DIPARTIMENTO DI SCIENZE E TECNOLOGIE

CORSO DI LAUREA IN INFORMATICA APPLICATA
(Machine Learning – Big Data)

Neural Machine Translation

Student :
Ciro Panariello
Matricola 0120000209

Academic Year 2021/2022

Summary

- ❑ Introduction to NLP
- ❑ What is NMT
- ❑ Application I : intro
- ❑ Application II : all we need
- ❑ Application III : code
- ❑ Application IV : Attention Mechanism
- ❑ Application V : possible improvements

Introduction to NLP

- **Natural Language Processing**, or **NLP**, is a subfield of Artificial Intelligence research that is focused on developing models and points of interaction between humans and computers based on natural language. This includes text, but also speech-based systems.

- **Why NLP is difficult**

❑ There is an infinite number of **different ways to arrange words in a sentence**.

❑ Words can have several meanings and contextual **information is necessary to correctly interpret sentences**. Every language is more or less unique and ambiguous.

❑ SYNTACTIC & SEMANTIC ANALYSIS : Syntax is the grammatical structure of the text, whereas semantics is the meaning being conveyed. A sentence that is syntactically correct, however, is not always semantically correct.

Introduction to NLP

NLP Applications

- ❑ **Chatbox** is used for automatic question answering, designed to understand natural language and deliver an appropriate response through natural language generation.
- ❑ **Sentiment analysis** is able to recognize subtle nuances in emotions and opinions – and determine how positive or negative they are.
- ❑ **Text Classification** involves automatically understanding, processing, and categorizing unstructured text.
- ❑ **Text Extraction** or information extraction, automatically detects specific information in a text, such as names, companies, places, and more. This is also known as named entity recognition.
- ❑ **Text Summarization** : it summarizes text, by extracting the most important information.
- ❑ **Speech Recognition** is used for transform spoken language into a machine-readable format.
- ❑ **Machine Translation** : automatic translation of text from one language to another.

What is NMT

- Machine translation is the task of automatically converting source text in one language to text in another language.

- Machine translation is perhaps one of the most challenging artificial intelligence tasks given the fluidity of human language. Classically, rule-based systems were used for this task, which were replaced in the 1990s with statistical methods. More recently, deep neural network models achieve state-of-the-art results in a field that is aptly named **Neural Machine Translation**.

Application I : intro

- Dataset source : http://www.manythings.org/anki/

- It consists of 300k translated Italian/English sentences.

- Italian vocabulary has 26179 words compared to ~600000 in real world.
- English vocabulary has 12849 words compared to ~1000000 in real world.

- The maximum length of an Italian sentence is 12 words.
- The maximum length of an English sentence is 13 words.

Application II : all we need

Prerequisites

- Recurrent Neural Network
- LSTM

In NLP we use

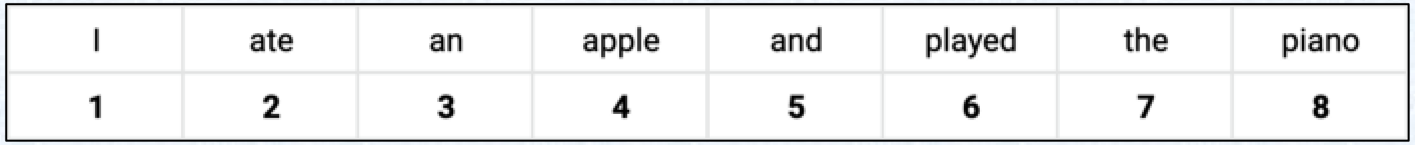
- Word Embedding
- Encoder/Decoder model

Word Embedding

- One **hot encoding** is a simple representation for the input data.

Example :

- The dataset consists of 8 words and we assign a unique code to each word.

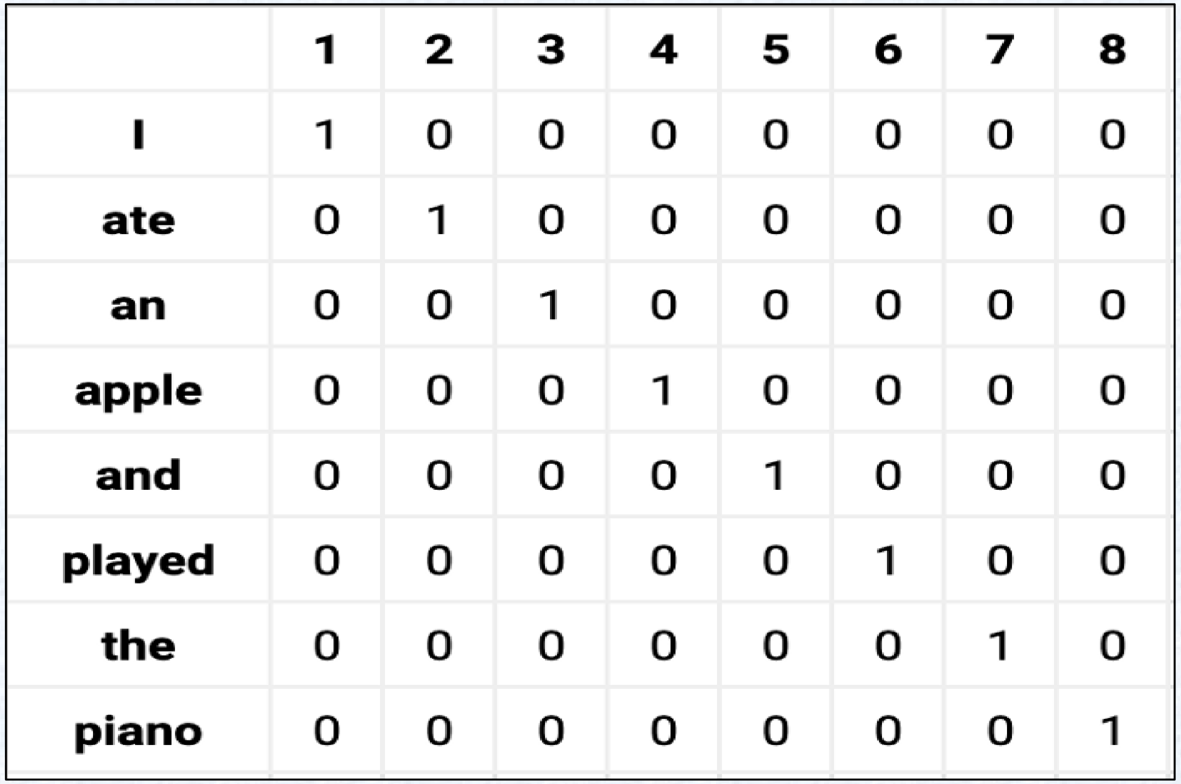


I	ate	an	apple	and	played	the	piano
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**

- The word “I” is at position 1, so its one-hot vector representation would be [1, 0, 0, 0, 0, 0, 0, 0]. Similarly, the word “ate” is at position 2, so its one-hot vector would be [0, 1, 0, 0, 0, 0, 0, 0].

Word Embedding

- The one-hot embedding matrix for the example text would look like this:



	1	2	3	4	5	6	7	8
I	1	0	0	0	0	0	0	0
ate	0	1	0	0	0	0	0	0
an	0	0	1	0	0	0	0	0
apple	0	0	0	1	0	0	0	0
and	0	0	0	0	1	0	0	0
played	0	0	0	0	0	1	0	0
the	0	0	0	0	0	0	1	0
piano	0	0	0	0	0	0	0	1

We have 2 major problems :

1. each pattern will have a number of features equal to the number of words in the dictionary → not computationally efficient.
2. we lose the semantic relationship that exists between the words of the sentence.

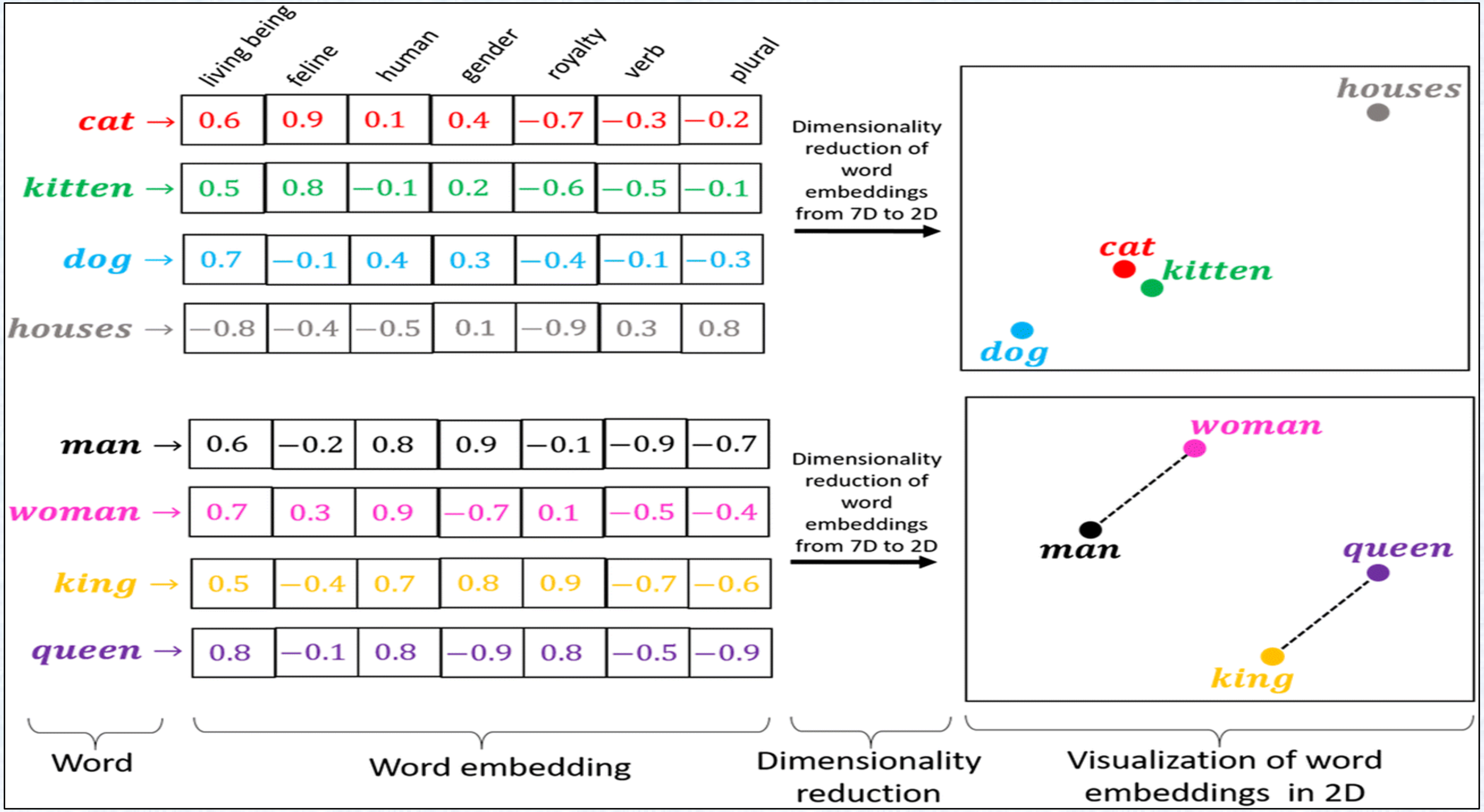
Word Embedding

- The purpose of word embedding is that **semantic similar words have a shorter distance** (euclidean, cosine, etc.) than words that have no semantic relationship. For example, words like "mom" and "dad" should be closer than the words "mom" and "ketchup" or "dad" and "butter".

- The goal is to have a **word vector** for each word of the dataset in order to properly identify its relationship with all the other words.

- We have several neural network based approaches to train word embedding, including CBOW and Skip-Gram models.

Word Embedding



	living being	feline	human	gender	royalty	verb	plural
cat →	0.6	0.9	0.1	0.4	−0.7	−0.3	−0.2
kitten →	0.5	0.8	−0.1	0.2	−0.6	−0.5	−0.1
dog →	0.7	−0.1	0.4	0.3	−0.4	−0.1	−0.3
houses →	−0.8	−0.4	−0.5	0.1	−0.9	0.3	0.8
man →	0.6	−0.2	0.8	0.9	−0.1	−0.9	−0.7
woman →	0.7	0.3	0.9	−0.7	0.1	−0.5	−0.4
king →	0.5	−0.4	0.7	0.8	0.9	−0.7	−0.6
queen →	0.8	−0.1	0.8	−0.9	0.8	−0.5	−0.9

Dimensionality reduction of word embeddings from 7D to 2D
houses
cat
kitten
dog
Dimensionality reduction of word embeddings from 7D to 2D
woman
man
queen
king
Word
Word embedding
Dimensionality reduction
Visualization of word embeddings in 2D

Encoder/Decoder

- The **encoder/decoder model** is a way to use recurrent neural networks for sequence-sequence prediction problems.

- The approach involves two recurrent neural networks, one to **encode the input sequence**, called the encoder, and a second to **decode the input sequence encoded** in the output sequence, called the decoder.

- The encoder/decoder architecture is widely used in NMT → in general, widely used when dealing with sequential data in which the input and output can have a different sequence length; it is precisely the case of the NMT where we have that the input (sentence to be translated) will most likely have a different length from the output (translated sentence).

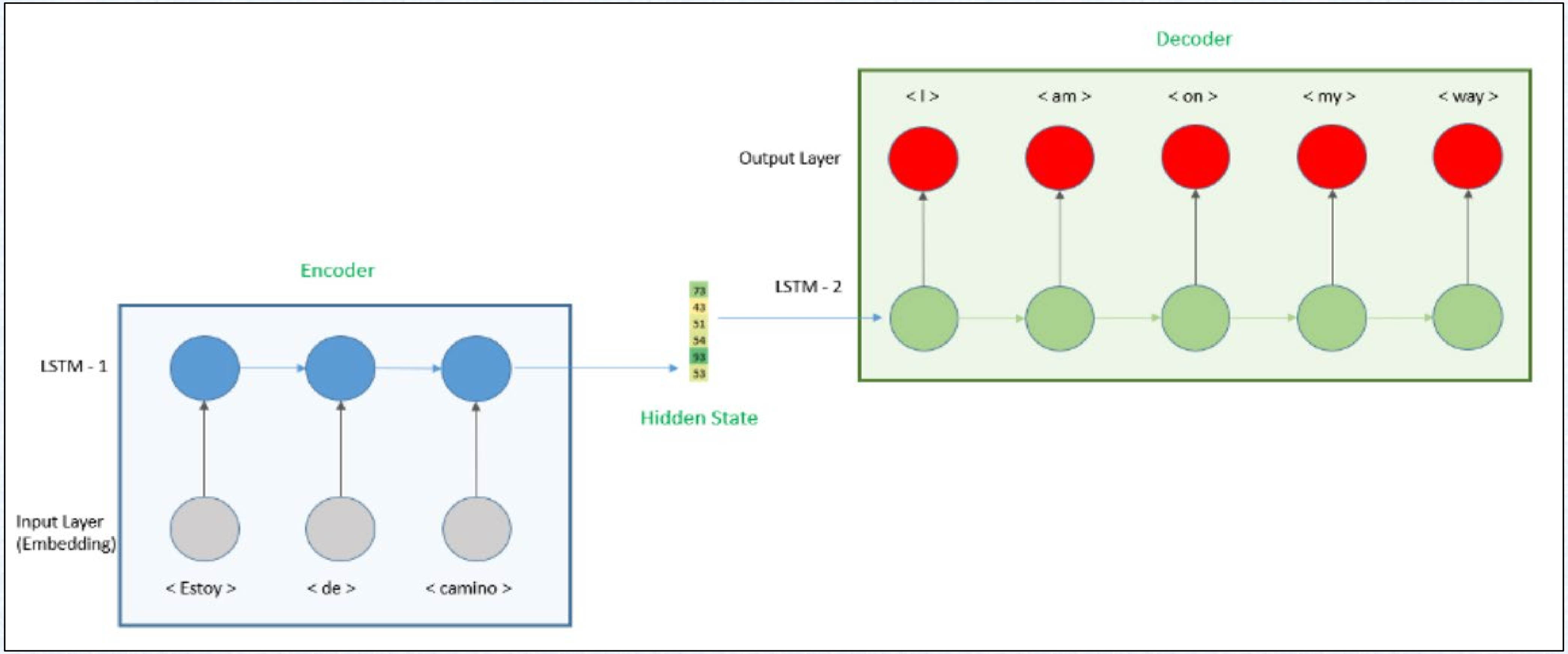
Encoder/Decoder

There are two ways to implement an encoder/decoder model :

1. Using repeat encoded vector
2. Using Teacher forcing

Encoder/Decoder

- Using **Repeat Encoded Vector**



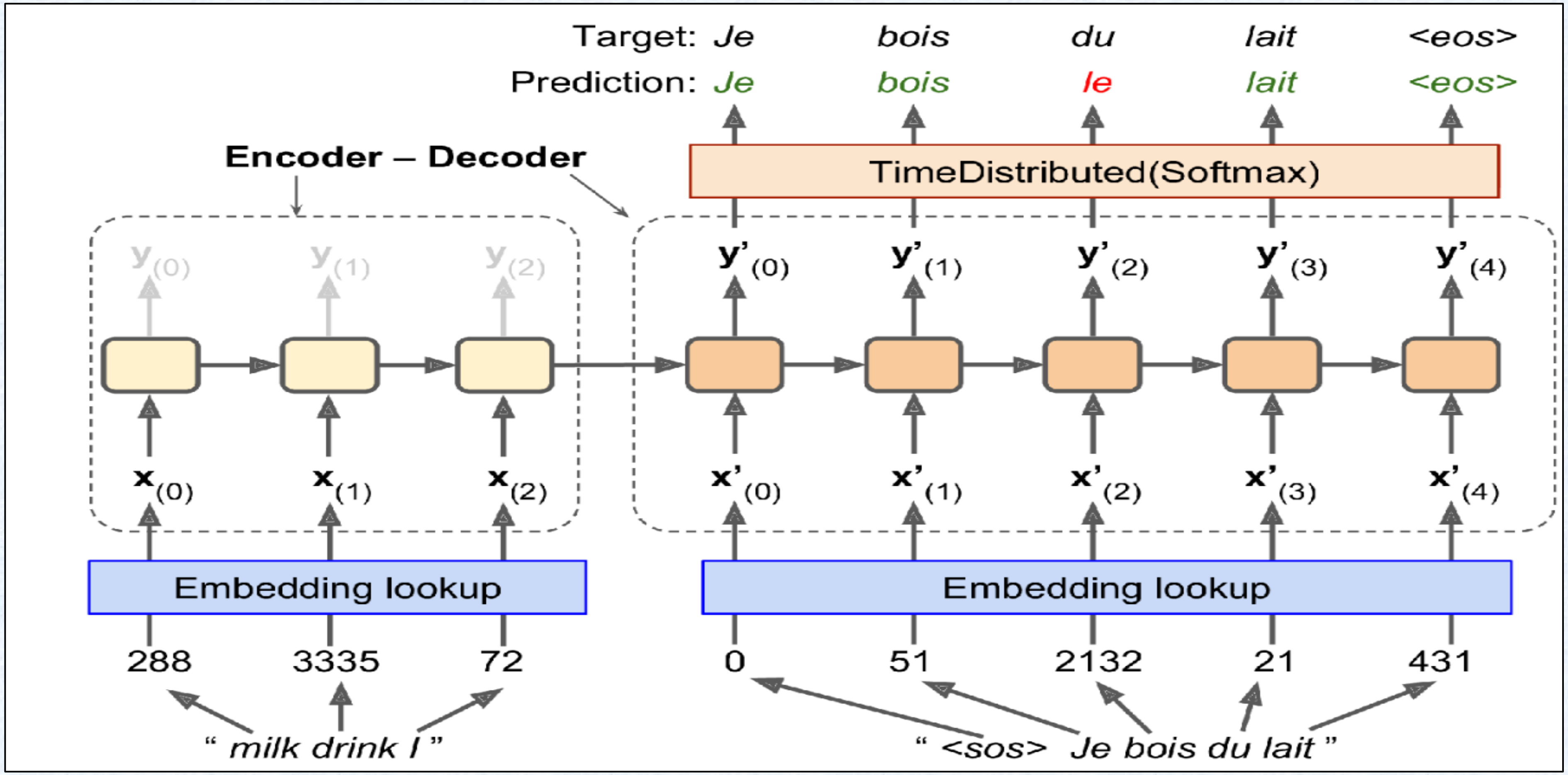
Encoder/Decoder

- Using **Teacher forcing**:

The encoder phase remains the same, only this time we have a slightly different method to bind the encoder output to the decoder and we need to distinguish the training phase from the testing phase.

Encoder/Decoder

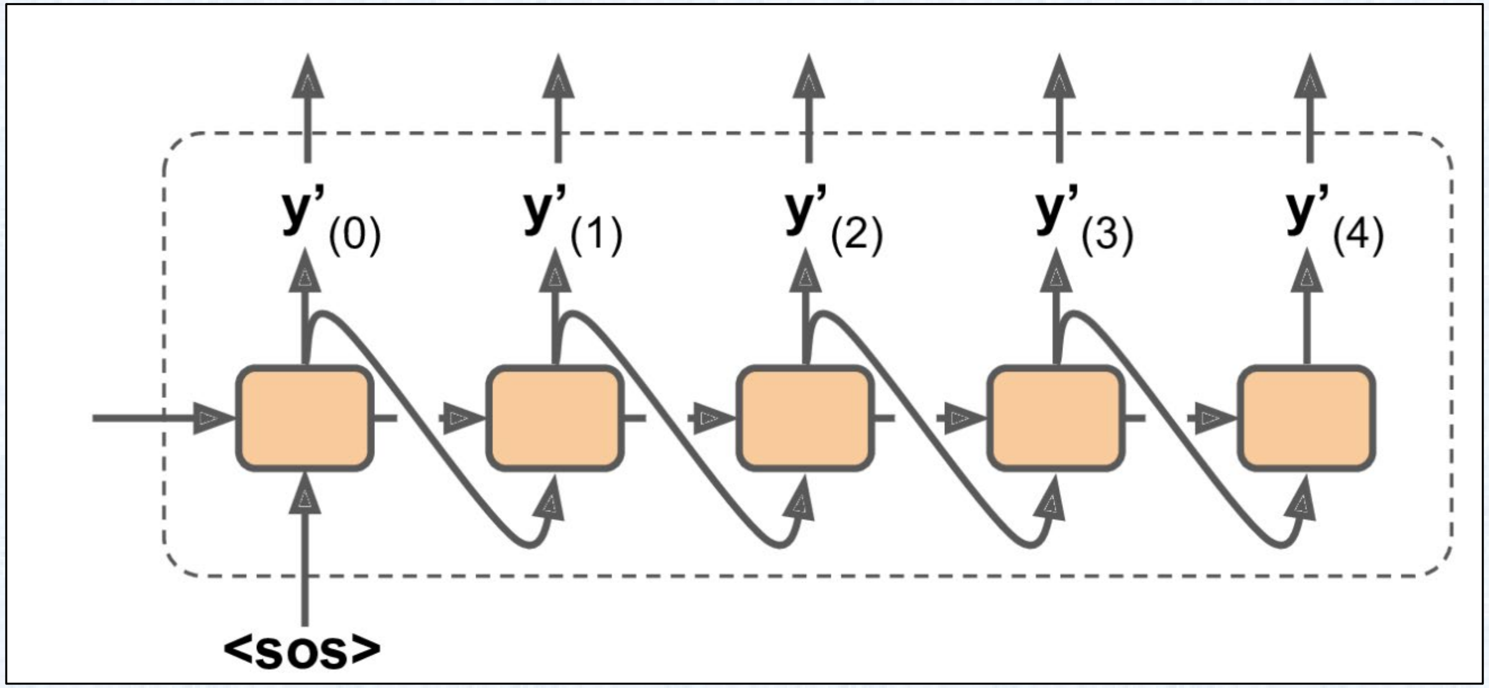
- Using **Teacher forcing** → Training phase



Target: Je bois du lait <eos>
Prediction: Je bois le lait <eos>
TimeDistributed(Softmax)
Encoder – Decoder
$y_{(0)}$ $y_{(1)}$ $y_{(2)}$
$y'_{(0)}$ $y'_{(1)}$ $y'_{(2)}$ $y'_{(3)}$ $y'_{(4)}$
$x_{(0)}$ $x_{(1)}$ $x_{(2)}$
$x'_{(0)}$ $x'_{(1)}$ $x'_{(2)}$ $x'_{(3)}$ $x'_{(4)}$
Embedding lookup
Embedding lookup
288 3335 72
0 51 2132 21 431
" milk drink I "
" <sos> Je bois du lait "

Encoder/Decoder

- Using **Teacher forcing** → Inference phase



Application III : code

- https://github.com/Dantekk/Neural-Machine-Translation-with-attention-mechanism

- There are two versions of the project :

1. OOP
2. Jupyter notebook : https://github.com/Dantekk/Neural-Machine-Translation-with-attention-mechanism/blob/main/Neural_Machine_Translation.ipynb

Application IV : Attention Mechanism

- The **attention mechanism** has changed the way we work with deep learning algorithms.

- Fields such as NLP and even Computer Vision have been revolutionized by the attention mechanism.

- It spawned the rise of so many recent breakthroughs in NLP, including the Transformer architecture and Google's BERT.

Application IV : Attention Mechanism

What is Attenion?

- In psychology, attention **is the cognitive process of selectively focusing on one or a few things while ignoring others**.

- You are looking at a group photo of your first school. Typically, there will be a group of children. Now, if someone asks the question, “How many people are there?”

- How will you answer?

Application IV : Attention Mechanism

What is Attenion?

- Just by counting heads, right? There is no need to consider other things in the photo.
- This is precisely the mechanism of attention → concentrate on the parts that we consider important and exclude everything else.

Application IV : Attention Mechanism

Encoder/Decoder model problems :

1. The LSTM output of the last step of the encoder is used to bind the encoder to the decoder → this is called the **context vector** and we can consider it as a summary of the encoder. All the other output steps of the encoder are ignored → we do not exploit them in any way in the decoder.
2. If the encoder makes a wrong summary, the translation will also be wrong. And in fact it has been observed that the encoder creates a bad summary when trying to understand longer sentences. It is called the **long-range dependence problem** of RNN/LSTM.
3. There is no way to give some input words more importance than others when translating the sentence.

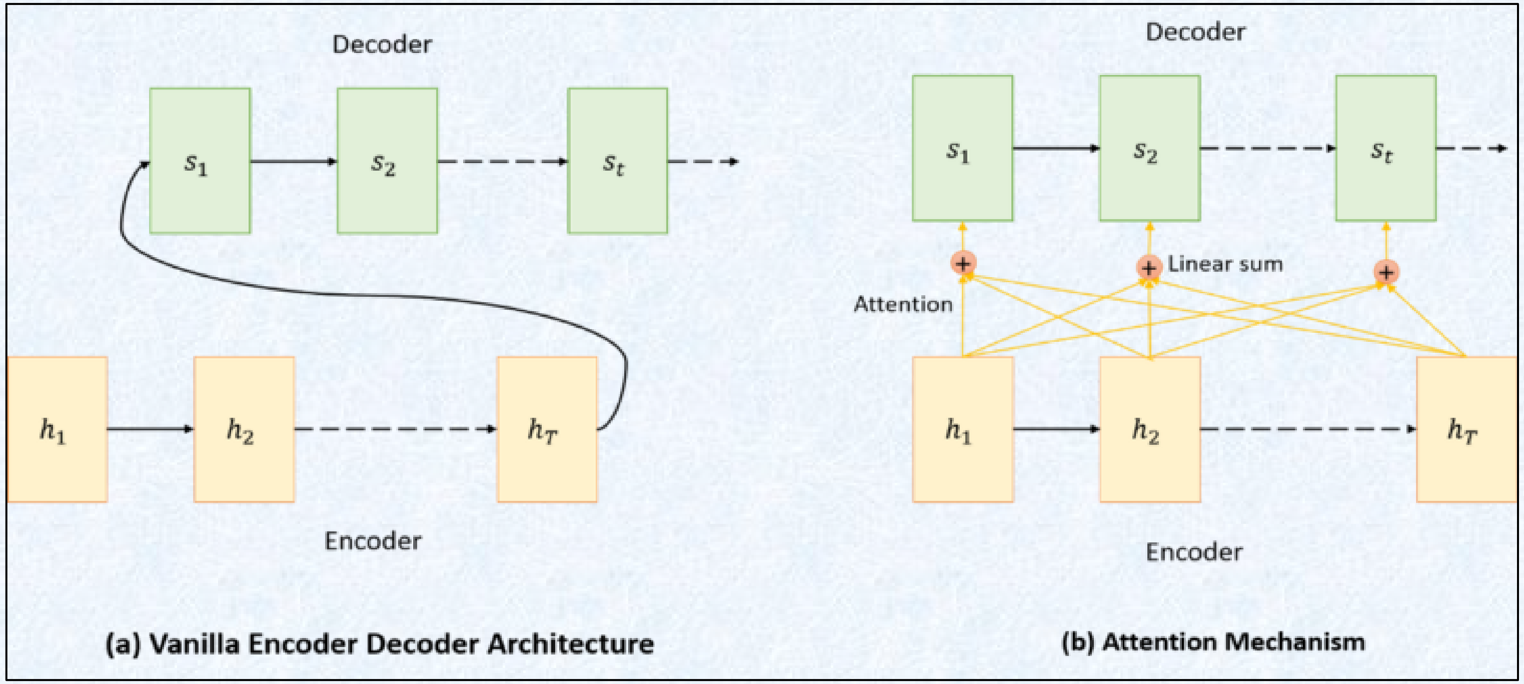
Basically, starting with a context vector we are asking to get too much to the decoder.

Application IV : Attention Mechanism

- The **attention mechanism** was introduced in 2015 in this article by **Bahdanau** (KyungHyun Cho, Yoshua Bengio).
- https://arxiv.org/pdf/1409.0473.pdf
- It was the first time that the concept of attention for neural networks was approached, for this reason **the author used it in the paper with great caution**.

Application IV : Attention Mechanism

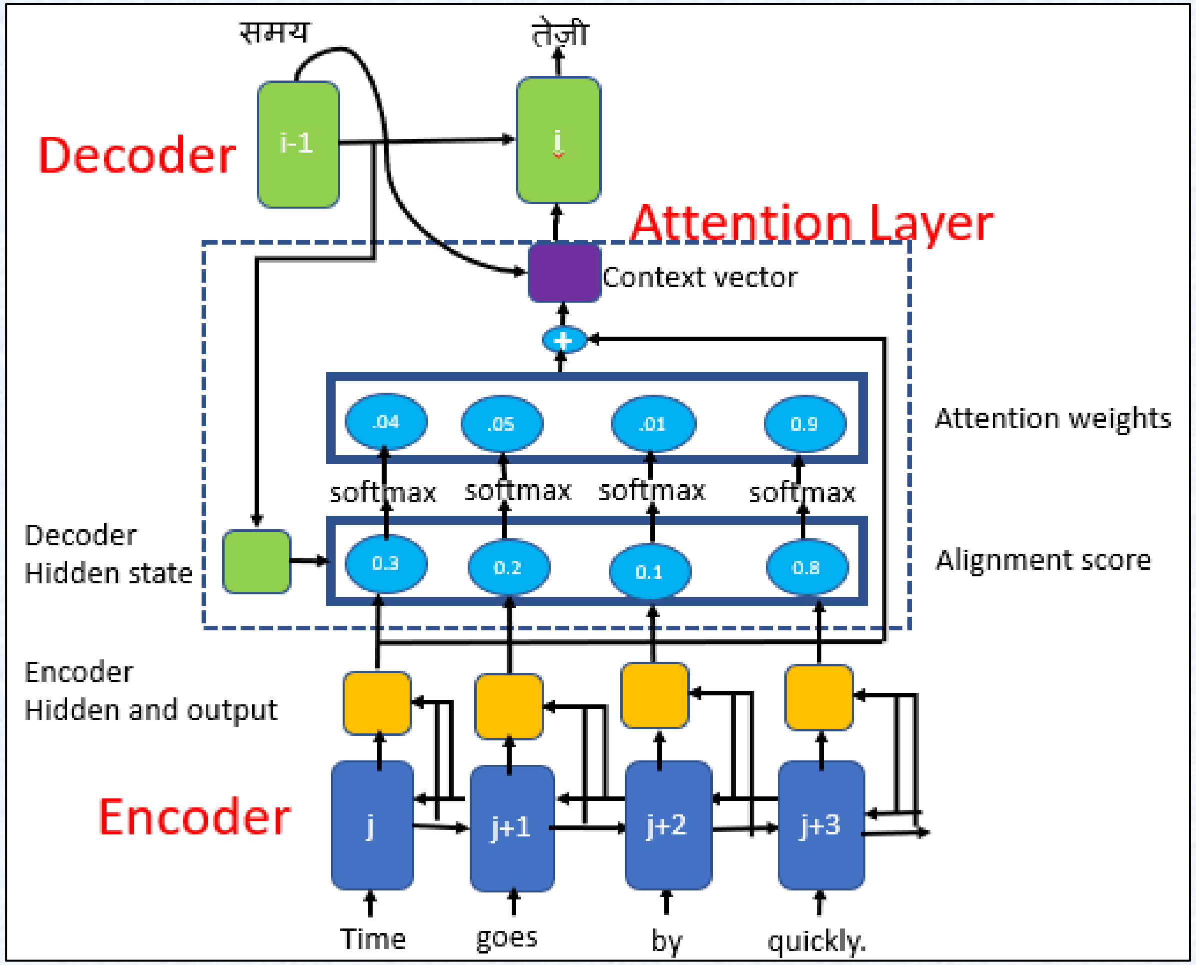
With the attention mechanism, **the decoder at each step uses all the hidden states of the entire encoder sequence** (and not just the last step of the encoder) to make predictions, unlike the vanilla encoder/decoder approach.



(a) Vanilla Encoder Decoder Architecture

(b) Attention Mechanism

Application IV : Attention Mechanism



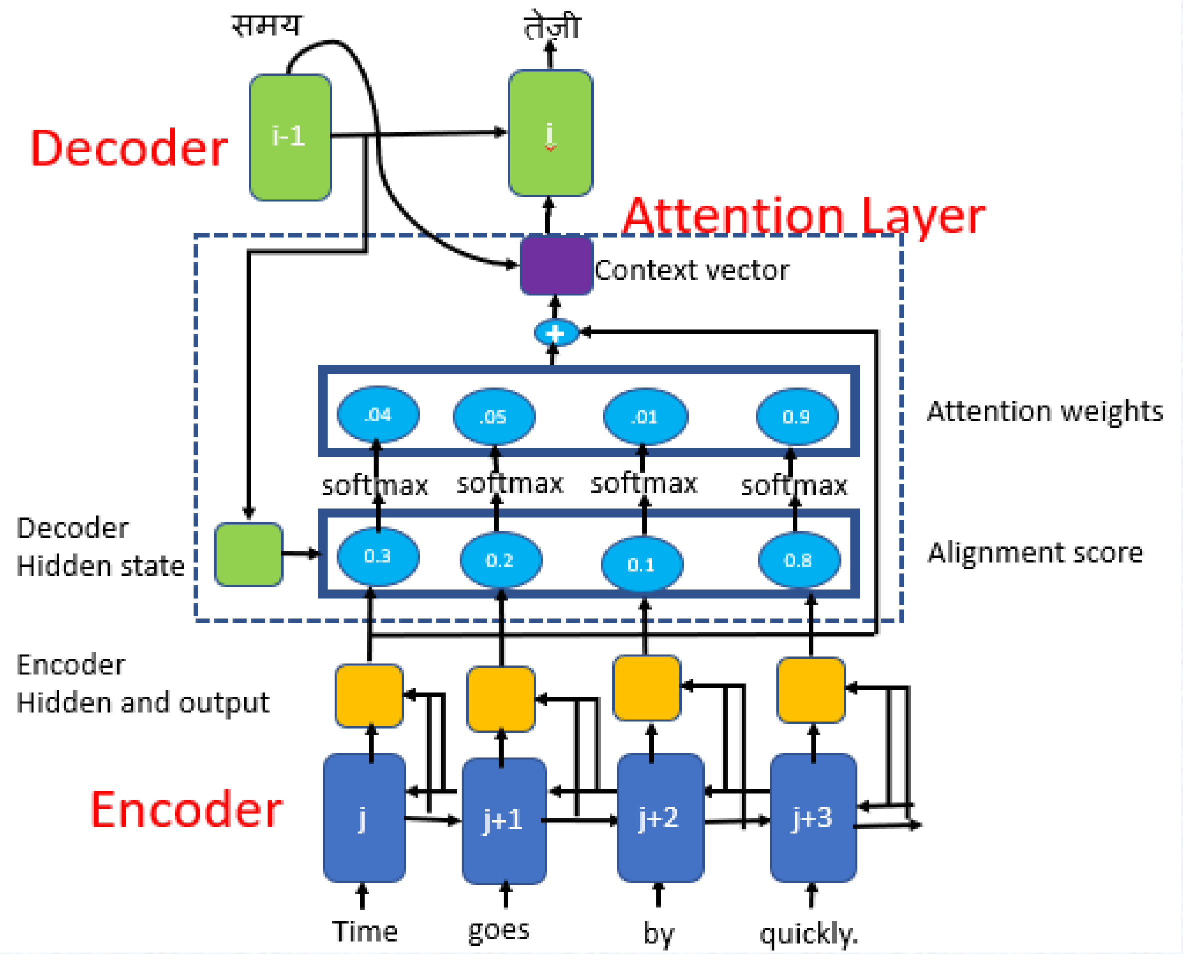
A neural network is used to calculate the vector of the **alignment scores** :

$$e_{ij.} = a(s_{i-1}, h_j) \qquad \text{Alignment Score}$$

Let's apply a softmax function to the alignment scores to get **attention weights**:

$$\alpha_{i_j} = {\exp(e_{ij})}\Big/{\sum_{k=1}^{Tx} \exp(e_{ik})} \qquad \text{Attention weight}$$

Application IV : Attention Mechanism



❑ **alignment scores** :

$$e_{ij.} = a(s_{i-1}, h_j) \qquad \text{Alignment Score}$$

❑ **attention weights**:

$$\alpha_{i_j} = {}^{\exp(e_{ij})}\Big/_{\sum_{k=1}^{Tx} \exp(e_{ik})} \qquad \text{Attention weight}$$

Finally, we calculate the **context vector** as a linear combination of the outputs of each step of the encoder for the attention weights just calculated.

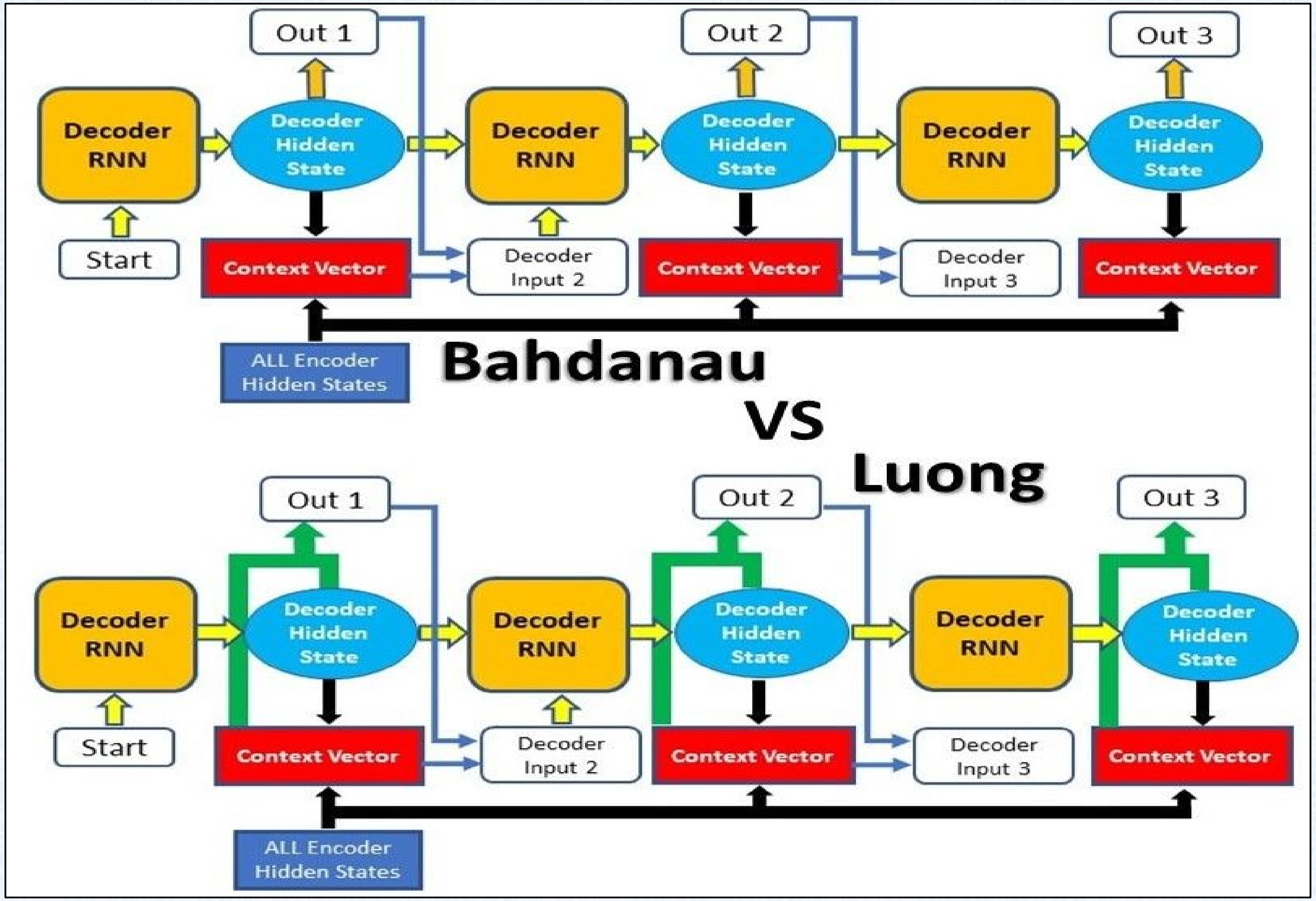
$$C_i = \sum_{j=1}^{Tx} \alpha_{ij}\, h_j \quad \text{Context vector}$$

Application IV : Attention Mechanism

- **Bahdanau attention** has a high overhead → comparable results but with lower computational cost are obtained with **Luong Attention**.

- https://arxiv.org/pdf/1508.04025.pdf

Application IV : Attention Mechanism



Application V : possible improvements

- We have faced the NMT task using an encoder/decoder architecture with teacher forcing and we have seen the improvements that can be obtained by introducing the attention mechanism.

Possible improvements

- ✓ Improve the dataset.
- ✓ Parameter tuning.
- ✓ Use the latest attention mechanism → Transformers.
- ✓ **Scheduled Sampling for Sequence Prediction with Recurrent Neural Networks :** https://arxiv.org/pdf/1506.03099.pdf (2015, Bengio at al)
  GOAL? Decrease discrepancy between training and inference.

Thanks for your attention