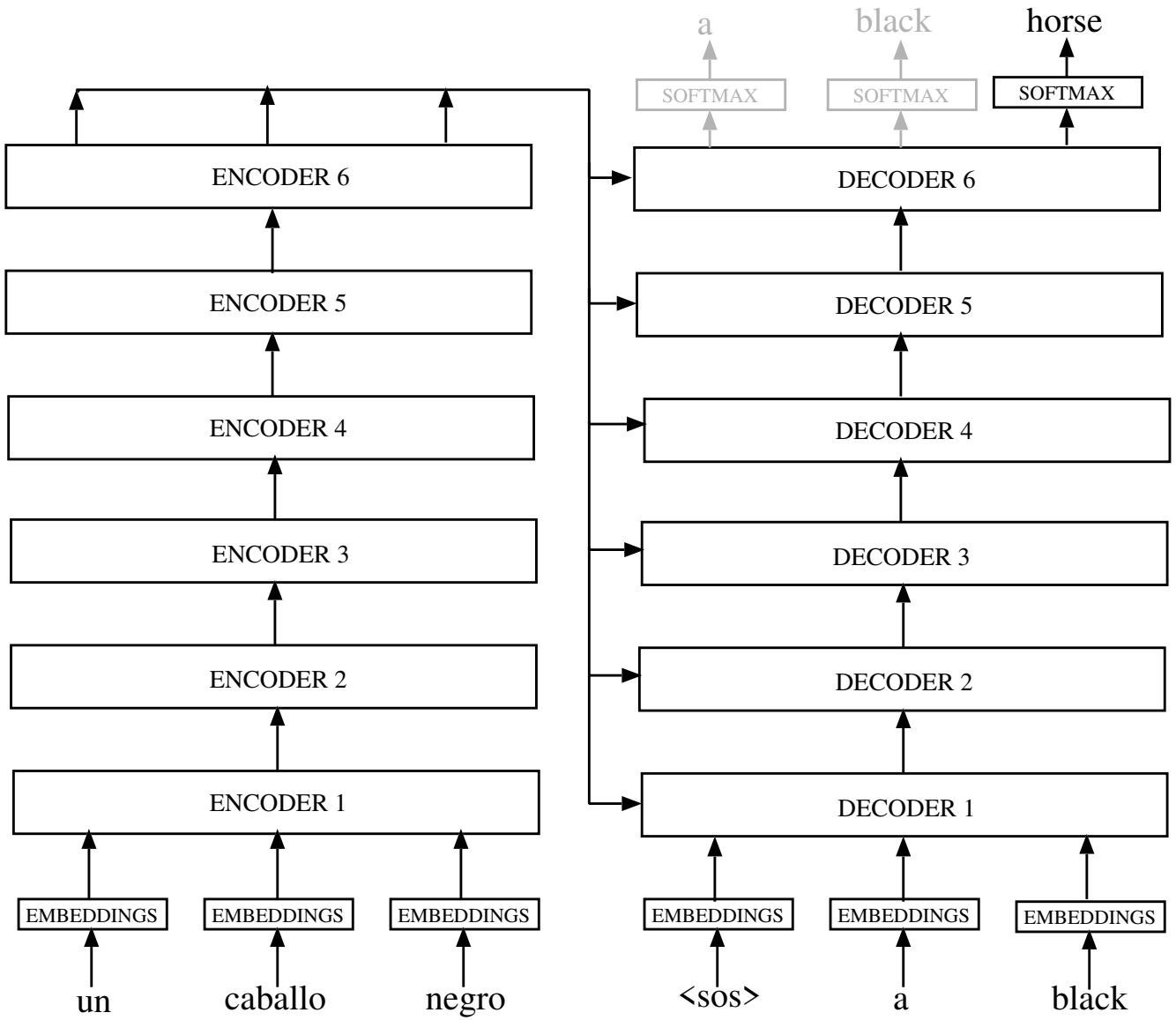
http://visualqa.org/roe.html



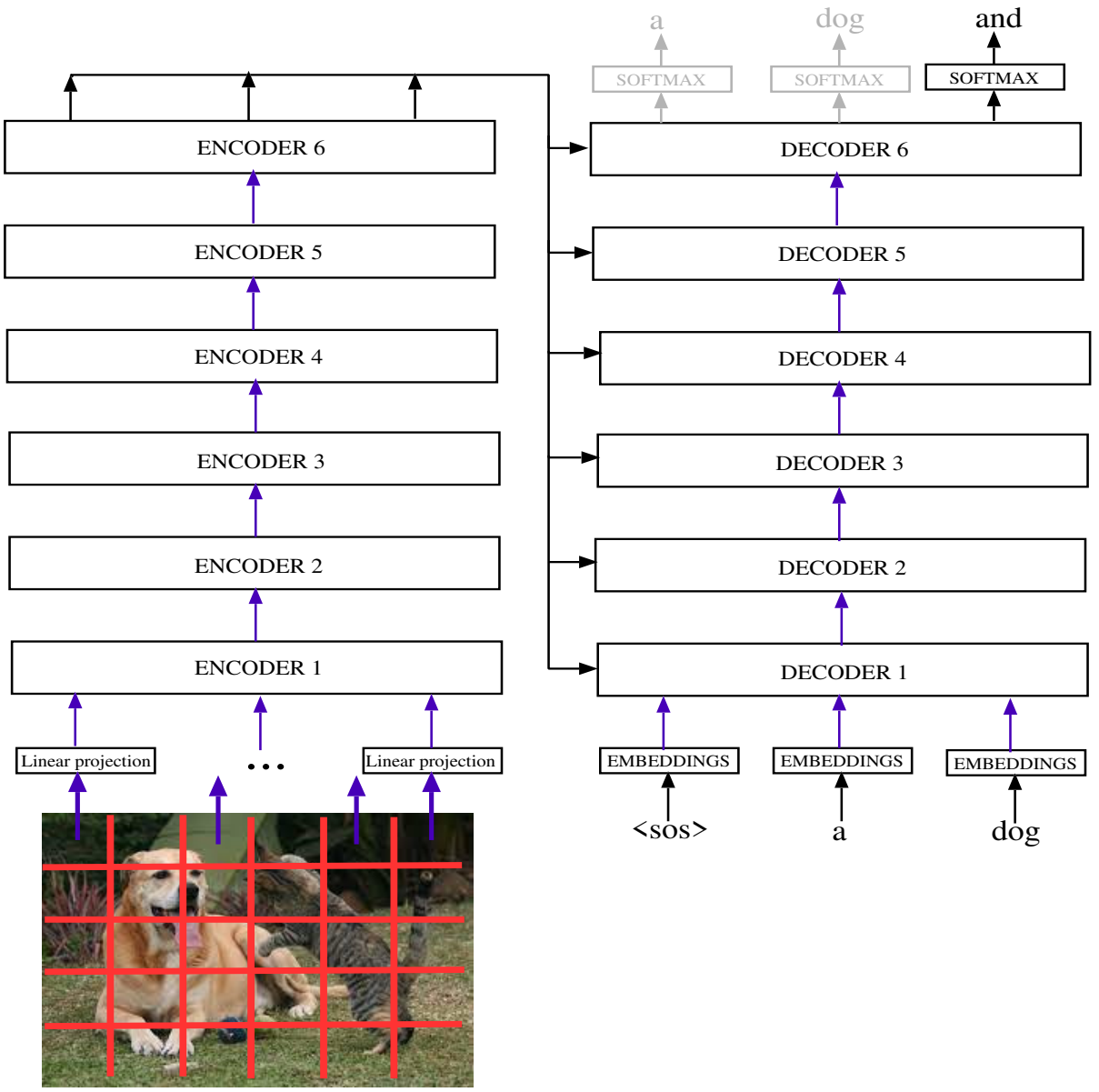
Visual question answering



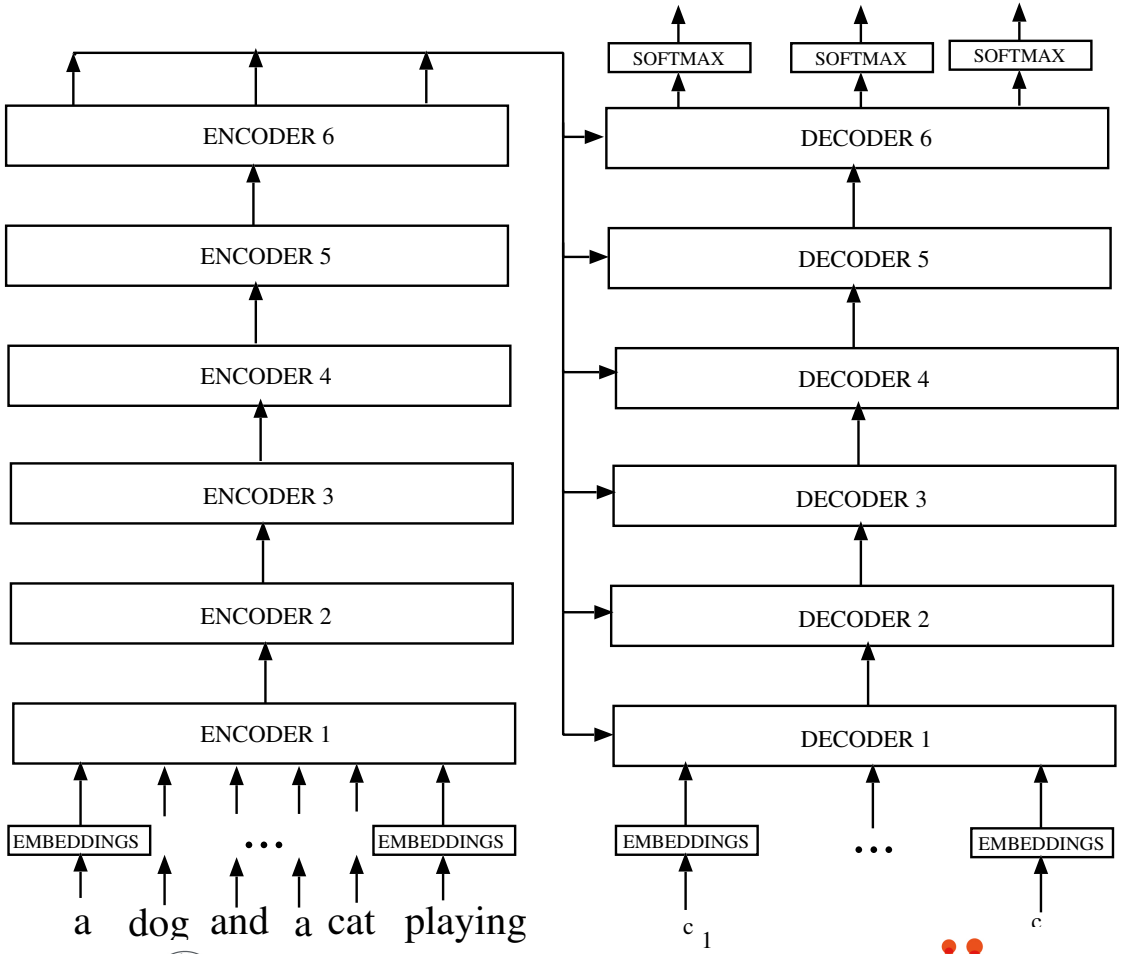
Machine Translation with Transformer (Vaswani 2017)



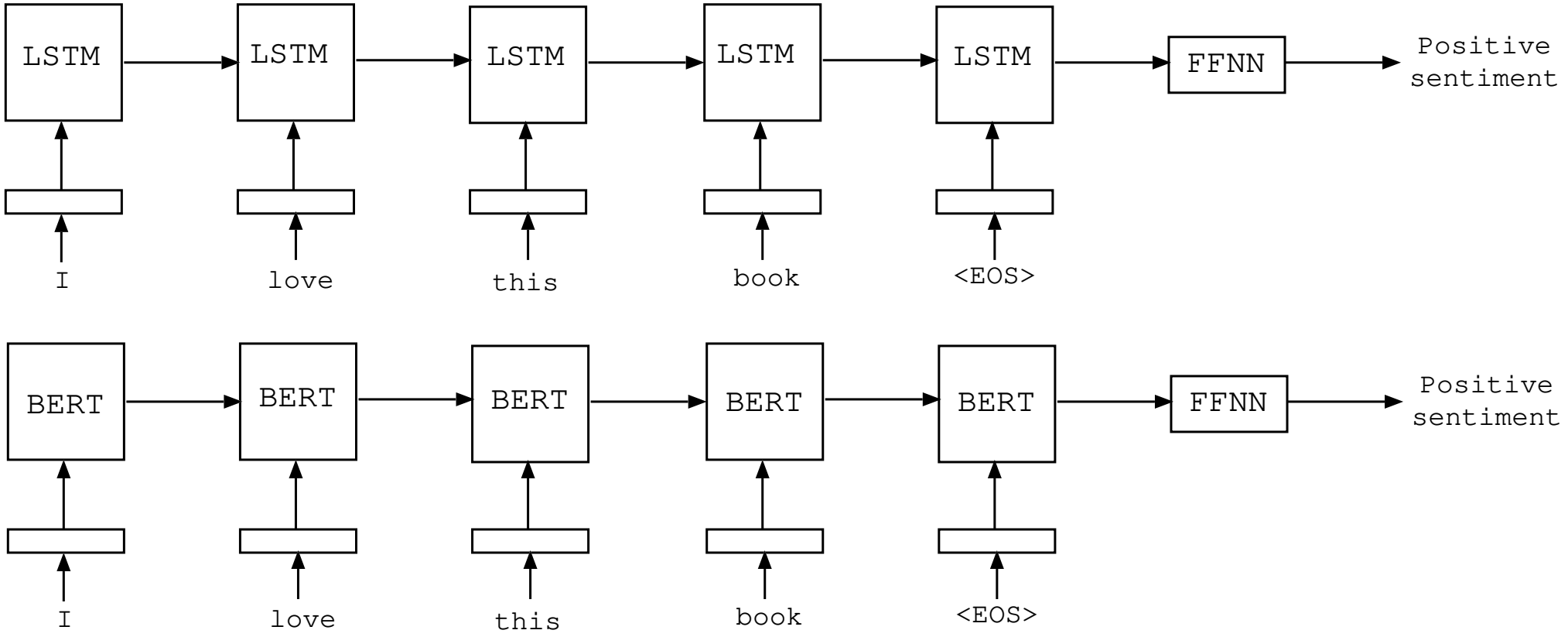
Transformer for image caption [Wang 2022]



Transformer for text-to-image generation



Sentence-level classification [Aggarwal 2018]



Index

1 Sequence-to-sequence architectures ▷ 3

2 Pre-trained models ▷ 35

3 Applications ▷ 54

○ 4 *Bibliography* ▷ 69

Bibliography (I)

- Aggarwal. Neural Networks and Deep Learning. Chap. 7. Springer. 2018.
- Alaycrac et al. Flamingo: a Visual Language Model for Few-Shot Learning. NeuroNIPS. 2022.
- D. Bahdanau, K. Cho, Y. Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. ICLR. 2015.
- Bengio et al. A Neural Probabilistic Language Model. JMLR. 2003.
- Bolaños, A. Peris, F. Casacuberta, S. Soler, P. Radeva. Egocentric Video Description based on Temporally-Linked Sequences. Journal of Visual Communication and Image Representation, 2018.
- Castaño, F. Casacuberta. A Connectionist Approach to Machine Translation. Eurospeech. 1997.
- Cho et. al On the properties of neural machine translation: Encoder-decoder approaches. arXiv. 2014.
- Chowdhery et al. PaLM: Scaling Language Modeling with Pathways. arXiv. 2022.

Bibliography (II)

- Conneau & Lample. Cross-lingual Language Model Pretraining. NeurNIPS
- Devlin et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. NAACL. 2019.
- Dosovitskiy et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. ICLR 2020.
- Elliott et. al. Multilingual Image Description with Neural Sequence Models. ICLR. 2016.
- Graves. Supervised Sequence Labelling with Recurrent Neural Networks. Springer. 2012.
- A. Graves, S. Fernández, F. Gómez. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. ICML 2006.
- Hawthorne et al. General-purpose, long-context autoregressive modeling with Perceiver AR. ICML 2022.
- Hochreiter & Schmidhuber. "Long short-term memory. Neural Computation. 1997.
- Hoffman et. al. Training Compute-Optimal Large Language Models. arXiv. 2022.

Bibliography (III)

- Koehn. Neural Machine Translation. Cambridge University Press. 2020.
- Kondratenko & Kuperin. Using Recurrent Neural Networks To Forecasting of Forex. arXiv. 2003.
- Katzer et. al. Prediction of Human Full-Body Movements with Motion Optimization and Recurrent Neural Networks. IEEE ICRA. 2020.
- Lewis et al. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. ACL. 2020.
- Li & Jia. Visual Question Answering with Question Representation Update. NeurNIPS. 2016.
- Lin et. al. Pre-training Multilingual Neural Machine Translation by Leveraging Alignment Information. arXiv. 2021.
- Liu et al. Multilingual Denoising Pre-training for Neural Machine Translation. TACL. 2020.

Bibliography (IV)

- Liwicki et al.A Novel Approach to On-Line Handwriting Recognition Based on Bidirectional Long Short-Term Memory Networks. ICDAR 2007.
- Lundsteed. A prototype real-time forecast service of space weather and effects using knowledge-based. Neurocomputing, 2001.
- Luong et al. Effective Approaches to Attention-based Neural Machine Translation. EMNLP. 2015.
- C. Martínez, A. Juan, F. Casacuberta. Iterative Contextual Recurrent Classification of Chromosomes. Neural Processing Letters. 2007.
- Mandic & Chambers. Recurrent Neural Networks for Prediction. John Wiley & Sons 2001.
- Medsker & Jain. Recurrent Neural Networks. CRC Press. 2001.
- Mikolov et al. Distributed Representations of Words and Phrases and their Compositionality. NIPS 2013.

Bibliography (V)

- Mishra & Viradiya. Survey of Sentence Embedding Methods. JASC. 2019.
- Peters et. al. Deep contextualized word representations. NAACL. 2018.
- Qiu et. al. Pre-trained Models for Natural Language Processing: A Survey. SCTS. 2020.
- Radford et. al. Improving Language Understanding by Generative Pre-Training. OpenAI. 2018.
- Radford et. al. Learning Transferable Visual Models From Natural Language Supervision. ICML. 2021.
- Rae et. al. Scaling Language Models: Methods, Analysis & Insights from Training Gopher. DeepMind. 2021
- Raffel et. al. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. JMLR. 2020.
- Reed at al. A Generalist Agent. MLR. 2022.

Bibliography (VI)

- Reimers & Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. EMNLP 2019.
- Romero. Ultra-Large AI Models Are Over. https://thealgorithmicbridge.substack.com/p/ultra-large-ai-models-are-over. 2022.
- Schuster & Paliwal. Bidirectional Recurrent Neural Networks. IEEETSP. 1997.
- Sejnowski & Rosenberg. NETtalk: a parallel network that learns to read aloud. The Johns Hopkins University electrical engineering and computer science technical report. 1986.
- Schuster & Paliwal. Bidirectional recurrent neural networks. IEEE TSP. 1995.
- Sinha et. al. Extractive Text Summarization using Neural Networks. arXiv. 2018.
- Sundermeyer. Improvements in Language and Translation Modeling. Ph.D. RWTH Aachen. 2016.
- Tunstall et al. Efficient Few-Shot Learning Without Prompts. arXiv 2022.
- A. Vaswani et al. Attention Is All You Need. NIPS 2017.

Bibliography (VII)

- Venugopalan et al. Sequence to Sequence - Video to Text. arXiv. 2015.
- Vinyals et al. A Neural Image Caption Generator (NIC). arXiv. 2015
- Wang et al. End-to-End Transformer Based Model for Image Captioning. AAAI 2022.
- Waibel et. al. Phoneme Recognition Using Time-Delay Neural Networks. IEEE TASSP 1989.
- Williams & Zipser. Gradient-based learning algorithms for recurrent networks and Their Computational Complexity. In "Back-propagation: Theory, Architectures and Applications". 1995.
- Xiong et al. On Layer Normalization in the Transformer Architecture. arXiv. 2020.
- Yang wt. al. XLNet: Generalized Autoregressive Pretraining for Language Understanding. NIPS. 2020.
- Yao & Wan. Multimodal Transformer for Multimodal Machine Translation. ACL. 2020.