# *Natural Language Processing (NLP) with Deep NLP*

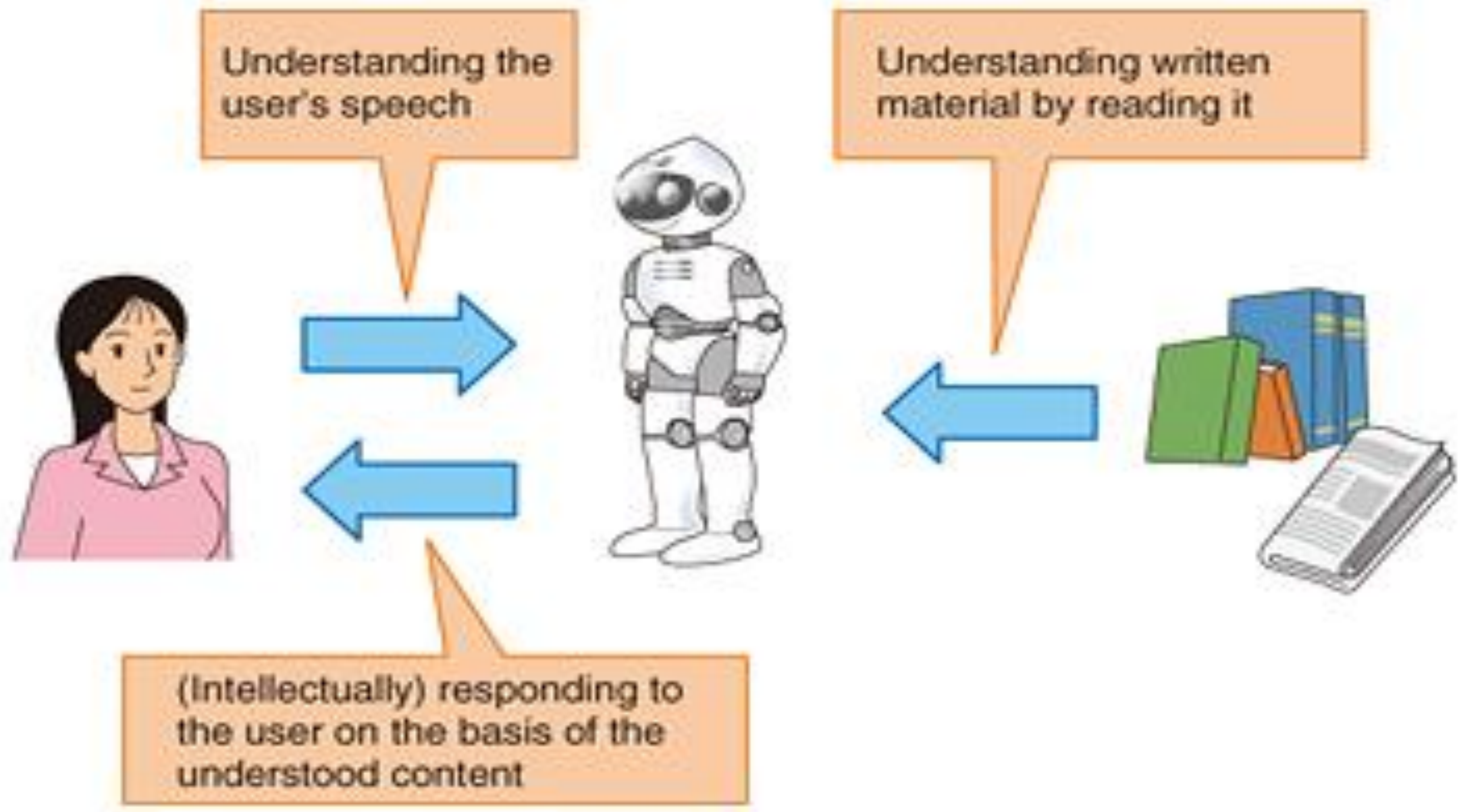
**AMIT DHOMNE**

***Computer Science Instructor***

**Machine Learning and Deep learning** Practitioner

# Prerequisite
# for
# Natural language processing

- ***Python***
- ***Basic Concept of Machine Learning and Deep Learning***

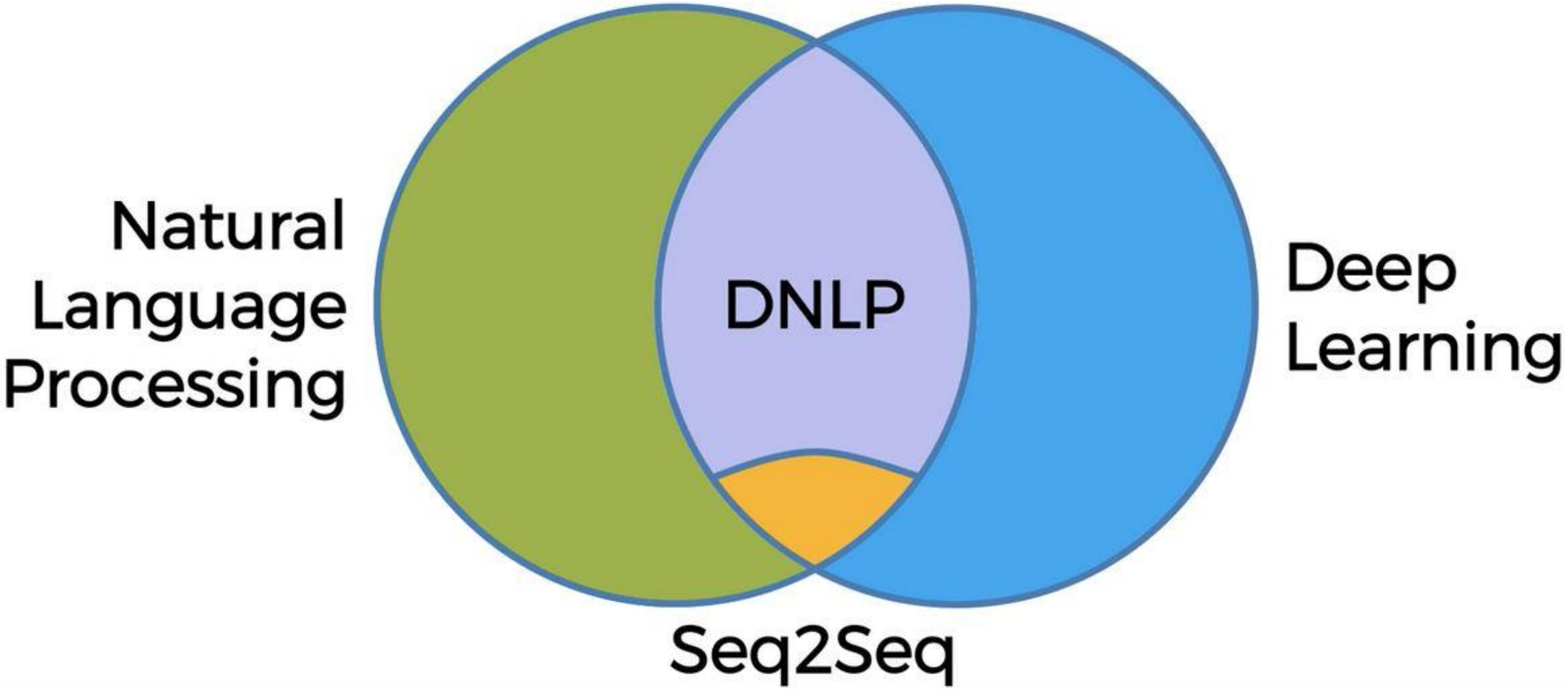

Understanding the user's speech
Understanding written material by reading it
(Intellectually) responding to the user on the basis of the understood content

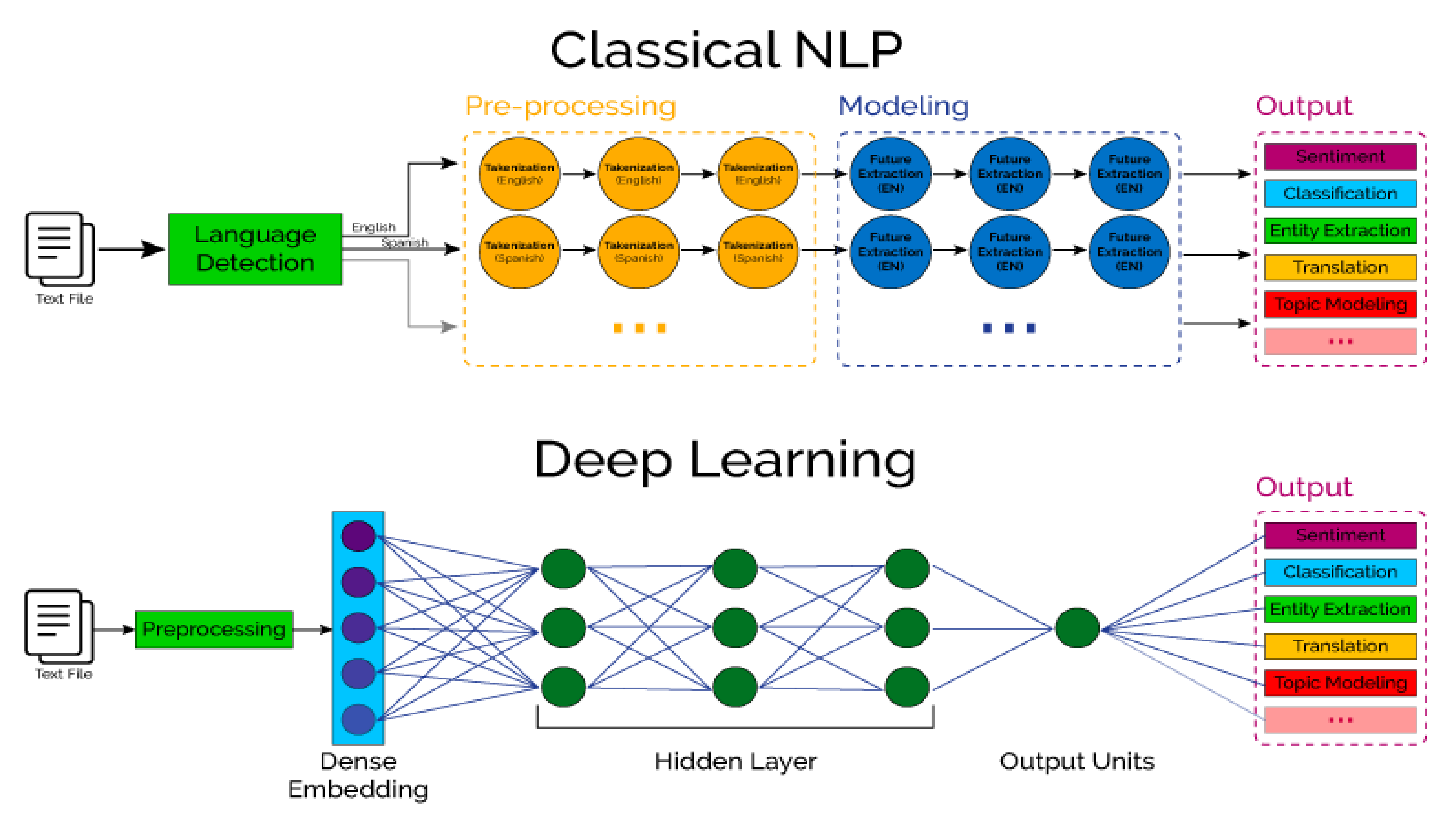
# Natural language processing

Natural language processing (NLP) is a subfield of linguistics, computer science, information engineering, and artificial intelligence concerned with the interactions between computers and human (natural) languages, in particular how to program computers to process and analyze large amounts of natural language data.

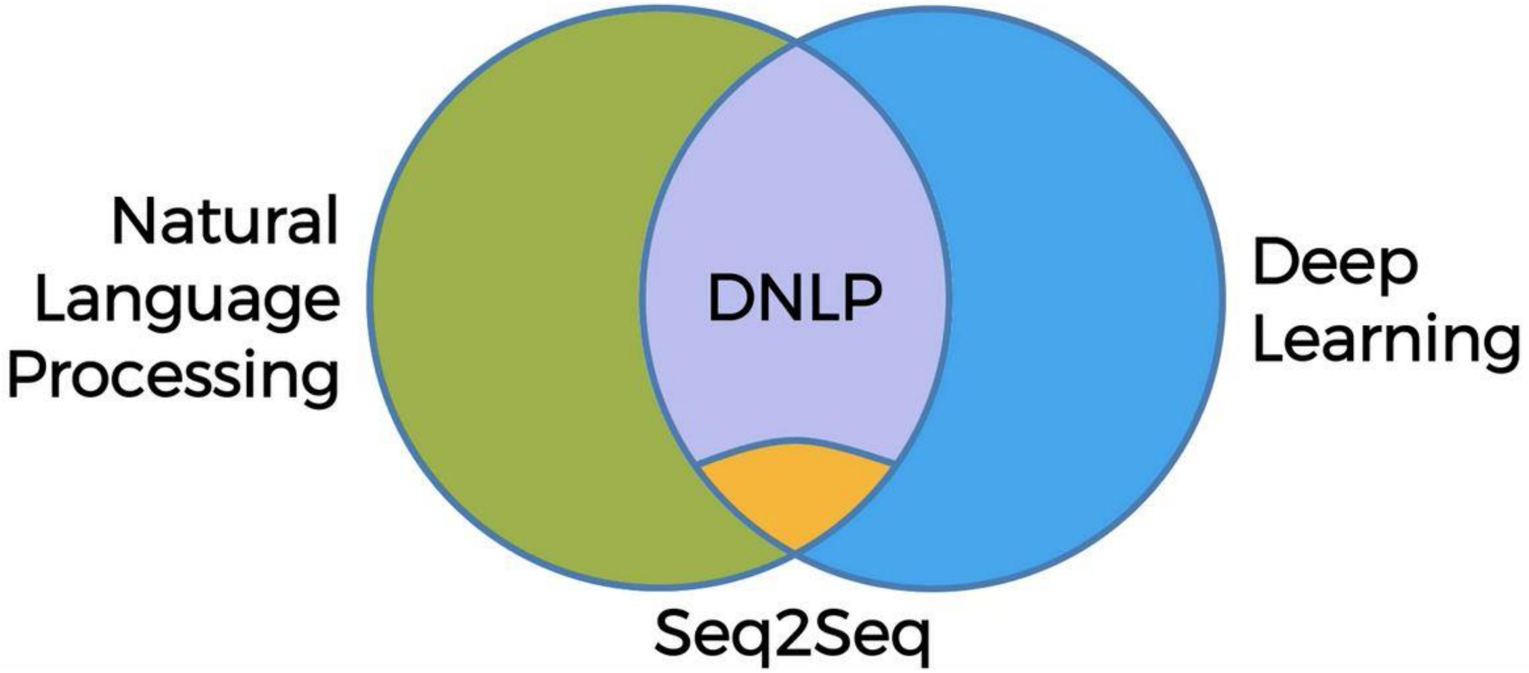
Challenges in natural language processing frequently involve speech recognition, natural language understanding, and natural language generation.

# Types of NLP




Classical NLP
Pre-processing
Modeling
Output
Text File
Language Detection
English
Spanish
Tokenization (English)
Tokenization (English)
Tokenization (English)
Tokenization (Spanish)
Tokenization (Spanish)
Tokenization (Spanish)
Future Extraction (EN)
Future Extraction (EN)
Future Extraction (EN)
Future Extraction (EN)
Future Extraction (EN)
Future Extraction (EN)
Sentiment
Classification
Entity Extraction
Translation
Topic Modeling
...
Deep Learning
Output
Text File
Preprocessing
Dense Embedding
Hidden Layer
Output Units
Sentiment
Classification
Entity Extraction
Translation
Topic Modeling
...

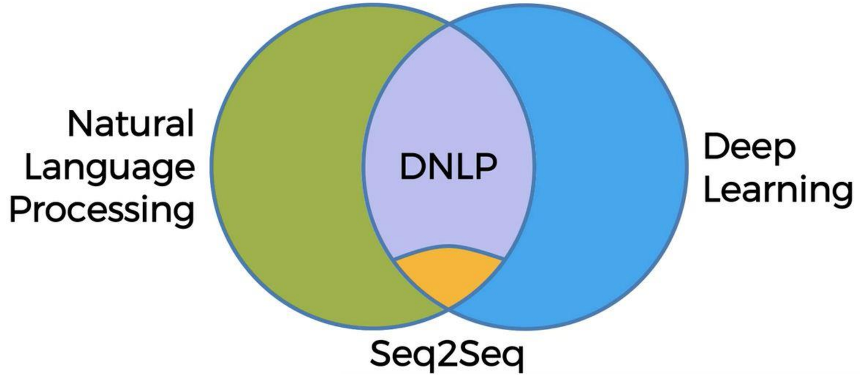
# How NLP, DNLP and DL involves in!!!

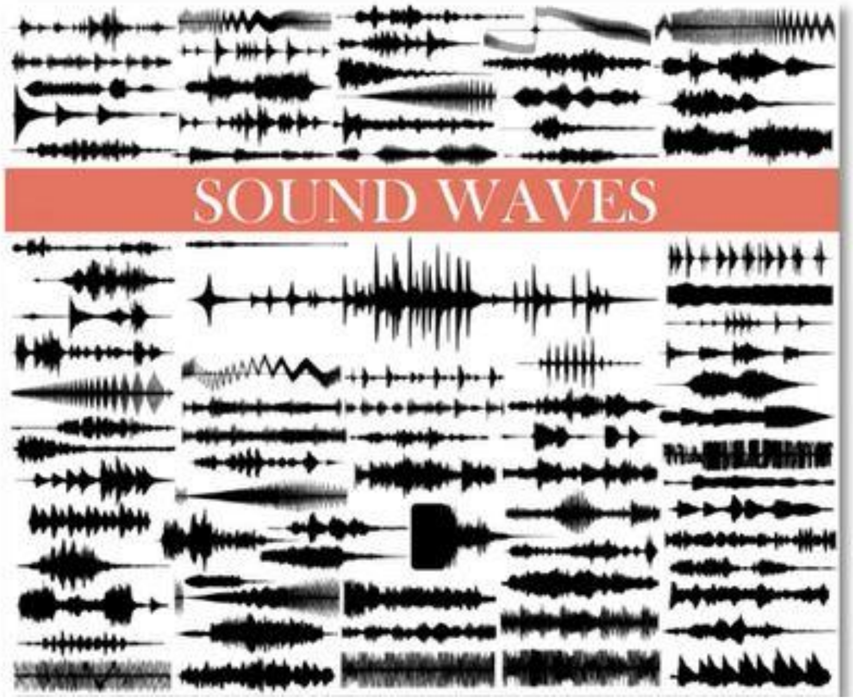


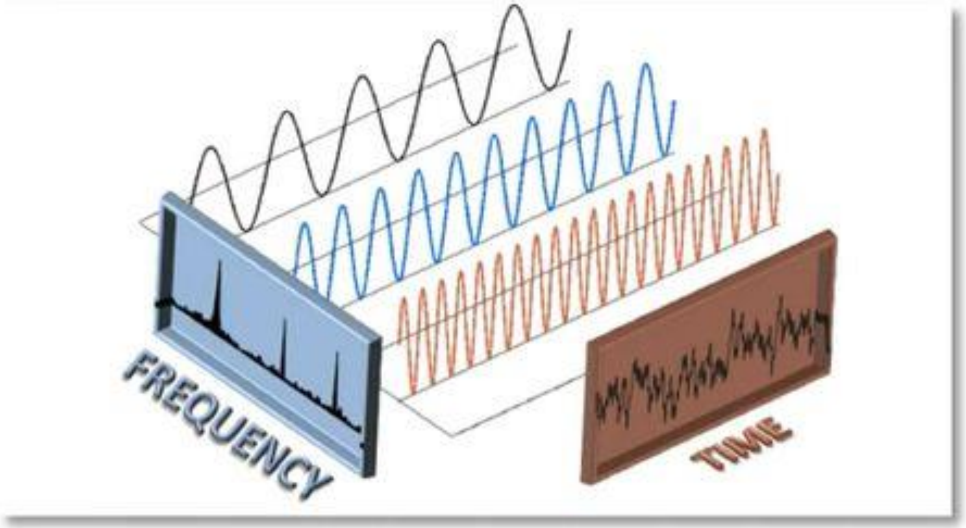

1. If / Else Rules (Chatbot)
2. Audio frequency components analysis (Speech Recognition)
3. Bag-of-words model (Classification)
4. CNN for text Recognition (Classification)
5. Seq2Seq (many applications)

s

# How NLP, DNLP and DL involves in!!!



# Applications

Speech
Transcription

Neural Machine
Translation (NMT)

Chatbots


Q&A

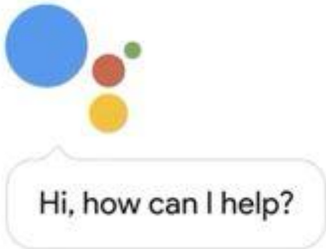
Text
Summarization

Image
Captioning

Video
Captioning

# Used by

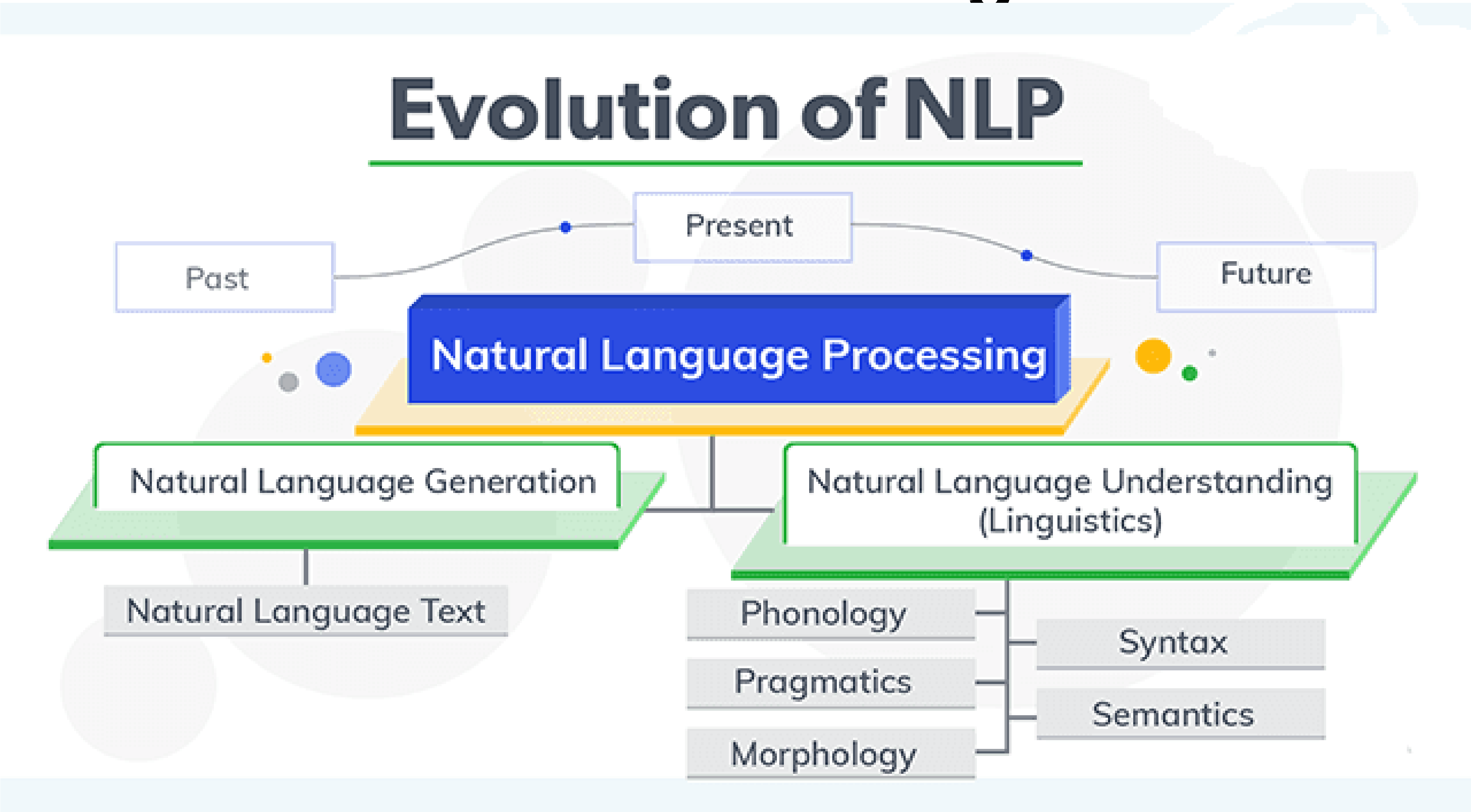




# NLP Working


NLP
NLG
NLU
Natural Language Processing is computers reading language
Natural Language Generation is computers writing language.
Natural Language Understanding is computers understanding language. e.g. Siri & Amazon Alexa.

# NLP Working



Evolution of NLP
Past
Present
Future
Natural Language Processing
Natural Language Generation
Natural Language Understanding (Linguistics)
Natural Language Text
Phonology
Pragmatics
Morphology
Syntax
Semantics

# Natural Language Understanding

**Ambiguity:**

- **Lexical Ambiguity : The Tank is full of water.**
- **Syntactic Ambiguity : ill men and women get to hospital.**
- **Semantic Ambiguity : The Bike hit the pole while it was running.**
- **Pragmatic Ambiguity : The Army is coming.**

**Phonology – This science helps to deal with patterns present in the sound and speeches related to the sound as a physical entity.**

**Pragmatics – This science studies the different uses of language.**

**Morphology – This science deals with the structure of the words and the systematic relations between them.**

**Syntax – This science deal with the structure of the sentences.**

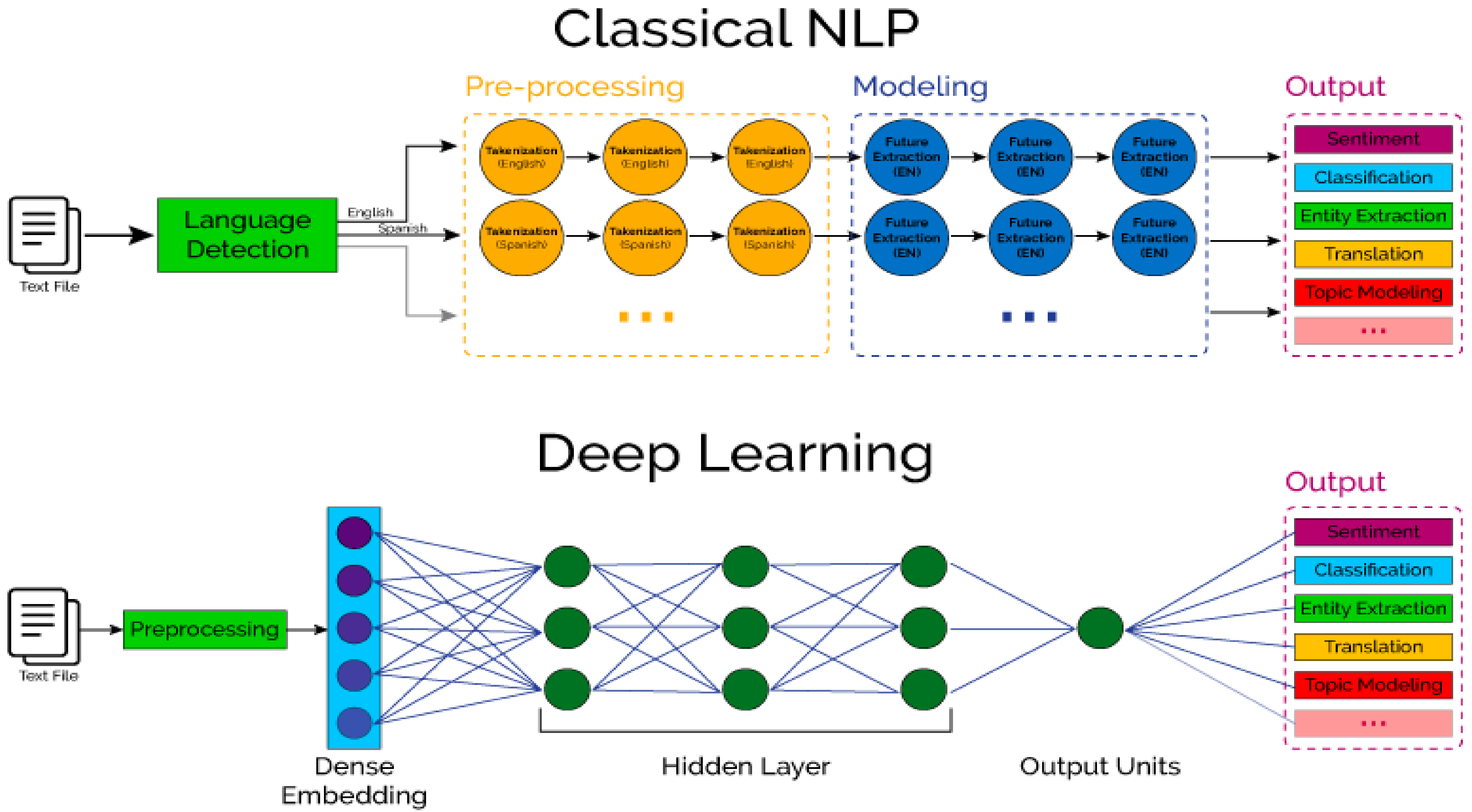
**Semantics – This science deals with the literal meaning of the words, phrases as well as sentences.**

# Natural Language Generation

Based on NL-Understanding, it will suggest about:

- **What should say to user.**
- **Should be Intelligent and Covervational as like human**
- **Usage of Structured data.**
- **With text and Sentence like planning.**

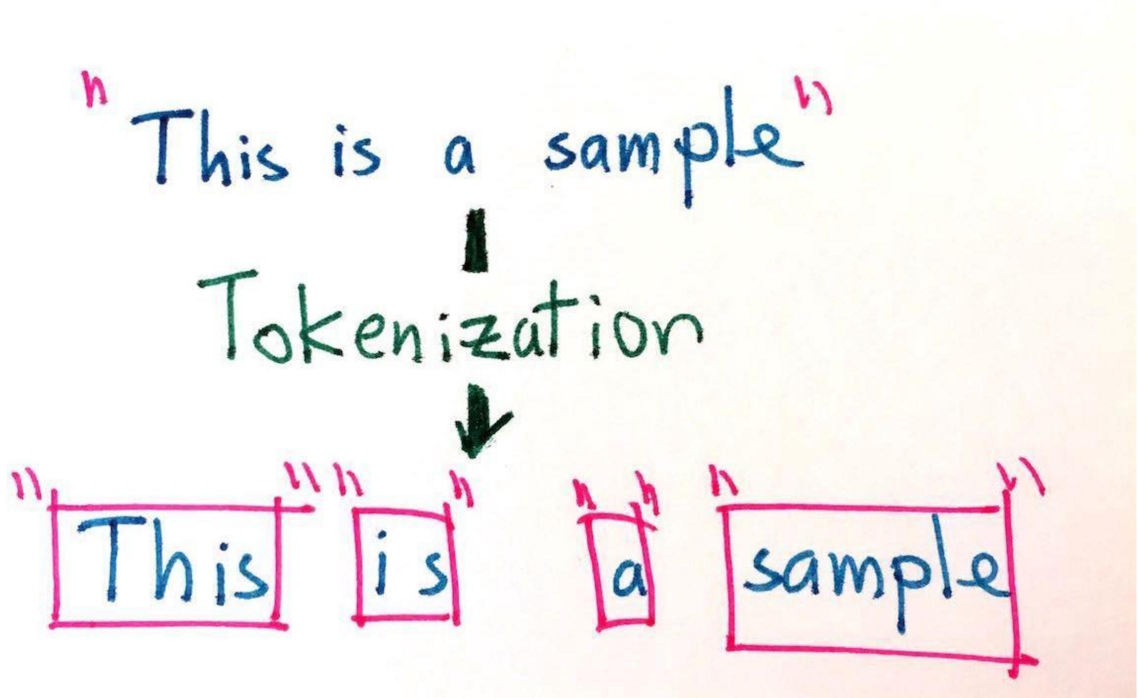
**NLP = NLU + NLG**



Classical NLP
Pre-processing
Modeling
Output
Text File
Language Detection
English
Spanish
Tokenization (English)
Tokenization (Spanish)
Future Extraction (EN)
Sentiment
Classification
Entity Extraction
Translation
Topic Modeling
...
Deep Learning
Output
Text File
Preprocessing
Dense Embedding
Hidden Layer
Output Units
Sentiment
Classification
Entity Extraction
Translation
Topic Modeling
...

# Tokenization

**Tokenization** is the process of replacing sensitive data with unique identification symbols that retain all the essential information about the data without compromising its security.



# Tokenization

**There are many library / framework for NLP problem solution**

1. **Natural Language Toolkit (NLTK)**
2. **TextBlob**
3. **CoreNLP**
4. **Gensim**
5. **spaCy**
6. **polyglot**
7. **scikit–learn**
8. **Pattern**

**...**

# Stemming and Lemmatization

Before understanding **Stemming and Lemmatization**, first let's understand the following…

**Prefix:** Character(s) at the beginning ▸ $ ( “ ¿

**Suffix:** Character(s) at the end ▸ km ) , . ! ”

**Infix:** Character(s) in between ▸ - -- / ...

**Exception:** Special-case rule to split a string into several tokens or prevent a token from being split when punctuation rules are applied St. U.S.

# Stemming

**Stemming** is the process of reducing a word to its word stem that affixes to suffixes and prefixes or to the roots of words known as a lemma. **Stemming** is important in natural language understanding (NLU) and **natural language processing (NLP)**.

| (F) | Rule | | | Example | | |
|---|---|---|---|---|---|---|
| | SSES | → | SS | caresses | → | caress |
| | IES | → | I | ponies | → | poni |
| | SS | → | SS | caress | → | caress |
| | S | → | | cats | → | cat |

| Form | Suffix | Stem |
|---|---|---|
| studies | -es | studi |
| studying | -ing | study |
| niñas | -as | niñ |
| niñez | -ez | niñ |

# Porter Stemming

One of the most common - and effective - stemming tools is *Porter's Algorithm* developed by Martin Porter in 1980. The algorithm employs five phases of word reduction, each with its own set of mapping rules

e.g: caresses reduces to caress but not cares

# Snowball Stemming

**Snowball** is a small string processing language designed for creating **stemming** algorithms for use in Information Retrieval. This site describes **Snowball**, and presents several useful **stemmers** which have been implemented using it

Useful link for additional reading...

https://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html

# Lemmatization

**Lemmatization** usually refers to doing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the lemma .

***The lemma of 'was' is 'be' and the lemma of 'mice' is 'mouse'. Further, the lemma of 'meeting' might be 'meet' or 'meeting' depending on its use in a sentence.***

# Lemmatization

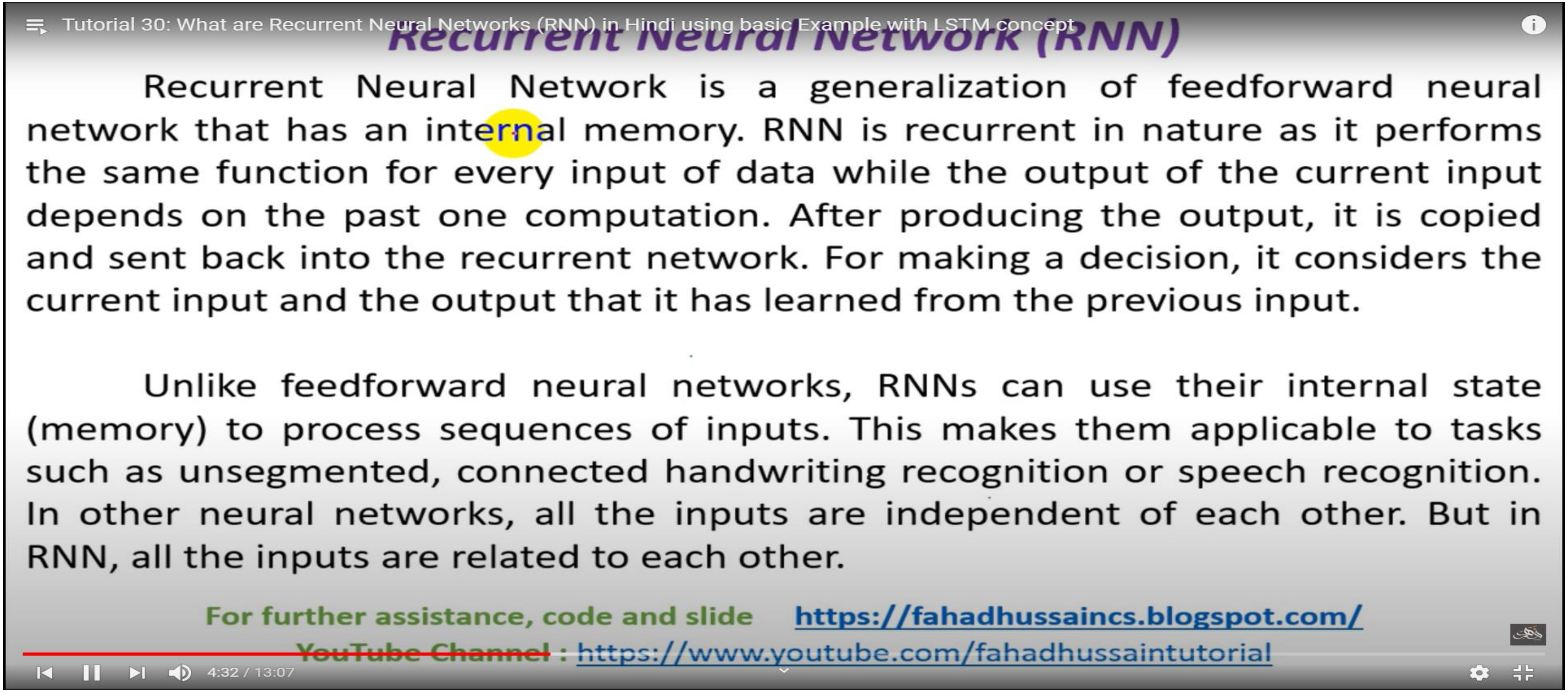
Lemmatization

Mapping from text-word to lemma
***help (verb)***

| **text-word** | to | **lemma** |
|---|---|---|
| help | | help (v) |
| helps | | help (v) |
| helping | | help (v) |
| helped | | help (v) |

https://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html

# Recurrent Neural Network



Tutorial 30: What are Recurrent Neural Networks (RNN) in Hindi using basic Example with LSTM concept

## Recurrent Neural Network (RNN)

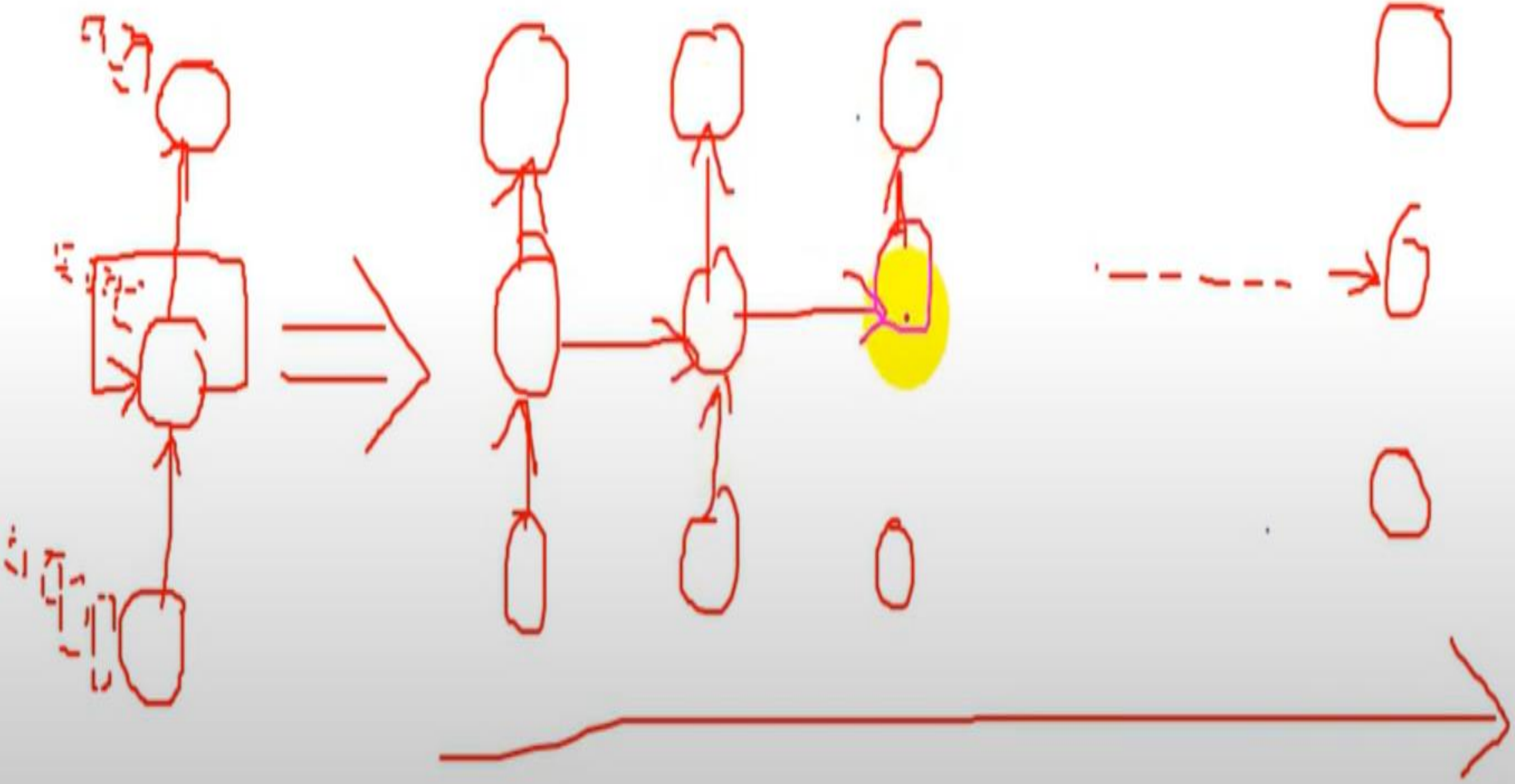
Recurrent Neural Network is a generalization of feedforward neural network that has an internal memory. RNN is recurrent in nature as it performs the same function for every input of data while the output of the current input depends on the past one computation. After producing the output, it is copied and sent back into the recurrent network. For making a decision, it considers the current input and the output that it has learned from the previous input.

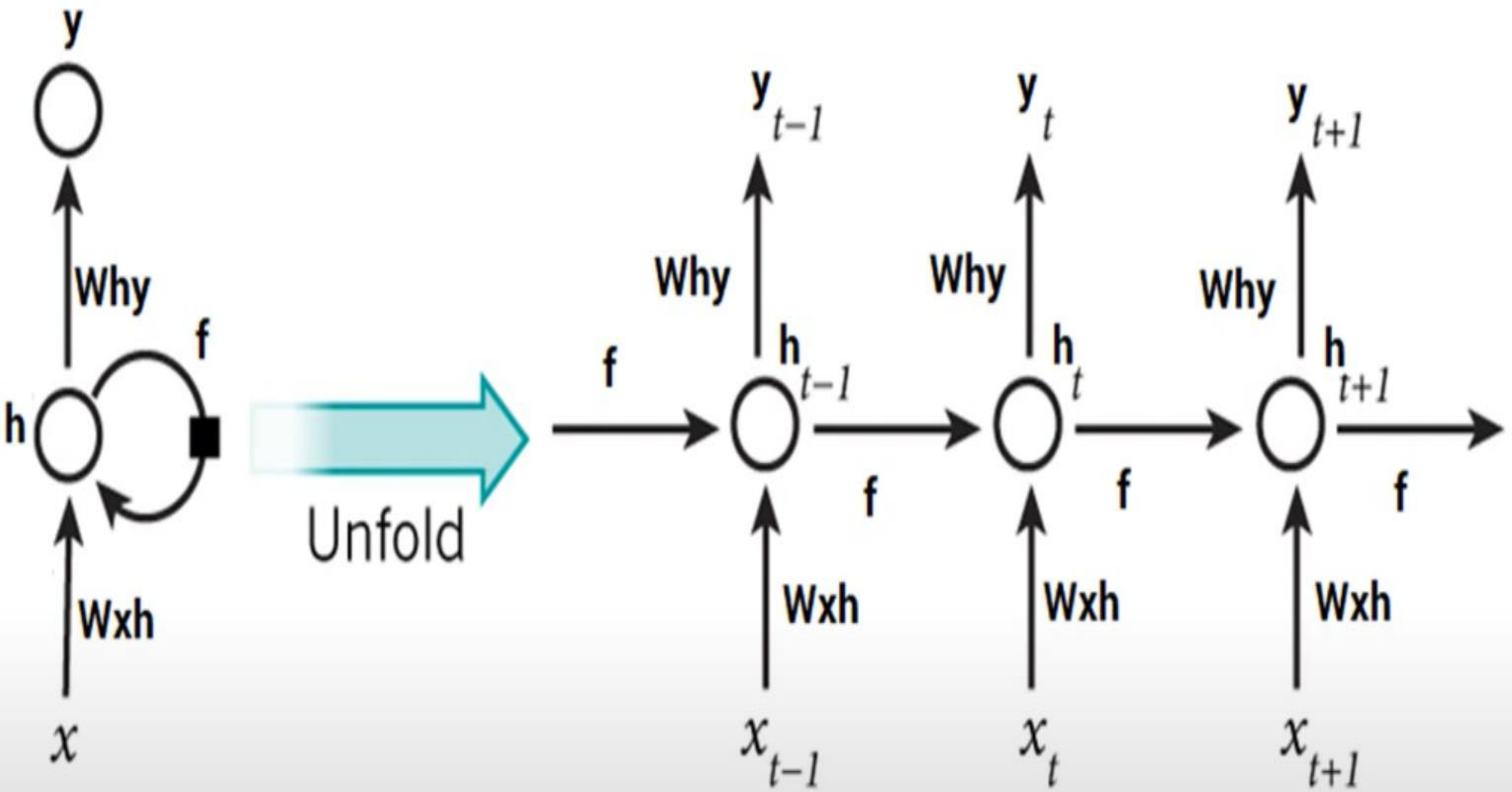
Unlike feedforward neural networks, RNNs can use their internal state (memory) to process sequences of inputs. This makes them applicable to tasks such as unsegmented, connected handwriting recognition or speech recognition. In other neural networks, all the inputs are independent of each other. But in RNN, all the inputs are related to each other.

For further assistance, code and slide https://fahadhussaincs.blogspot.com/

YouTube Channel : https://www.youtube.com/fahadhussaintutorial

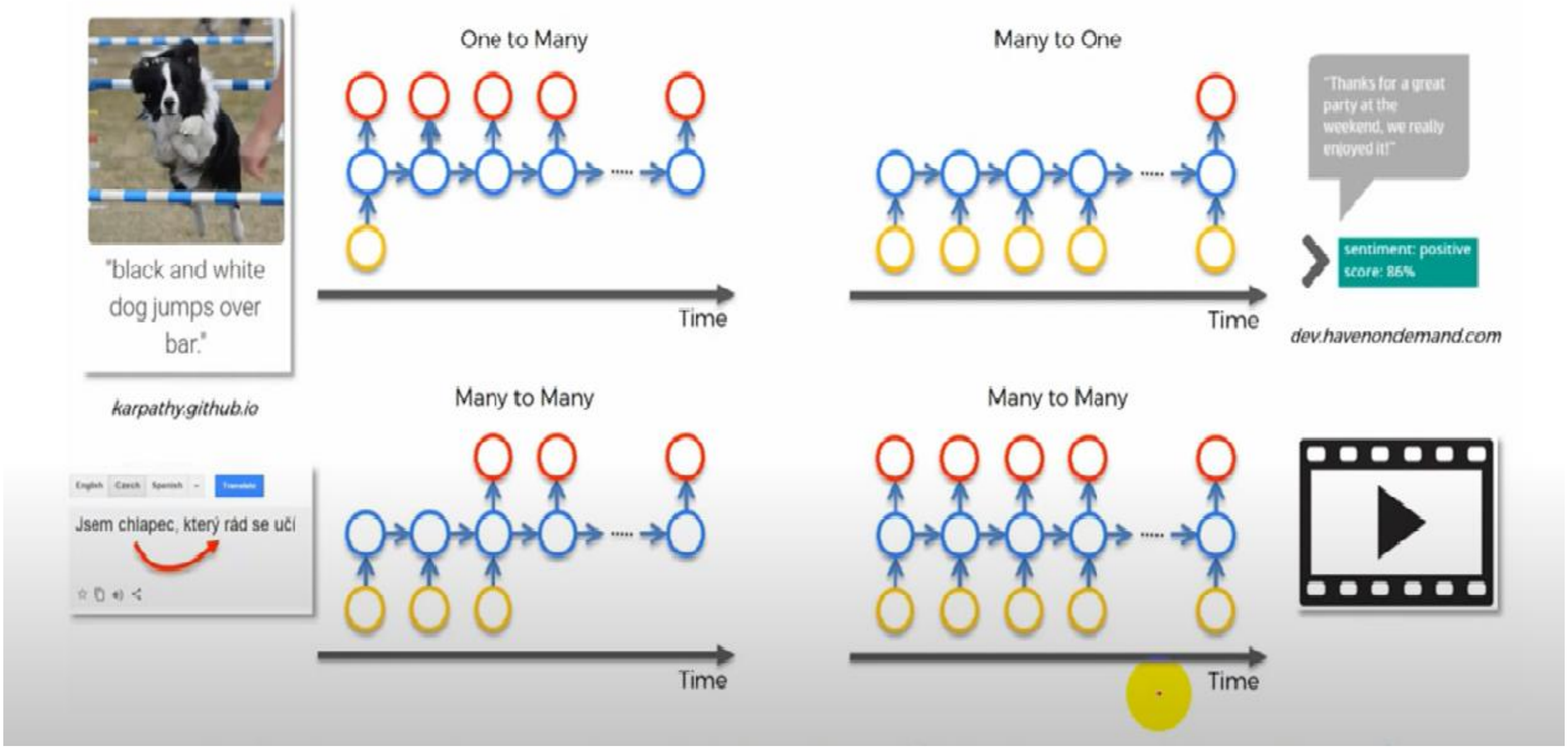
4:32 / 13:07

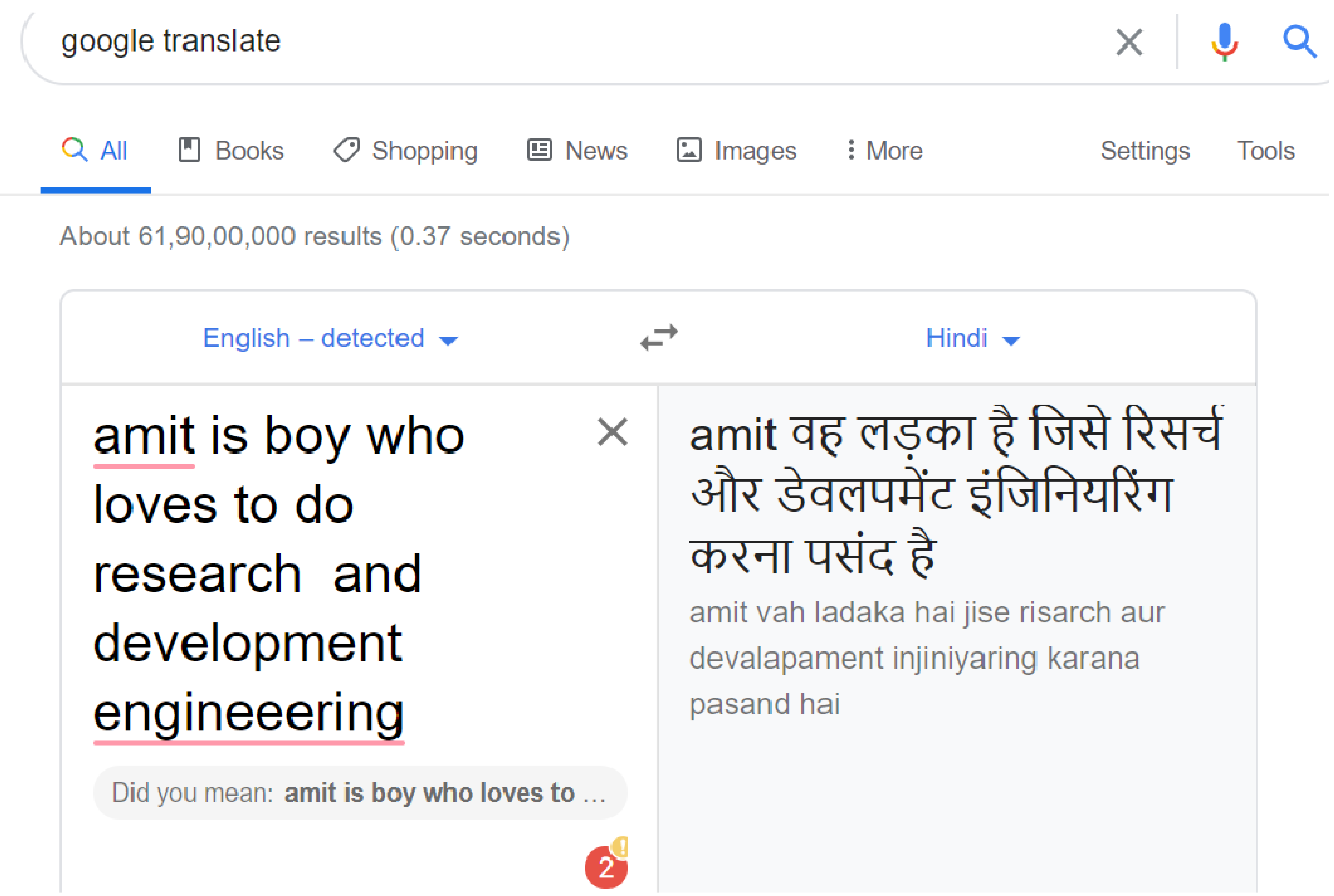




y
Why
f
h
Wxh
x
Unfold
f
y t−1
Why
h t−1
f
Wxh
x t−1
y t
Why
h t
f
Wxh
x t
y t+1
Why
h t+1
f
Wxh
x t+1

# Types of RNN


"black and white dog jumps over bar."
karpathy.github.io
One to Many
Time
Many to One
Time
"Thanks for a great party at the weekend, we really enjoyed it!"
sentiment: positive
score: 86%
dev.havenondemand.com
Jsem chlapec, který rád se učí
Many to Many
Time
Many to Many
Time


google translate
All
Books
Shopping
News
Images
More
Settings
Tools
About 61,90,00,000 results (0.37 seconds)
English – detected
Hindi
amit is boy who loves to do research  and development engineeering
amit वह लड़का है जिसे रिसर्च और डेवलपमेंट इंजिनियरिंग करना पसंद है
amit vah ladaka hai jise risarch aur devalapament injiniyaring karana pasand hai
Did you mean: amit is boy who loves to …
2

# What is Time series Analysis, How relate it is RNN to

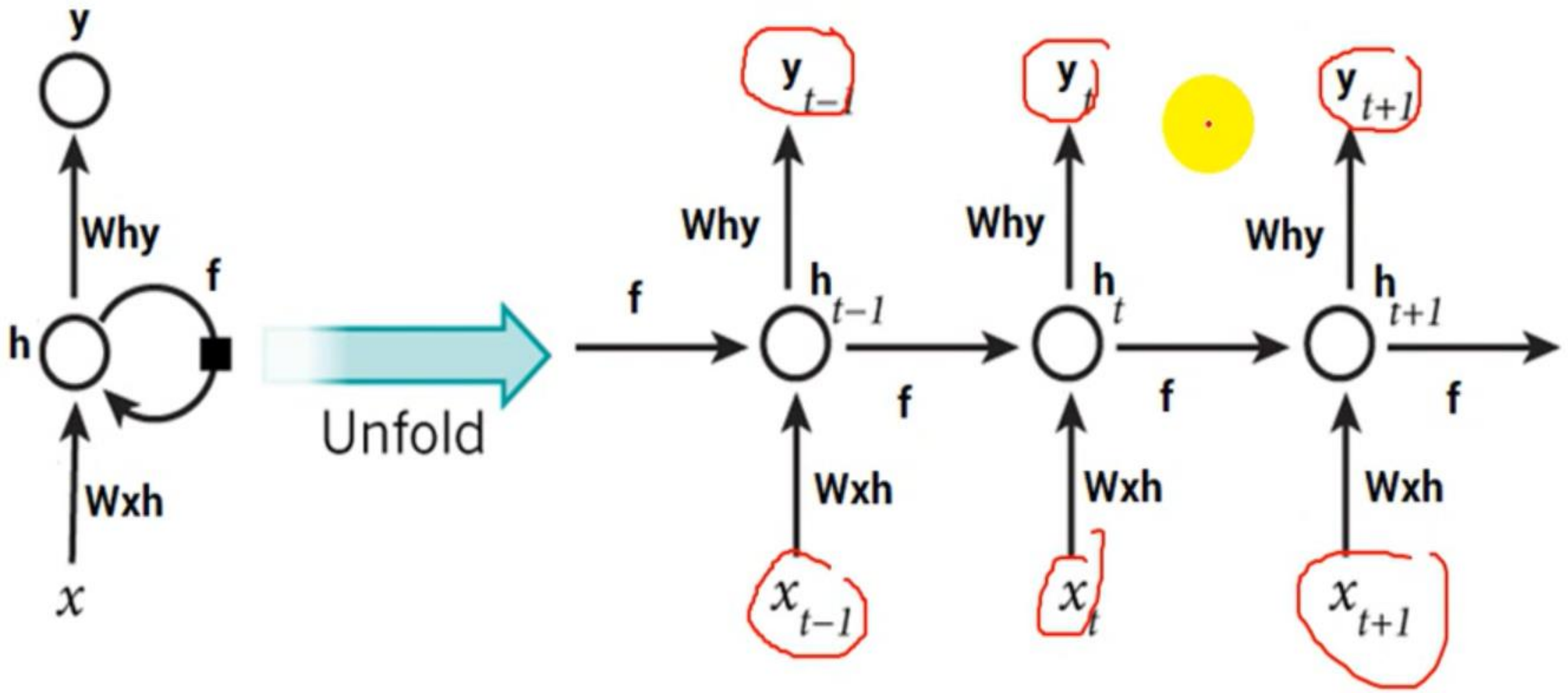
A time series is a series of data points indexed in time order. Most commonly, a time series is a sequence taken at successive equally spaced points in time. Thus it is a sequence of discrete-time data

| | A | B | C |
|---|---|---|---|
| 1 | Year | Quarter | Sales |
| 2 | 2012 | 1 | $165,000.00 |
| 3 | | 2 | $253,000.00 |
| 4 | | 3 | $316,000.00 |
| 5 | | 4 | $287,000.00 |
| 6 | 2013 | 1 | $257,000.00 |
| 7 | | 2 | $308,000.00 |
| 8 | | 3 | $376,000.00 |
| 9 | | 4 | $351,000.00 |

Time series model is purely dependent on the idea that past behavior and price patterns can be used to predict future price behavior.

# *Vanishing gradient problem*

The **vanishing gradient** makes the **gradient** very close to zero, so it's difficult to know where to move in the state space; the exploding **gradient** makes the **gradient** a very large value, so it makes learning unstable. This **problem** is more pronounced in recurrent networks since they use the same matrix at each time step.



## Exploding Gradient:

The working of the exploding gradient is similar but the weights here change drastically instead of negligible change. Notice the small change.

### *Truncated BTT*

Instead of starting backpropagation at the last time stamp, we can choose a smaller time stamp like 10 (we will lose the temporal context after 10 time stamps)

### *Clip gradients at threshold*

Clip the gradient when it goes higher than a threshold

### *RMSprop to adjust learning rate*

## Vanishing Gradient:

When making use of back-propagation the goal is to calculate the error which is actually found out by finding out the difference between the actual output and the model output and raising.

### *ReLU activation function*

We can use activation functions like ReLU, which gives output one while calculating gradient

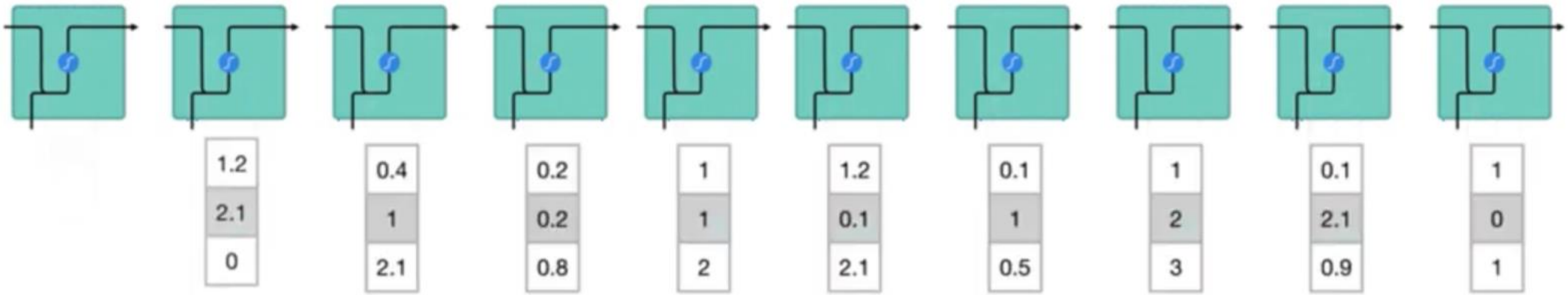
### *RMSprop*

Clip the gradient when it goes higher than a threshold

### *LSTM, GRUs*

Different network architectures that has been specially designed can be used to combat this problem

# First Understand the RNN Works




***This is a cat, and ______ is a good pet animal***

# Basic LSTM

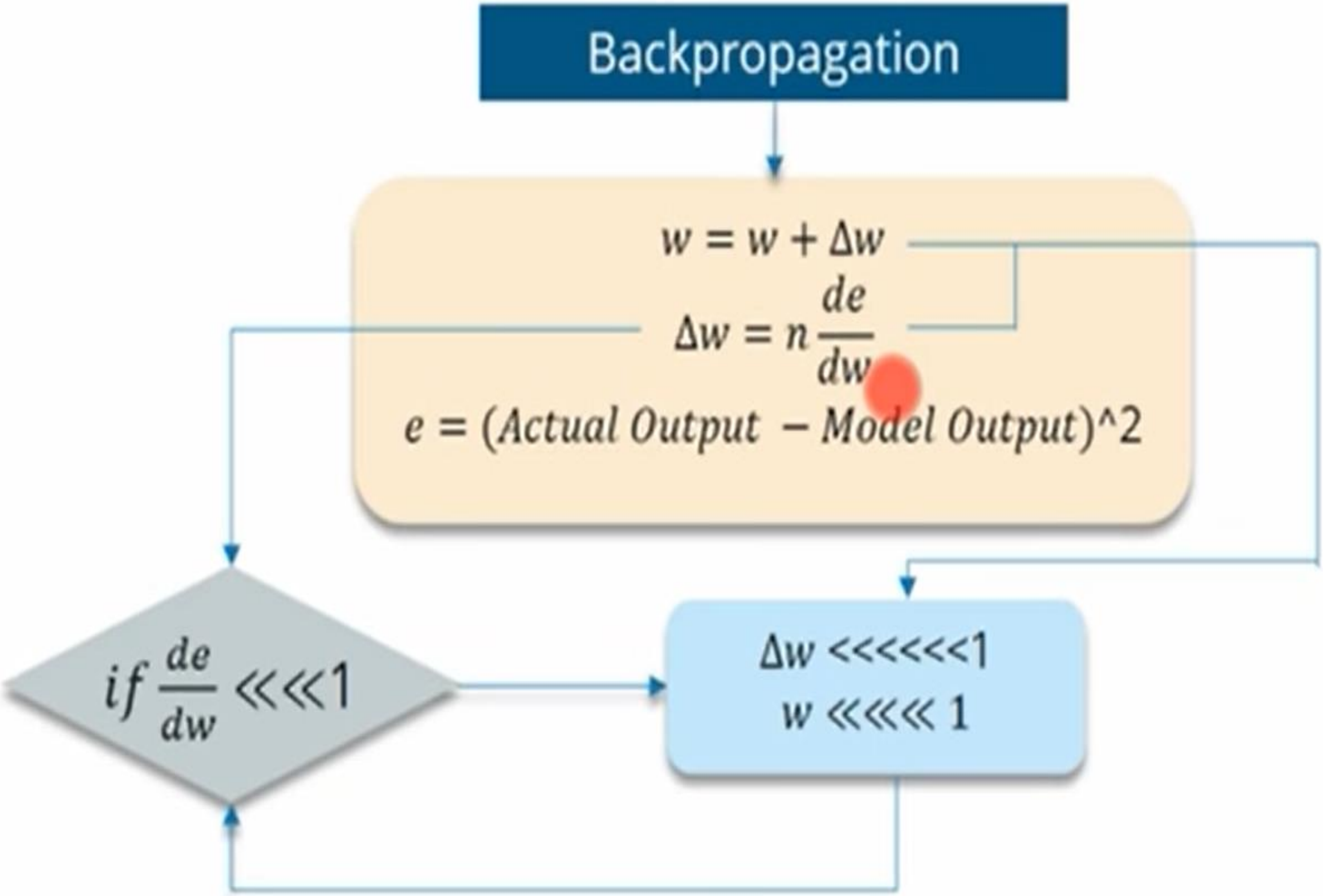
Long short-term memory network was first introduced in 1997 by Sepp Hochreiter and his supervisor for a Ph.D. thesis.

LSTM is a special kind of RNN, capable of learning long term dependencies.

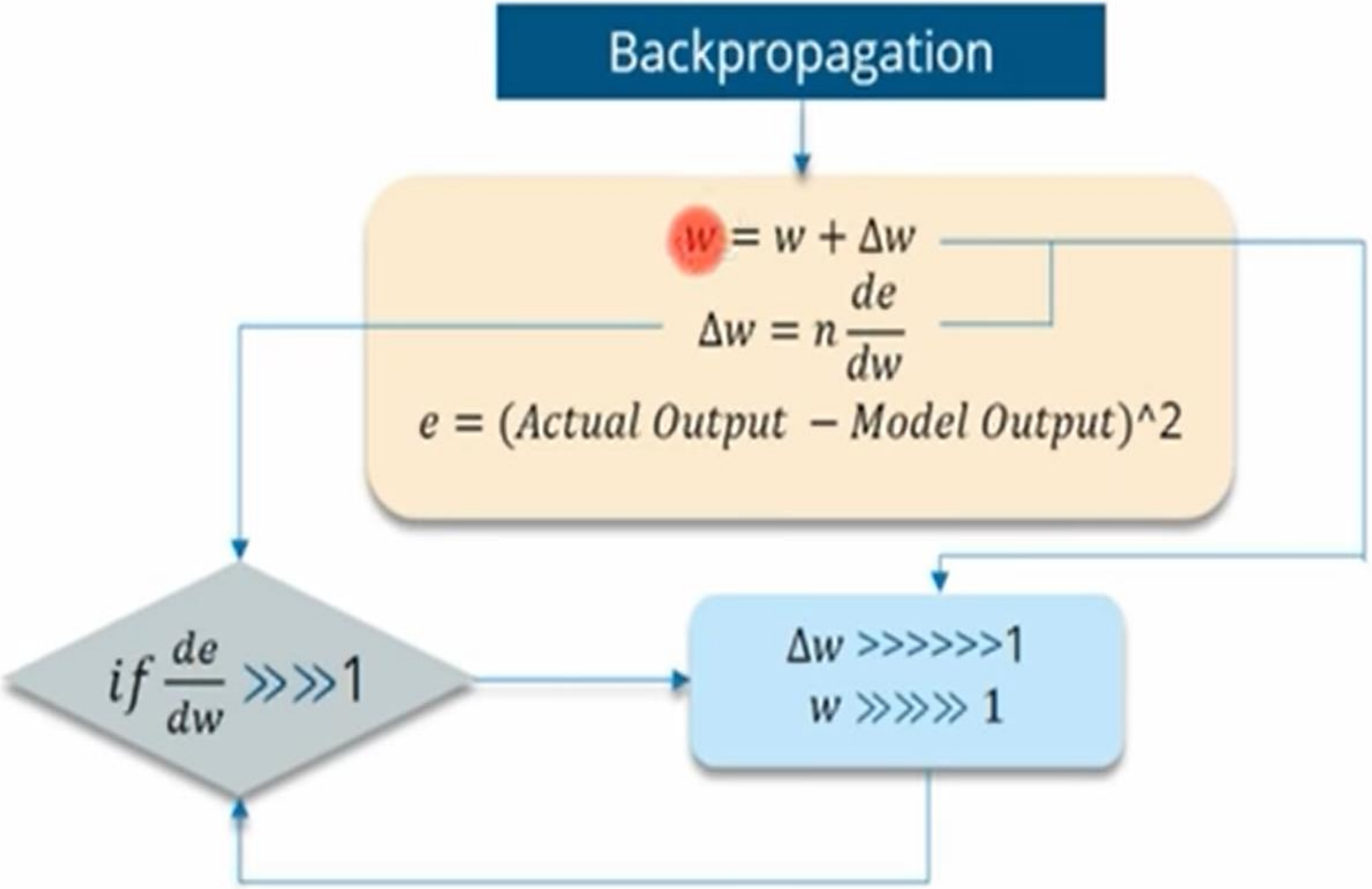
Remembering information for long period of time is it's default behaviour.

Long short-term memory (LSTM) network is the most popular solution to the vanishing gradient problem.

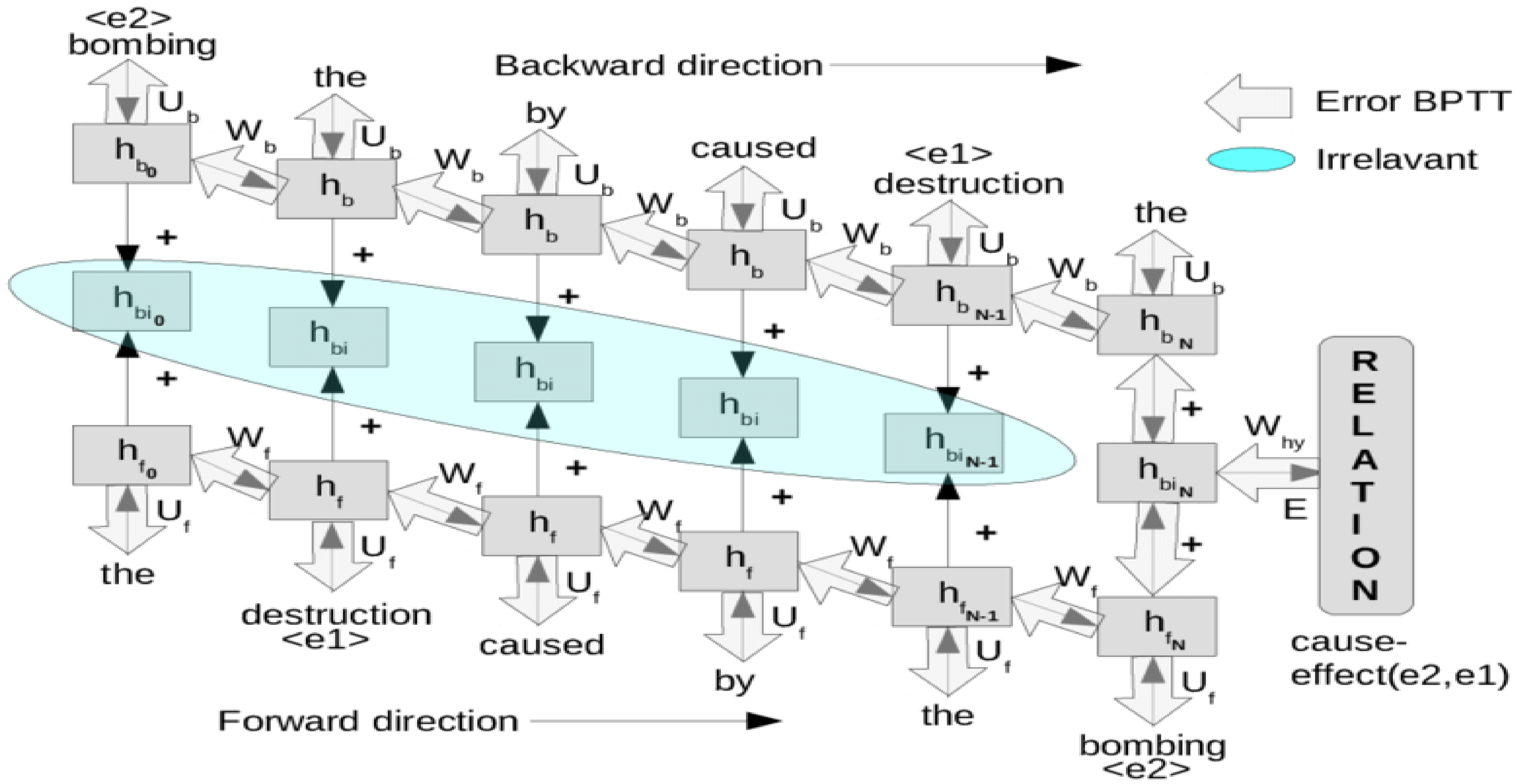
# Vanishing Gradient



# Exploding Gradient



**Back Propagation Through Time(BTT) : Backpropagation through time is a gradient-based technique for training certain types of recurrent neural networks.**



# How To Overcome These Challenges?

## Exploding gradients

- *Truncated BTT*

  Instead of starting backpropagation at the last time stamp, we can choose a smaller time stamp like 10 (we will lose the temporal context after 10 time stamps)

- *Clip gradients at threshold*

  Clip the gradient when it goes higher than a threshold

- *RMSprop to adjust learning rate*

## Vanishing gradients

- *ReLU activation function*

  We can use activation functions like ReLU, which gives output one while calculating gradient

- *RMSprop*

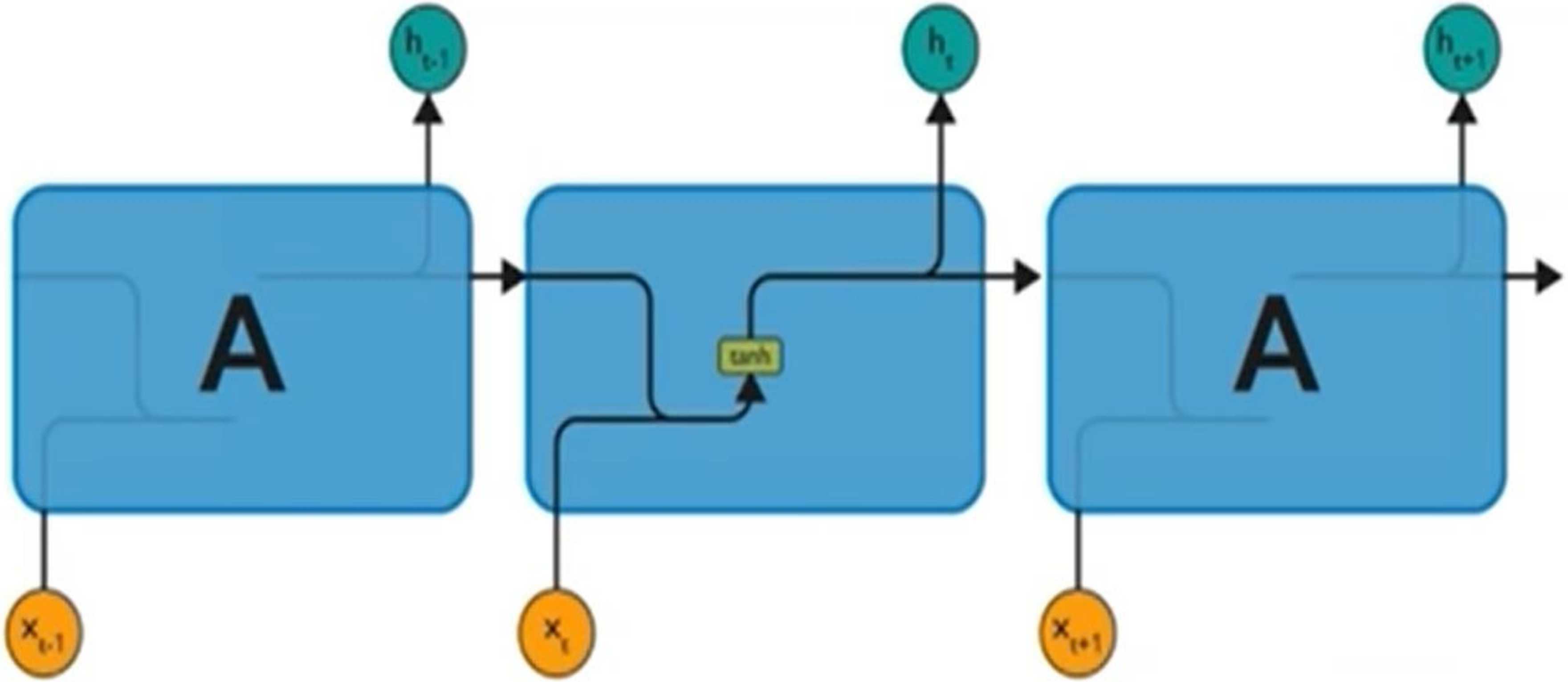
  Clip the gradient when it goes higher than a threshold

- *LSTM, GRUs*

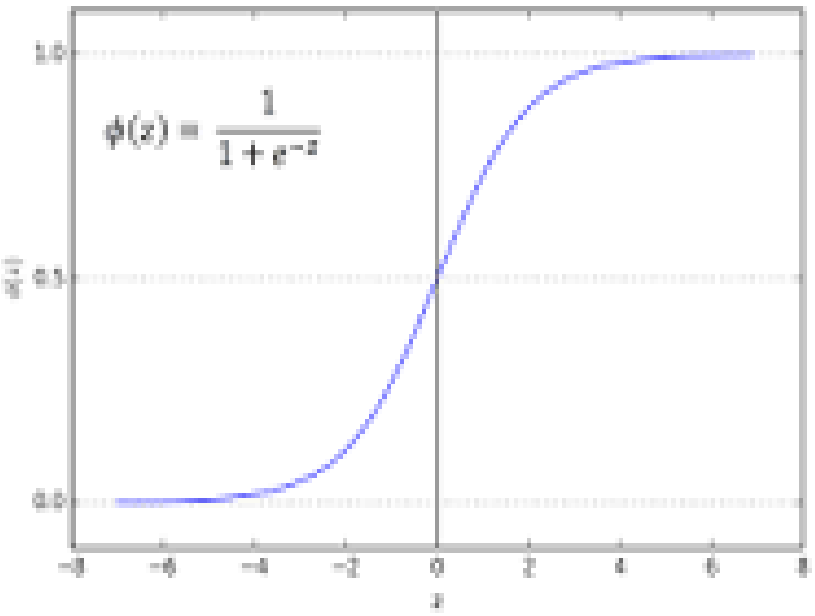
  Different network architectures that has been specially designed can be used to combat this problem

# Long Short Term Memory Networks

- ✓ Long Short Term Memory networks – usually just called "LSTMs" – are a special kind of RNN.
- ✓ They are capable of learning long-term dependencies.

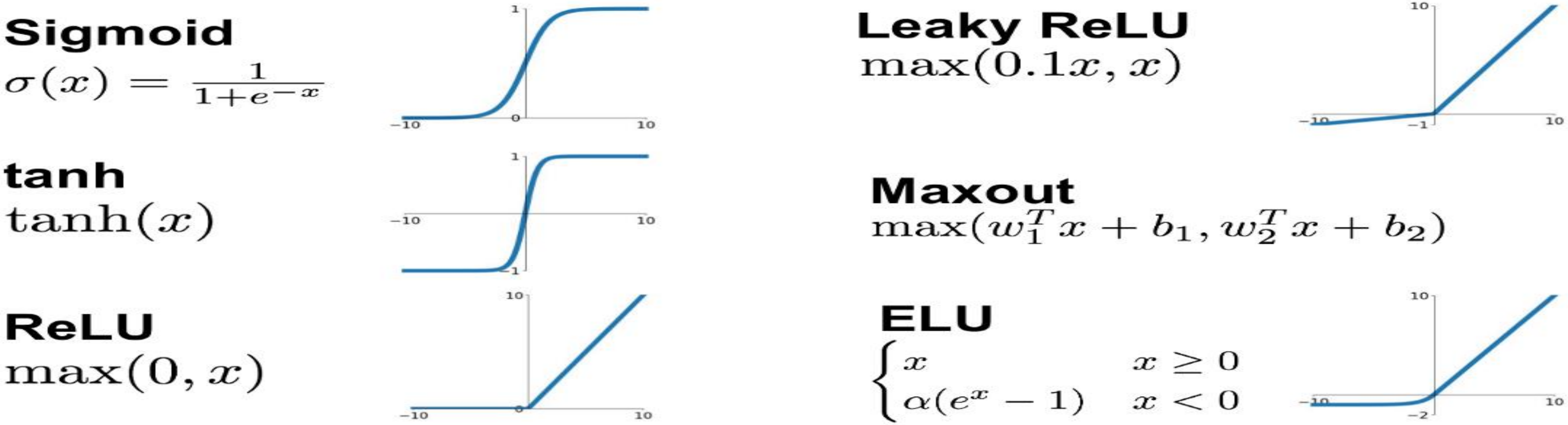



The repeating module in a standard RNN contains a single layer

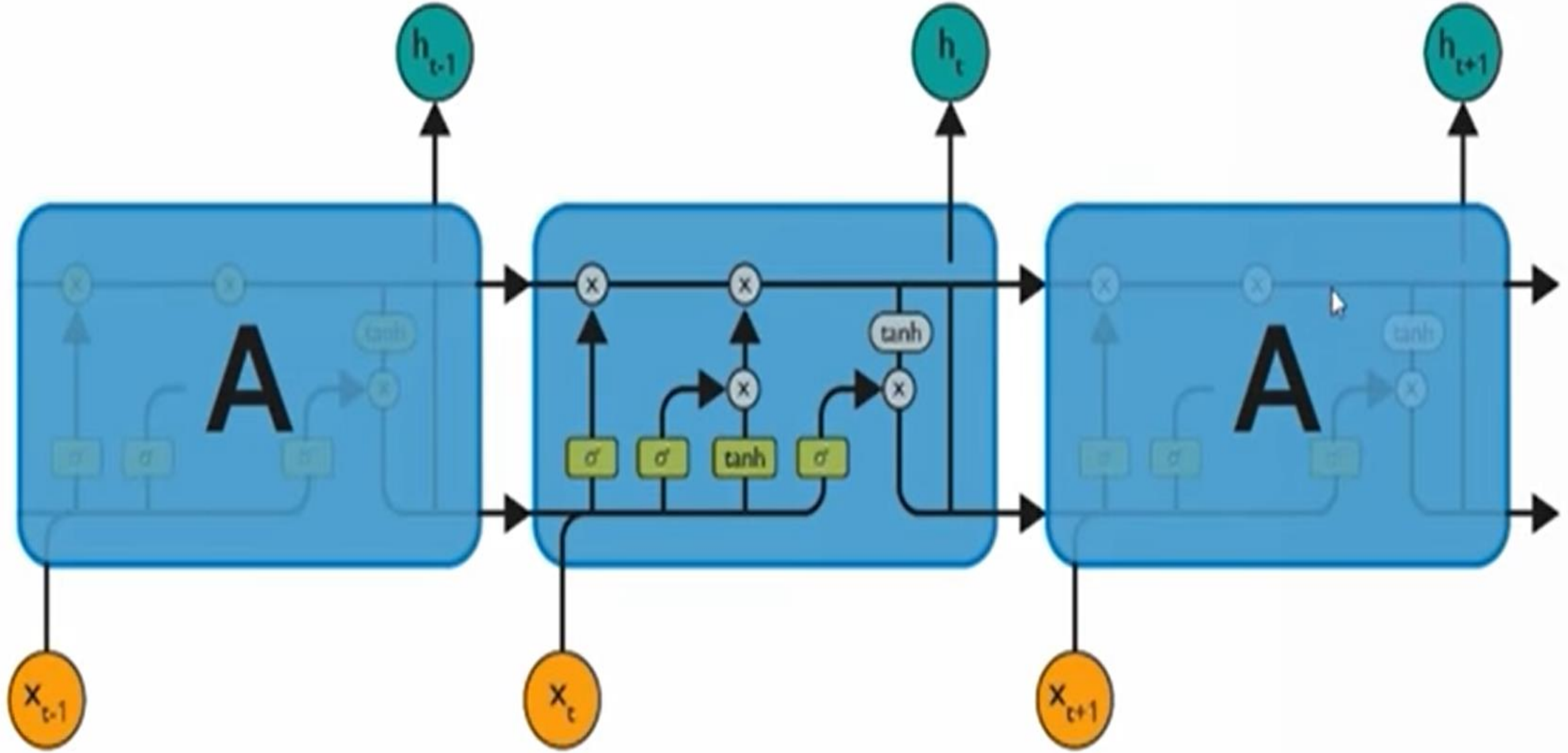



**Sigmoid :** -The **sigmoid activation function**, also called the logistic **function**, is traditionally a very popular **activation function** for neural networks. The input to the **function** is transformed into a value between 0.0 and 1.0

**Tanh :**The range of the **tanh function** is from (-1 to 1)



# Long Short Term Memory Networks

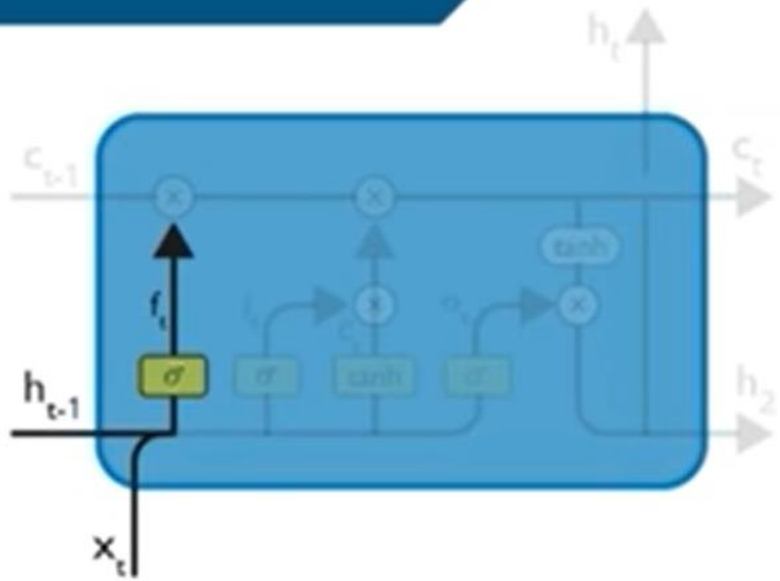


# Long Short Term Memory Networks

Step-1

The first step in the **LSTM** is to identify those information that are not required and will be thrown away from the cell state. This decision is made by a sigmoid layer called as forget gate layer.

$$w_f = Weight$$
$$h_{t-1} = Output\ from\ the\ previous\ time\ stamp$$
$$x_t = New\ input$$
$$b_f = Bias$$



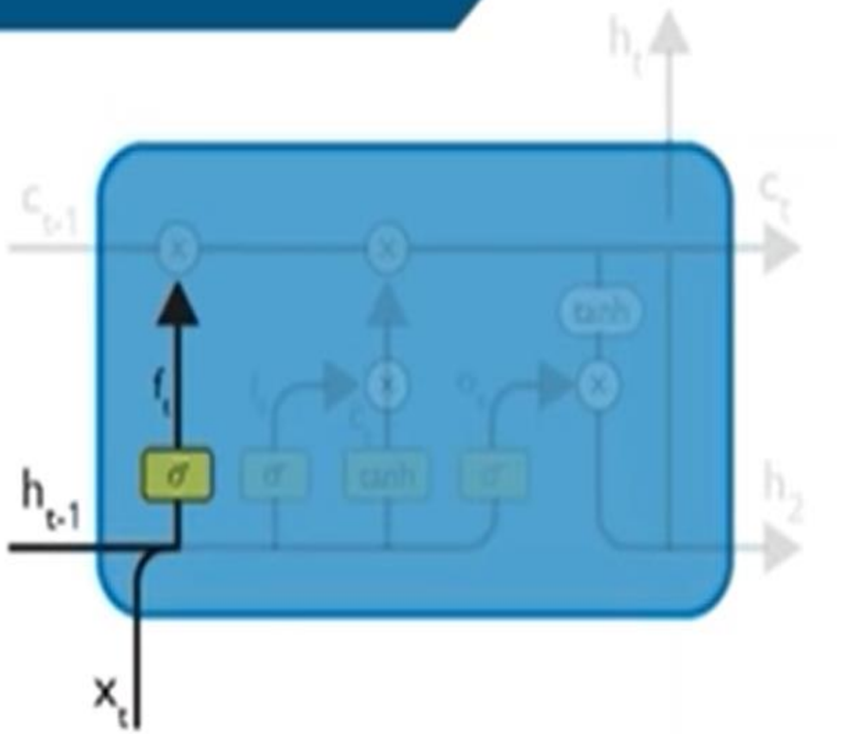

$$f_t = \sigma(w_f[h_{t-1}, x_t] + b_f)$$

# Long Short Term Memory Networks

**Step-1**

The first step in the **LSTM** is to identify those information that are not required and will be thrown away from the cell state. This decision is made by a sigmoid layer called as forget gate layer.

$$w_f = Weight$$
$$h_{t-1} = Output\ from\ the\ previous\ time\ stamp$$
$$x_t = New\ input$$
$$b_f = Bias$$

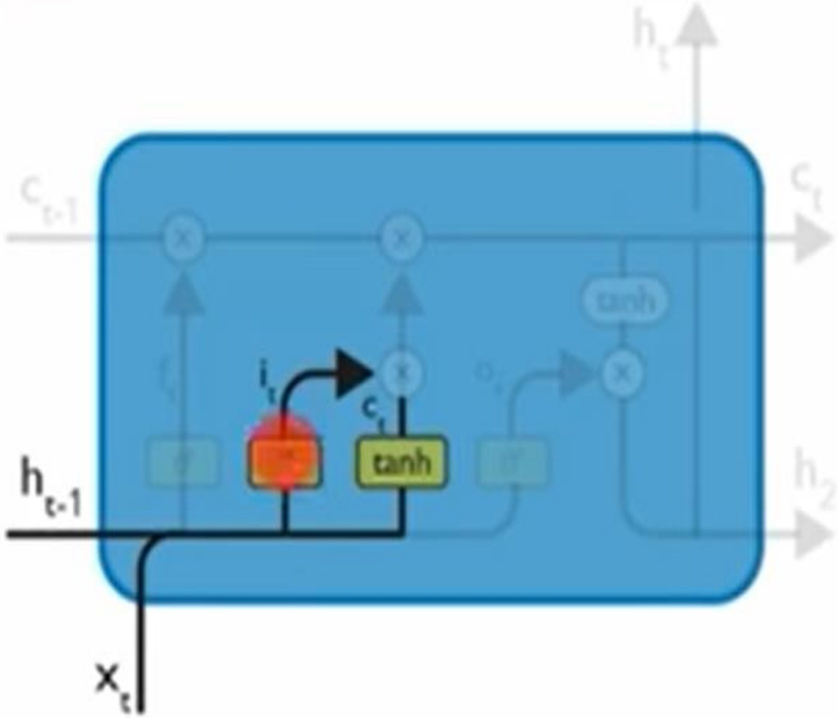



$$f_t = \sigma(w_f[h_{t-1}, x_t] + b_f)$$

# Long Short Term Memory Networks

**Step-2**

The next step is to decide, what new information we're going to store in the cell state. This whole process comprises of following steps. A **sigmoid layer** called the "input gate layer" decides which values will be updated. Next, a **tanh layer** creates a vector of new candidate values, that could be added to the state.




$$i_t = \sigma(w_i[h_{t-1}, x_t] + b_i)$$

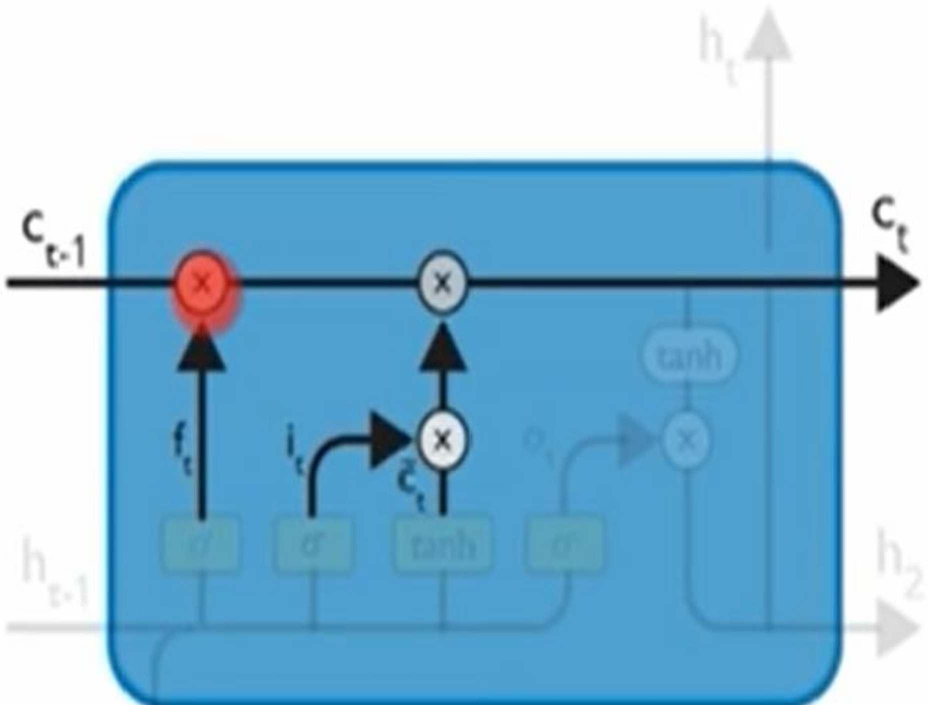
$$\tilde{c}_t = tanh(w_c[h_{t-1}, x_t] + b_c)$$

In the next step, we'll combine these two to update the state.

# Long Short Term Memory Networks

Step-3

Now, we will update the old cell state, $C_{t-1}$, into the new cell state $C_t$. First, we multiply the old state ($C_{t-1}$) by $f_t$, forgetting the things we decided to forget earlier. Then, we add $i_t * \tilde{c}_t$. This is the new candidate values, scaled by how much we decided to update each state value.

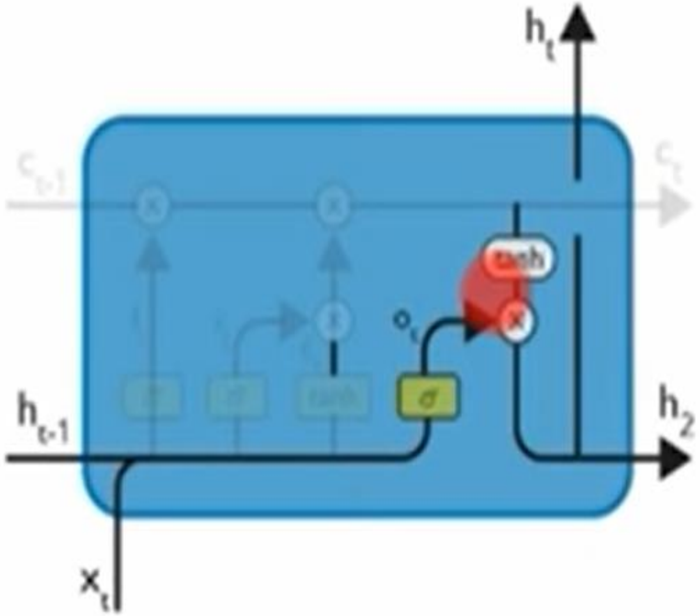



$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t$$

# Long Short Term Memory Networks

**Step-4**

We will run a sigmoid layer which decides what parts of the cell state we're going to output. Then, we put the cell state through tanh (push the values to be between −1 and 1) and multiply it by the output of the sigmoid gate, so that we only output the parts we decided to.

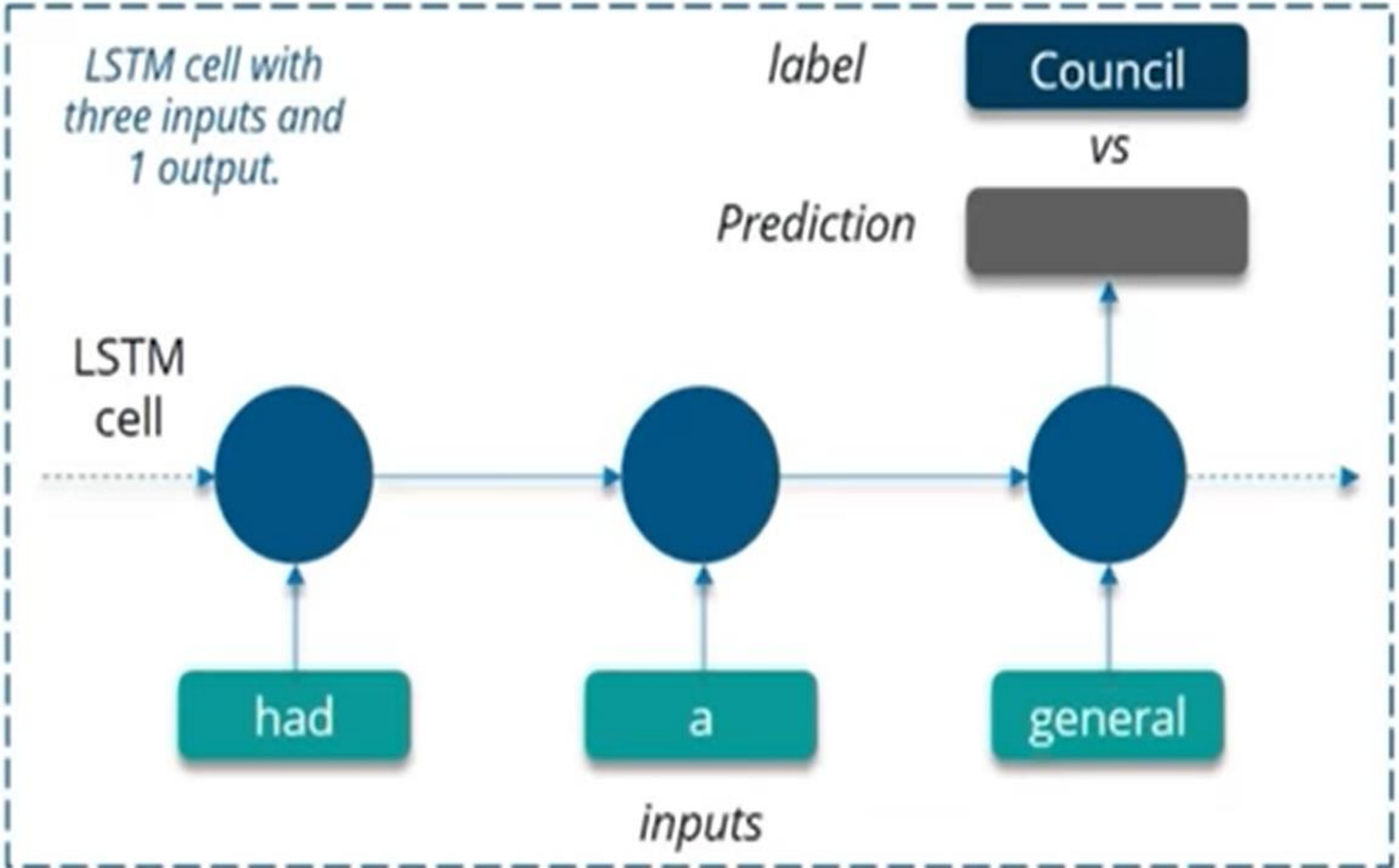



$$o_t = \sigma(w_o[h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh(c_t)$$

# Long Short Term Memory Networks Use-Case

We will feed a LSTM with correct sequences from the text of 3 symbols as inputs and 1 labeled symbol, eventually the neural network will learn to predict the next symbol correctly



# Long Short Term Memory Networks Use-Case


long ago , the mice had a general council to consider what measures they could take to outwit their common enemy , the cat . some said this , and some said that but at last a young mouse got up and said he had a proposal to make , which he thought would meet the case . you will all agree , said he , that our chief danger consists in the sly and treacherous manner in which the enemy approaches us . now , if we could receive some signal of her approach , we could easily escape from her . i venture , therefore , to propose that a small bell be procured , and attached by a ribbon round the neck of the cat . by this means we should always know when she was about , and could easily retire while she was in the neighborhood . this proposal met with general applause , until an old mouse got up and said that is all very well , but who is to bell the cat ? the mice looked at one another and nobody spoke . then the old mouse said it is easy to propose impossible remedies .

A short story from Aesop's Fables
with 112 unique symbols

# Long Short Term Memory Networks Use-Case

A unique integer value is assigned to each symbol because LSTM inputs can only understand real numbers.