Master's degree ICT Internet Multimedia Engineering

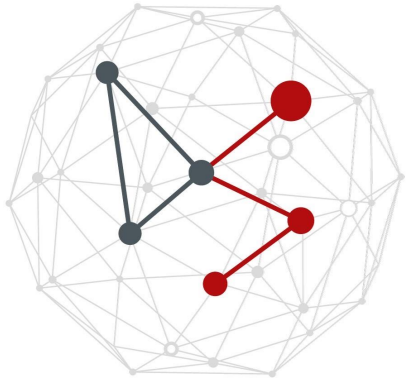
ADVANCED ARCHITECTURES: SEQ2SEQ AND VISUAL ATTENTION

Michele Rossi

michele.rossi@unipd.it

Dept. of Information Engineering

University of Padova, IT



DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

DIPARTIMENTO MATEMATICA

800 ANNI 1222·2022

UNIVERSITÀ DEGLI STUDI DI PADOVA

Outline

- The attention mechanism
  - Attention for temporal sequences
    - Example: Natural Language Translation (NLT)
  - Problem with *plain* Seq-2-Seq models
  - Solution
    - the **attention mechanism**
  - Alignment of hidden states @encoder with input sequence
    - Jointly learning to *align* and *translate*
    - Experimental results
  - Attention for 2D images
    - Example results

Bibliography

[Bahdanau15] D. Bahdanau, K. Cho, Y. Bengio, “Neural machine translation by jointly learning to align and translate,” International Conference on Learning Representations (ICLR), San Diego, CA, US, May 2015. [25918 citations, Oct. 2022]

Example TensorFLow implementation

https://github.com/hlamba28/NMT-with-Attention-Mechanism

Published as a conference paper at ICLR 2015

NEURAL MACHINE TRANSLATION BY JOINTLY LEARNING TO ALIGN AND TRANSLATE

Dzmitry Bahdanau
Jacobs University Bremen, Germany

KyungHyun Cho **Yoshua Bengio***
Université de Montréal

ABSTRACT

Neural machine translation is a recently proposed approach to machine translation. Unlike the traditional statistical machine translation, the neural machine translation aims at building a single neural network that can be jointly tuned to maximize the translation performance. The models proposed recently for neural machine translation often belong to a family of encoder–decoders and encode a source sentence into a fixed-length vector from which a decoder generates a translation. In this paper, we conjecture that the use of a fixed-length vector is a bottleneck in improving the performance of this basic encoder–decoder architecture, and propose to extend this by allowing a model to automatically (soft-)search for parts of a source sentence that are relevant to predicting a target word, without having to form these parts as a hard segment explicitly. With this new approach, we achieve a translation performance comparable to the existing state-of-the-art phrase-based system on the task of English-to-French translation. Furthermore, qualitative analysis reveals that the (soft-)alignments found by the model agree well with our intuition.

Attention is used at all times by humans

- Motivated by how we pay visual attention to, e.g., the *details in an image* or *words in a sentence*

Attention is used at all times by humans

- ID of the spaceship (I570DLV), name of the fleet (Galactica)

Attention is used at all times by humans

- Shape and colors of the nose, including coat of arms

Attention mechanism

- Initially developed for **Neural Machine Translation (NMT)** using the Seq-2-Seq architecture [Bahdanau15]

- Seq-2-Seq
  - Encoder: processes the input sequence and encodes/summarizes it into a *single* context vector (the code) of a **fixed length**
  - Decoder: is initialized with the context vector and must generate the transformed output sequence from the sole information contained in the context

- Critical point: encoder
  - **Encoder:** The generated context vector is likely to be insufficient (alone) to capture the structure and content of long sequences (verified *empirically*)

Attention mechanism

- Initially developed for **Neural Machine Translation (NMT)** using the Seq-2-Seq architecture [Bahdanau15]
- Seq-2-Seq
  - Encoder: processes the input sequence and encodes/summarizes it into a *single* context vector (the code) of a **fixed length**
  - Decoder: is initialized with the context vector and must generate the transformed output sequence from the sole information contained in the context
- Critical for the decoder
  - **Decoder:** The decoder sees a single compressed representation of the source. However, at each generation step different parts of source can be more useful than others. The decoder has to extract relevant information from the same fixed representation

Attention - intuition

Attention – which part of the text we should focus on?

focus			attention vectors (sum to 1)
The	→	“The big red dog”	[0.71 0.20 0.07 0.02]
Big	→	“The big red dog”	[0.17 0.65 0.17 0.01]
Red	→	“The big red dog”	[0.01 0.17 0.65 0.17]
Dog	→	“The big red dog”	[0.02 0.07 0.20 0.71]

Attention – long term dependencies

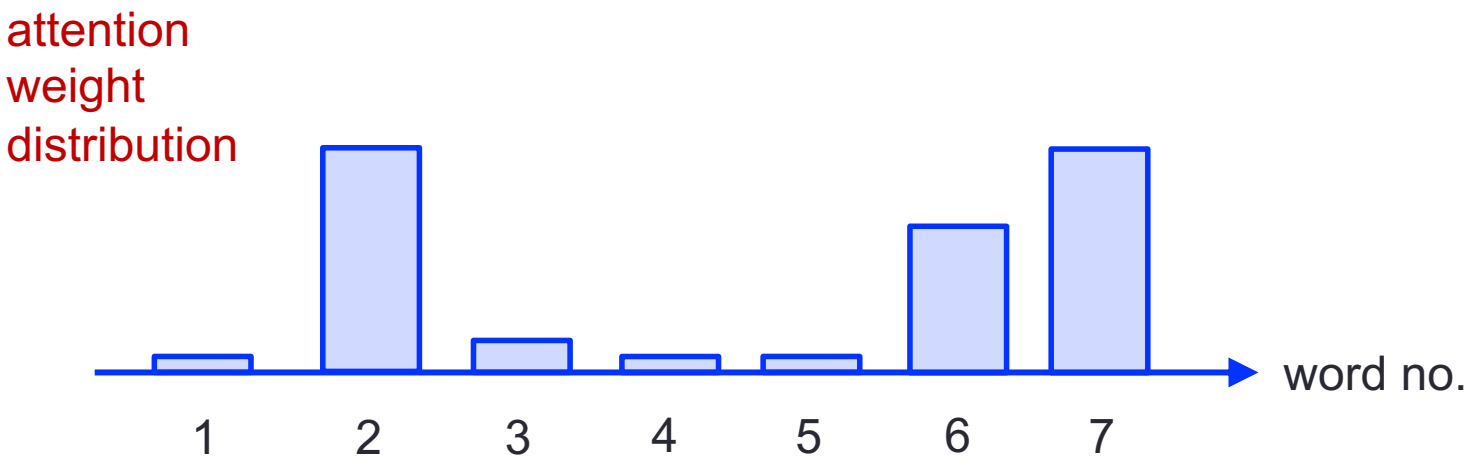
We would like to capture long term dependencies

focus

"This noise is caused by road construction"

attention vector (sum to 1)

[0.01 0.35 0.02 0.01 0.01 0.25 0.35]



Intermission: word embeddings

- What are they? They amount to representing an *input sequence of words* as a *sequence of vectors* (of the same dimension, say, m)
- In Natural Language Processing (NLP) tasks a vocabulary (also called dictionary) is required
- It can be represented using one-hot vectors, e.g., for d=10 words

word	index	one-hot encoding
ant	1	(1,0,0,0,0,0,0,0,0,0)
blue	2	(0,1,0,0,0,0,0,0,0,0)
car	3	(0,0,1,0,0,0,0,0,0,0)
flag	4	(0,0,0,1,0,0,0,0,0,0)
is	5	(0,0,0,0,1,0,0,0,0,0)
juice	6	(0,0,0,0,0,1,0,0,0,0)
kite	7	(0,0,0,0,0,0,1,0,0,0)
open	8	(0,0,0,0,0,0,0,1,0,0)
the	9	(0,0,0,0,0,0,0,0,1,0)
zebra	10	(0,0,0,0,0,0,0,0,0,1)

Word embeddings

- With one-hot representations:
  - the dot product between any two words is zero
  - the distance between any two words is the same
  - this indicates that we have no prior knowledge of the similarity between words
  - we would like to build a new embedding space where this similarity exists

word	index	one-hot encoding
ant	1	(1,0,0,0,0,0,0,0,0,0)
blue	2	(0,1,0,0,0,0,0,0,0,0)
car	3	(0,0,1,0,0,0,0,0,0,0)
flag	4	(0,0,0,1,0,0,0,0,0,0)
is	5	(0,0,0,0,1,0,0,0,0,0)
juice	6	(0,0,0,0,0,1,0,0,0,0)
kite	7	(0,0,0,0,0,0,1,0,0,0)
open	8	(0,0,0,0,0,0,0,1,0,0)
the	9	(0,0,0,0,0,0,0,0,1,0)
zebra	10	(0,0,0,0,0,0,0,0,0,1)

Word embeddings

- However, using one-hot vectors is inefficient
  - And also not very informative
  - We would like to represent words in a space where similar words are close (similarity and closeness)
- For this, we use embedding matrices
- An embedding matrix **E** of size mxd (m<<d)
  - Transforms one-hot vectors of size d into vectors of size m
  - d is the number of words in the vocabulary, it is large, e.g., 50,000
  - m is the size of each word in an embedded space, e.g., 300
- So, we have

$$\boldsymbol{x}_i = \boldsymbol{E}\boldsymbol{w}_i$$

word in embedding space

word in original one-hot encoding space

Embedding space

- It is a list of all words in dictionary and their corresponding m-dimensional embeddings
- Each row corresponds to a column of matrix **E** (the column selected by $\mathbf{w}_i$ expressed in one-hot form)

hello	12	45	43	26	78	532	...
there	43	25	778	43	53	78	...
texas	34	56	23	12	56	74	...
world	342	54	23	5	7	423	...
...	...						

How to build matrix E

- We list the words in the vocabulary (one-hot-representation)
- For each: we identify a number of features

Feature	Man	Woman	King	Queen	Apple	Orange
Gender	-1	1	-0.95	0.97	0.00	0.01

How to build matrix E

- We list the words in the vocabulary (one-hot-representation)
- For each: we identify a number of features

Feature	Man	Woman	King	Queen	Apple	Orange
Gender	-1	1	-0.95	0.97	0.00	0.01
Royal	0.01	0.02	0.93	0.95	-0.01	0.00
Age	0.03	0.02	0.7	0.69	0.03	-0.02
Food	0.04	0.01	0.02	0.01	0.97	0.95
…						

How to build matrix E

- We list the words in the vocabulary (one-hot-representation)
- For each: we identify a number of features

Feature	Man	Woman	King	Queen	Apple	Orange
Gender	-1	1	-0.95	0.97	0.00	0.01
Royal	0.01	0.02	0.93	0.95	-0.01	0.00
Age	0.03	0.02	0.7	0.69	0.03	-0.02
Food	0.04	0.01	0.02	0.01	0.97	0.95
…						

- From this representation
  - We can use the columns as the word embeddings
  - We see that apple and orange are similar than, e.g., king and orange

How to build matrix E

- Real E matrices
  - Do not contain clearly interpretable features
  - Are automatically obtained through a supervised ML algorithm
- Since labeling word similarity is hard/impossible
  - Embedding and similarity learning is obtained indirectly
  - Solving a classification problem depending on the context of words when they appear into sentences
  - E.g., the context implied by apple and orange in the following two sentences is of the same kind
    - “I drink an apple juice”
    - “We are drinking orange juice”
  - If the embedding is good at representing context is supposedly good at representing the word in an embedded space

Example problem

Translate the input sentence

"Joe is a good boy" (english)

Into the output sequence

"Joe è un bravo ragazzo" (italian)

Using a Seq-2-Seq model

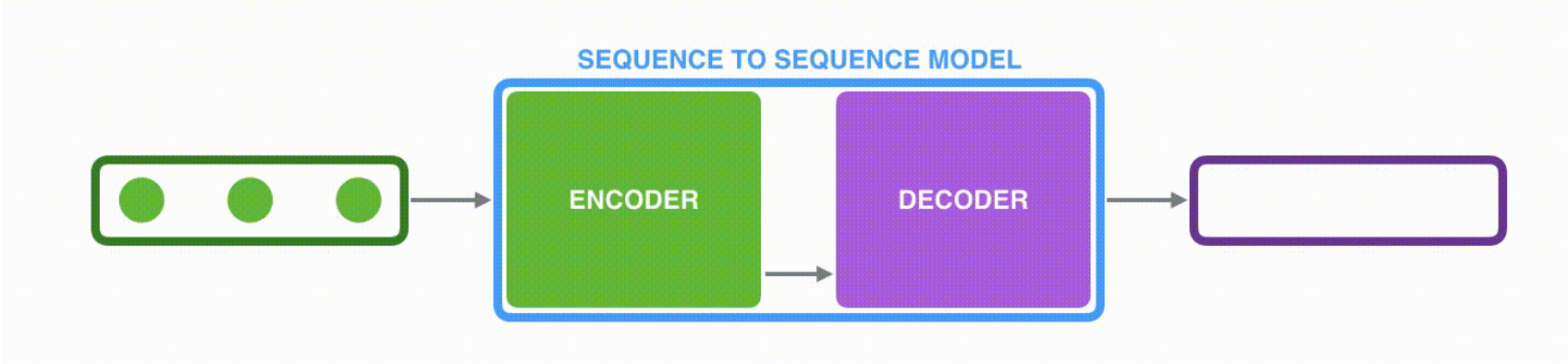
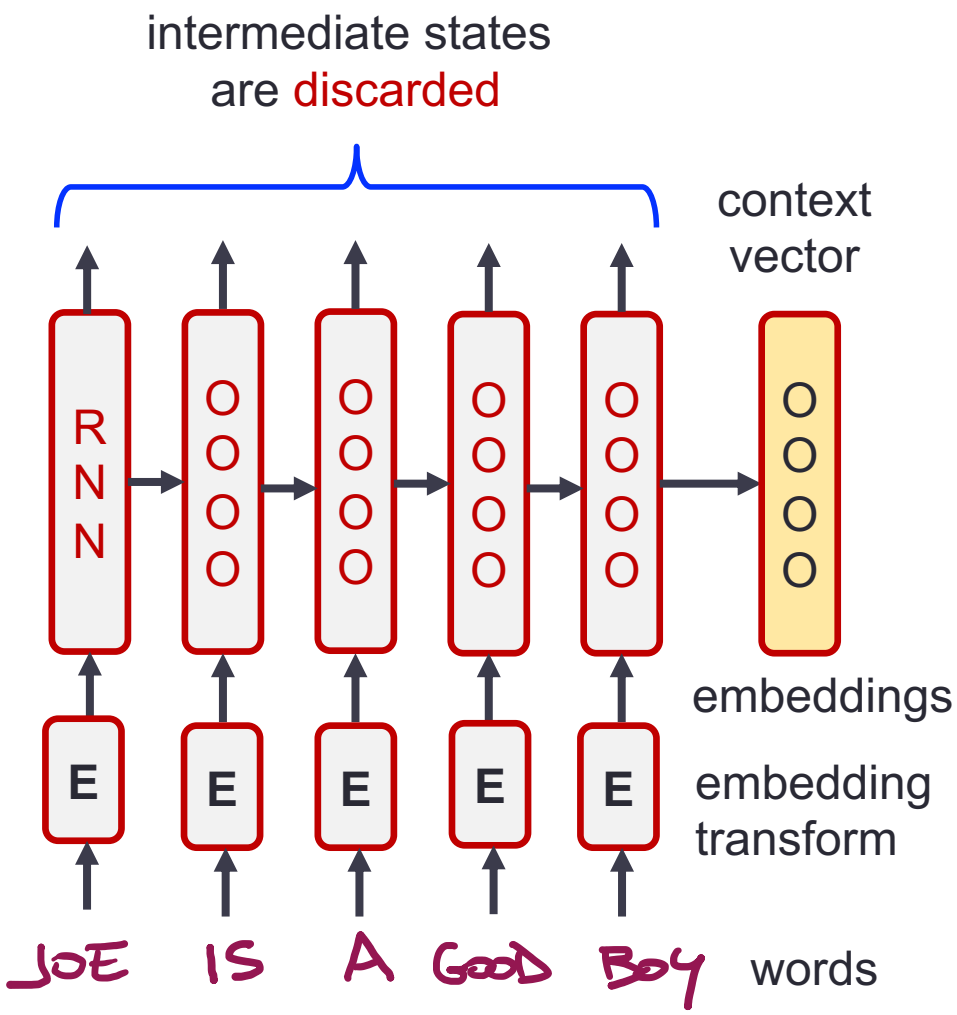


Figure: Jay Alammar, "Visualizing a Neural Machine Translation Model," 2018

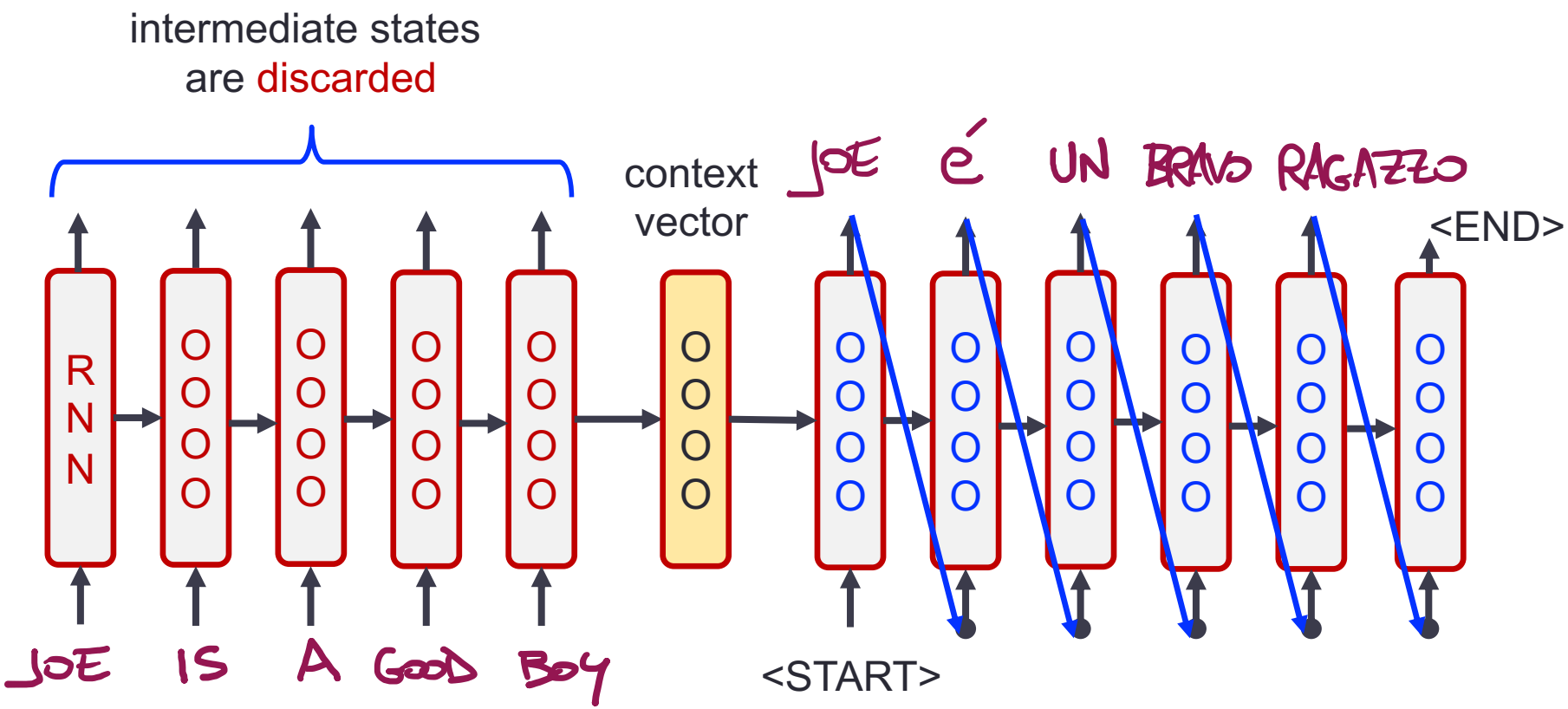
Seq-2-Seq: encoder



1. The encoder encodes the entire input sequence into a single context vector

2. Send the final state of the encoder as the initial state of the decoder

Seq-2-Seq: encoder + decoder



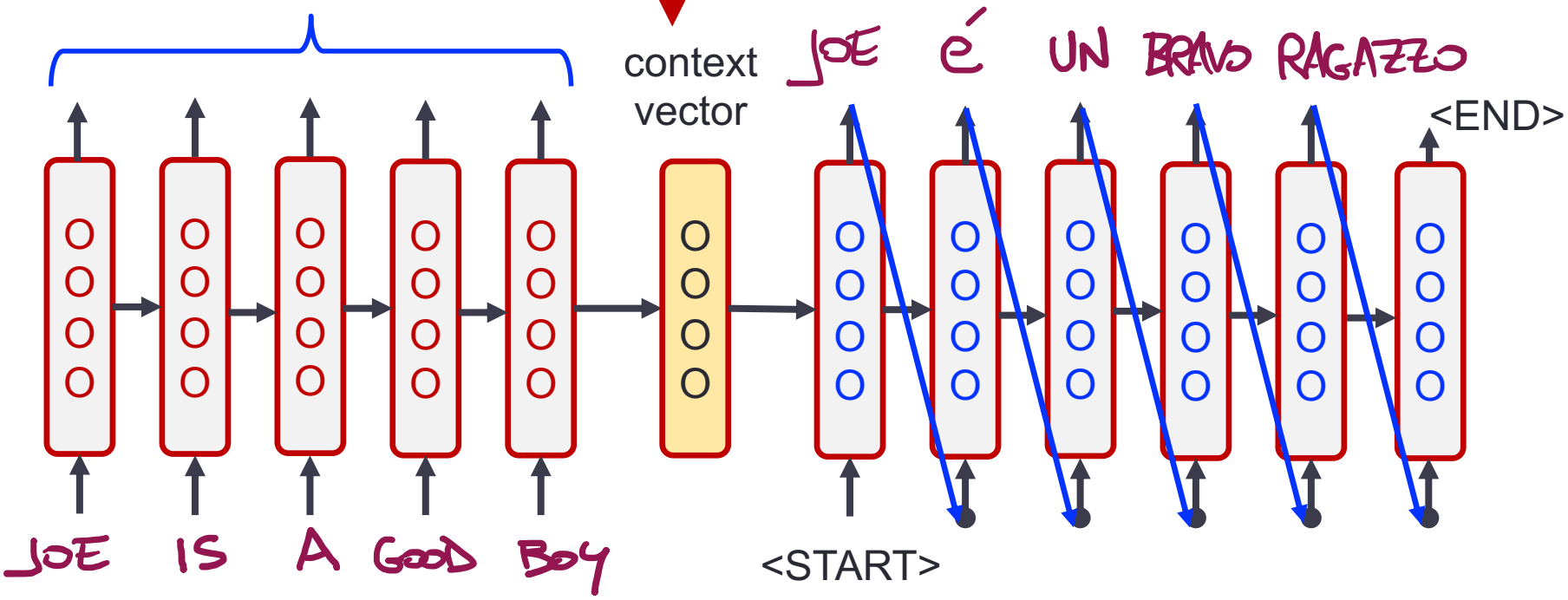
(of course, we always use word embeddings in real implementations)

states and output at each time step become the state and input at the next time step

Seq-2-Seq: encoder + decoder

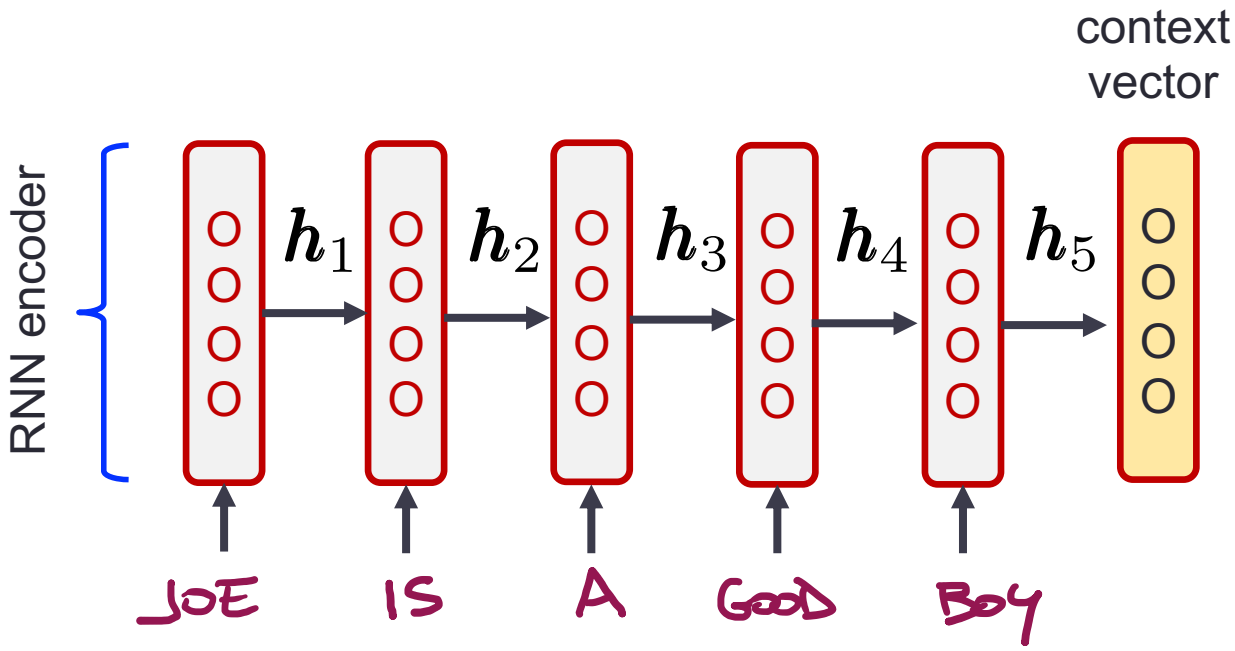
PROBLEM: this is a bottleneck

SOLUTION: do not discard intermediate hidden states



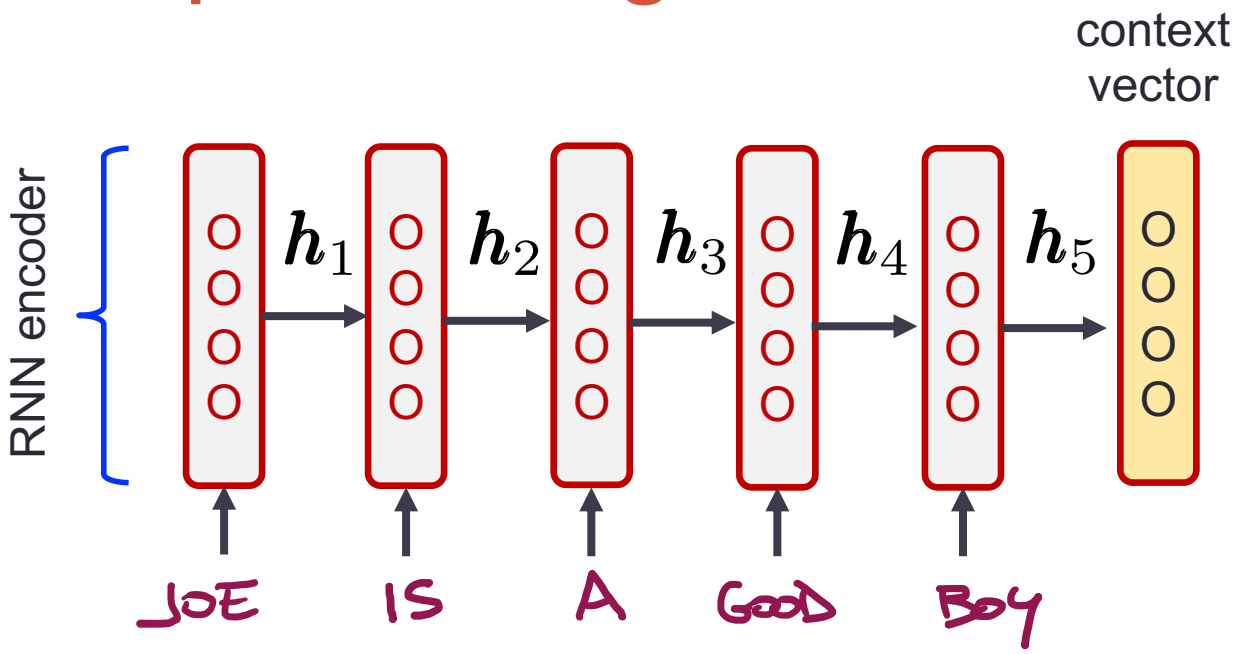
PROBLEM: as the length of the input sequence increases → a single context vector can hardly capture the entire input effectively

Intermediate states



- Intermediate states $\mathbf{h}_1$-$\mathbf{h}_5$ are *hidden* state vectors (fixed size)
  - Store local & global information about the whole source input sentence

Encoder processing



- Encoder processing
  - In 5 time steps, the encoder goes through the entire sequence
  - Obtaining the 5 hidden states $\mathbf{h}_1$, $\mathbf{h}_2$, …, $\mathbf{h}_5$
  - These hidden states are retained and will be all used to decode

Attention mechanism

- At different decoding steps
  - Let the model focus on different parts of the input
  - Deciding which source parts are important
  - This is implemented by computing a distribution of attention weights over the source tokens (input words or encoder states)
- The attention mechanism
  - Is part of a neural network
  - It is jointly trained with the other neural network weights
  - It is given as input all the “source tokens” – in the implementation here described, they correspond to all the encoder states $\mathbf{h}_1, \ldots, \mathbf{h}_5$

Query, keys and values

- Example from *database management*
  - We have a database of pictures – each indexed via a key
  - We issue a query to search for a picture with specific content
  - The picture with the best match between query and key is selected
  - The query could also return a distribution of the best matching pictures

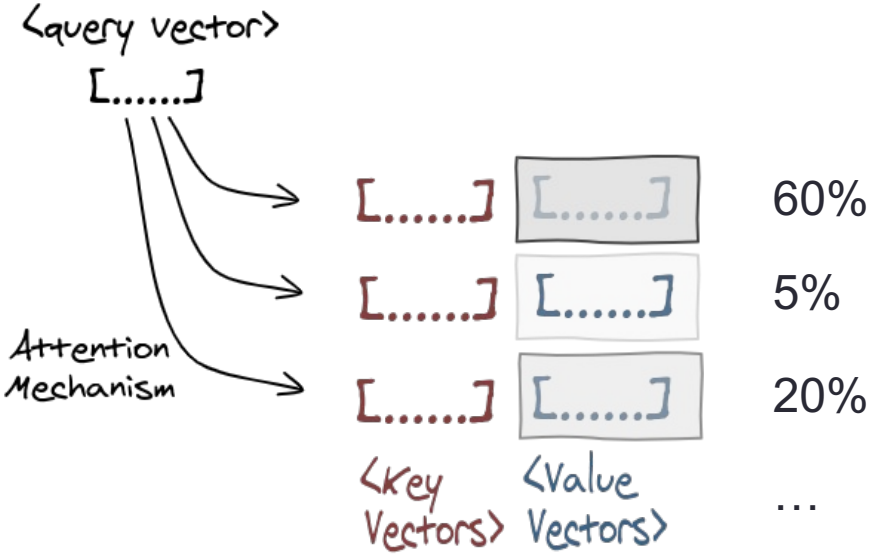
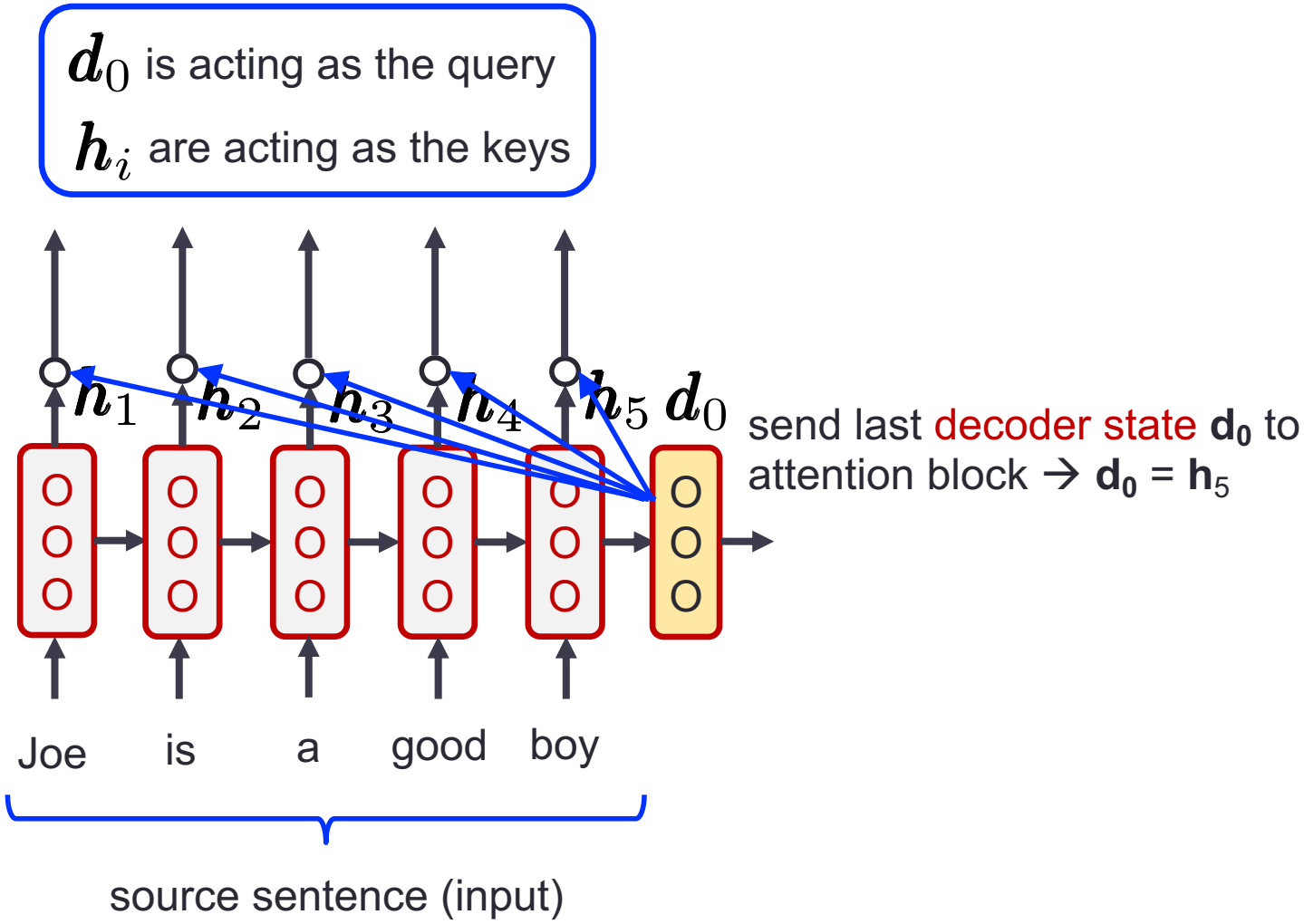
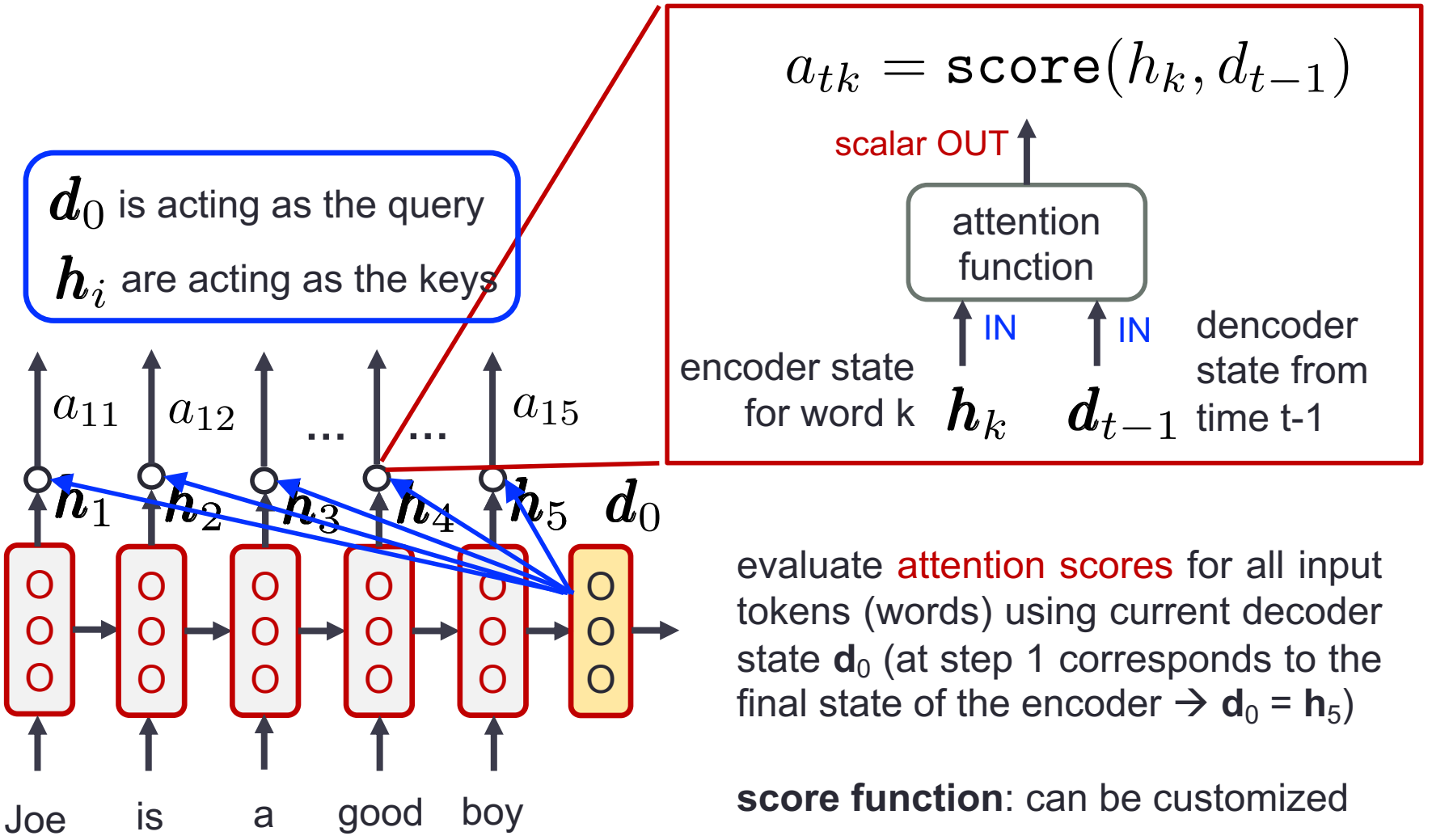


Figure: Nikhil Shah, "Attention? Another perspective," 2020.

First decoding step t=1



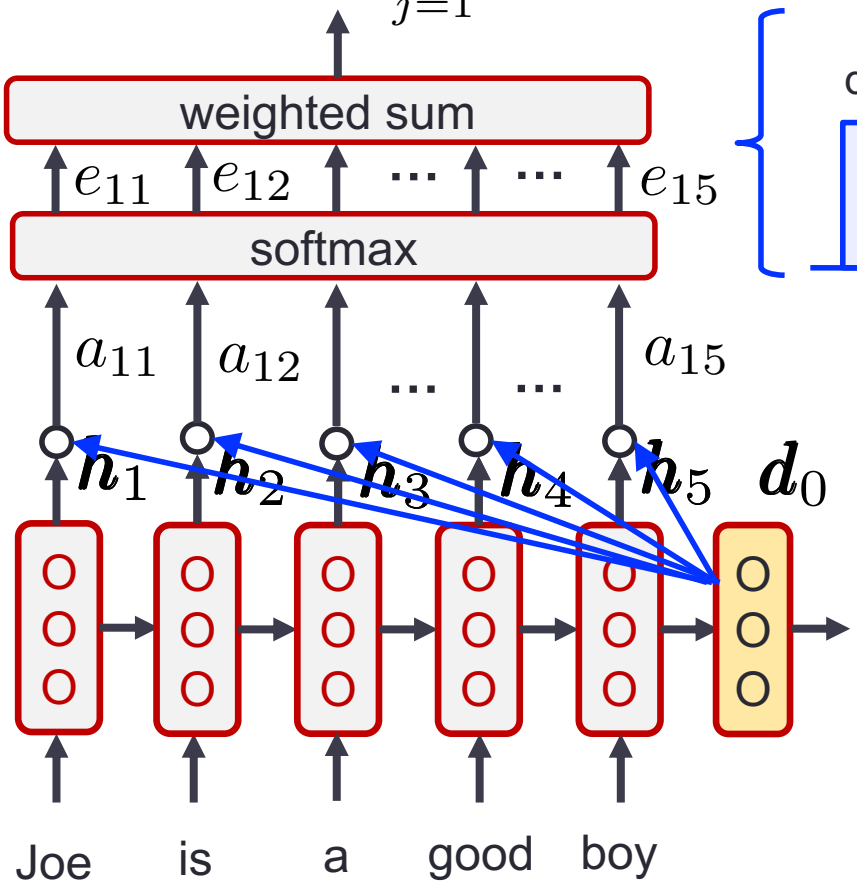


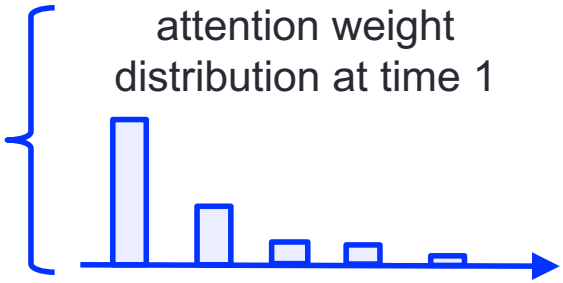
evaluate attention scores for all input tokens (words) using current decoder state $\mathbf{d}_0$ (at step 1 corresponds to the final state of the encoder → $\mathbf{d}_0 = \mathbf{h}_5$)

score function: can be customized

$$CV_1 = \sum_{j=1}^{5} e_{tj} h_j$$

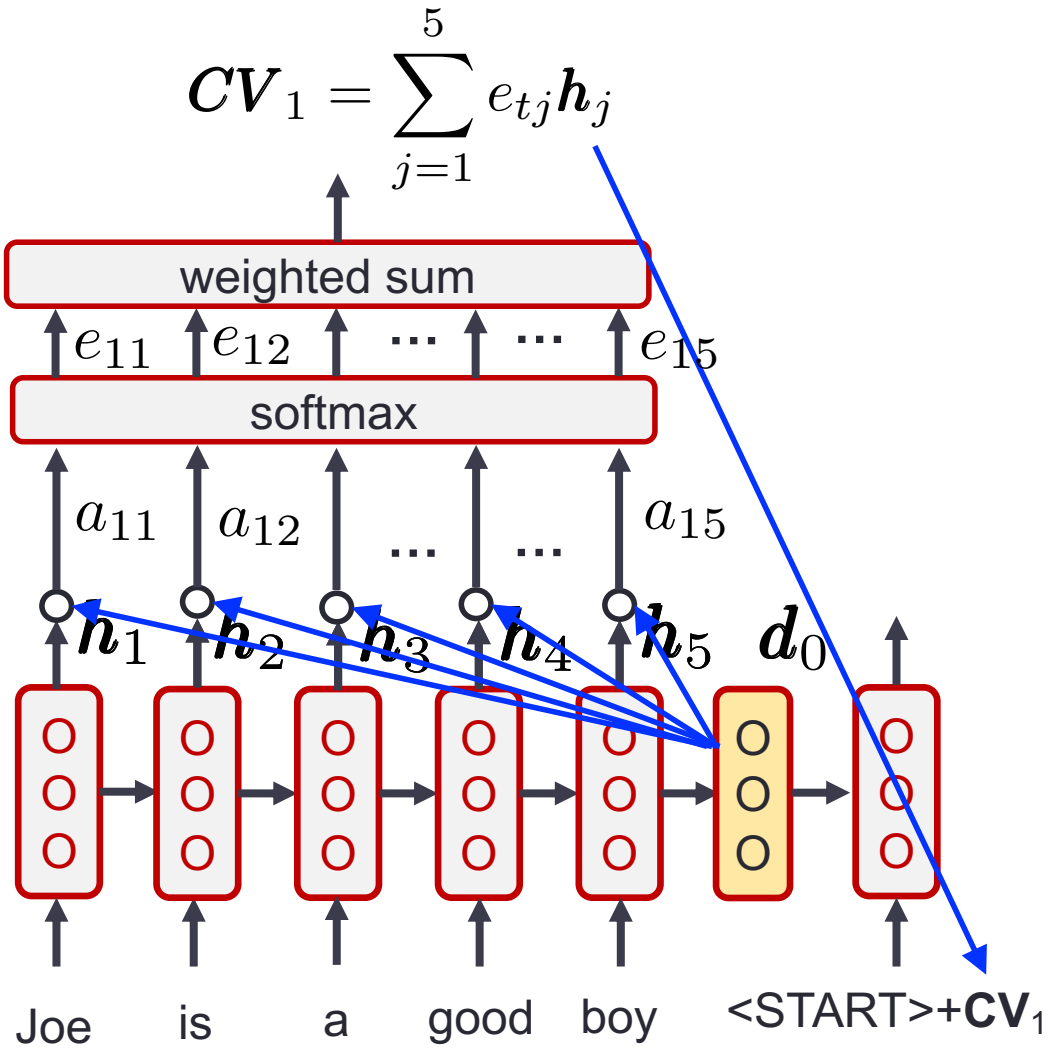
h_i are now acting as the values



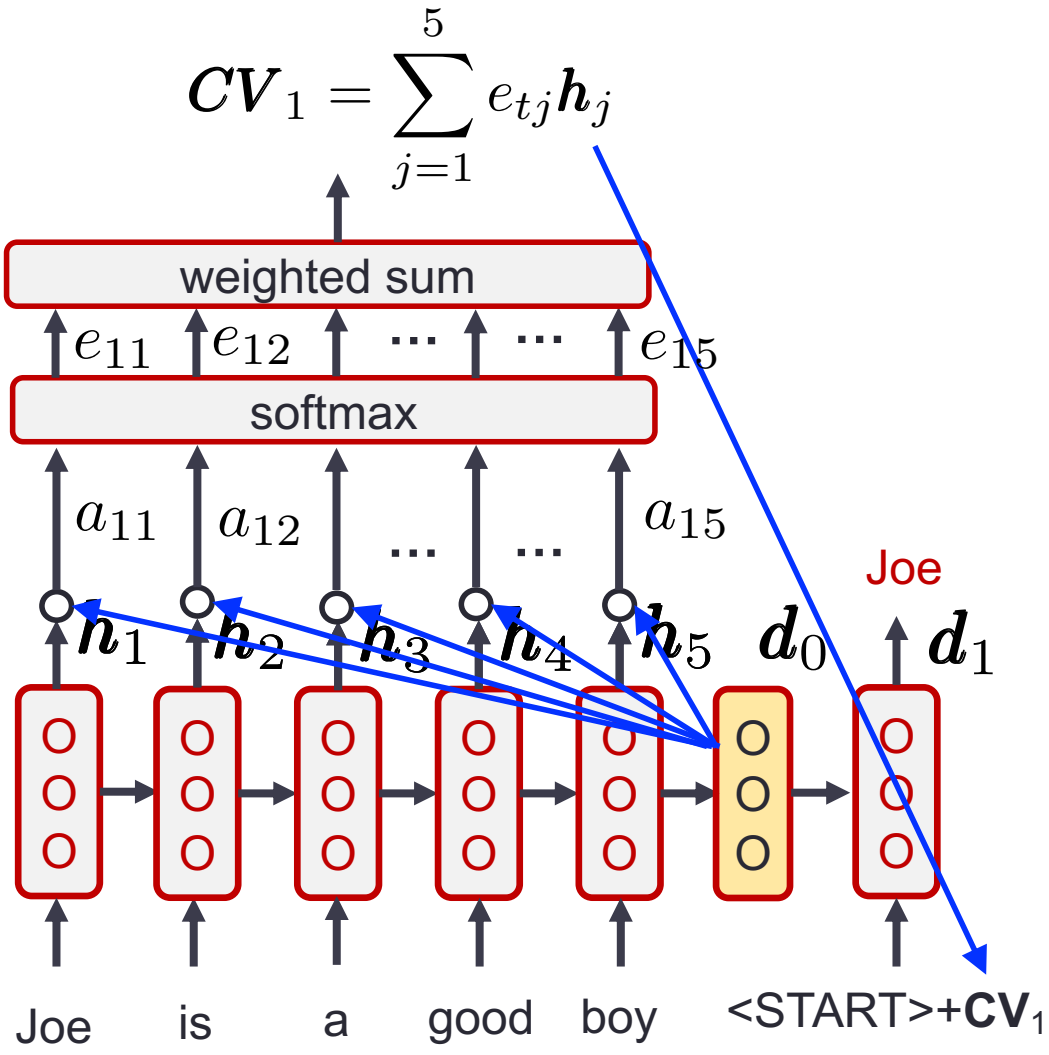


evaluate softmax of attention scores for all input tokens

calculate context vector **CV**$_1$: is is a weighted sum of the encoder states, according to their relative importance at this decoding step

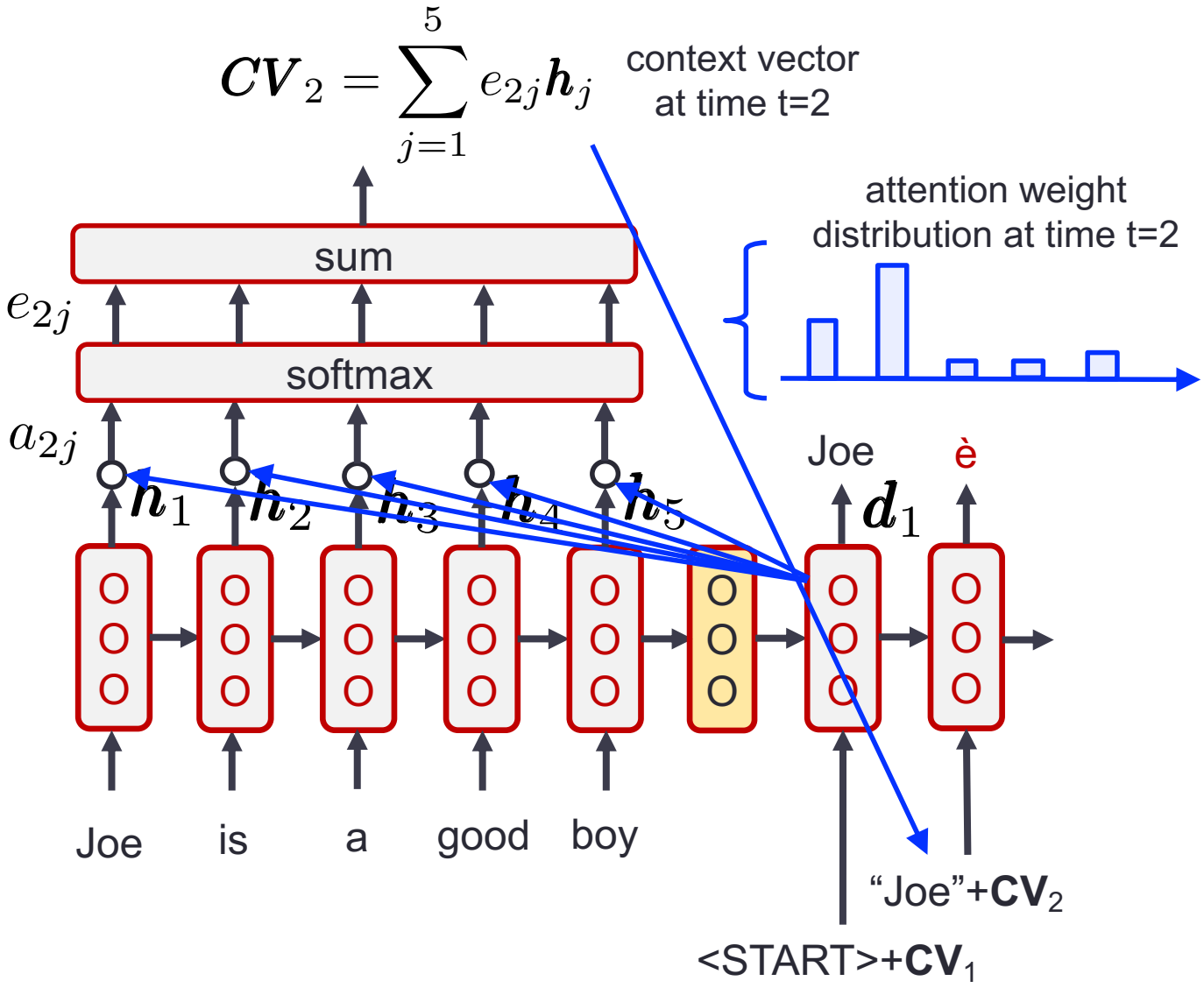


concatenate context vector **CV**$_1$ with current input at the receiver to obtain the next prediction

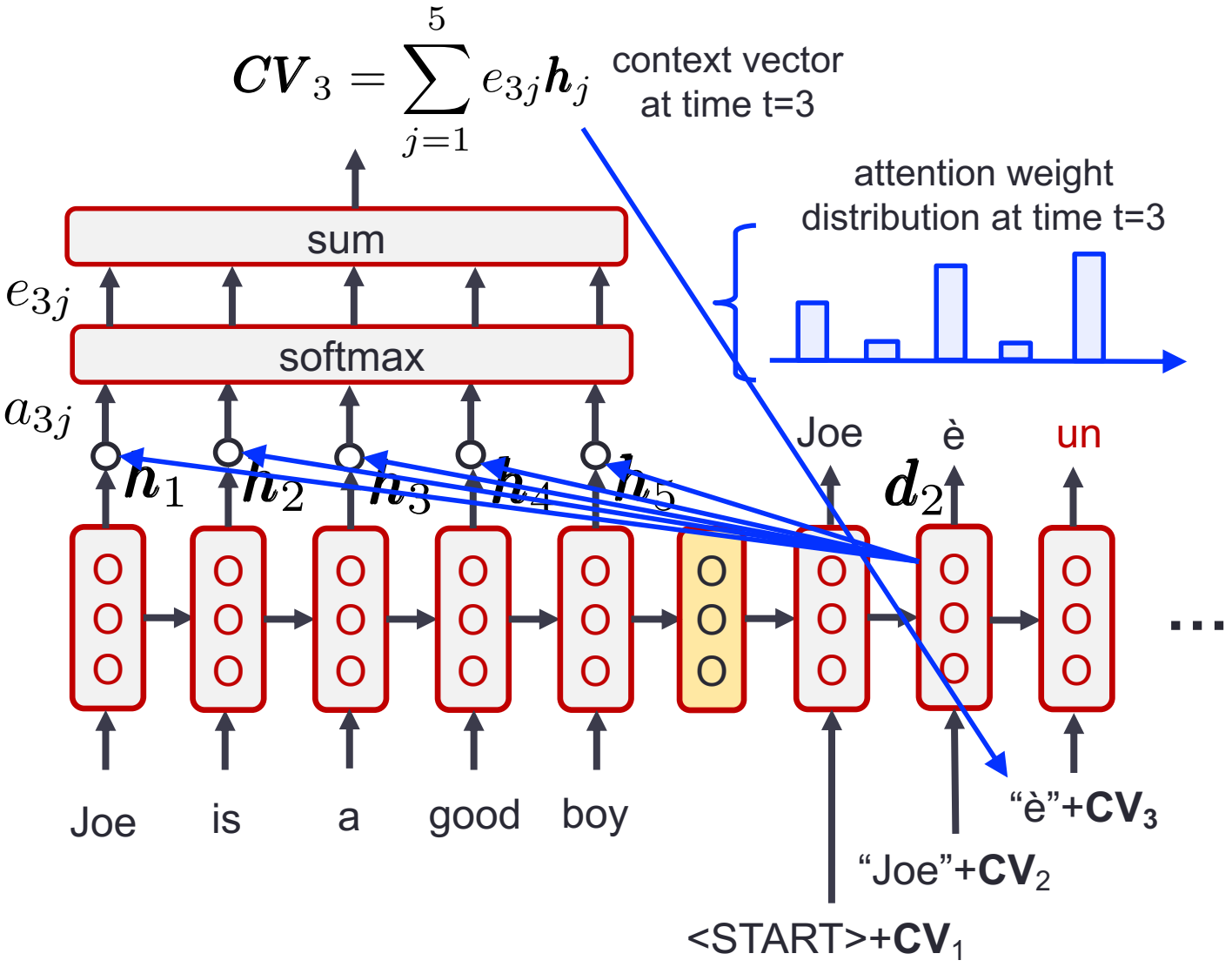


obtain the next prediction "Joe" and update the decoder state $d_0 \rightarrow d_1$

Decoding step t=2



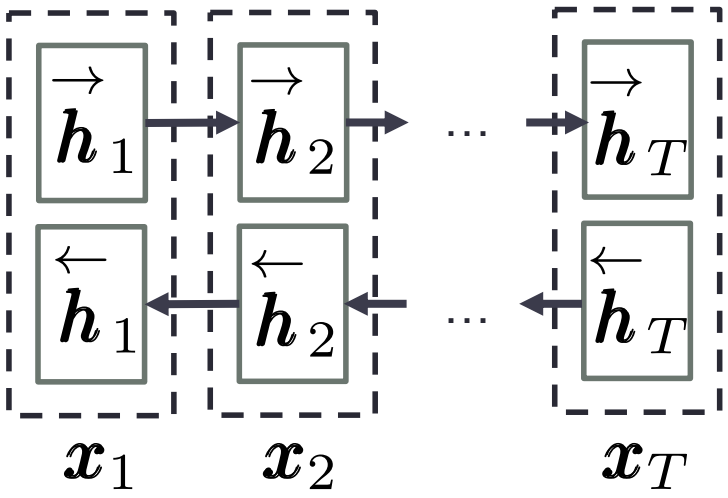
Decoding step t=3



Bi-directional RNN as encoder

- The entire source sequence of length T is inputted into and fully processed by a bi-rectional RNN
  - T hidden states are obtained for each direction (**forward & backward**)
  - The hidden state to be used at time t is obtained by concatenating forward and backward hidden states at that time instant, namely:

$$\boldsymbol{h}_t = [(\overleftarrow{\boldsymbol{h}}_t)^T; (\overrightarrow{\boldsymbol{h}}_t)^T]$$



Input sequence (T words)

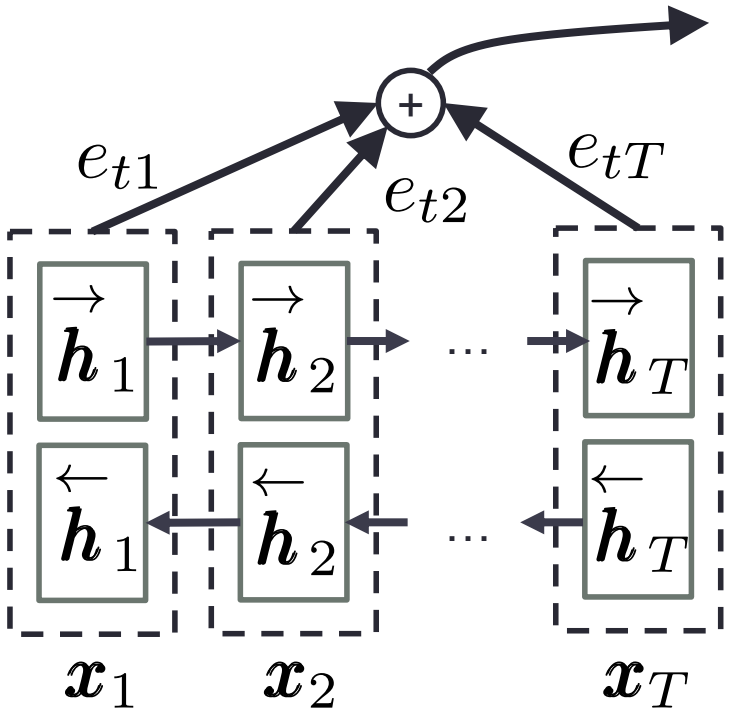
A bi-directional RNN allows capturing more complex dependencies within the input sequence

Context vector – CV

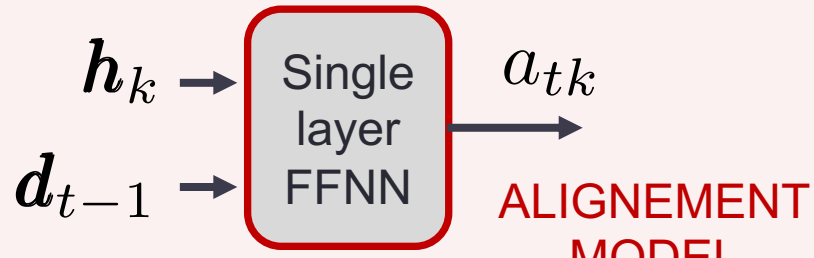
- CV is obtained through the so called alignment model
- It is a FFNN jointly trained with all the system components

$$\boldsymbol{CV}_t = \sum_{j=1}^{T} e_{tj}\boldsymbol{h}_j$$

attention weights

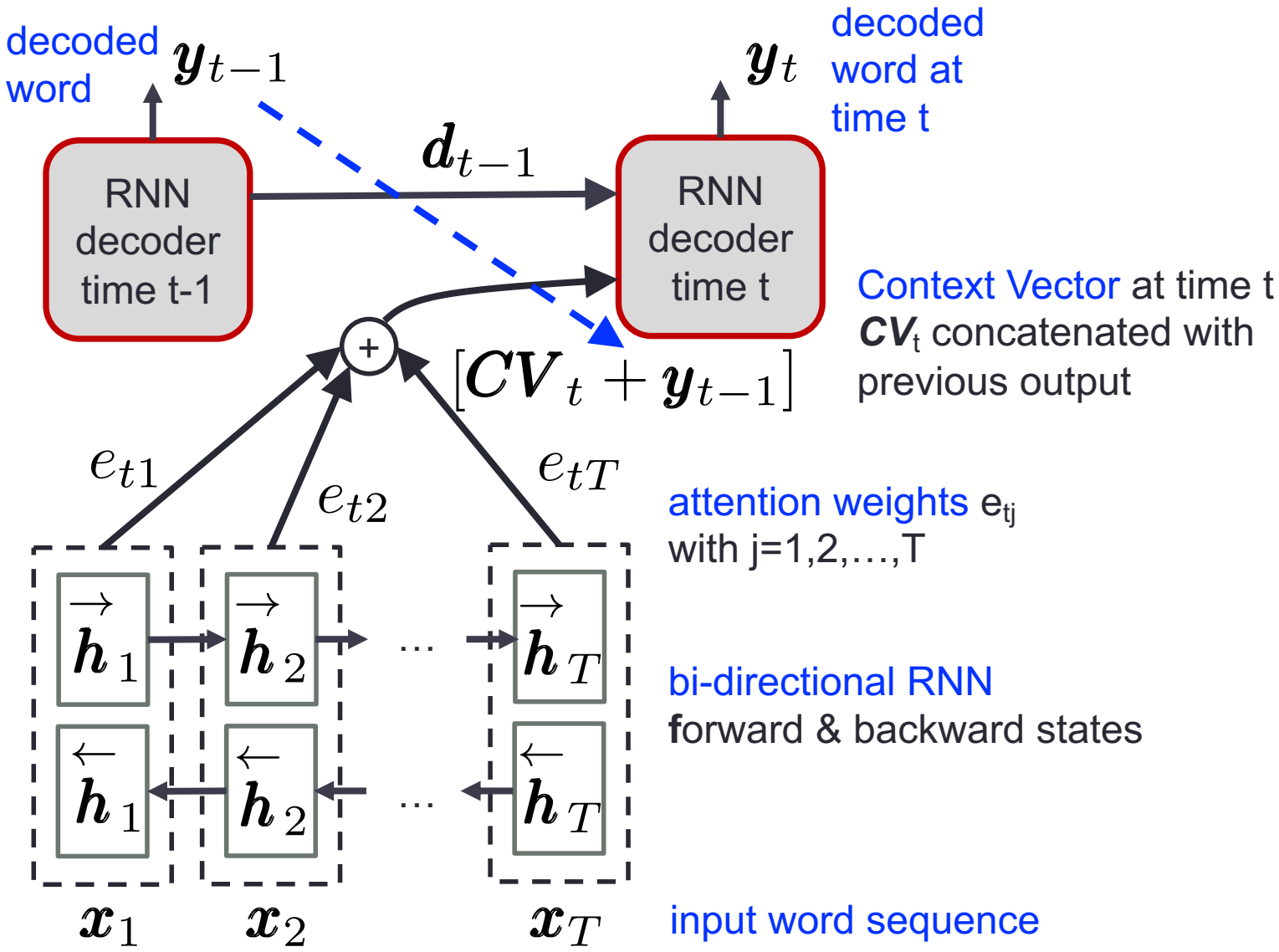


encoder state for token k



decoder internal state from previous step

In general: bi-directional RNN



BLEU scores

BLEU – Bilingual Evaluation Understudy

- Is a score for comparying a candidate translation of a text against one or more reference translations
- It can be used to numerically evaluate the text quality generated by a natural language processing tasks. Main qualities: quick and inexpensive to compute, language independent, correlates highly with human evaluation, has been widely adopted

[Papineni-2002] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, "BLEU: a Method for Automatic Evaluation of Machine Translation," 40-th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, PA, US, 2002.

For Python code & additional details

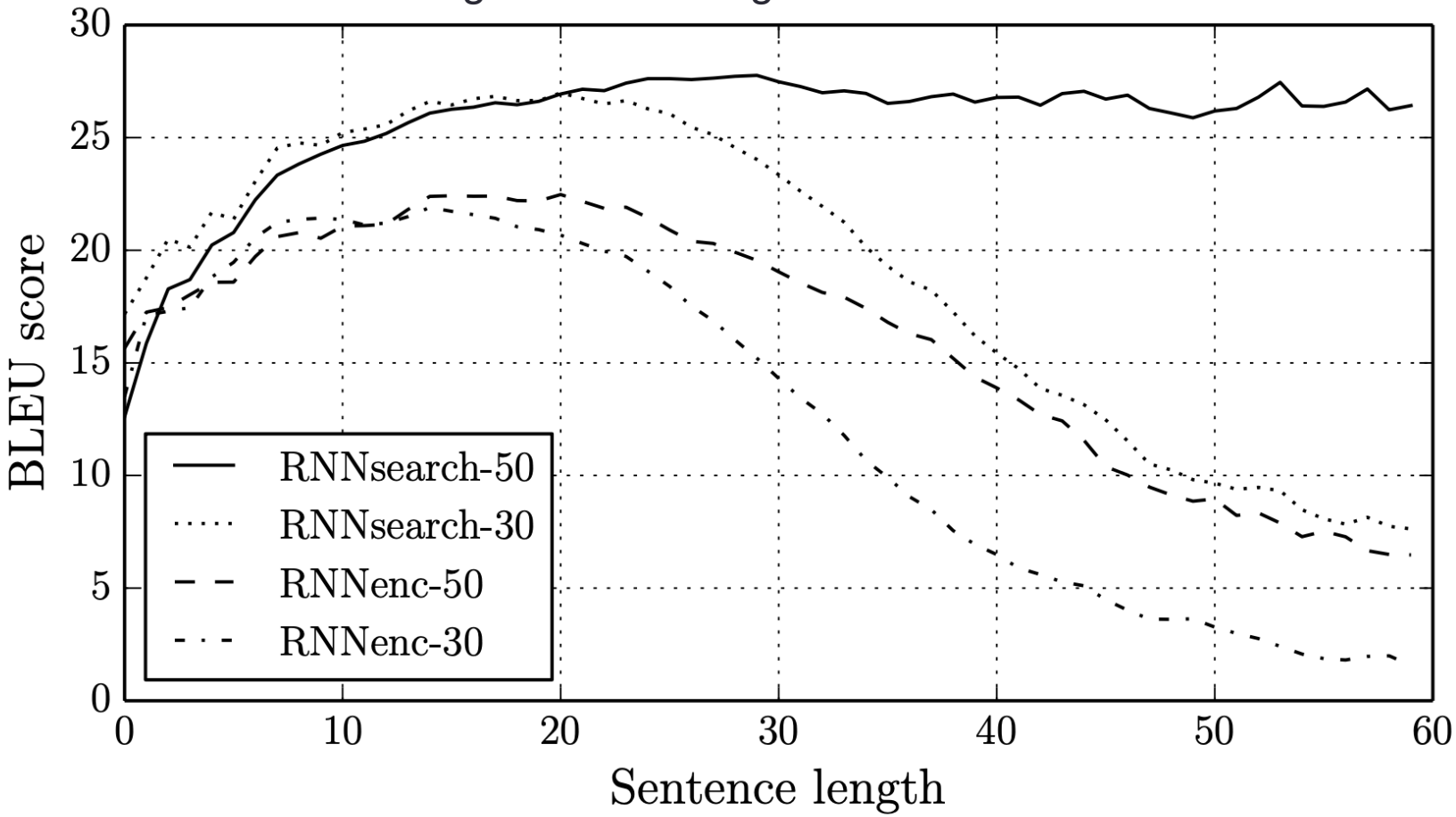
https://machinelearningmastery.com/calculate-bleu-score-for-text-python/

RNN enc-dec architectures

- RNN encoder-decoder (dubbed RNNenc, no attention)
  - 1,000 hidden units
  - Trained with sentences of length up to 30 and 50 words
- RNN encoder-decoder with attention (RNN-search)
  - Encoder: forward + backward RNN each with 1,000 hidden units
  - Trained with sentences of length up to 30 and 50 words
- For both models
  - Training used SGD with mini-batches of 80 sentences each
  - Training lasted approx. 5 days

Results from [Bahdanau15]

WMT ‘14 used for training/testing (348 M words) and
Google translator Eglish → French used as a reference



Translation example (1/2)

EN: ”An admitting privilege is the right of a doctor to admit a patient to a hospital or a medical centre to carry out a diagnosis or a procedure, based on his status as a health care worker at a hospital”

FR(RNN-EncDec-50): “Un privilège d’admission est le droit d’un médecin de reconnaître un patient à l’hôpital ou un centre médical d’un diagnostic ou de prendre un diagnostic en fonction de son état de santé”

- Translation is correct up to medical centre, after that, it replaced [based on his status as a health care worker at a hospital] with [based on his state of health]

→ changing the meaning

Translation example (2/2)

EN: ”An admitting privilege is the right of a doctor to admit a patient to a hospital or a medical centre to carry out a diagnosis or a procedure, based on his status as a health care worker at a hospital”

FR(RNN+attention-50): Un privilège d’admission est le droit d’un médecin d’admettre un patient à un hôpital ou un centre médical pour effectuer un diagnostic ou une procédure, selon son statut de travailleur des soins de santé a` l’hôpital

- Translation is correct and meaning is preserved

Attention matrix

- The attention matrix provides a visual representation of which input tokens are attended to produce each output token
- Shows a sort of “alignement” between input and output words

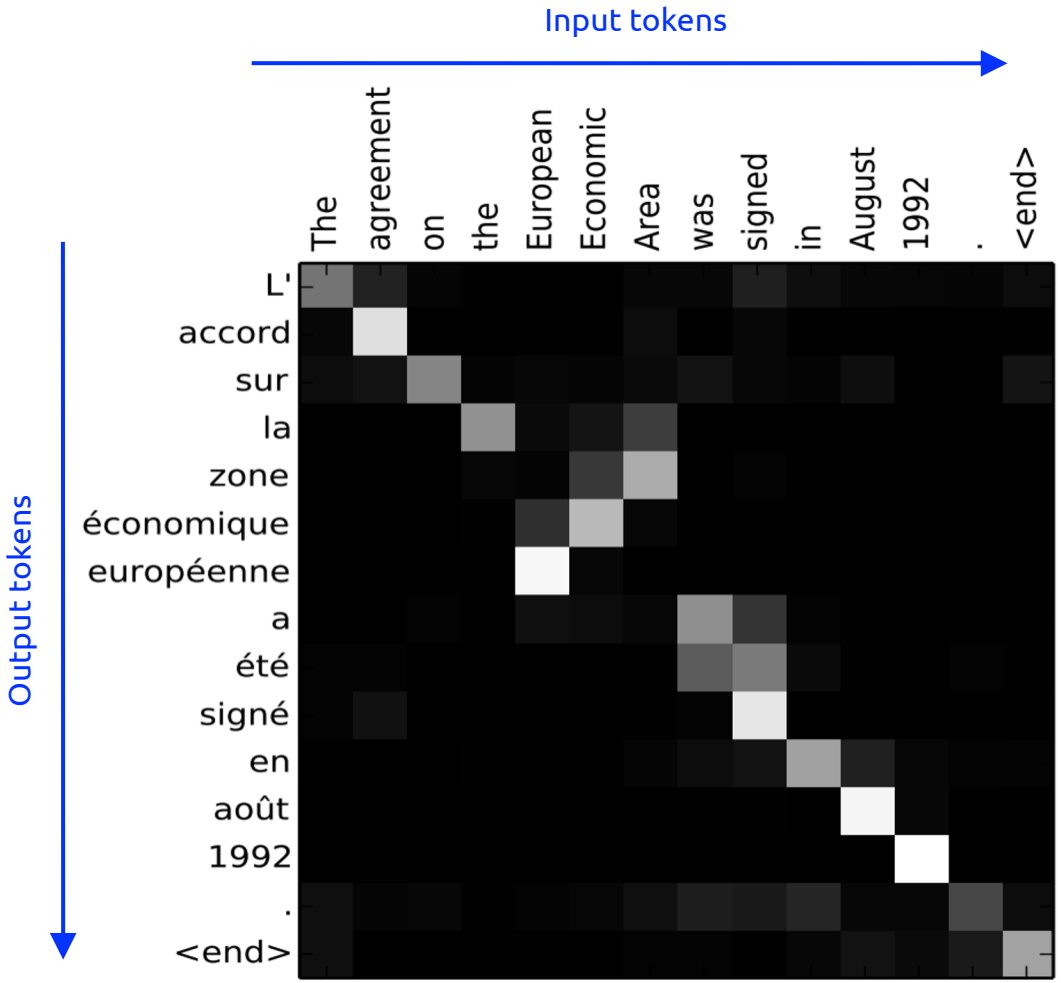


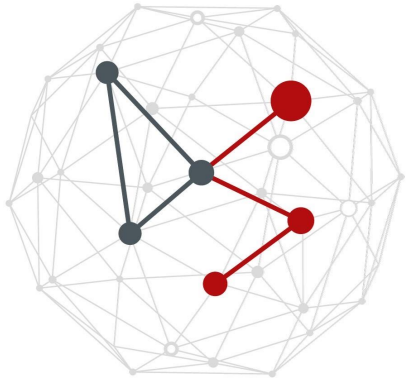
Figure: [Bahdanau2015] D. Bahdanau, K. Cho, Y. Bengio, “Neural machine translation by jointly learning to align and translate”. ICLR 2015.

Recap of benefits brough by attention

- Bypasses the bottleneck problem of plain Seq-2-Seq RNNs
- Reduces RNN problems
  - Long memory losses
  - Vanishing gradients
- Improves performance, especially for long sentences
- Works more similarly to what humans do
  - Look back to the source, rather than remembering it all

Master's degree ICT Internet Multimedia Engineering

VISUAL ATTENTION EXAMPLE



DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

DIPARTIMENTO MATEMATICA

1222 · 2022 800 ANNI

UNIVERSITÀ DEGLI STUDI DI PADOVA

Visual attention

- Thorax disease classification on chest X-ray images
- General methods use global images of the thorax
- This has limitations
  - Generally the disease is evident within localized smaller areas
  - The use of global images may affect the classification performance
    - Excessive & irrelevant noisy regions
    - Poor alignment of diseased portions, that may be located along the borders
- Solution from [Guan2018] Attention Guided CNN (AG-CNN)

[Guan2018] Qingji Guan, Yaping Huang, Zhun Zhong, Zhedong Zheng, Liang Zheng, Yi Yang, “Diagnose like a radiologist: Attention guided convolutional neural network for thorax disease classification,” arXiv preprint arXiv:1801.09927.

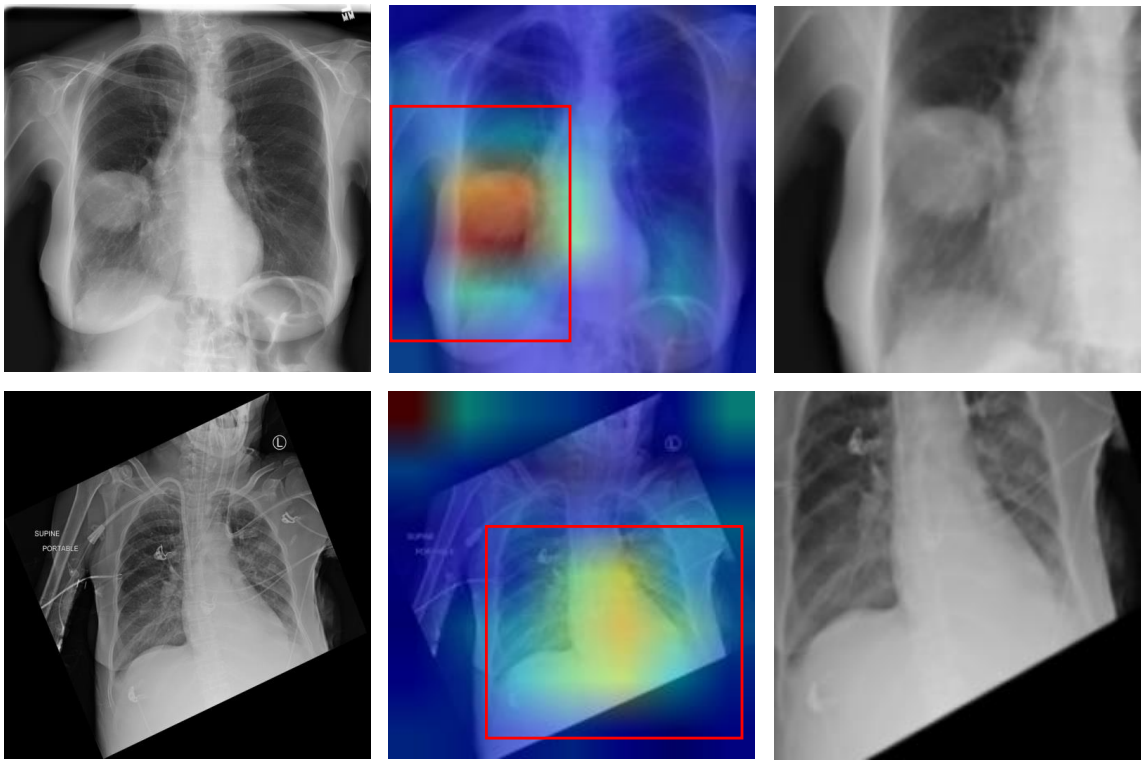
- Code on Github: https://github.com/Ien001/AG-CNN

Visual insight

(a) Original global X-ray image

(b) Heatmap is extracted from original global image

(b) The relevant sub-regions are cropped into local images



(a) original global image (b) heatmap (c) cropped local image

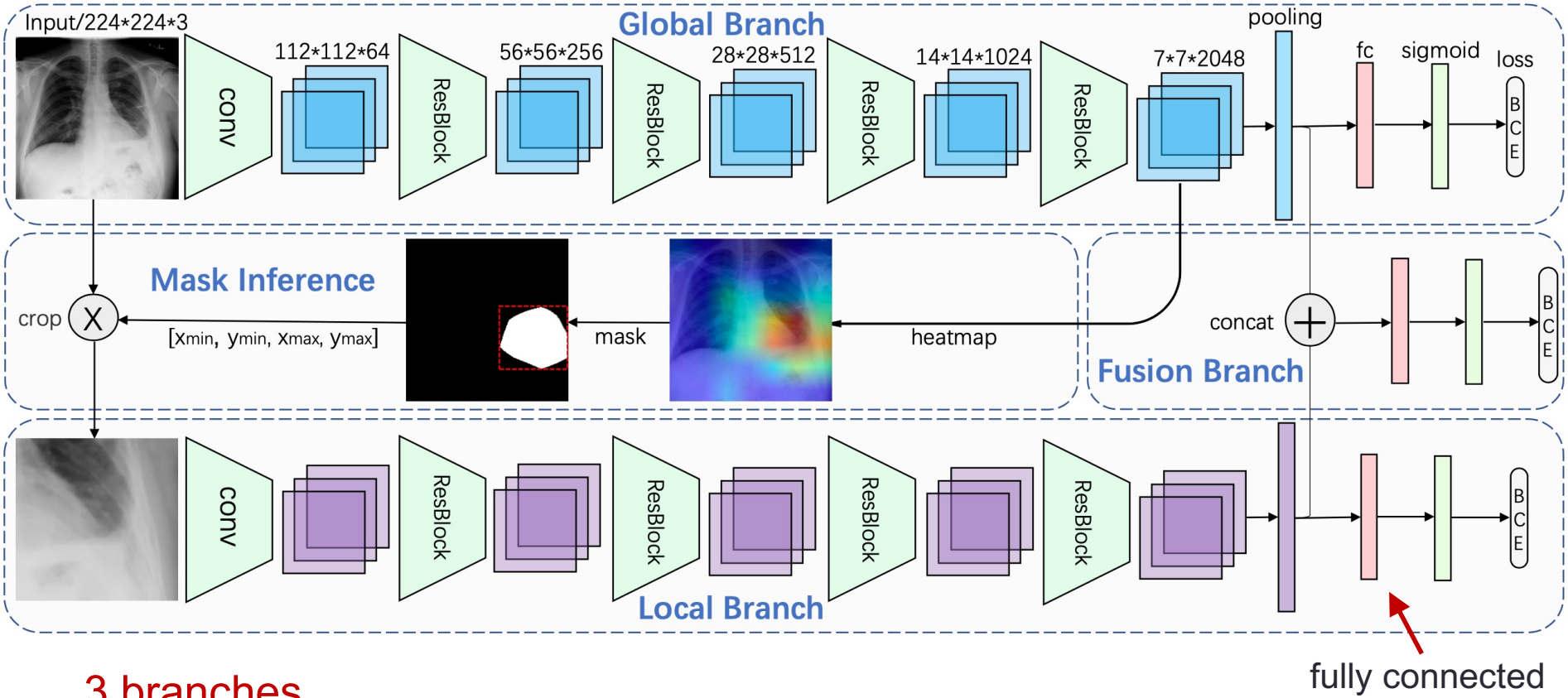
Dataset & labels

- Labeled dataset of X-ray chest images
  - 15 disease types (classes)
- Each image is labeled with a 15-dimensional vector

$$\boldsymbol{\ell} = [\ell_1\, \ell_2\, \cdots\, \ell_{15}]^\top,\ \ell_i \in \{0, 1\}$$

  - each entry: 0 if corresponding pathology is absent and 1 otherwise
- Pathologies (15 classes)
  - atelectasis, cardiomegaly, effusion, infiltration, mass, nodule, pneumonia, pneumothorax, consolidation, edema, emphysema, fibrosis, pleural thickening, hernia
- ChestX-ray14 dataset
  - https://www.rsna.org/en/education/ai-resources-and-training/ai-image-challenge/RSNA-Pneumonia-Detection-Challenge-2018

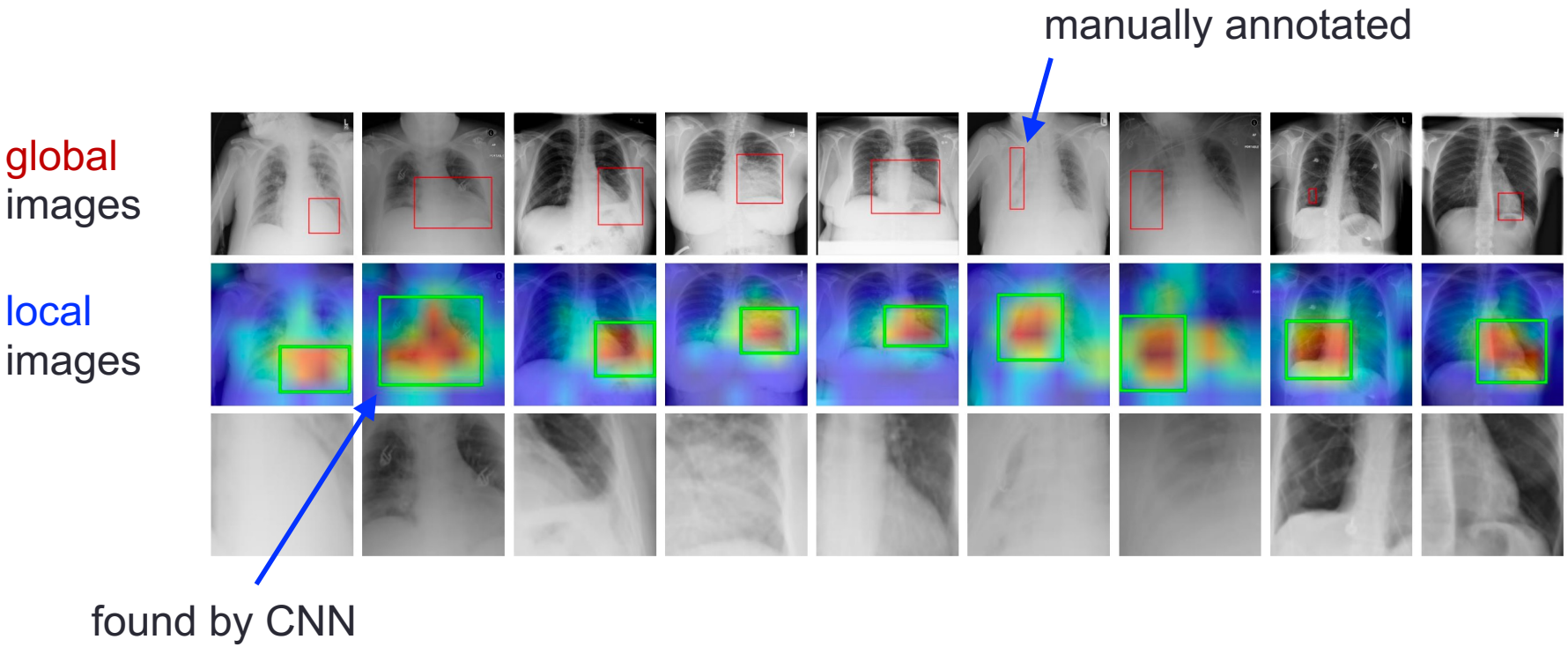
Neural network: based on ResNet



3 branches

- Global branch: learns to attend to a certain area (problem area)
- Local branch: performs classification using the cropped local area
- Fusion branch: uses *global* and *local* features to output the final class

Results of “global branch CNN”



- Global images: are manually annotated (see the red bounding boxes)
- Local images: are found by the “global branch CNN”

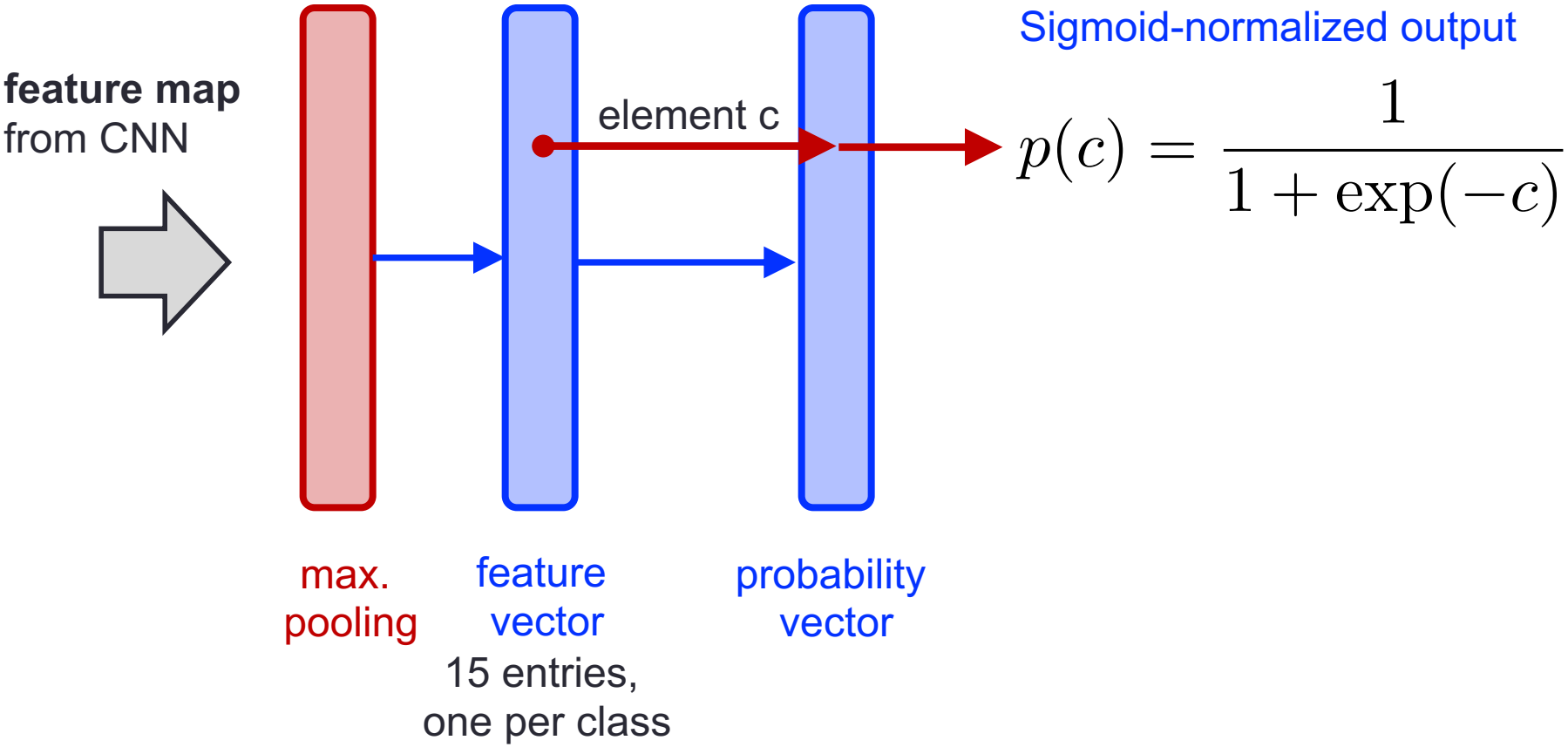
Important: the annotated regions (red boxes on top images) are neither used for learning, nor for testing purposes

Training procedure – step 1

- Global Branch CNN
  - Based on ResNet design
  - Processes the entire image and determines the region of interest (ROI) → to be cropped
  - The network weights are learned in a supervised manner using *backpropagation* and the *disease labels*
- When learning is complete
  - The feature map right before the final pooling layer is extracted and used to crop the ROI (according to the intensities of the pixels in such feature map)

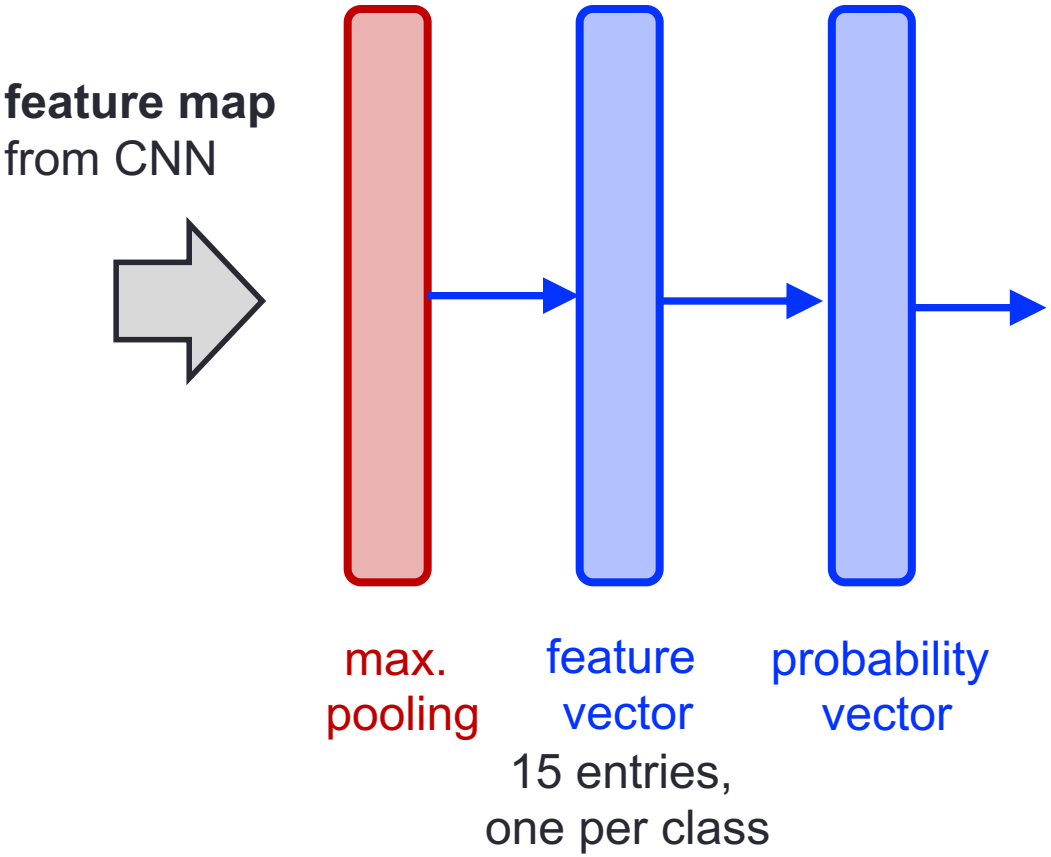
Output of global branch

- Pooling → feature vector (15 entries = num. of classes)



Output of global branch

- Pooling → feature vector (15 entries = num. of classes)



Binary Cross-Entropy (BCE) loss

$$L = \sum_{c=1}^{15} \Big[\ell_c \log(p(c)) + (1-\ell_c) \log(1-p(c)) \Big]$$

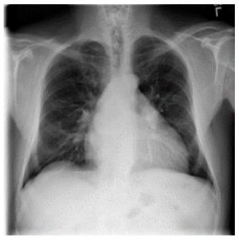
Training procedure – step 2

- Local Branch CNN
  - Is also based on a ResNet design
  - Same max. pooling, feature vector and BCE loss as in step 1
  - It takes as input the cropped ROI
  - Its weights are trained in a supervised manner (backprop, after step 1 and **NOT** jointly with it) using again the labels from the dataset
  - Uses entirely independent weights wrt the Global Branch CNN
  - Trained **after training** the Global Branch CNN

Training procedure – step 3

- Fusion branch
  - Is a fully connected layer
  - Same max. pooling, feature vector and BCE loss as in steps 1 & 2
  - Again, the weights are independent of those of the other branches
  - Uses the features extracted by the global and the local branches
    - Fusing features to obtain better classification results
    - Funsion layer: concatenates the features (**pooling output**) from the *global* and *local* branches, then a fully connected layer is applied to this concatenated vector → obtaining the 15-dimensional output vector. This last vector is passed (element-wise) through a sigmoid nonlinearity to obtain the 15 output probabilities for the considered diseases
  - Trained **after training** Global and Local Branches

Example classification results

Images							
							
Classification results	Effusion 0.770 Atelectasis 0.732 Infiltration 0.352 Consolidation 0.205 No Finding 0.127 Pneumonia 0.017 Mass 0.014 Nodule 0.014 Edema 0.014 Cardiomegaly 0.013	Emphysema 0.831 Pneumothorax 0.754 Effusion 0.106 Infiltration 0.101 Mass 0.087 No Finding 0.082 Atelectasis 0.075 Nodule 0.030 PT 0.027 Consolidation 0.024	Effusion 0.902 Atelectasis 0.727 Consolidation 0.207 Infiltration 0.193 No Finding 0.074 Pneumothorax 0.058 Emphysema 0.017 PT 0.016 Mass 0.012 Cardiomegaly 0.010	Effusion 0.820 Mass 0.780 Atelectasis 0.201 Infiltration 0.130 Nodule 0.115 No Finding 0.065 Consolidation 0.051 PT 0.046 Pneumothorax 0.028 Edema 0.011	Cardiomegaly 0.752 No Finding 0.304 Effusion 0.133 Infiltration 0.108 Atelectasis 0.068 Hernia 0.054 Nodule 0.048 Fibrosis 0.037 PT 0.035 Mass 0.022	Emphysema 0.854 Pneumothorax 0.810 Atelectasis 0.264 Effusion 0.139 No Finding 0.138 Infiltration 0.085 PT 0.054 Nodule 0.034 Mass 0.018 Fibrosis 0.016	Effusion 0.915 Cardiomegaly 0.807 Infiltration 0.415 Edema 0.144 Atelectasis 0.089 PT 0.078 Consolidation 0.052 Pneumonia 0.037 Mass 0.029 Nodule 0.029

True classes are reported in blue color

- The true labels are always within the labels with the highest probability

Experimental results

Method	CNN	Atel	Card	Effu	Infi	Mass	Nodu	Pne1	Pne2	Cons	Edem	Emph	Fibr	PT	Hern	Mean
Wang *et al.* [9]	R-50	0.716	0.807	0.784	0.609	0.706	0.671	0.633	0.806	0.708	0.835	0.815	0.769	0.708	0.767	0.738
Yao *et al.* [19]	D-/	0.772	0.904	0.859	0.695	0.792	0.717	0.713	0.841	0.788	0.882	0.829	0.767	0.765	0.914	0.803
Rajpurkar *et al.* [8]*	D-121	0.821	0.905	0.883	0.720	0.862	0.777	0.763	0.893	0.794	0.893	0.926	0.804	0.814	0.939	0.842
Kumar *et al.* [7]*	D-161	0.762	0.913	0.864	0.692	0.750	0.666	0.715	0.859	0.784	0.888	0.898	0.756	0.774	0.802	0.795
Global branch (baseline)	R-50	0.818	0.904	0.881	0.728	0.863	0.780	0.783	0.897	0.807	0.892	0.918	0.815	0.800	0.889	0.841
Local branch	R-50	0.798	0.881	0.862	0.707	0.826	0.736	0.716	0.872	0.805	0.874	0.898	0.808	0.770	0.887	0.817
AG-CNN	R-50	0.844	0.937	0.904	0.753	0.893	0.827	0.776	0.919	0.842	0.919	0.941	0.857	0.836	0.903	0.868
Global branch (baseline)	D-121	0.832	0.906	0.887	0.717	0.870	0.791	0.732	0.891	0.808	0.905	0.912	0.823	0.802	0.883	0.840
Local branch	D-121	0.797	0.865	0.851	0.704	0.829	0.733	0.710	0.850	0.802	0.882	0.874	0.801	0.769	0.872	0.810
AG-CNN	D-121	0.853	0.939	0.903	0.754	0.902	0.828	0.774	0.921	0.842	0.924	0.932	0.864	0.837	0.921	0.871

* We compute the AUC of each class and the average AUC across the 14 diseases. $*$ denotes that a different train/test split is used: 80% for training and the rest 20% for testing. All the Other methods split the dataset with 70% for training, 10% for validation and 20% for testing. Each pathology is denoted with its first four characteristics, *e.g.*, Atelectasis with *Atel*. Pneumonia and Pneumothorax are denoted as *Pneu1* and *Pneu2*, respectively. PT represents Pleural Thickening. We report the performance with parameter $\tau = 0.7$. ResNet-50 (R-50) and Desnet-121 (D-121) are used as backbones in our approach. For each column, the best and second best results are highlighted in red and blue, respectively.

Other interesting works

[Liu2019] J. Liu, G. Zhao, Y. Fei, M. Zhang, Y. Wang, Y. Yu, “Align, Attend and Locate: Chest X-ray Diagnosis via Contrast Induced Attention Network with Limited Supervision,” IEEE/CVF International Conference on Computer Vision (**ICCV**), Seoul, Korea, October 2019.

[Vaswani2017] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.-N. Gomez, Ł. Kaiser, I. Polosukhin, “Attention is all you need,” Conference on Neural Information Processing Systems (NIPS), Long Beach, CA, USA, December 2017.

Master's degree ICT Internet Multimedia Engineering

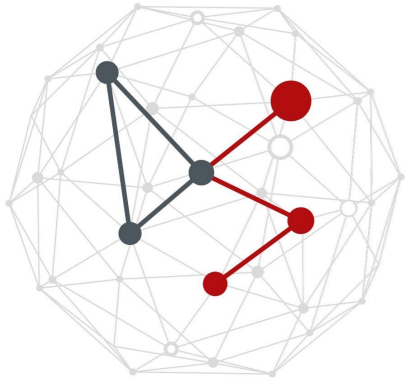
ADVANCED ARCHITECTURES: SEQ2SEQ AND VISUAL ATTENTION

Michele Rossi

michele.rossi@unipd.it

Dept. of Information Engineering

University of Padova, IT



DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

DIPARTIMENTO MATEMATICA

800 ANNI 1222·2022

UNIVERSITÀ DEGLI STUDI DI PADOVA