

# Semantic representations

## Sahar Ghannay

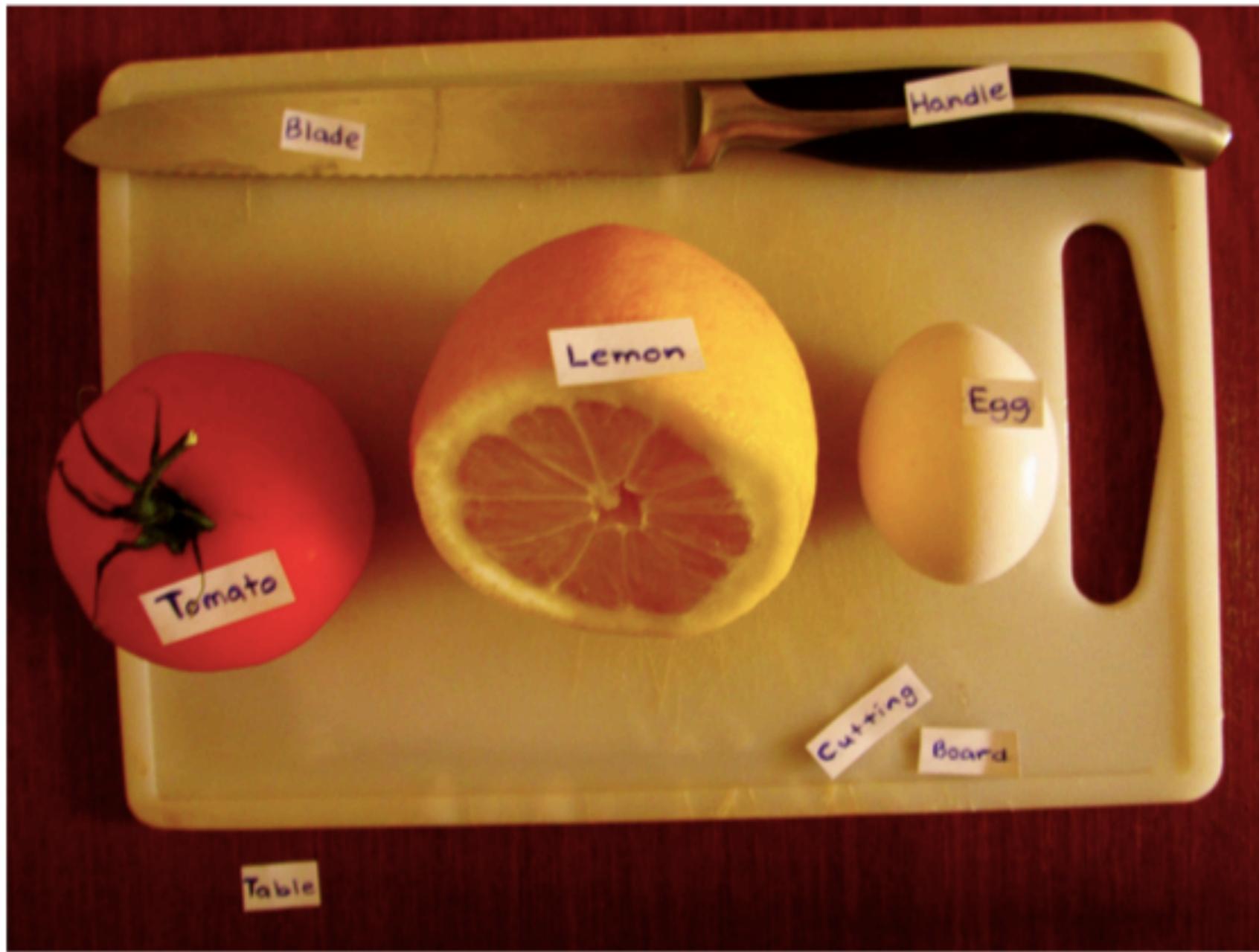
sahar.ghannay@universite-paris-saclay.fr

# Introduction

# Looking for meaning: what are we talking about? What is the meaning?

- ♣ Définition in a dictionary:
  - ♦ what is meant by a word, text, concept, or action.
- ♣ For an application:
  - ♦ The same, but restricted to the concepts needed for the application

Looking for meaning: what are we talking about? What is the meaning?



# Semantic analysis in linguistic technology

- ♣ Information extraction
  - ♦ Analyzing texts to obtain information for a specific application:
    - Example: Named entity recognition
- ♣ Natural language understanding
- ♣ Semantic textual similarity
- ♣ *Etc.*

# Semantic analysis in linguistic technology

## Named Entity recognition

### ❖ Named entity recognition:

- ♦ Seeks to locate and classify **named entities** (names of people/places/ organisations, dates) mentioned in unstructured text into pre-defined categories such as **person, organizations, locations, time, ....**

# Semantic analysis in linguistic technology

## Named Entity recognition

### ❖ Exemple :

On the invitation of the Festival de Cannes, the Italian actress Monica Belucci has agreed to play the role of Mistress of the Opening and Closing Ceremonies of the 70th festival de Cannes to be held from 17 to 28 May 2017, under the presidency of Spanish filmmaker Pedro Almodovar. [...] Monica Bellucci's friendship with the Festival de Cannes goes back a long way: in 2000, she walked up the steps for the first time to present *Under Suspicion* by Stephen Hopkins.

# Semantic analysis in linguistic technology

## Named Entity recognition

❖ Exemple :

On the invitation of the **Festival de Cannes**, the Italian actress **Monica Belucci** has agreed to play the role of Mistress of the Opening and Closing Ceremonies of the 70th festival de **Cannes** to be held from 17 to 28 **May 2017**, under the presidency of Spanish filmmaker **Pedro Almodovar**. [...] **Monica Bellucci**'s friendship with the **Festival de Cannes** goes back a long way: in **2000**, she walked up the steps for the first time to present *Under Suspicion* by **Stephen Hopkins**.

PERSON, ORGANIZATION, LOCATION, DATE

# Semantic analysis in linguistic technology

## Natural language understanding

- ❖ Natural language understanding system (NLU):
  - ♦ Can be considered as a system that translates a sequence of words to one or more actions:
    1. Associate the sequence of words in the input of the system to intermediate semantic language often called **concepts**.
      - ♦ A concept is a class of words dealing with the same subject and sharing common properties.
        - For example, the words hotel and room can all correspond to the concept of “accommodation” in a tourism application.
    2. Translate the concepts obtained into actions or responses during a sentence interpretation step to respond to the entry request.

### ❖ Example:

Hyp	I	want	to	book	a	room
Concept		command			number	object
Label	command-B	command-I	command-I	command-I	number-B	object-B
Valeur		Booking			I	room

# Semantic analysis in linguistic technology

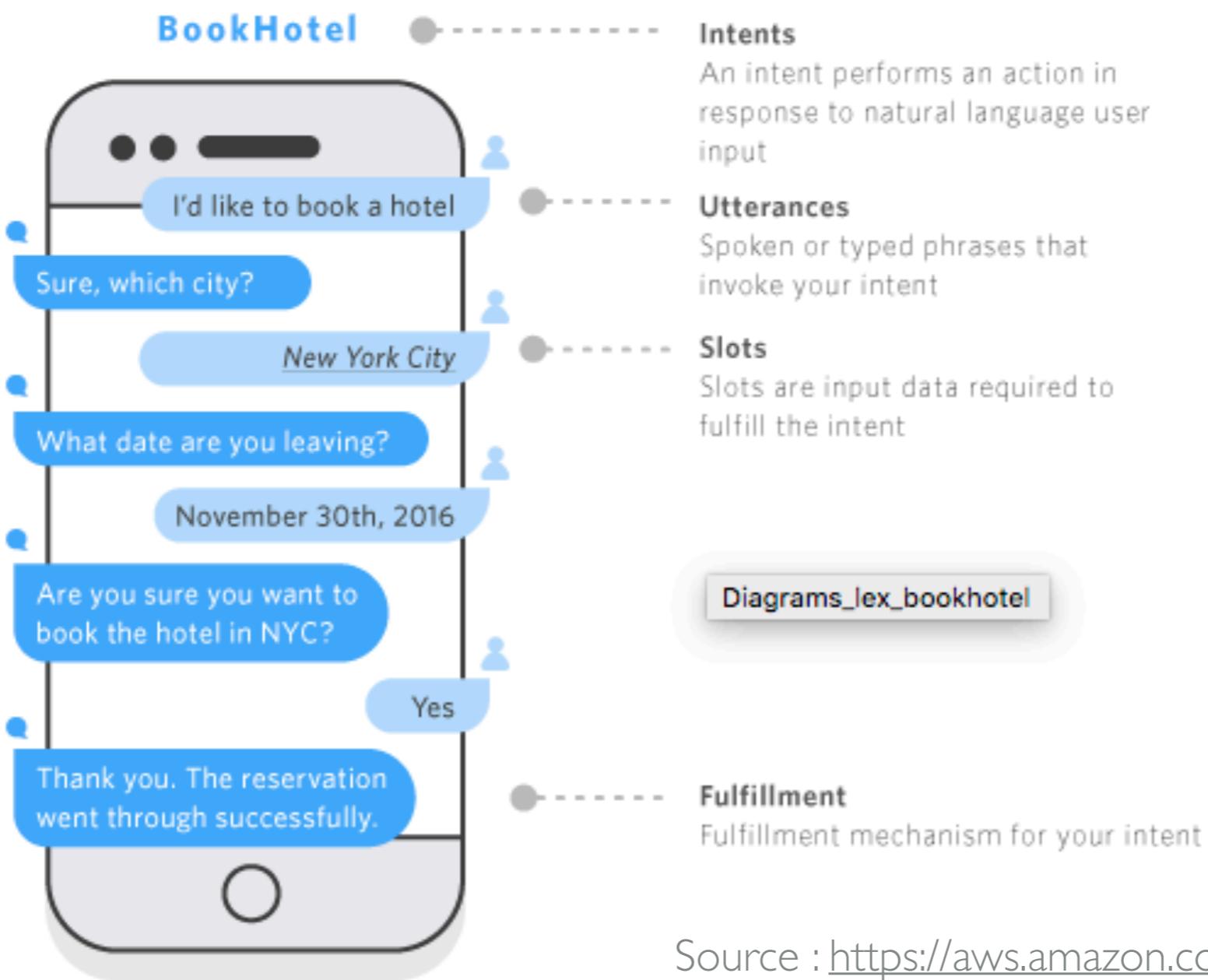
## Natural language understanding

- ♣ NLU is an important module in a dialogue system
  - ♦ Provides direct human-machine interaction
  - ♦ Allows the computer to understand human languages

# Semantic analysis in linguistic technology

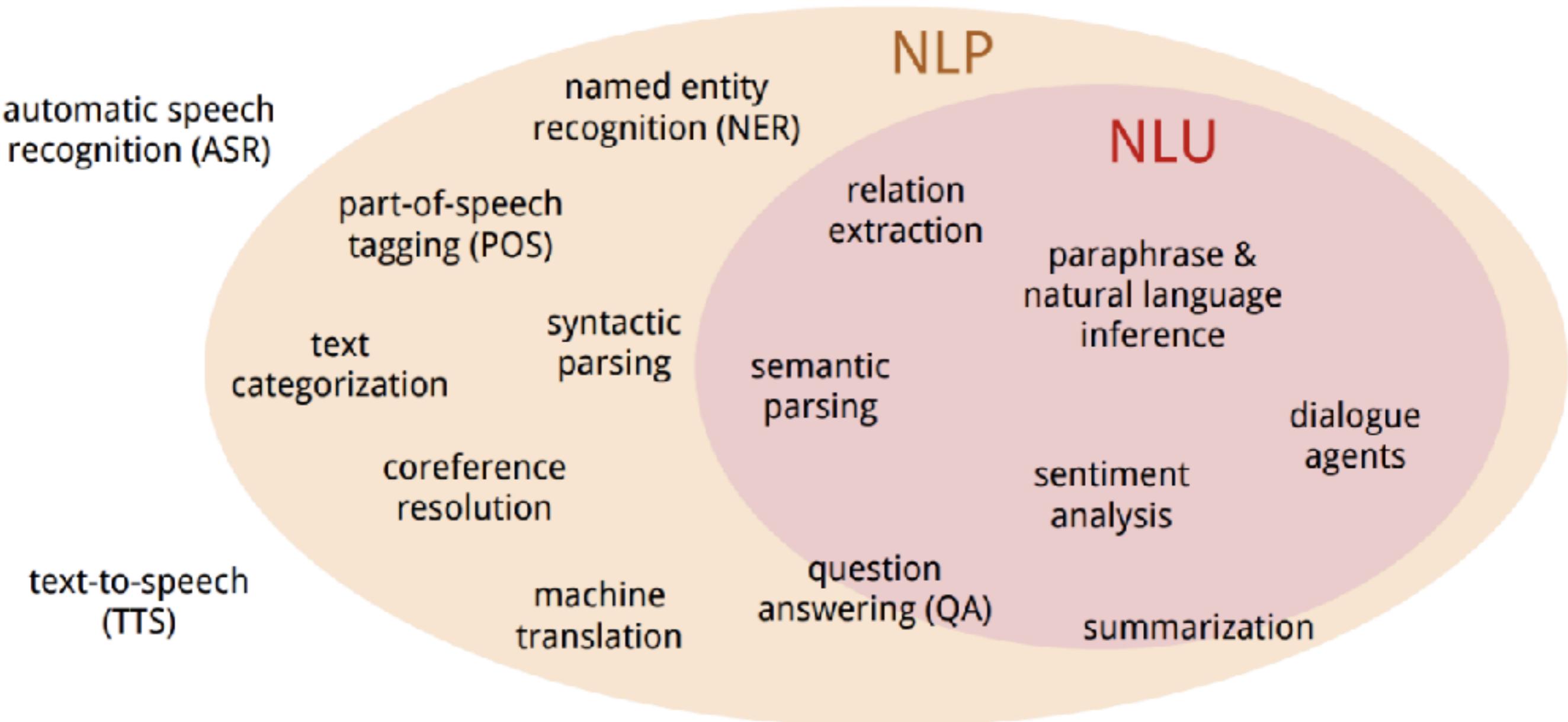
## Natural language understanding

### ❖ Example : Amazon Lex



Source : <https://aws.amazon.com/fr/lex/details/>

# Terminology: NLU vs. NLP vs. ASR



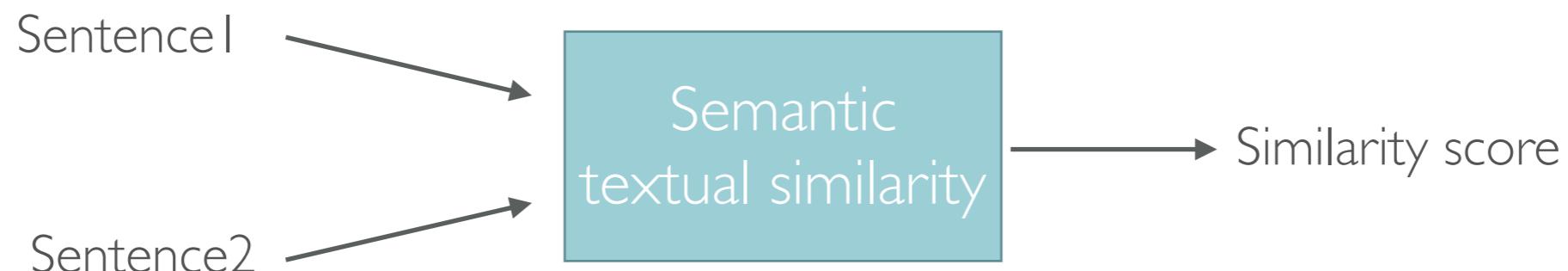
# Semantic analysis in linguistic technology

## Semantic textual similarity

- ❖ SemEval 2017 task 5

### Exemple de paire de phrases

Three men are playing chess.	Two men are playing chess.
A man is playing the cello.	A man seated is playing the cello.
Some men are fighting.	Two men are fighting.



- Evaluate how these sentences are close on a scale of 1 to 5

# Recent approaches

❖ Recent approaches for :

- ◆ Information extraction
  - Named entity recognition
- ◆ Natural language understanding
- ◆ Semantic textual similarity
- ◆ *Etc.*

→ Are based on **deep learning approaches**

# Deep Learning

# Deep Learning

- ❖ Deep learning: part of a broader family of machine learning methods based on artificial neural networks with representation learning. Learning can be supervised, semi-supervised or unsupervised
- ❖ Deep learning architectures: Deep Neural Network(DNN), recurrent neural networks, convolutional neural networks, transformers, ...

# Machine learning

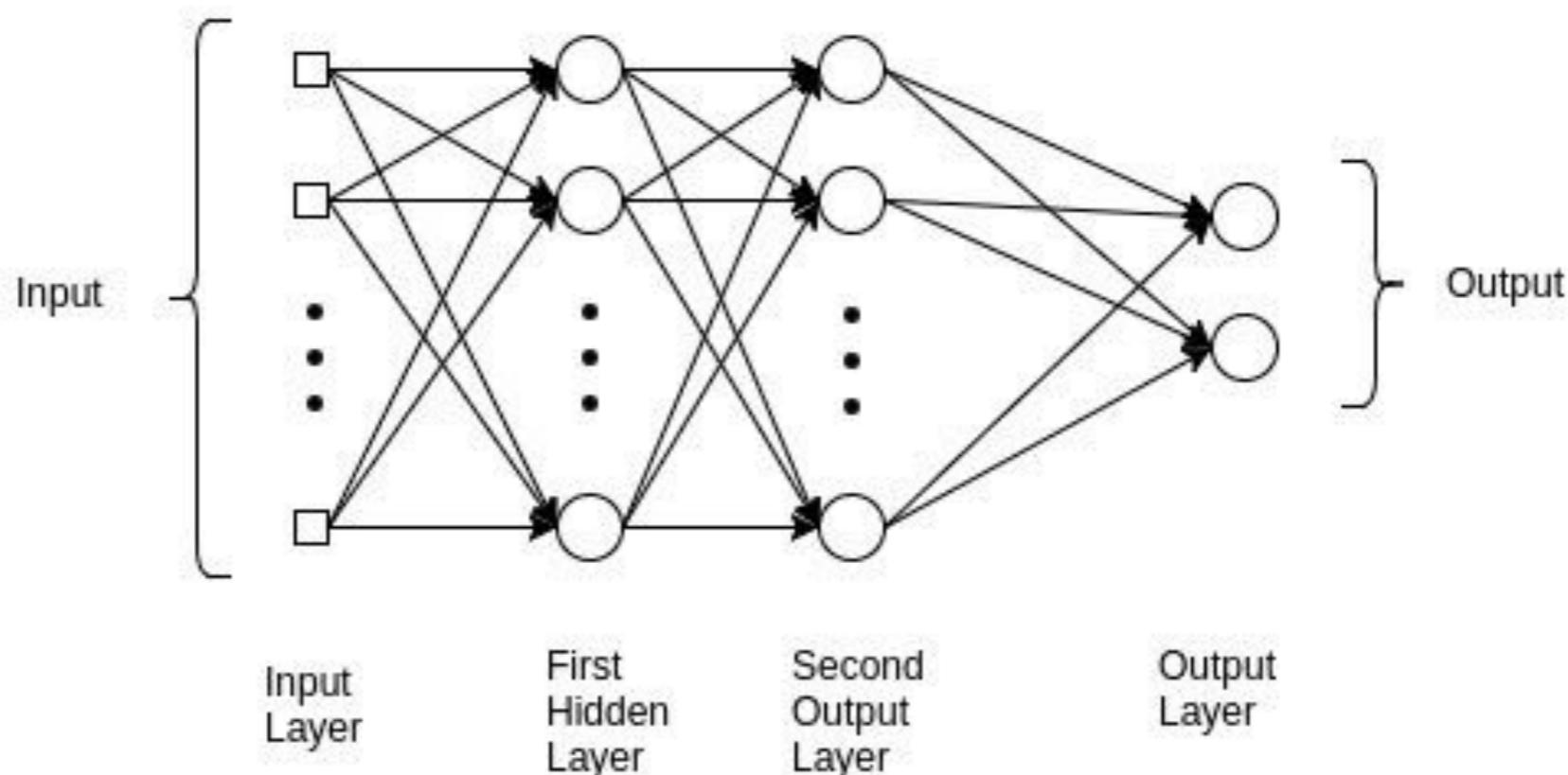
## ♣ Machine learning:

- ♦ The study of computer algorithms that improve automatically through experience
- ♦ Based on statistical approaches to give the systems the ability to “learn” from data, in order to make predictions or decisions without being explicitly programmed to do so
- ♦ It is seen as a subset of **artificial intelligence** (IA)

# Neural network architectures

## ❖ Multilayer perceptron

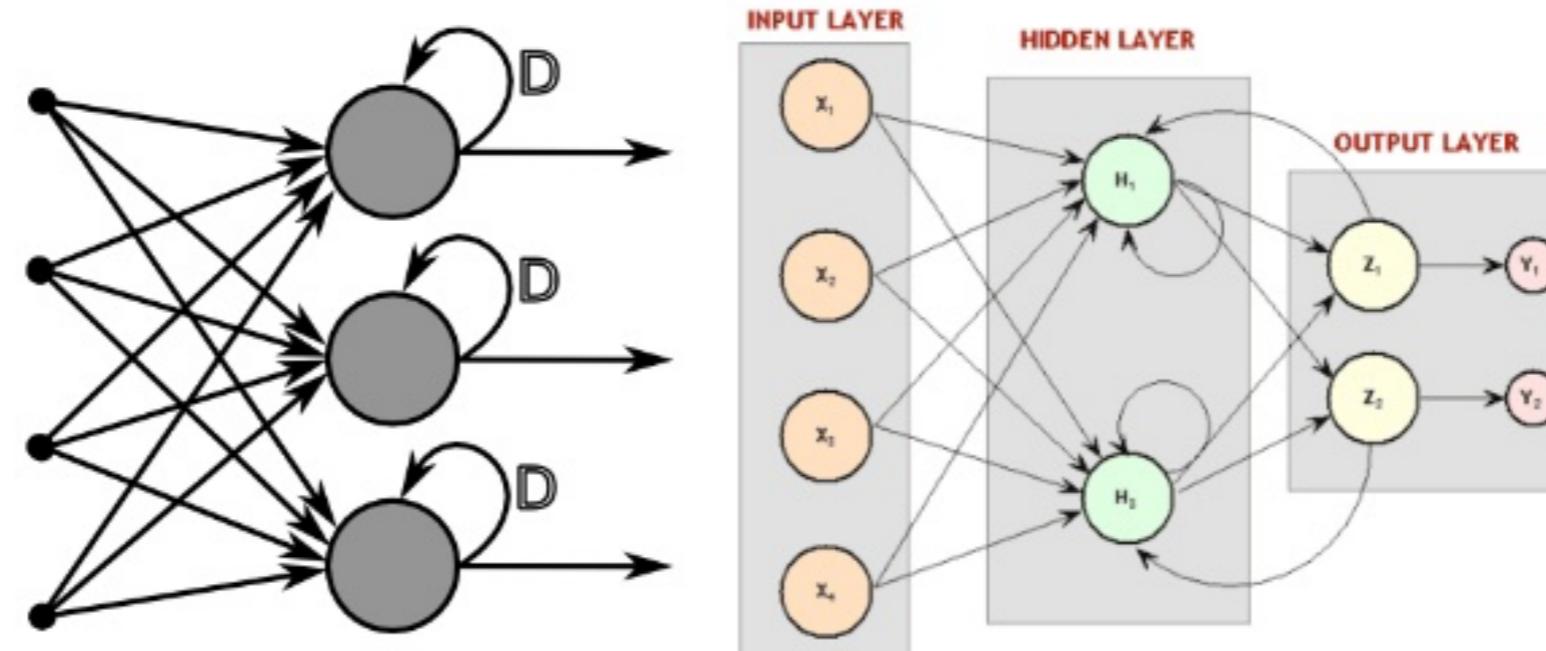
- ♦ Is a class of feedforward artificial neural network, wherein connections between the nodes do *not* form a cycle



# Neural network architectures

## ❖ Recurrent neural network (RNN)

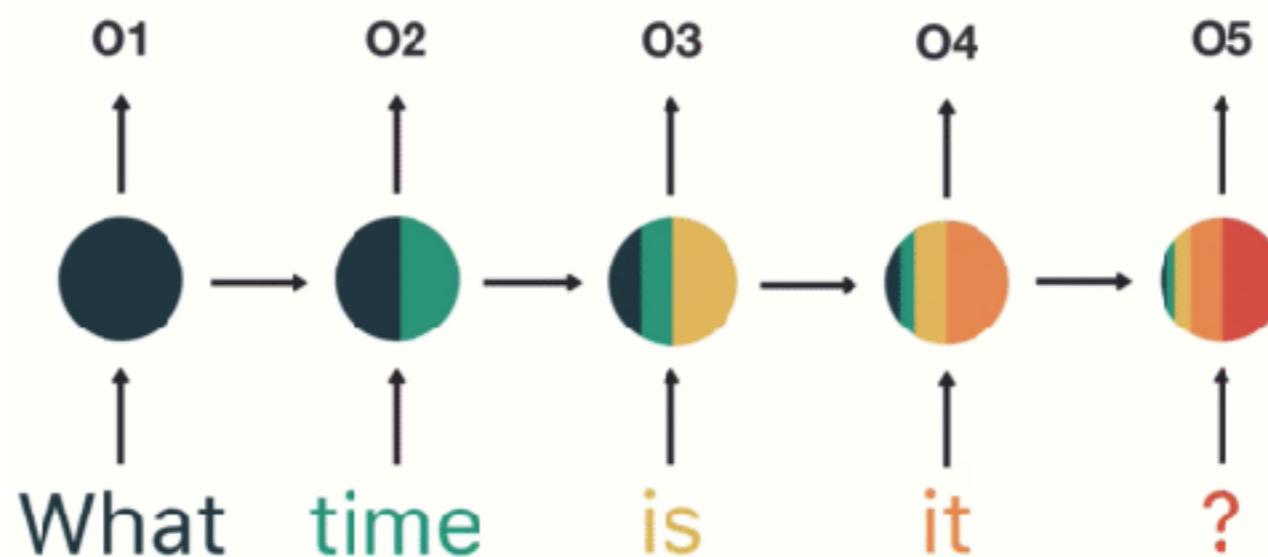
- ♦ Is a type of neural network that contains loops, allowing information to be stored within the network
- ♦ Use their reasoning from previous experiences to inform the upcoming events
- During training, gradients may explode (tend to infinity) or vanish (tend to zero) because of temporal depth



# Neural network architectures

## ❖ Recurrent neural network (RNN)

- ♦ Is a type of neural network that contains loops, allowing information to be stored within the network
- ♦ Use their reasoning from previous experiences to inform the upcoming events
- During training, gradients may explode (tend to infinity) or vanish (tend to zero) because of temporal depth

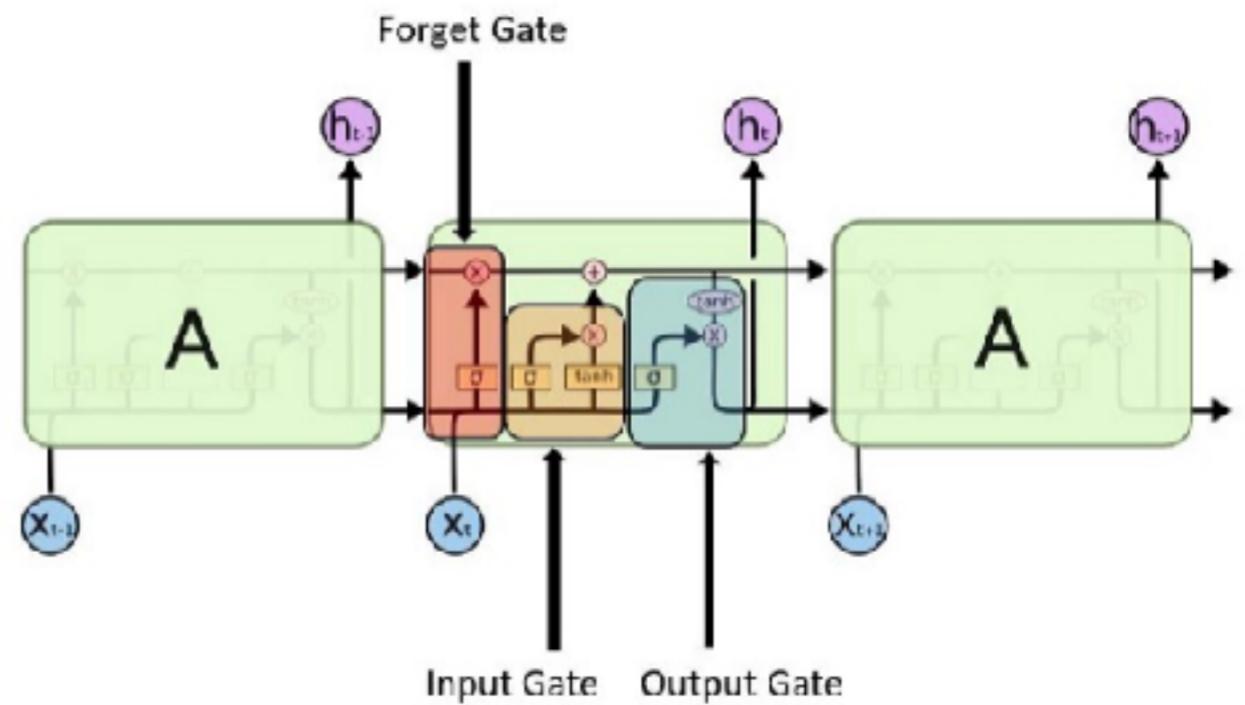


Source: Michael Nguyen / Learned Vector

# Neural network architectures

## ❖ LSTM (log short term memory)

- ◆ is a special kind of RNN's, capable of learning long-term dependencies.
- ◆ Solve vanishing gradient problem
- ◆ LSTM's have skills to remember the information for a long periods of time.
- ◆ LSTMs' core component is the memory cell
  - can maintain its state over time, consisting of an explicit memory and gating units.
  - Gating units regulate the information flow into and out of the memory.

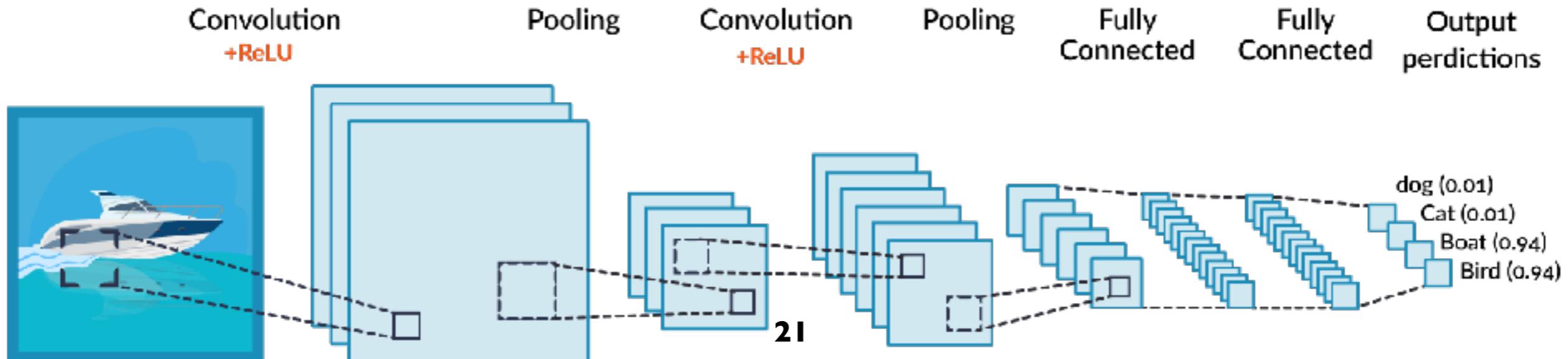


LSTM module has 3 gates named as Forget gate, Input gate, Output gate.

# Neural network architectures

## ❖ Convolution neural network

- ◆ is the foundation of most computer vision technologies.
- ◆ uses two basic operations:
  - convolution using multiple filters is able to extract features (feature map) from the data set, through which their corresponding spatial information can be preserved.
  - pooling, also called subsampling, is used to reduce the dimensionality of feature maps from the convolution operation.
    - max pooling and average pooling are the most common pooling operations used in the CNN.



# Neural network architectures

## ♣ Transformers

- ♦ Is based entirely on **attention mechanism**, replacing the recurrent layers most commonly used in encoder-decoder architectures with **multi-headed self-attention**
- ♦ The main characteristics are:
  - Non sequential: sentences are processed as a whole rather than word by word.
  - Self Attention: this is the newly introduced 'unit' used to compute similarity scores between words in a sentence.
  - Positional embeddings: another innovation introduced to replace recurrence. The idea is to use fixed or learned weights which encode information related to a specific position of a token in a sentence.

# Neural network architectures

## ❖ Transformers

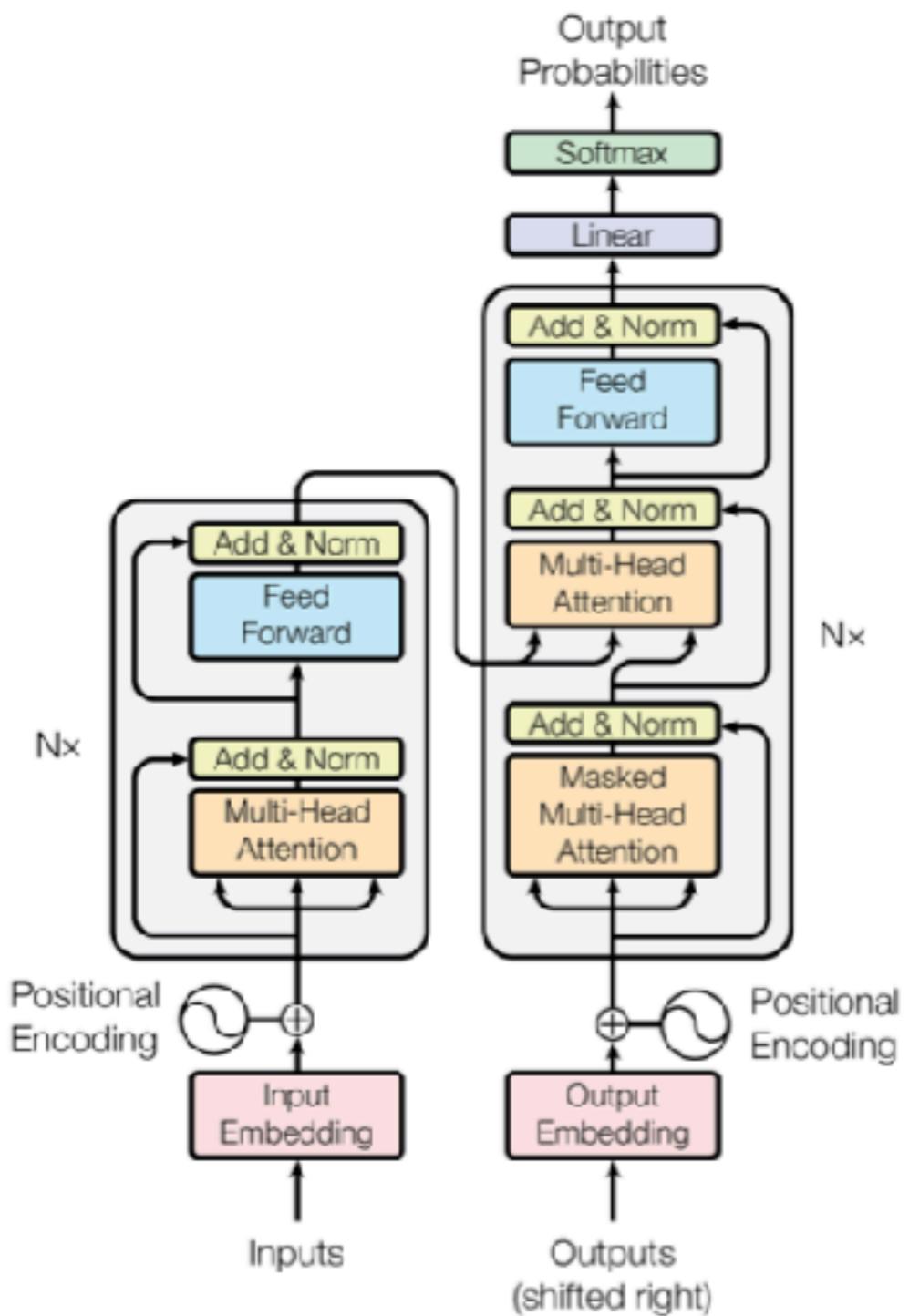


Figure 1: The Transformer - model architecture.

Attention Is All You Need

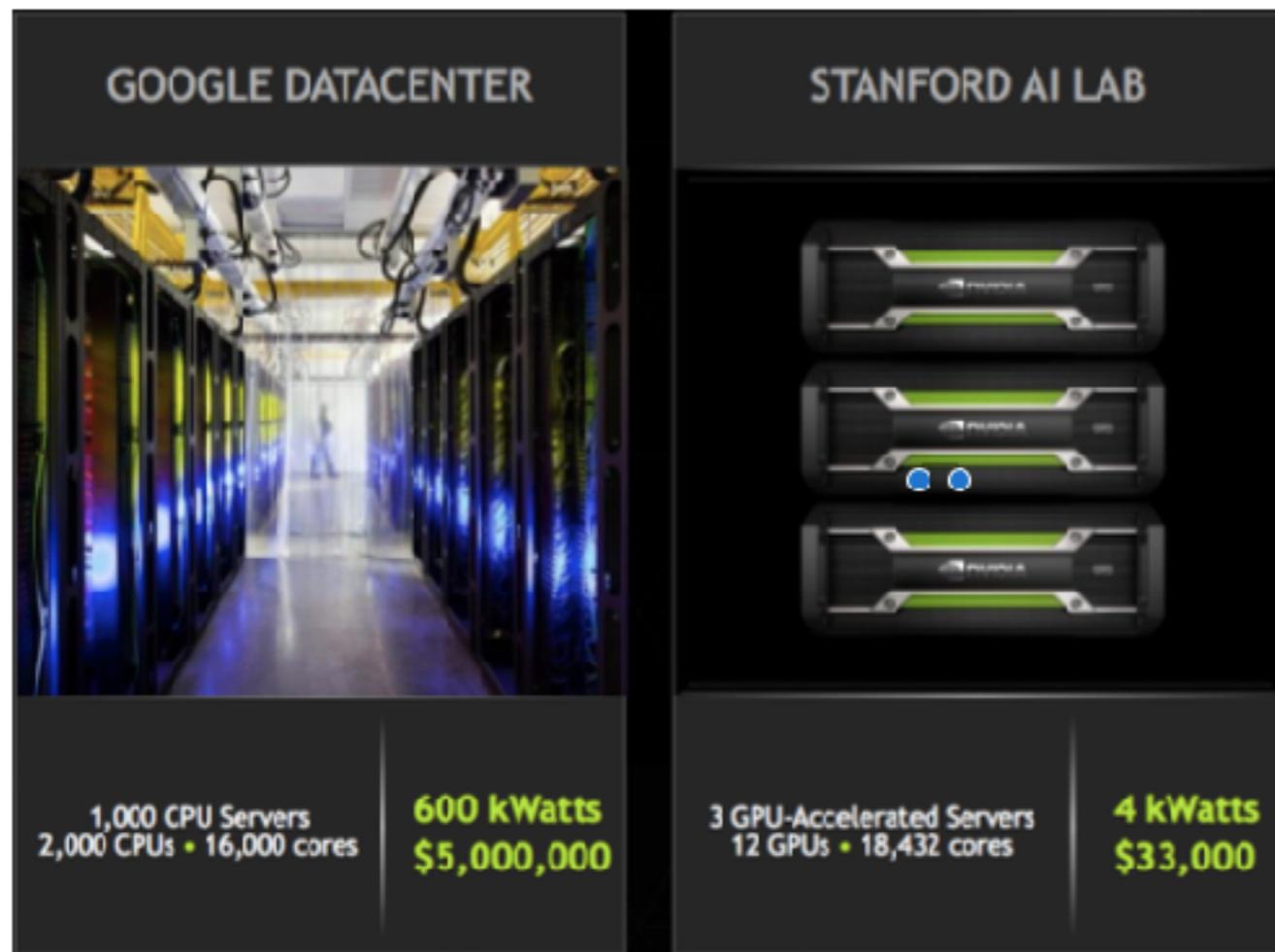
Why DNNs Work Better Now?

# DNN and computing power

- ♣ Why did DNNs not work so well before, when neural networks have been experimented since the 1980s for acoustic modeling?
- ♣ A first element of response:
  - ♦ DNNs are particularly well suited to graphics cards (GPUs) that achieve phenomenal computing power ...

# DNN and computing power

- ❖ ... at a very competitive price!
  - This makes accessible reasonable training time and therefore feasible experiments.



# DNN and data

- ❖ Availability of large amount of data

- ❖ i.e. text, images, audio published via news sites, social media, collaborative platforms, smartphones, etc.



# DNN and learning algorithms

- ♣ A third answer comes from the progress made by researchers for training DNNs:
  - ♦ best learning algorithms (pre-training RBM, SDAE, learning rate dynamique,...)
  - ♦ New architectures (transformers, attention mechanism, LSTM, GRU...)
    - ➡ use continuous representations able to encode different types of hidden relations (syntactic, semantic, contextual,..)

# Continuous word representations

# WORD REPRESENTATIONS

## I. One hot:

- ◆ Example : Merci ID=3, vocabulary=10 words

*Merci* : 

0	0	1	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---

- ◆ Difficulty:
  - can not detect the relationships between words
  - Vector size depends on vocabulary size

# WORD REPRESENTATIONS

## 2. Continuous word representations:

- ◆ Capture relationships between words
- ◆ Different approaches have been studied in the NLP (Natural language processing) community:
  - Clustering
  - distributional word representations
  - Distributed word representations (neural) (known by: word embeddings, continuous word representations)

# CONTINUOUS WORD REPRESENTATIONS

# CONTINUOUS WORD REPRESENTATIONS

- ❖ Clustering (Brown *et al.* 1992) :

- ✓ Grouping words into clusters (group) based on their contexts (**bigrams**)
- ✓ **Disadvantage:** does not consider the use of words in a larger context

# CONTINUOUS WORD REPRESENTATIONS

- ❖ Clustering (Brown et al. 1992) :

- ✓ Grouping words into clusters (group) based on their contexts (**bigrams**)
- ✓ **Disadvantage:** does not consider the use of words in a larger context

- ❖ Distributional word representations exp. PMI (Pointwise Mutual Information):

- ✓ Known as *Count based models* ou *global matrix factorization*
- ✓ Use of word co-occurrence matrix
- ✓ The word is represented by a vector in which each entry is a measure of association between the word and a particular context
- ✓ **Disadvantage:** very high dimensional **sparse** vector (most elements are zero) (same vocabulary size)

# CONTINUOUS WORD REPRESENTATIONS

- ❖ Clustering (Brown et al. 1992) :

- ✓ Grouping words into clusters (group) based on their contexts (**bigrams**)
  - ✓ **Disadvantage:** does not consider the use of words in a larger context

- ❖ Distributional word representations exp. PMI (Pointwise Mutual Information):

- ✓ Known as *Count based models* ou *global matrix factorization*
  - ✓ Use of word co-occurrence matrix
  - ✓ The word is represented by a vector in which each entry is a measure of association between the word and a particular context
  - ✓ **Disadvantage:** very high dimensional **sparse** vector (most elements are zero) (same vocabulary size)

- ❖ Distributed word representations (word embeddings) :

- ✓ Low dimensional **dense** vector with real values

# CONTINUOUS WORD REPRESENTATIONS

- ❖ Clustering (Brown et al. 1992) :

- ✓ Grouping words into clusters (group) based on their contexts (**bigrams**)
- ✓ **Disadvantage:** does not consider the use of words in a larger context

- ❖ Distributional word representations exp. PMI (Pointwise Mutual Information):

- ✓ Known as *Count based models* ou *global matrix factorization*
- ✓ Use of word co-occurrence matrix
- ✓ The word is represented by a vector in which each entry is a measure of association between the word and a particular context
- ✓ **Disadvantage:** very high dimensional **sparse** vector (most elements are zero) (same vocabulary size)

- ❖ Distributed word representations (word embeddings) :

- ✓ Low dimensional **dense** vector with real values

# CONTINUOUS WORD REPRESENTATIONS

# CONTINUOUS WORD REPRESENTATIONS

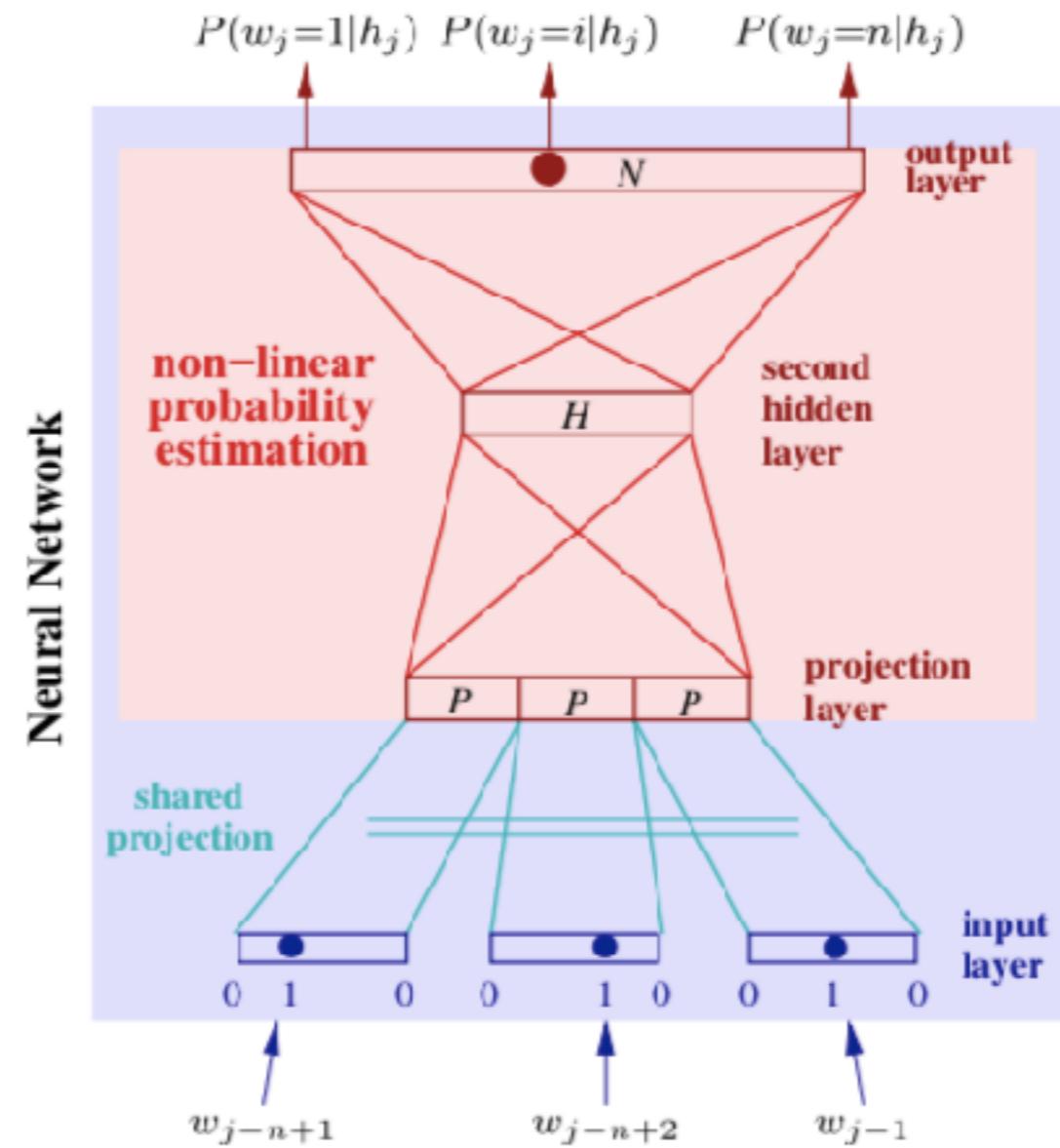
- \* Why dense vectors?:

- ♦ Short vectors can be easily used in machine learning (less weight to optimize)
- ♦ Dense vectors can generalize better
- ♦ etc.

# CONTINUOUS WORD REPRESENTATIONS

Word embeddings :

- Introduced through neural language models training [Y.Bengio et al. 2003, H.Schwenk et al. 2006]



# CONTINUOUS WORD REPRESENTATIONS

Word embeddings :

- ❖ Introduced through neural language models training [Y.Bengio et al. 2003, H.Schwenk et al. 2006]
- ❖ it involves a mathematical embedding from a space with many dimensions per word to a continuous vector space with a much lower dimension, so as to preserve **semantic, syntactic similarities**, etc

$$R : Words = \{W_1, \dots, W_n\} \rightarrow Vectors = \{R(W_1), \dots, R(W_n)\} \subset R^d$$

# CONTINUOUS WORD REPRESENTATIONS

Word embeddings :

- ❖ Introduced through neural language models training [Y.Bengio et al. 2003, H.Schwenk et al. 2006]
- ❖ it involves a mathematical embedding from a space with many dimensions per word to a continuous vector space with a much lower dimension, so as to preserve **semantic, syntactic similarities**, etc

$$R : Words = \{W_1, \dots, W_n\} \rightarrow Vectors = \{R(W_1), \dots, R(W_n)\} \subset R^d$$

- ❖ If the word vectors are close to each other in terms of distance, the words must be semantically or syntactically close.

$$R(W_1) \approx R(W_n) \rightarrow W_1 \approx W_n$$

# CONTINUOUS WORD REPRESENTATIONS

Word embeddings :

- ❖ Introduced through neural language models training [Y.Bengio et al. 2003, H.Schwenk et al. 2006]
- ❖ it involves a mathematical embedding from a space with many dimensions per word to a continuous vector space with a much lower dimension, so as to preserve **semantic, syntactic similarities**, etc

$$R : Words = \{W_1, \dots, W_n\} \rightarrow Vectors = \{R(W_1), \dots, R(W_n)\} \subset R^d$$

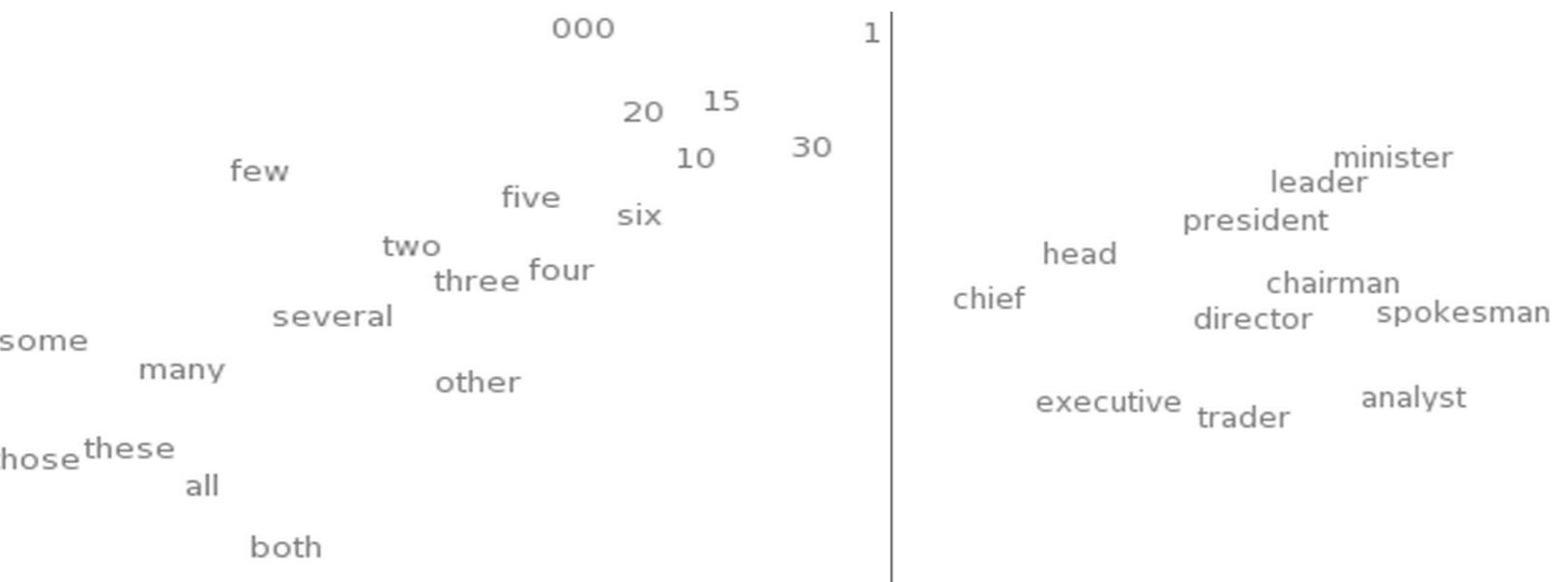
- ❖ If the word vectors are close to each other in terms of distance, the words must be semantically or syntactically close.

$$R(W_1) \approx R(W_n) \rightarrow W_1 \approx W_n$$

- ❖ Each dimension represents a latent characteristic of the word, which can capture syntactic and semantic properties.

# CONTINUOUS WORD REPRESENTATIONS

Word embeddings visualisation:



2D t-SNE visualizations of word embeddings. Left:  
Number Region; Right: Jobs Region [J.Turian et al . 2010]

# CONTINUOUS WORD REPRESENTATIONS

Word embeddings visualisation:



2D t-SNE visualizations of word embeddings. Left:  
Number Region; Right: Jobs Region [J.Turian et al . 2010]

What words have embeddings closest to a given word? [R.Collobert et al . 2011]

# CONTINUOUS WORD REPRESENTATIONS

Word embeddings :

- ❖ Efficient for many tasks [R. Collobert et al 2011], [M. Bansal 2014] et [J. Turian et al 2010]:
  - ◆ Named entity recognition
  - ◆ Part of speech tagging
  - ◆ Natural language understanding
  - ◆ Question answering
  - ◆ etc.

# CONTINUOUS WORD REPRESENTATIONS

## RECENT APPROACHES

### ❖ Context-independent (static) word embeddings:

- The occurrences of the same word have the same representation
- Skipgram, CBOW, Glove, w2vf-deps, fasttext

### ❖ Contextual word embeddings:

- Each occurrence of the word has a different representation calculated based on its context
- Internal representations of words are a function of the entire input sentence
- Exemple : ELMo, BERT, GPT-2



# CONTINUOUS WORD REPRESENTATIONS CONTEXT-INDEPENDENT

Skip-gram (Mikolov et al. 2013a) and CBOW (Mikolov et al. 2013a)

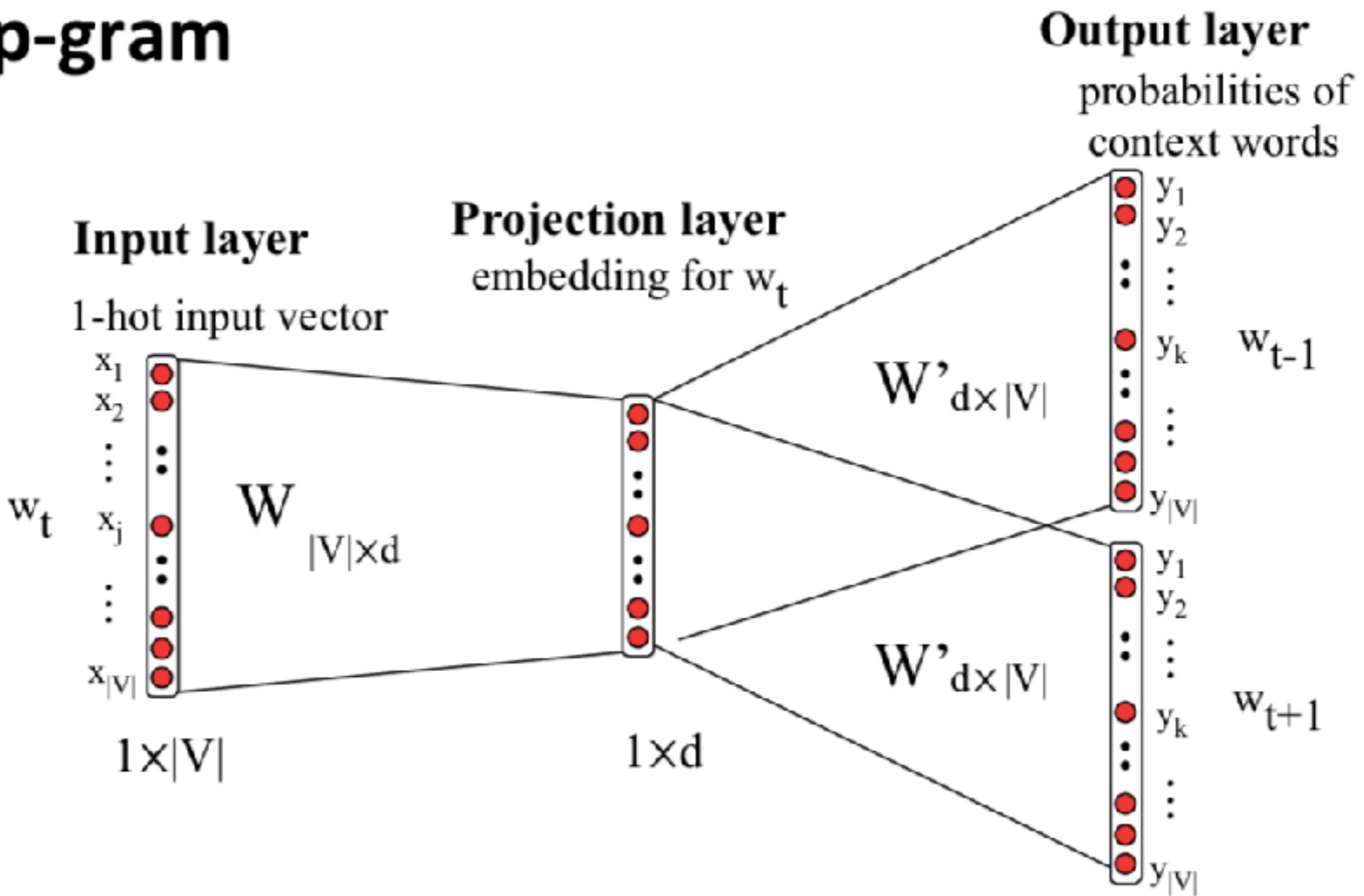
- ❖ Learn word embeddings as part of the word prediction process.
  - ◆ Inspired by neural language models.



# CONTINUOUS WORD REPRESENTATIONS

## CONTEXT-INDEPENDENT

## Skip-gram



# CONTINUOUS WORD REPRESENTATIONS

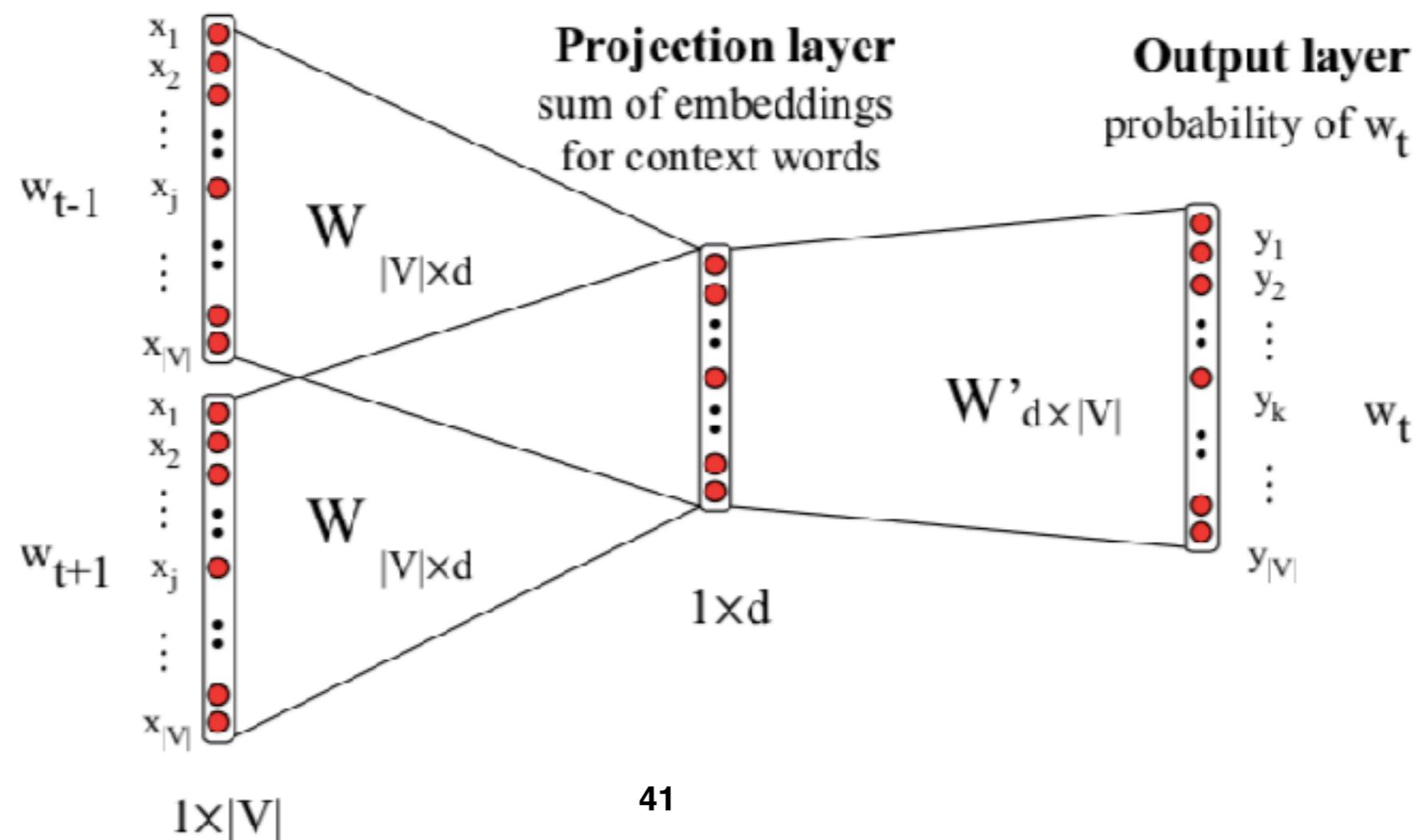
## CONTEXT-INDEPENDENT



## CBOW (Continuous Bag of Words)

### Input layer

1-hot input vectors  
for each context word

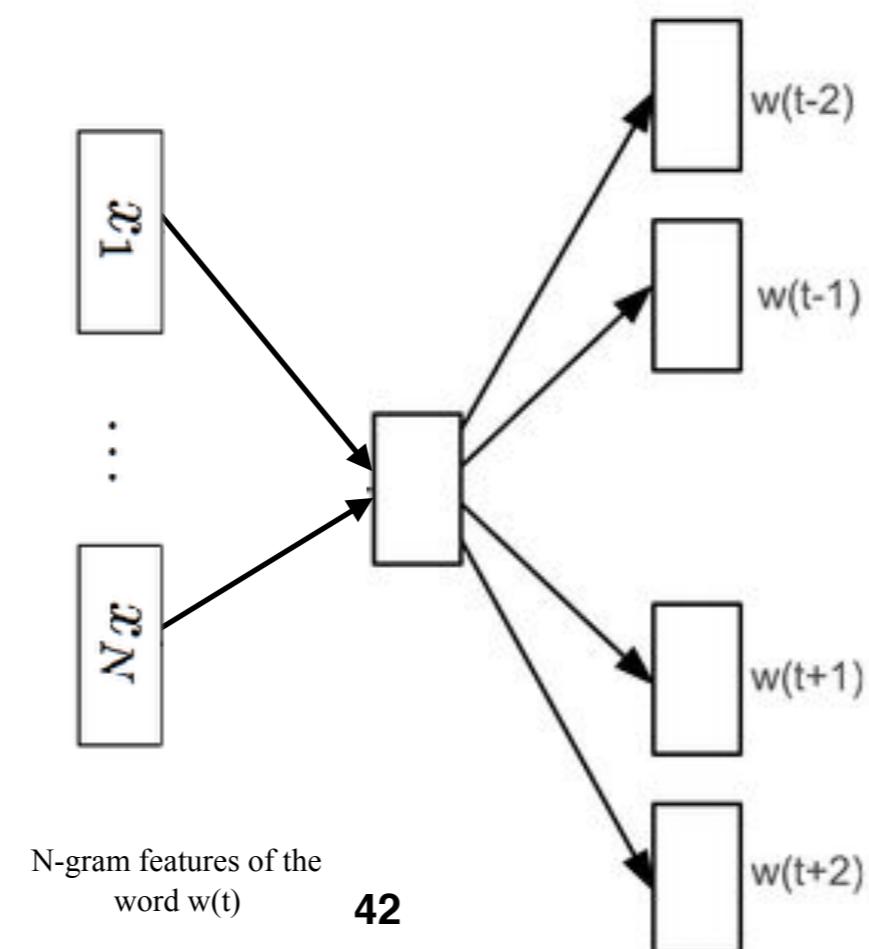


# CONTINUOUS WORD REPRESENTATIONS

## CONTEXT-INDEPENDENT

FastText (P. Bojanowski et al. 2017)

- ❖ Learn word embeddings as part of the word prediction process based on the skipgram architecture
    - ◆ Takes into account the morphological information of a word represented in the form of character ngrams
    - ◆ Each word is represented as the sum of representations of its characters n grams.
    - ◆ Computes word representations for words that do not appear in the training data, which is not the case for other approaches (word2vec)

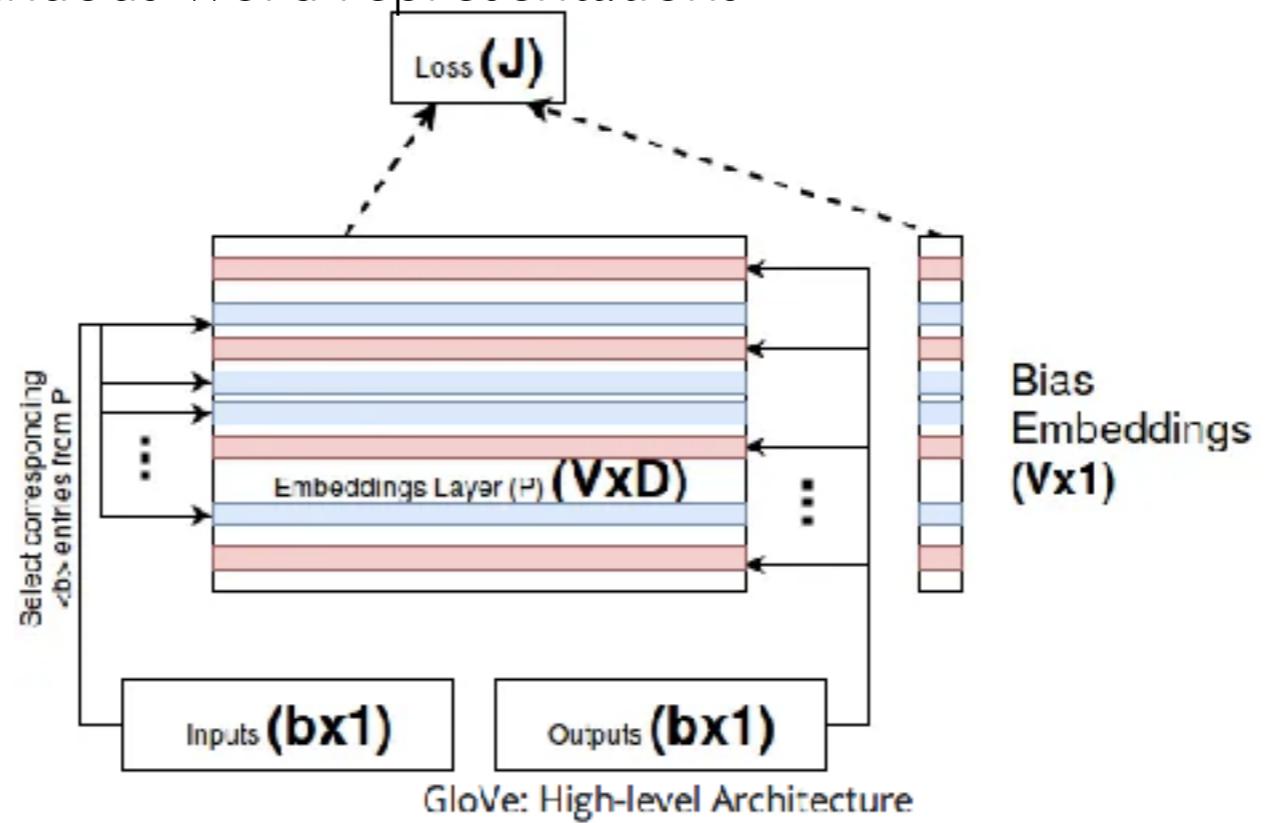


# CONTINUOUS WORD REPRESENTATIONS

## CONTEXT-INDEPENDENT

Glove (J. Pennington et al 2014)

- ❖ Count based approach
  - ❖ Based on the analysis of the co-occurrences of words in the corpus
    - ◆ Co-occurrence matrix construction: use of sliding window
    - ◆ Estimation of continuous word representations

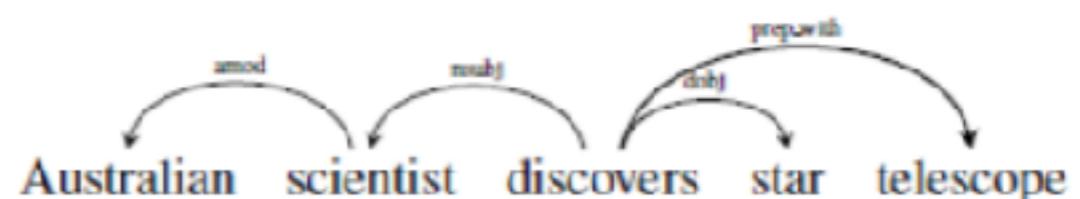


# CONTINUOUS WORD REPRESENTATIONS

## CONTEXT-INDEPENDENT

w2vf-deps : dependency based word embeddings (O. Levy et al. 2014)

- ❖ Generalization of the skip-gram model with negative samples
  - ❖ Use syntactic relationships (root words, dependency between root word and dependent words) to extract words in context
    - ◆ capture relations between distant words: can not be detected in a small context window



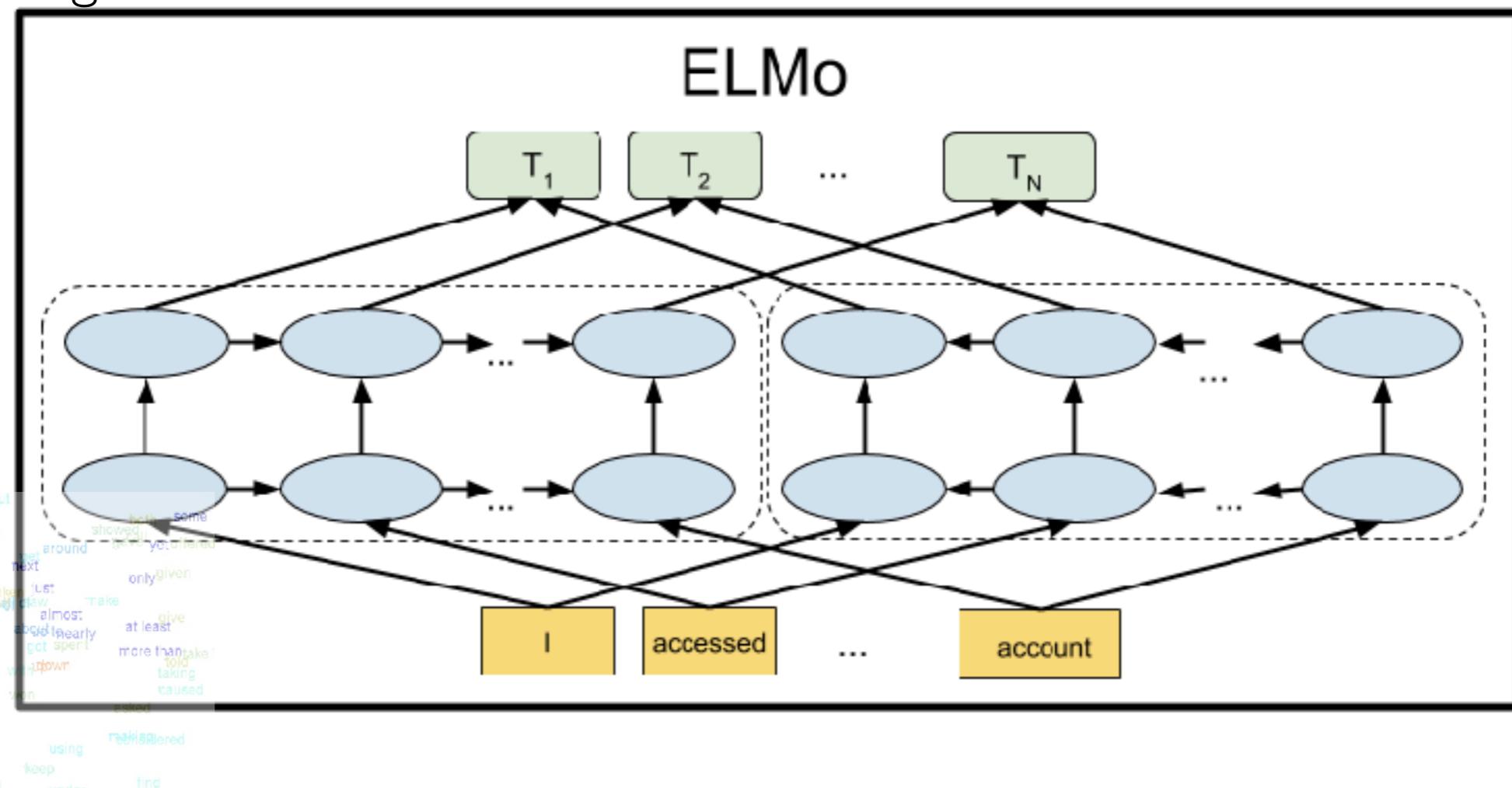
WORD	CONTEXTS
australian	scientist/amod <sup>-1</sup>
scientist	australian/amod, discovers/nsubj <sup>-1</sup>
discovers	scientist/nsubj, star/dobj, telescope/prep_with
star	discovers/dobj <sup>-1</sup>
telescope	discovers/prep_with <sup>-1</sup>

# CONTINUOUS WORD REPRESENTATIONS

## CONTEXTUAL

ELMo (Embeddings from Language Models)(Peters, et al, 2018)

- Learn word embeddings by creating bidirectional language models
- creates contextualized representations of each token by concatenating the internal states of a 2-layer biLSTM trained on a bidirectional language modeling task



# CONTINUOUS WORD REPRESENTATIONS

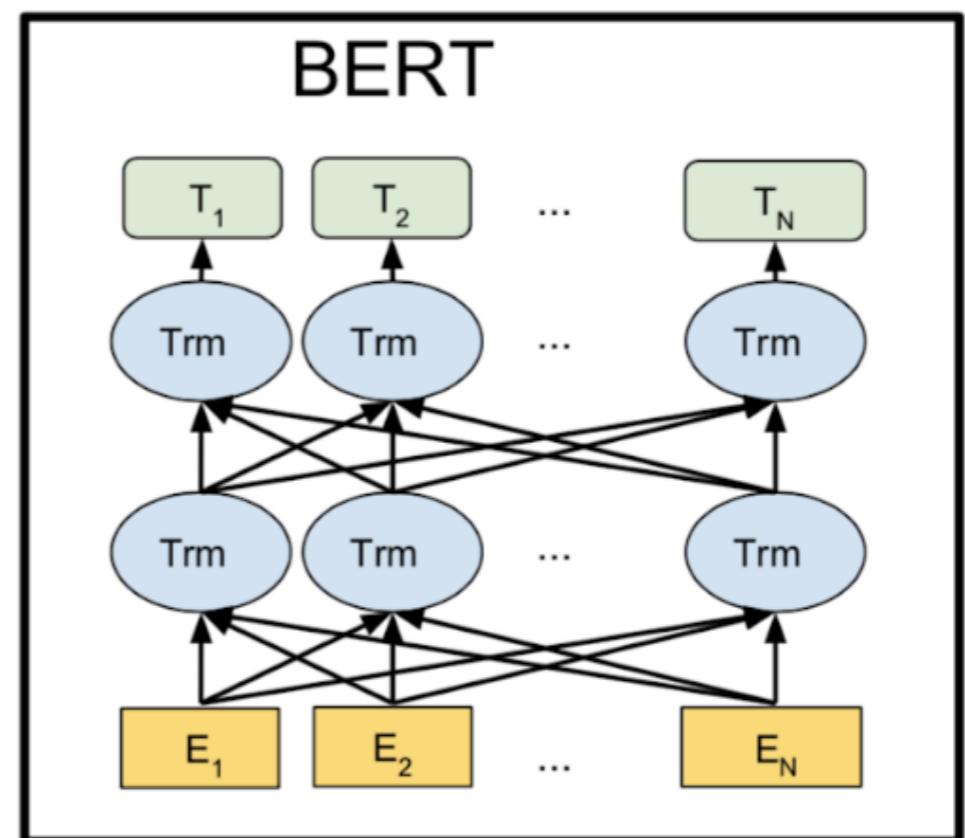
## CONTEXTUAL



BERT (Bidirectional Encoder Representations from Transformers)(Devlin, et al., 2019)

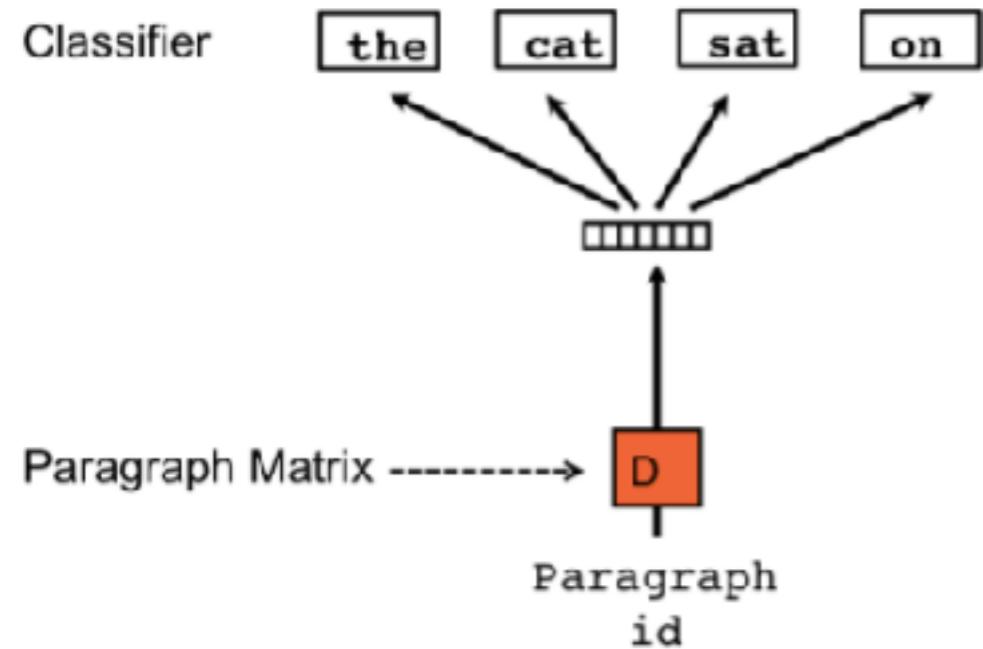
- ❖ It's a bidirectional transformer pre-trained using a combination of masked language modeling objective and next sentence prediction on a large corpus comprising.
- ❖ Each transformer layer of 12- layer creates a contextualized representation of each token by attending to different parts of the input sentence

try  
able  
seeking  
forced  
called  
put  
select  
look  
next  
around  
both  
some  
showed  
travel  
ycoffered  
only  
given  
make  
almost  
about  
nearly  
got  
spent  
down  
with  
won  
more  
at least  
take  
told  
taking  
caused  
asked  
asked  
reconsidered  
reconsidered  
very  
as  
against  
for  
win  
using  
keep  
under  
find



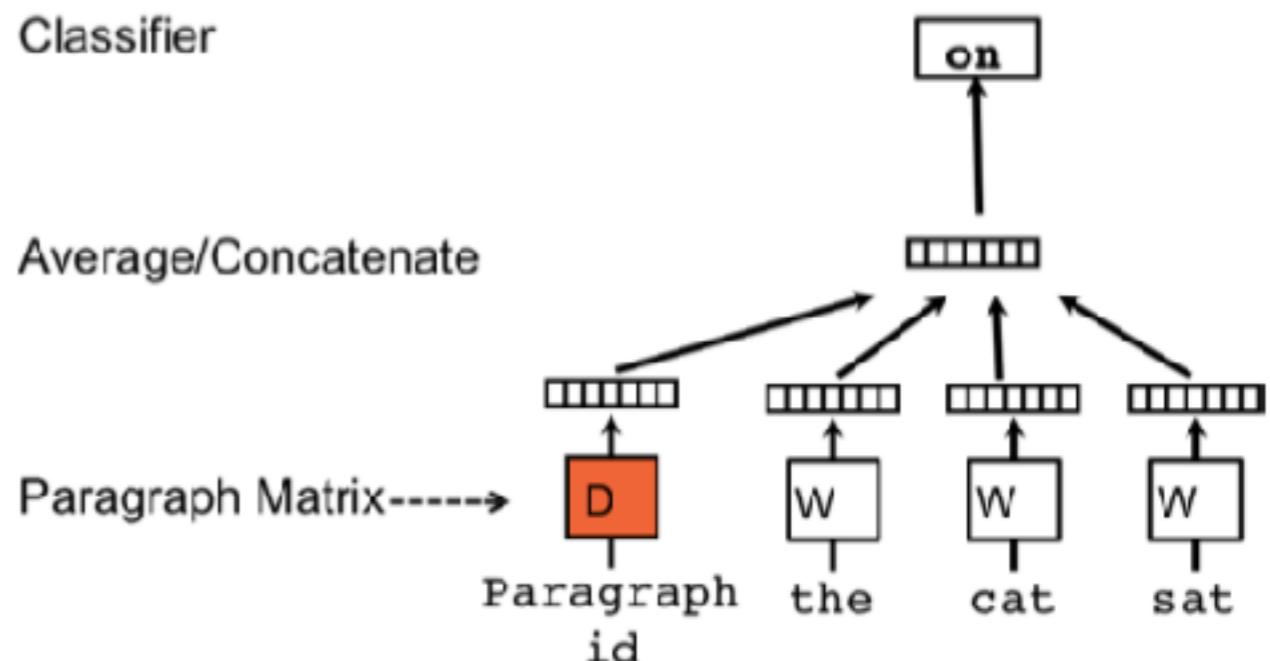
# CONTINUOUS DOCUMENT REPRESENTATIONS

Distributed bag of words



(Mikolov et al. 2013b)

Distributed Memory model



# CONTINUOUS WORD REPRESENTATIONS

## EVALUATION TASKS

### I. Natural language processing tasks (NLP)

- ♦ Part-of-speech tagging (POS): syntactic roles (noun, adverb, etc.)
- ♦ Chunking (CHK): syntactic constituent (noun phrase, verb phrase, etc.)
- ♦ Named Entity recognition (NER): person, company, etc.
- ♦ Mention detection (MENT): begin, inside, and outside
- ♦ Spoken language understanding task (SLU)

# CONTINUOUS WORD REPRESENTATIONS

## EVALUATION TASKS

## 2. Word semantic similarity (Kiela et al., 2015)

# CONTINUOUS WORD REPRESENTATIONS

## EVALUATION TASKS

### 2. Word semantic similarity (Kiela et al., 2015)

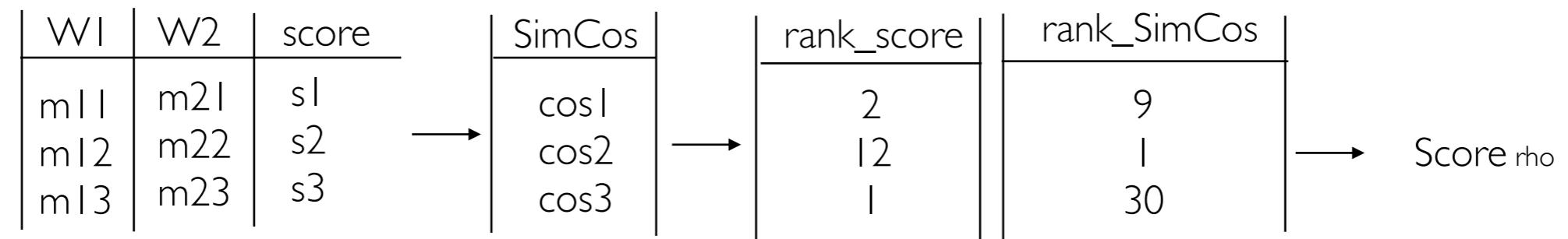
- ♦ Is based on an idea that the distances between words in an embedding space could be evaluated through the human heuristic judgments on the actual semantic distances between these words
  - e.g., the distance between *cup* and *mug* defined in an continuous interval  $\{0, 1\}$  would be 0.8 since these words are synonymous, but not really the same thing.
- ♦ The assessor is given a set of pairs of words and asked to assess the degree of similarity for each pair.
- ♦ The distances between these pairs are also collected in a word embeddings space, and the two obtained distances sets are compared.
- ♦ The more similar they are, the better are embeddings

# CONTINUOUS WORD REPRESENTATIONS

## EVALUATION TASKS

### 2. Word semantic similarity (Kiela et al., 2015)

- ♦ Is based on an idea that the distances between words in an embedding space could be evaluated through the human heuristic judgments on the actual semantic distances between these words
  - e.g., the distance between *cup* and *mug* defined in an continuous interval  $\{0, 1\}$  would be 0.8 since these words are synonymous, but not really the same thing.
- ♦ The assessor is given a set of pairs of words and asked to assess the degree of similarity for each pair.
- ♦ The distances between these pairs are also collected in a word embeddings space, and the two obtained distances sets are compared.
- ♦ The more similar they are, the better are embeddings



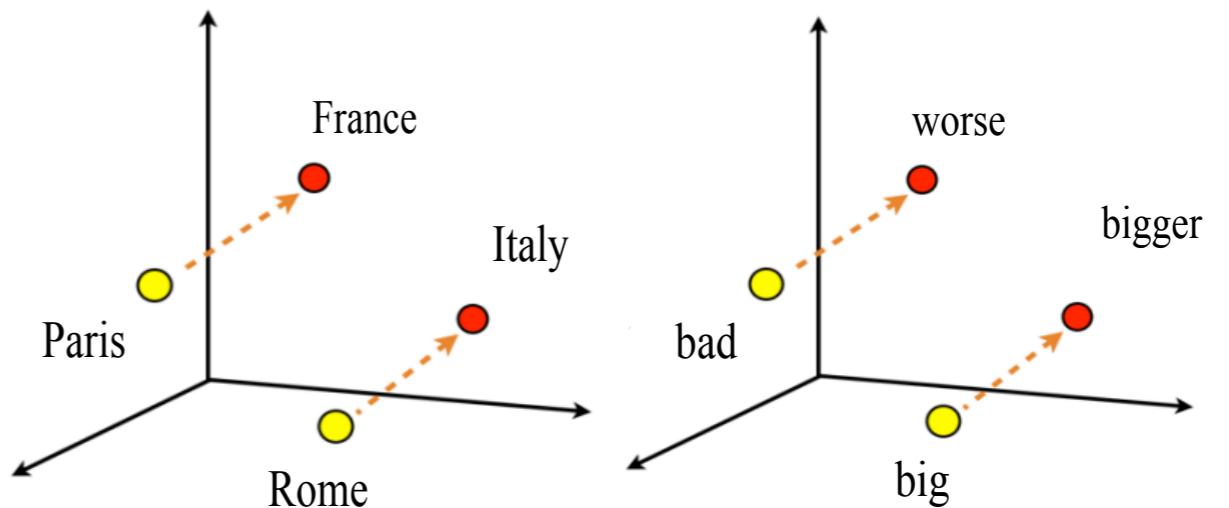
# CONTINUOUS WORD REPRESENTATIONS

## EVALUATION TASKS

- Is based on the idea that arithmetic operations in a word vector space could be predicted by humans:
  - given a set of three words,  $a, a^*$  and  $b$ , the task is to identify such word  $b^*$  that the relation  $b:b^*$  is the same as the relation  $a:a^*$ .
  - For instance, one has words  $a=Paris, a^*=France, b=Rome$ .
  - Then the target word would be  $Italy$  since the relation  $a:a^*$  is  $\text{capital:country}$ , so one needs to find the  $\text{capital}$  of which country is  $Rome$

Semantic: Paris:France → Rome:?

Syntactic: bad:worse → big:?



# CONTINUOUS WORD REPRESENTATIONS

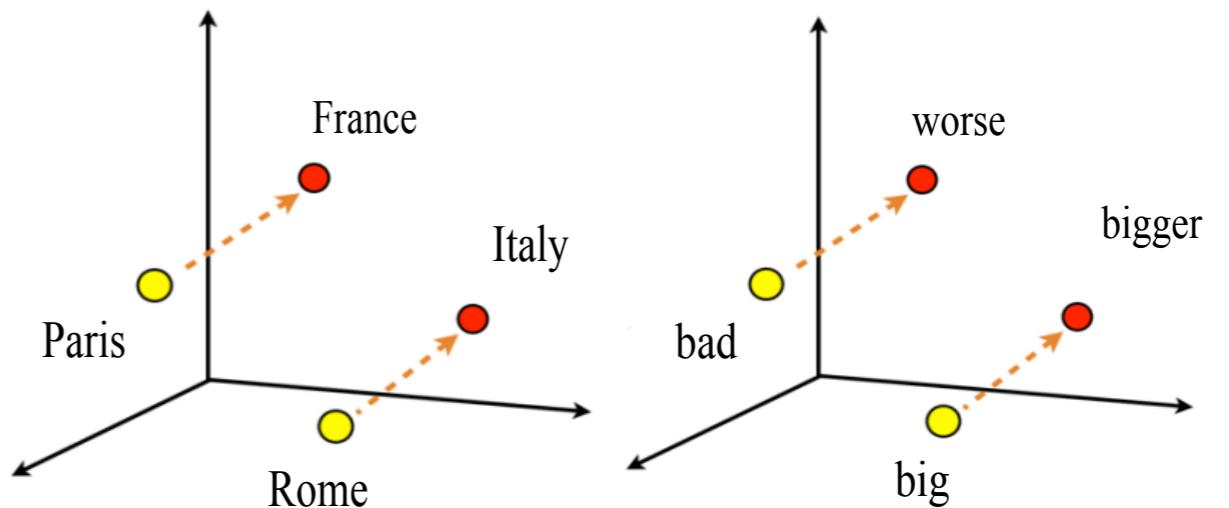
## EVALUATION TASKS

### 3. Word analogy

- Is based on the idea that arithmetic operations in a word vector space could be predicted by humans:
  - given a set of three words,  $a, a^*$  and  $b$ , the task is to identify such word  $b^*$  that the relation  $b:b^*$  is the same as the relation  $a:a^*$ .
  - For instance, one has words  $a=Paris, a^*=France, b=Rome$ .
  - Then the target word would be Italy since the relation  $a:a^*$  is  $\text{capital:country}$ , so one needs to find the  $\text{capital}$  of which country is  $Rome$

Semantic: Paris:France → Rome:?

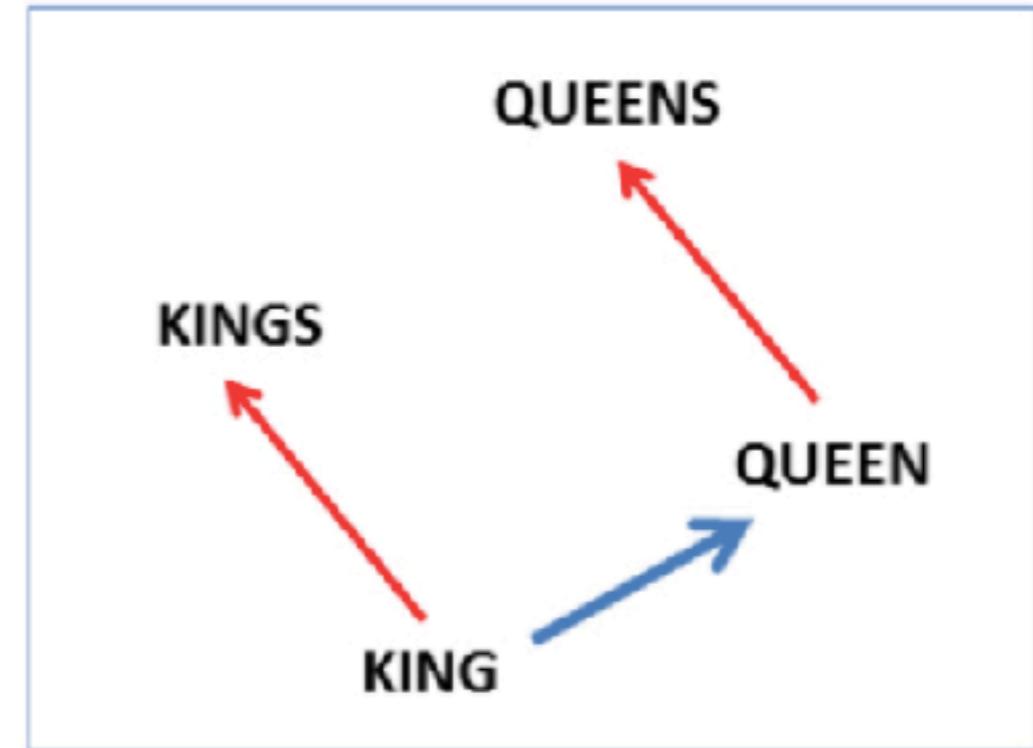
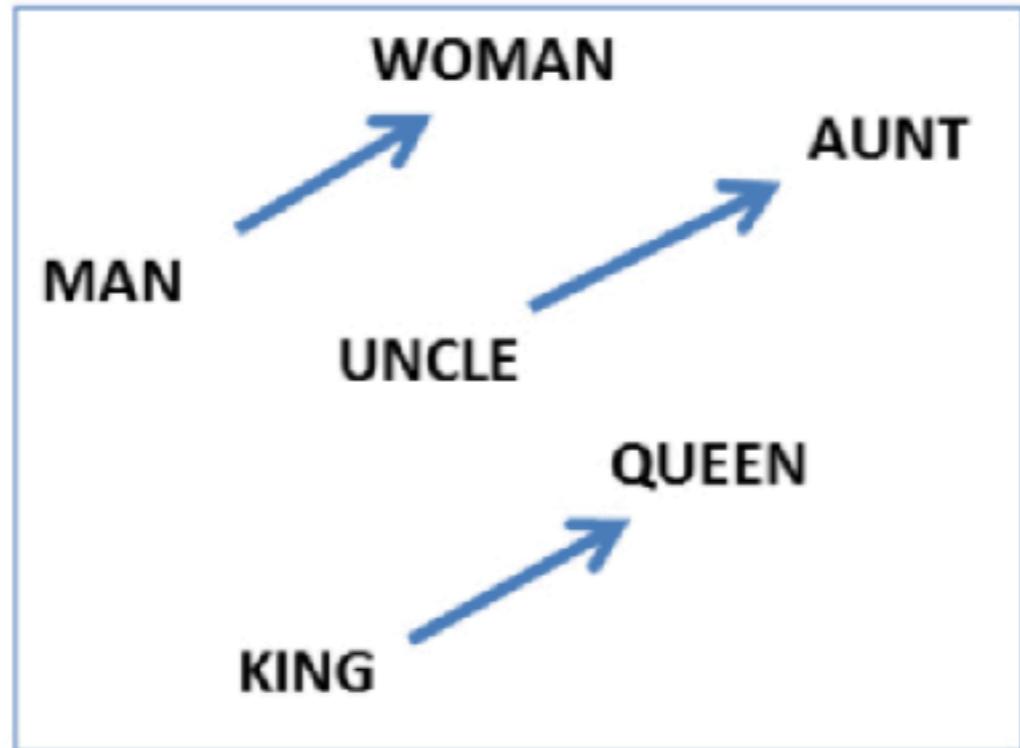
Syntactic: bad:worse → big:?



# CONTINUOUS WORD REPRESENTATIONS

## EVALUATION TASKS

vector('king') - vector('man') + vector('woman')  $\approx$  vector('queen')  
vector('Paris') - vector('France') + vector('Italy')  $\approx$  vector('Rome')



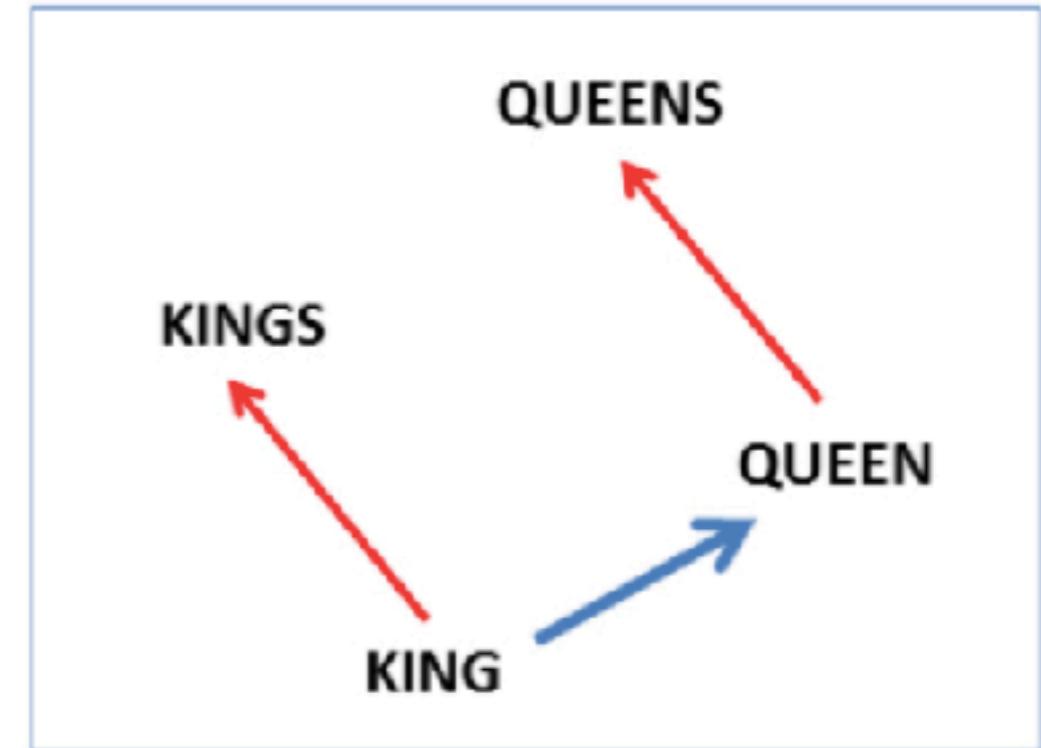
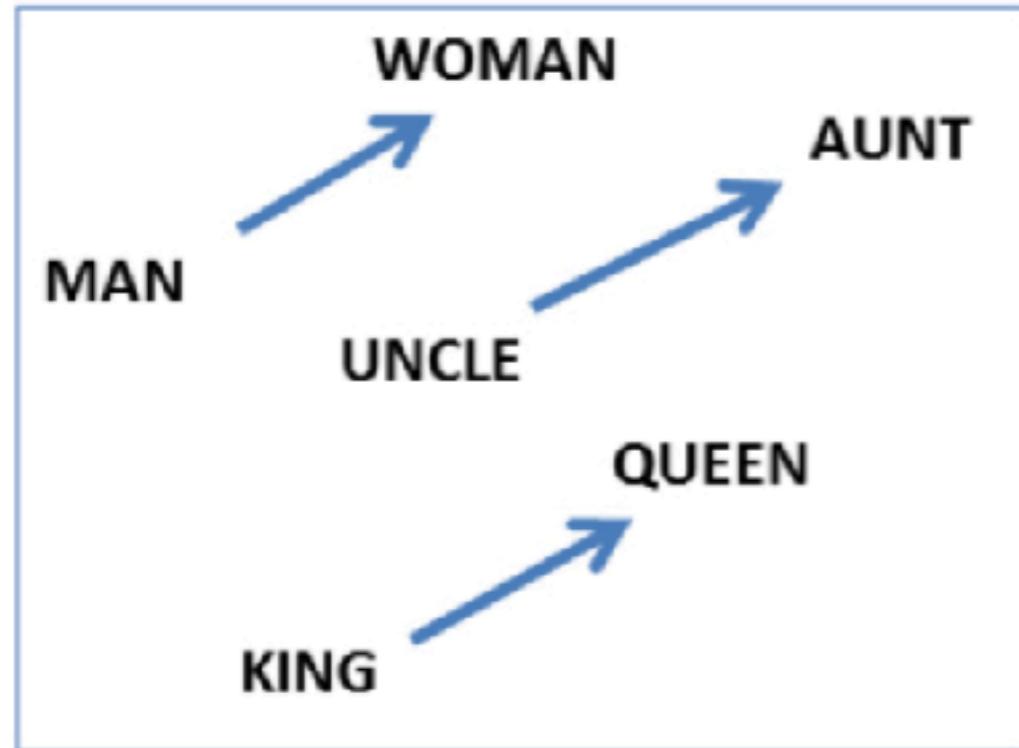
# CONTINUOUS WORD REPRESENTATIONS

## EVALUATION TASKS

### 3. Word analogy

$\text{vector('king')} - \text{vector('man')} + \text{vector('woman')} \approx \text{vector('queen')}$

$\text{vector('Paris')} - \text{vector('France')} + \text{vector('Italy')} \approx \text{vector('Rome')}$



# CONTINUOUS WORD REPRESENTATIONS

## EVALUATION TASKS

ble seeking  
decided planned held issued all scheduled while over  
as against for in inducing inside between win using keep under find  
selected took around next get taken list off day walk eat live at least more than take told taking caused asked making considered

### Evaluation data

- ❖ Word embeddings:
  - ◆ data: corpus Gigaword (english)
  - ◆ Vocabulary: 239k words
  - ◆ Parameters:

Embeddings	Tail.	Dim.	Neg.
CBOW			5
Skip-gram	5		5
GloVe		200	-

w2vf-deps	-		5
-----------	---	--	---

- ❖ Analogy task
  - ◆ Questions:
    - 8869 semantic
    - 10675 syntactic

- ❖ Similarity task:
  - ◆ Data:
    - WordSim353
    - RW (2034)
    - MEN (3000)

- ❖ NLP tasks
  - ◆ Data:

Tâche	Benchmark	Train	Dev	Test
POS	Penn Treebank	958k	34k	58k
CHK	CoNLL 2000	191k	21k	47k
NER	CoNLL 2003	205k	52k	47k
MENT	Ontonotes	736k	102k	105k

# CONTINUOUS WORD REPRESENTATIONS

## EVALUATION TASKS

Embeddings	Tâches TALN				Tâche de similarité			Tâche d'analogie
	POS	CHK	NER	MENT	WS353	RW	MEN	
	Acc.	FI			Spearman's rank rho			
CBOW	96,01	0,904	0,783	0,554	<b>0,590</b>	0,465	0,609	57,20
Skip-gram	96,43	0,896	0,776	0,578	0,558	<b>0,502</b>	<b>0,662</b>	62,30
w2v-deps	<b>96,66</b>	<b>0,92</b>	<b>0,793</b>	<b>0,580</b>	0,523	0,435	0,557	42,70
GloVe	95,79	0,869	0,764	0,544	0,533	0,410	0,660	<b>65,50</b>

[Ghannay, et al., 2016]

# CONTINUOUS WORD REPRESENTATIONS

## EVALUATION TASKS

Embeddings	Tâches TALN				Tâche de similarité			Tâche d'analogie
	POS	CHK	NER	MENT	WS353	RW	MEN	
	Acc.	FI			Spearman's rank rho			
CBOW	96,01	0,904	0,783	0,554	<b>0,590</b>	0,465	0,609	57,20
Skip-gram	96,43	0,896	0,776	0,578	0,558	<b>0,502</b>	<b>0,662</b>	62,30
w2v-deps	<b>96,66</b>	<b>0,92</b>	<b>0,793</b>	<b>0,580</b>	0,523	0,435	0,557	42,70
GloVe	95,79	0,869	0,764	0,544	0,533	0,410	0,660	<b>65,50</b>

[Ghannay, et al., 2016]

→ These embeddings carry different information

# CONTINUOUS WORD REPRESENTATIONS EVALUATION TASKS

## TASK-DEPENDENT DATA VS OUT OF DOMAIN DATA

### QUANTITATIVE EVALUATION

Bench.	task-dependent					Out-of-domain				
	ELMo	FastText	GloVe	Skip-gram	CBOW	ELMo	FastText	GloVe	Skip-gram	CBOW
M2M	88.89	72.13	<b>92.54</b>	88.87	89.39	91.14	93.01	91.77	<b>93.19</b>	92.13
ATIS	<b>94.38</b>	85.72	92.95	90.84	91.87	94.93	<b>95.52</b>	95.35	95.62	<b>95.77</b>
SNIPS	78.68	76.35	<b>87.40</b>	82.10	83.94	90.29	<b>94.85</b>	93.90	94.43	94.05
SNIPS70	53.06	38.19	<b>63.65</b>	47.11	49.76	75.19	79.75	78.68	78.90	<b>80.13</b>
MEDIA	80.26	71.73	<b>82.66</b>	80.01	79.57	<b>86.42</b>	85.30	85.11	85.95	86.06

Tagging performance of different word embeddings trained on task-dependent corpus (ATIS, MEDIA, M2M, SNIPS or SNIPS70) and on huge and out of domain corpus (WIKI English or French) on all benchmark corpora in terms of F1 using conlleval scoring script (in %)

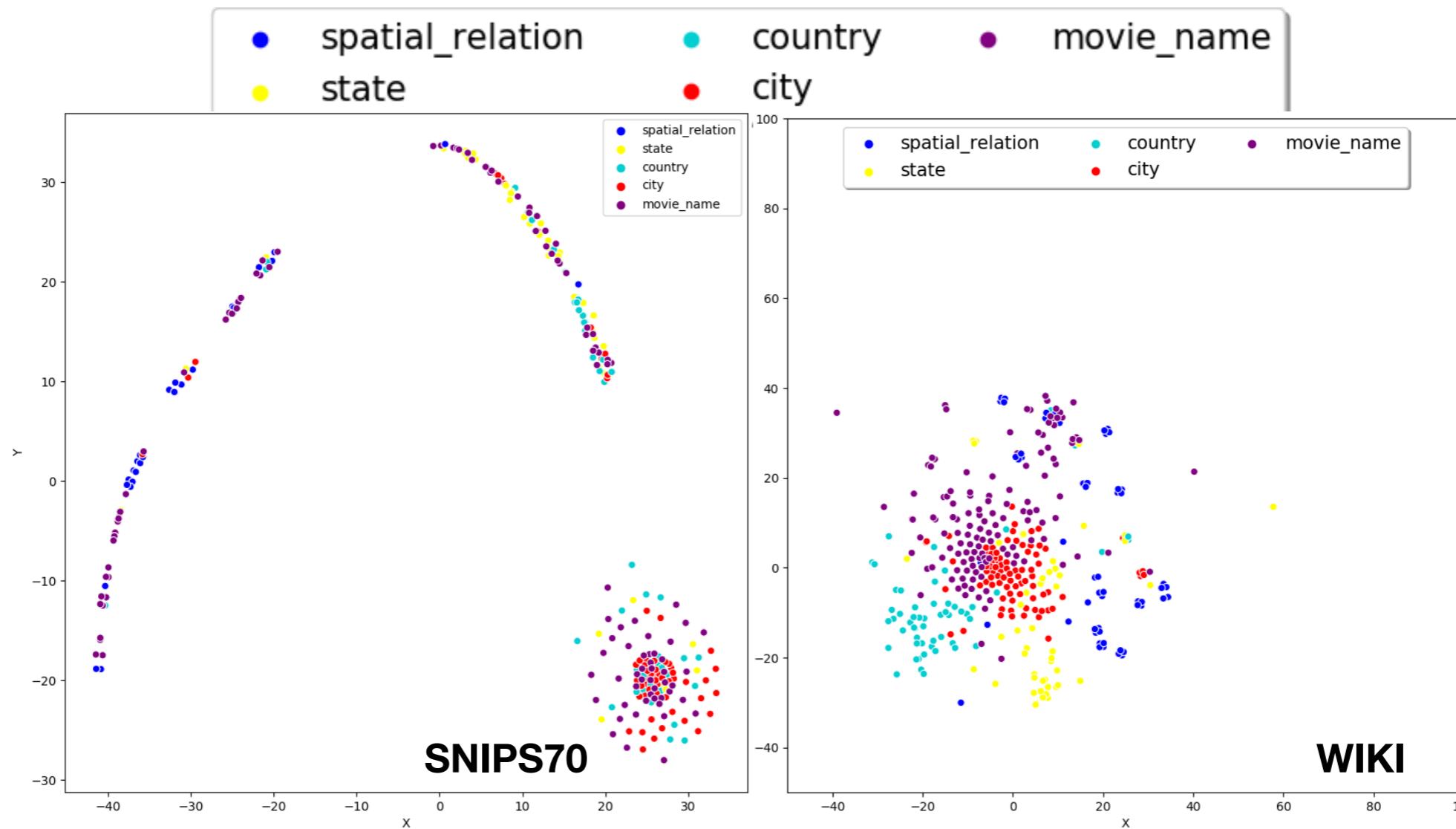
[Ghannay, et al., 2020]

# CONTINUOUS WORD REPRESENTATIONS

## EVALUATION TASKS

# TASK-DEPENDENT DATA VS OUT OF DOMAIN DATA

## Qualitative evaluation: CBOW

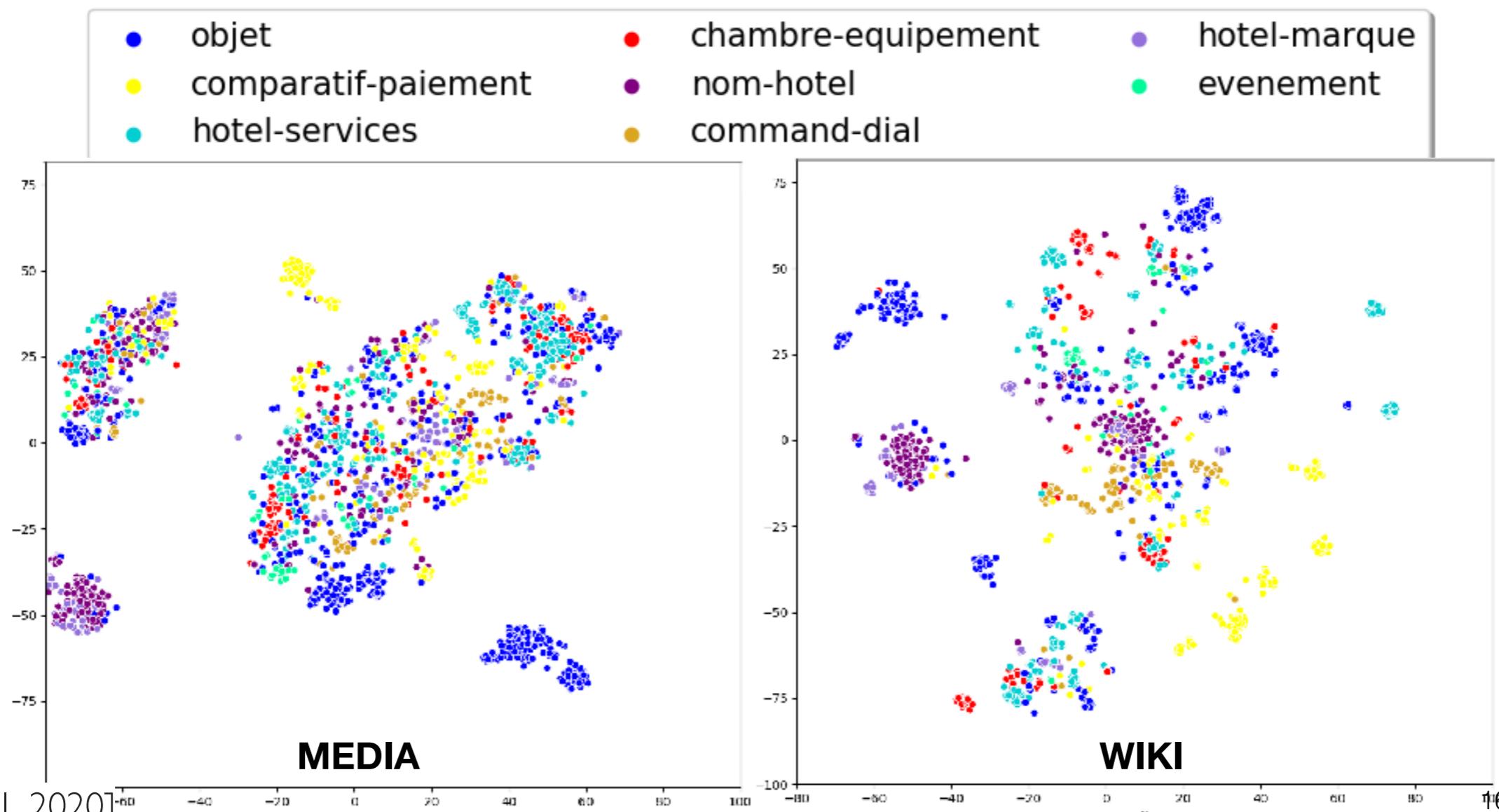


# CONTINUOUS WORD REPRESENTATIONS

## EVALUATION TASKS

# TASK-DEPENDENT DATA VS OUT OF DOMAIN DATA

# Qualitative evaluation: ELMo



# TOOLS

## Tools available to build the embeddings

- ♦ Word2vec Mikolov: <https://github.com/tmikolov/word2vec>
- ♦ W2vf-deps Levy: <https://levyomer.wordpress.com/2014/04/25/dependency-based-word-embeddings/>
- ♦ Glove Pennington: <https://nlp.stanford.edu/projects/glove/>
- ♦ Fasttext: <https://fasttext.cc/docs/en/support.html>
- ♦ ELMo: allenlp [https://github.com/allenai/allennlp/blob/master/tutorials/how\\_to/elmo.md](https://github.com/allenai/allennlp/blob/master/tutorials/how_to/elmo.md)
- ♦ BERT: <https://pypi.org/project/bert-embedding/>
- ♦ Python library: gensim

# Sitography/cknowledgment

- Lectures:
  - Dan Jurafsky: Vector Sementic, Standford university
  - Yannick Estève: Semantics, Avignon université
  - Sophie Rosset, LISN, CNRS :
    - <https://sophierosset.github.io/docs/eidi-dhm.pdf>
    - <https://bigdataspeech.github.io/EN/>
- Tutorials:
  - <https://ahmetozlu93.medium.com/long-short-term-memory-lstm-networks-in-a-nutshell-363cd470ccac>
  - <https://missinglink.ai/guides/convolutional-neural-networks/convolutional-neural-network-tutorial-basic-advanced/>

# Bibliography

- [Mikolov et al., 2013a] Mikolov, T., Chen, K., Corrado, G. et Dean, J. (2013a). Efficient Estimation of Word Representations in Vector Space. In Proceedings of Workshop at ICLR.
- [Levy et Goldberg, 2014] Levy, O. et Goldberg, Y. (2014). Dependencybased word embeddings. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, volume 2, pages 302–308.
- [Mikolov et al., 2013b] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. et Dean, J. (2013b). Distributed representations of words and phrases and their compositionality.
- [Pennington et al., 2014] Pennington, J., Socher, R. et Manning, C. D. (2014). Glove : Global vectors for word representation. In Proceedings of the Empirical Methods in Natural Language Processing (EMNLP 2014), volume 12.
- [Peters, et al, 2018] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. arXiv preprint arXiv:1802.05365.
- [Devlin, et al., 2019] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.
- [Ghannay, et al., 2016] Ghannay, S., Favre, B., Esteve, Y., & Camelin, N. (2016, May). Word embedding evaluation and combination. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)
- [Kiela et al., 2015] Kiela, D., Hill, F. et Clark, S. (2015). Specializing word embeddings for similarity or relatedness. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 2044–2048, Lisbon, Portugal. Association for Computational Linguistics.
- [Mikolov et al., 2013] Mikolov, T., Yih, W.-t. et Zweig, G. 2013. Linguistic regularities in continuous space word representations. In HLT-NAACL, pages 746–751.
- [P. Bojanowski et al. 2017] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov, “Enriching word vectors with subword information,” Transactions of the Association for Computational Linguistics, vol. 5, 2017.
- [Ghannay, et al., 2020] Ghannay, Sahar, Antoine Neuraz, and Sophie Rosset. "What is best for spoken language understanding: small but task-dependant embeddings or huge but out-of-domain embeddings?." ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020

# Practical session

Lectures: [https://saharghannay.github.io/files/Cours\\_Master\\_EN.pdf](https://saharghannay.github.io/files/Cours_Master_EN.pdf)

PS: <https://saharghannay.github.io/courses/cours1/example1/>