

# “IA et langage naturel”

## 1

Introduction

Prof. Stéphane DUPONT

2022-2023

# OUTLINE

- Introduction
  - Historique
  - Catégories d'applications et exemples
  - Elements de linguistique: domaine, structure de phrase, ambiguïtés (lexicales et syntaxiques), contexte
- Algorithmes et outils
  - Descripteurs (sparse, denses, contextuels,...)
  - Architectures des modèles (CNN, RNNs, Transformers, ...)
  - Classification de texte
  - Modèles de langage
- Méthodologie
  - Databases, data augmentation, métriques (y compris NMT)
  - Conclusions et point de vue critique

# 50's – Traduction automatique – Guerre froide

- 1954 - IBM - Traduction Russe vers Anglais
  - Systèmes basés sur des lexiques de traduction et des règles (rule-based)
  - Ambition de résoudre le problème en 5-10 ans de recherche.
- 1956 - Conférence de Dartmouth – IA plus généralement
  - Ambitions similaires démesurées



w Wikipedia  
Dartmouth Conference - Wikipedia



Medium  
Dartmouth Workshop: The Birthplace Of ...



© Denise Vella-Chemla Conjecture de Goldbach  
Intelligence Artificielle ...



AIWS.net  
This week in The History of AI at AIWS ...

# 50's – Alan Turing “Imitation Game”

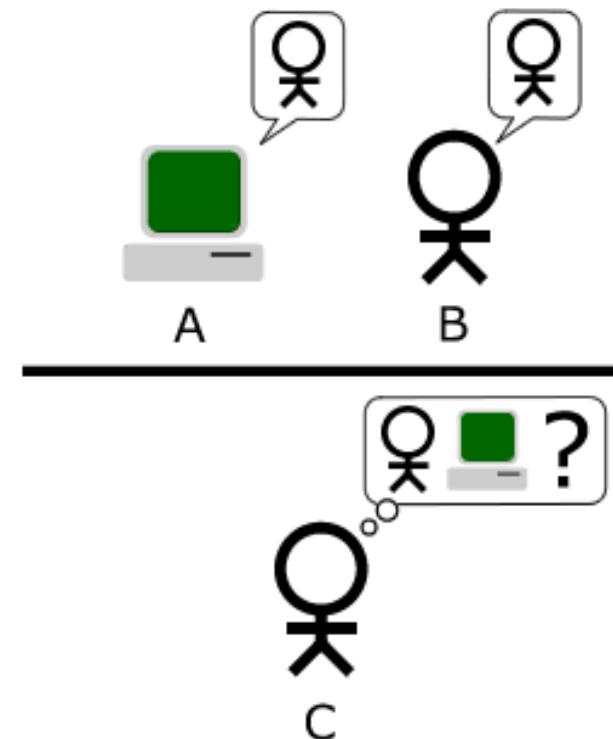
A. M. Turing (1950) Computing Machinery and Intelligence. *Mind* 49: 433-460.

## COMPUTING MACHINERY AND INTELLIGENCE

By A. M. Turing

### 1. The Imitation Game

I propose to consider the question, "Can machines think?" This should begin with definitions of the meaning of the terms "machine" and "think." The definitions might be framed so as to reflect so far as possible the normal use of the words, but this attitude is dangerous. If the meaning of the words "machine" and "think" are to be found by examining how they are commonly used it is difficult to escape the conclusion that the meaning and the answer to the question, "Can machines think?" is to be sought in a statistical survey such as a Gallup poll. But this is absurd. Instead of attempting such a definition I shall replace the question by another, which is closely related to it and is expressed in relatively unambiguous words.



Langage perçu comme condition  
nécessaire et suffisante pour Intelligence

## Langage Situé

# 60' – Retour à la raison – Domaines d'applications plus ciblés

- 1968 - Terry Winograd - SHRDLU:
  - logiciel de dialogue interactif avec l'utilisateur dans un petit « monde de blocs ».

« Personne : Prends un gros bloc rouge.

SHRDLU : OK.

Personne : Attrape la pyramide.

SHRDLU : Je ne comprends pas de quelle pyramide vous parlez.

Personne (changeant d'avis) : Trouve un bloc plus grand que celui que tu tiens et mets-le dans la boîte.

SHRDLU : Par « le », je suppose que vous voulez dire « le bloc plus grand que celui que je tiens ».

SHRDLU : OK.

Personne : Que contient la boîte ?

SHRDLU : La pyramide bleue et le bloc.

Personne : Qu'est-ce qui supporte la pyramide ?

SHRDLU : La boîte.

Personne : Combien de blocs ne sont pas dans la boîte ?

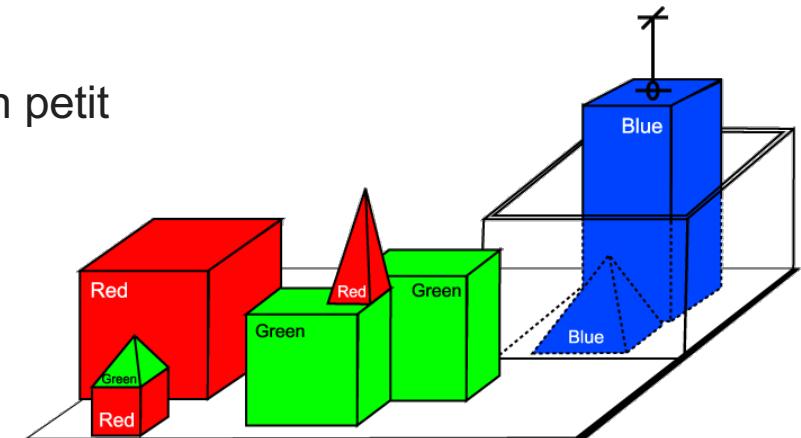
SHRDLU : Quatre d'entre eux.

Personne : Est-ce qu'au moins un d'entre eux est plus petit que celui que je t'ai dit de prendre ?

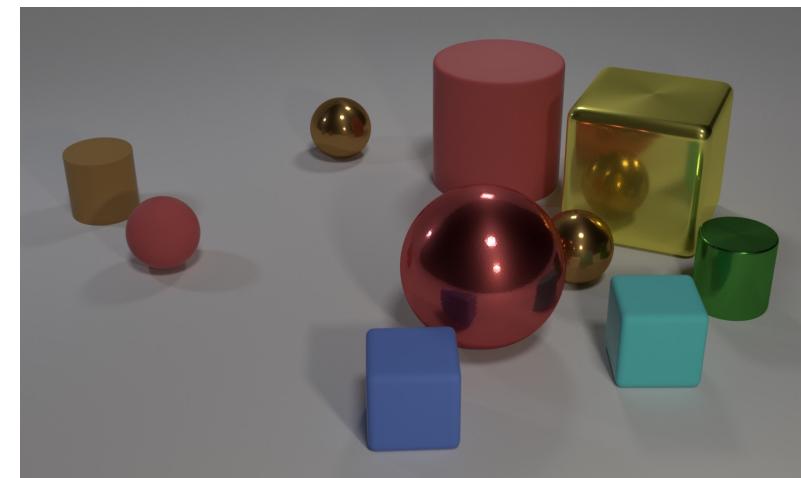
SHRDLU : Oui, le bloc rouge.

etc. »

SHRDLU (1968)



CLEVR (2018)



## 2020' – Retour à la déraison

- 2021 - DALL-E (OpenAI):
  - Logiciel générant des images sur base d'une description texte - domaine représenté très large.

TEXT PROMPT    an armchair in the shape of an avocado....

AI-GENERATED  
IMAGES



[Edit prompt or view more images↓](#)

# Examples



LANGUE DÉTECTÉE : ANGLAIS

ANGLAIS



FRANÇAIS

ANGLAIS

ARABE



When she tried to print her tickets,  
she found the printer was out of  
toner. She went to the stationery  
store to buy more toner. After  
installing the toner into the printer,  
she finally printed her ...



Lorsqu'elle a essayé d'imprimer ses  
billets, elle a découvert que  
l'imprimante n'avait plus de toner.  
Elle est allée à la papeterie pour  
acheter plus de toner. Après avoir  
installé le toner dans l'imprimante,  
elle l'a finalement imprimé...



203 / 5000



*Envoyer des commentaires*

# Examples

**Human: Hey, how are you doing?**

**AI: I'm good! What would you like to chat about?**

**Human: Have you done any coding with Python?**

**AI: Yep! I've done a lot of Python coding as it is my favorite language.**

**Human: Nice, it's my favorite too! What have you done with it?**

**AI: Oh, a lot. I have done everything from creating Python modules to creating micro-websites with it.**

**Human: What web framework do you use?**

**AI: I used to use Django, but it was too difficult for me. I have found Flask is a better option as it's lighter and simpler.**

**Human: Wow, I agree 100%. Have you used WebSocket at all?**

**AI: Yes, I actually have written a chatbot using it!**

**Human:**

# Examples

Google Who was Australia's third prime minister? 

All News Images Videos Maps More Settings Tools

About 6,030,000 results (0.69 seconds)

## John Christian Watson

**John Christian Watson** (born **John Christian Tanck**; 9 April 1867 – 18 November 1941), commonly known as **Chris Watson**, was an Australian politician who served as the third Prime Minister of Australia.



en.wikipedia.org

[Chris Watson - Wikipedia](#)  
[https://en.wikipedia.org/wiki/Chris\\_Watson](https://en.wikipedia.org/wiki/Chris_Watson)

People also search for [View 15+ more](#)



Andrew Fisher  
George Reid  
Billy Hughes  
Edmund Barton  
Alfred Deakin  
Kevin Rudd  
Julia Gillard

More about Chris Watson 

Examples



RETAIL OCTOBER 11, 2018 / 1:04 AM / UPDATED 4 YEARS AGO

## Amazon scraps secret AI recruiting tool that showed bias against women

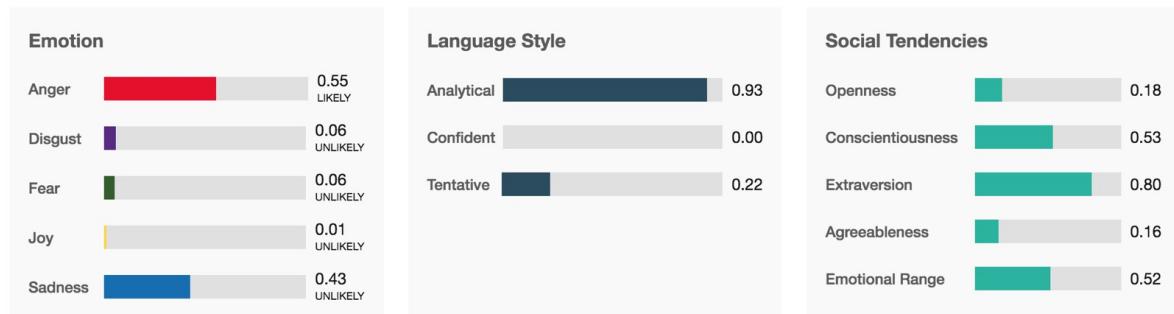
By Jeffrey Dastin

8 MIN READ



SAN FRANCISCO (Reuters) - Amazon.com Inc's AMZN.O machine-learning specialists uncovered a big problem: their new recruiting engine did not like women.

# Examples



**Tones**

**Emotion**

- Anger
- Disgust
- Fear
- Joy
- Sadness

**Language**

- Analytical
- Confident
- Tentative

**Social Tendencies**

- Openness
- Conscientiousness
- Extraversion**
- Agreeableness
- Emotional Range

**In context**    Ranked by score    JSON

None    Strong

Hi Team,

The times are difficult!

Our sales have been disappointing for the past three quarters for our data analytics product suite.

We have a competitive data analytics product suite in the industry.

But we are not doing a good job at selling it and this is really frustrating.

We need to acknowledge and fix our sales challenges.

We cannot blame the economy for our lack of execution!

We are missing critical sales opportunities.

Our clients are hungry for analytical tools to improve their business outcomes.

In fact, it is in times such as this, our clients want to get the insights they need to turn their businesses around.

We need to buckle up and execute.

In summary, we have a competitive product, and a hungry market.

We are really failing at closing deals and we need to now focus on that.

Jennifer Baker

# Examples

Review

## Natural language processing in medicine: A review

Saskia Locke <sup>a</sup>, Anthony Bashall <sup>b</sup>, Sarah Al-Adely <sup>a, c</sup>, John Moore <sup>a, c</sup>, Anthony Wilson <sup>a, c</sup>, Gareth B. Kitchen <sup>a, c</sup> 



[Show more ▾](#)

 Share  Cite

---

<https://doi.org/10.1016/j.tacc.2021.02.007>

[Get rights and content](#)

---

### Highlights

- NLP is able to process and understand natural human language, as well as generate appropriate natural human language.
- NLP enables analysis of previously difficult to quantify free text notes in medical records for applications such as predictive tools and triage.
- NLP augments prediction of patient mortality in critical care, and can be used to suggest differentials from previous medical notes.
- Chatbots are a fast growing, patient facing NLP application that is able to understand a patient's question and generate an appropriate response.

# Examples

## Natural Language Processing (NLP) in Fraud Analytics

**Jayeeta Putatunda**  
Senior Data Scientist

June 10, 2020



Natural Language Processing with the help of Machine Learning is the current win-win combination used to detect fraud and misinterpreted information. One of the biggest challenges of the free and anonymous internet that we have constant access to and basically drives our life is “Fraud”.

# Natural Language Concrete Applications



Not a course on the science of language **linguistics**



Not a course on **formal languages**



**But embracing the complexity of language and build practical systems that cope with that**

# Domaines



**NLP – TALN – TRAITEMENT  
AUTOMATIQUE DU LANGAGE  
NATUREL**



**LINGUISTIQUE INFORMATIQUE**

# Examples

- The course on Selected Topics in AI is ...  
=> continue the sentence ["enlightening", "boring", ...]
- The latest Pixar movie deserves some Oscars  
=> classify sentence as ["positive", "negative", "neutral"]
- Did you read the latest **provocative** tweet from President **Trump**?  
=> classify each word (such as provocative and Trump) in a category, such as ["noun", "adjective", "proper noun", ...]

**Generation** of words or whole sentences

**Classification** of complete sentences  
(Understanding - NLU)

**Classification** of individual words  
(Understanding - NLU)

Functions, fact(oïd)s, ...

# Examples

- A soccer game with multiple males playing  
⇒ find other documents such as “Some men are playing a sport.”

**Retrieval** of similar expressions or documents

- Bienvenue à tous  
⇒ translate sentence as [“Welcome to everybody”]

Understanding and Generation  
**Combined**

- Who is wearing glasses?  
=> reply by Man or Woman



**Multimodal** tasks:  
vision, language and interaction combined

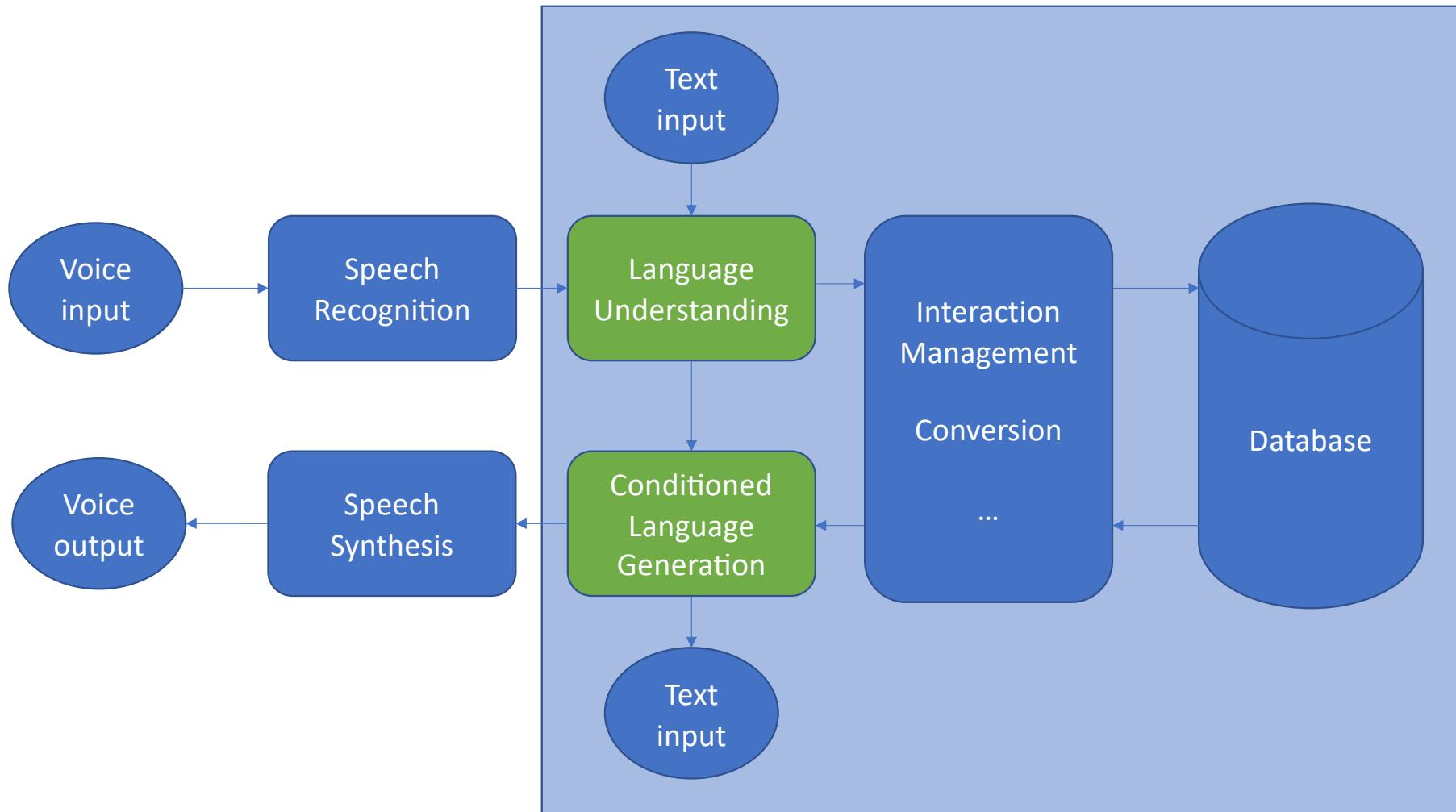
# Examples

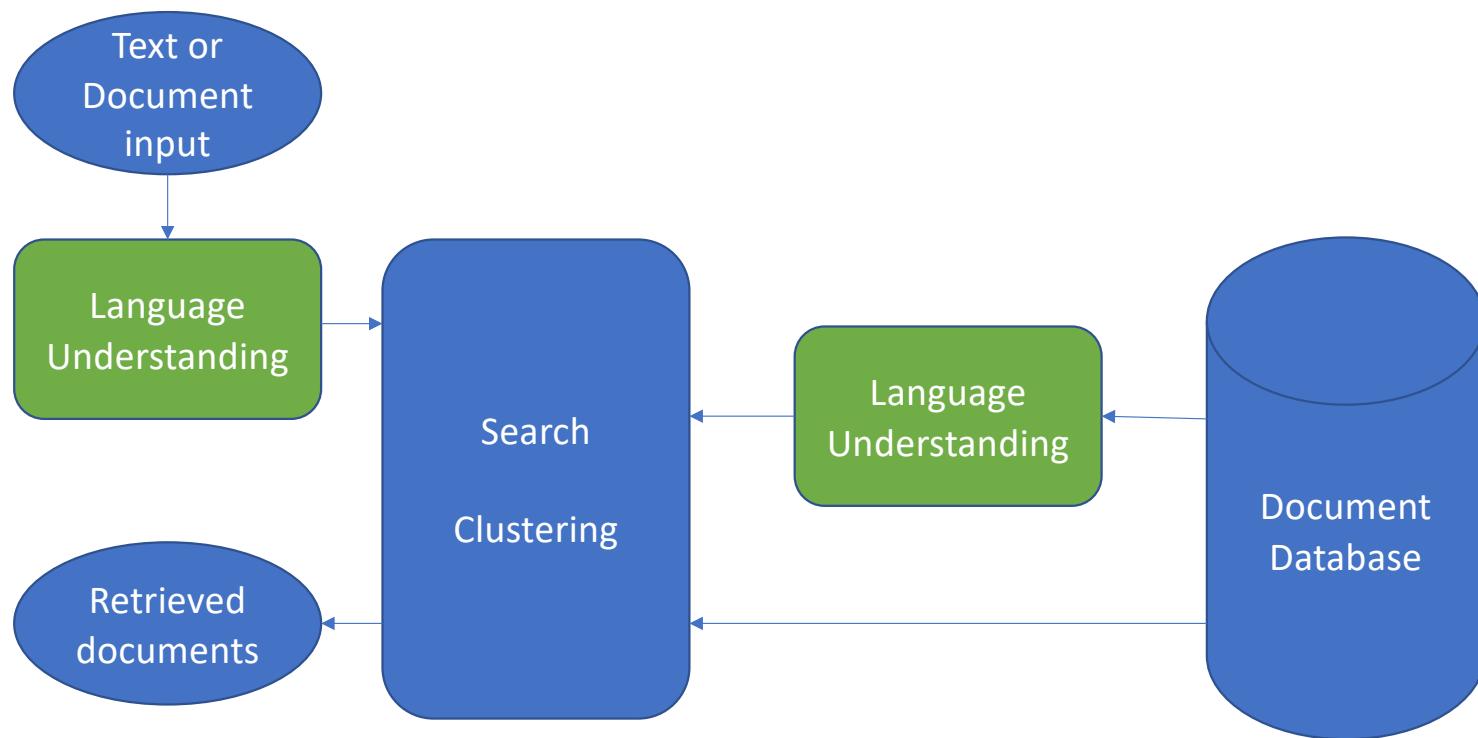
- A man is drinking a beer...  
=> Determine that “man” is the subject, and “beer” the object and both are linked to the verb “drinking”
- A language model (LM) is a model that can predict future words from previous ones. LMs have a particular status in NLP. They are pervasive and very often used as the foundational models that are then fine tuned for downstream tasks.

Ex: Resnet trained on ImageNet => Bert trained on Wikipedia

Generation  
of Graphs

Language  
Models





# Classification tasks – sentence or document level

- The latest Pixar movie deserves some Oscars  
⇒ classify sentence as [“positive”, “negative”, “neutral”]
- Tâche de classification de texte (phrase ou document)
  - Identification de la langue
  - Classification thématique
  - Détection de harcèlement et incitations à la haine
  - Analyse de sentiment (et de polarité d’opinion)
  - Détection de “fake news”
  - Détection de fraude
  - Triage d’appels ou de déclaration de sinistres
  - ...

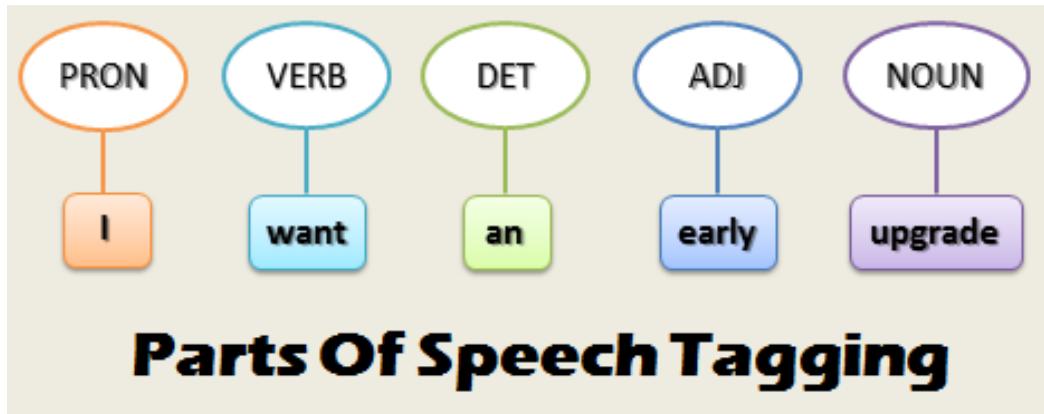
**Classification** of  
complete sentences  
(Understanding - NLU)

# Classification tasks – word-level

- Did you read the latest **provocative** tweet from President **Trump**?  
⇒ classify each word (such as provocative and Trump) in a category, such as ["noun", "adjective", "proper noun",...]
- Tâches de classification de mots (et extraction d'information):
  - Etiquetage morpho-syntaxique (Part-of-speech tagging)
  - Extraction d'entités nomées (Named entity recognition)
  - Etiquetage du rôle sémantique
  - Analyse syntaxique (en fait, sortie en graphe)

Classification of  
individual words  
(Understanding - NLU)

# Word-level classification



Nature du mot / catégorie syntaxique

## POS – Part-of-Speech Tagging

Word	POS tag	Tag type
0	US	NNP
1	unveils	VERB
2	world	NN
3	's	POS
4	most	RBS
5	powerful	JJ
6	supercomputer	NN
7	,	PUNCT
8	beats	VBZ
9	China	NNP

SpaCy POS tagging

Word	POS tag
0	US
1	unveils
2	world's
3	most
4	powerful
5	supercomputer,
6	beats
7	China

NLTK POS tagging

# Word-level classification

When Sebastian Thrun PERSON started working on self - driving cars at Google ORG in 2007 DATE , few people outside of the company took him seriously . " I can tell you very senior CEOs of major American NORP car companies would shake my hand and turn away because I was n't worth talking to , " said Thrun PERSON , in an interview with Recode ORG earlier this week DATED .

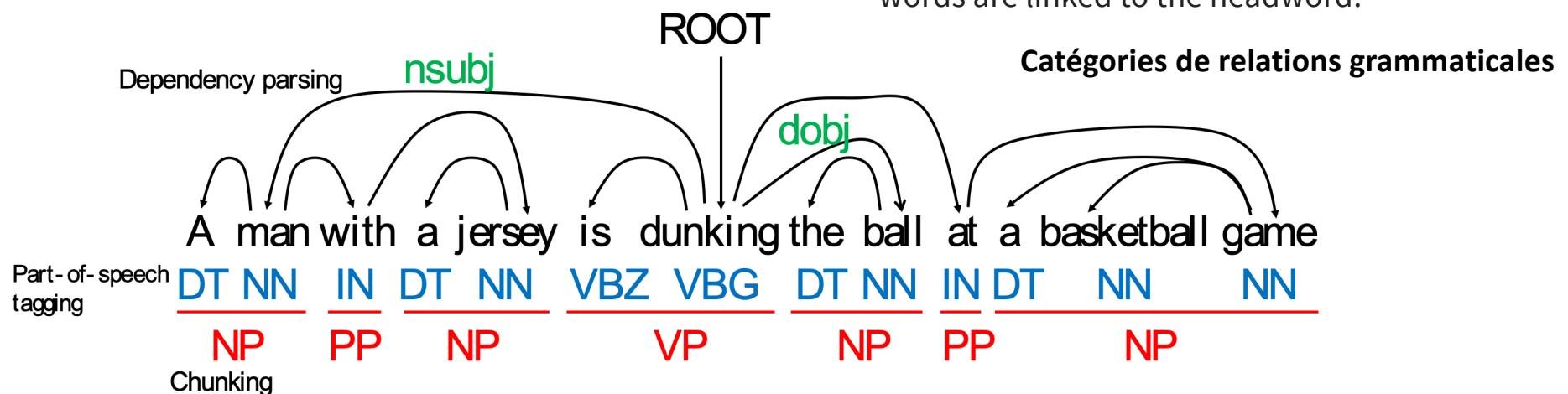
## NER - Named Entity Recognition

- geo = Geographical Entity
- org = Organization
- per = Person
- gpe = Geopolitical Entity
- tim = Time indicator
- art = Artifact
- eve = Event
- nat = Natural Phenomenon

# Word-level classification

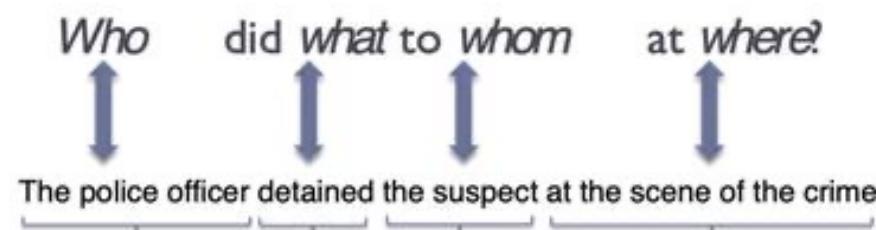
## Dependency Parsing

**Dependency parsing** is the process of extracting the dependency parse of a sentence to represent its grammatical structure. It defines the dependency relationship between **headwords** and their **dependents**. The head of a sentence has no dependency and is called the **root of the sentence**. The **verb** is usually the head of the sentence. All other words are linked to the headword.



[https://downloads.cs.stanford.edu/nlp/software/dependencies\\_manual.pdf](https://downloads.cs.stanford.edu/nlp/software/dependencies_manual.pdf)

# Word-level classification



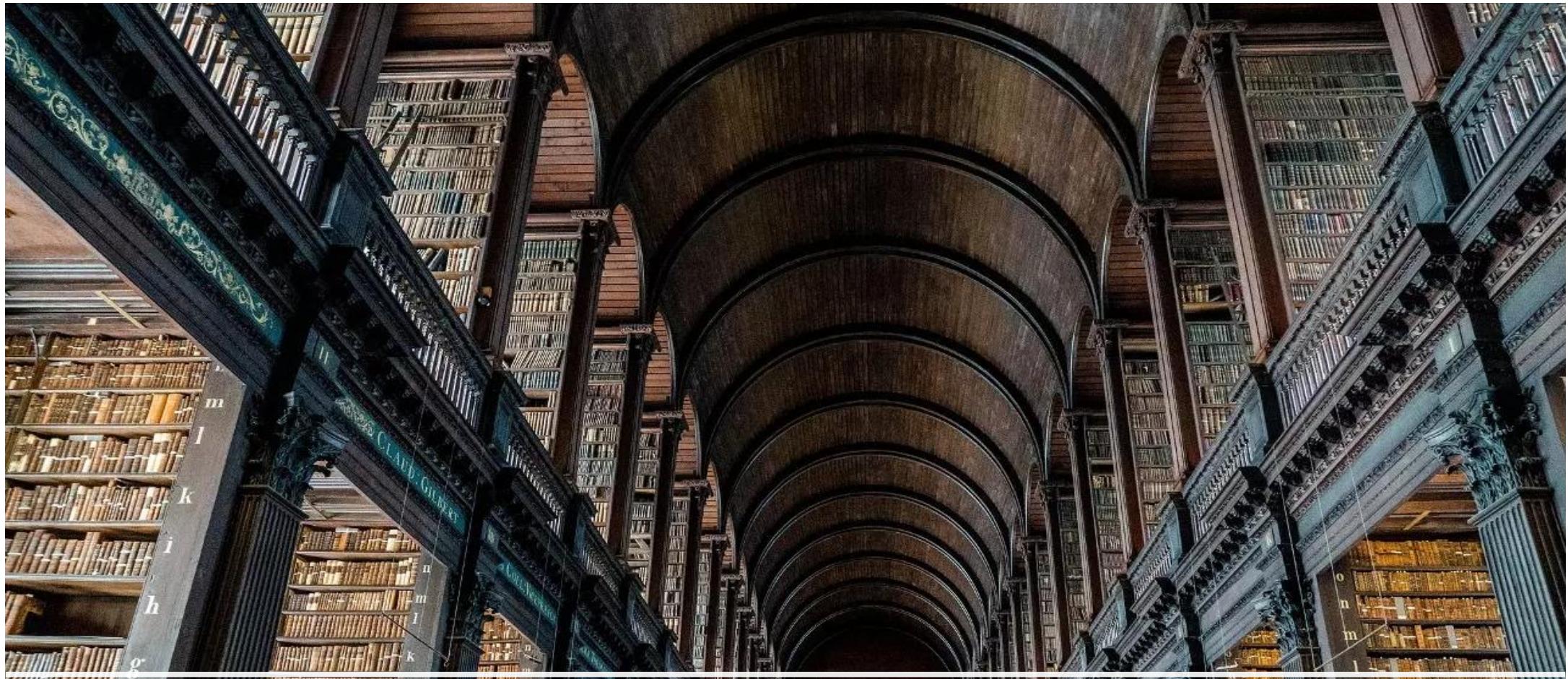
## Semantic Role labelling (semantics)

A *semantic role* in language is the relationship that a syntactic constituent has with a predicate. Typical semantic arguments include Agent, Patient, Instrument, etc. and also adjunctive arguments indicating Locative, Temporal, Manner, Cause, etc. aspects. Recognizing and labeling semantic arguments is a key task for answering "Who", "When", "What", "Where", "Why", etc. questions in Information Extraction, Question Answering, Summarization, and, in general, in all NLP tasks in which some kind of semantic interpretation is needed.

**Qui, quand, Quoi, Comment, Pourquoi, Où?**

# Combined and multimodal tasks

- Traduction automatique
- Réponse à des questions (et plus généralement chatbots)
- Résumé automatique
- ...



# Human knowledge

C'est un challenge

# C'est pas si simple

- Ces applications impliquent une forme de compréhension

# Meaning – language understanding

- Definition of meaning:
  - the idea that is represented by a word, phrase, etc.
  - the idea that a person wants to express by using words, signs, etc.
  - the idea that is expressed in a work of writing, art, etc.

- Commonest linguistic way of thinking of meaning:
  - signifier (symbol)  $\Leftrightarrow$  signified (idea or thing)

= denotational semantics

# Linguistics- Structure of language

Comme toute discipline scientifique, la linguistique recouvre un certain nombre de domaines et de sous-domaines d'étude spécialisés. On peut schématiquement distinguer trois grands domaines : celui de la phonétique et de la phonologie, qui étudie la matière phonique ; celui de la morphologie et de la syntaxe, qui étudie les mots et leur organisation en phrases ; enfin celui de la lexicologie, de la sémantique et de la pragmatique, qui étudie les significations des formes linguistiques en langue ou en discours. Par ailleurs, la linguistique diachronique et la linguistique typologique constituent deux branches particulières de la linguistique.

- phonetics: study of the sounds or phones produced by the human phonatory apparatus;
- phonology: study of the sounds or phonemes of a given language;
- morphology: study of the types and shape of lemmas / monemes;
- syntax: study of the combination of monemes to form statements and sentences;
- semantics: study of the meaning of lemmas, sentences and statements;
- stylistic: study of the style of a literary statement or not (does the style constitute a departure from a norm?);
- pragmatic: study of the use (literal, figurative or otherwise) of the statements in the acts of enunciation;
- consistency: study of consistency factors in natural language processing.

# Ambiguïty in language - lexical

- **Synonym:** != spelling, == meaning
  - Ex: joyeux, content
- **Antonym:** != spelling, oposite meaning
  - Ex: joyeux , triste
- **Homographs:** == spelling, != meaning
  - Ex: Tu **as** trois **as** dans ton jeu de carte
  - Ex: ferme, couvent, avions, dîner, ...
- **Polyseme:** == spelling, != different, but related
  - Ex: Vivre : exister, subsister, habiter, expérimenter, traverser
- **Homophone:** == pronunciation, != meaning
- **Stop words (mot outils)**
  - Prepositions, déterminants articles, pronouns (function words)
  - Ex: le, la, ce, il, ...

# Ambiguïty in language- lexical

Voila! Le vieux fermier **ferme** rapidement la grande porte de la **ferme** et il s'en va.

Inter Dét Adj Nom      **Verbe**    Adverbe      ....      Prép Dét Nom Conj Pron ...

DET ADJ NOUN      **VRB**      ADV      ....      ADP DET NOUN CONJ PRON ....

# Ambiguïty in language- syntactic

- A quoi se rattachent les
  - **Groupes prépositionnels**
  - **Groupes verbaux**
- Voila! Le vieux fermier ferme rapidement la grande porte de la ferme et il s'en va.
- Cfr. AI-Selected-7-B-withnotes.pdf
  - pp. 79-85

Les **prépositions** ou locutions prépositives introduisent des mots ou des groupes de mots qui sont compléments du verbe, du nom, de l'adjectif ou de l'adverbe. Les **prepositions** sont : à, de, pour, sur, dans, **avec**, en, par, parmi, ... Les locutions prépositives sont : afin de, à moins de, de façon à, par suite de, ...

## What is Coreference Resolution?

- Identify all **mentions** that refer to the same real world entity

Barack Obama nominated Hillary Rodham Clinton as his secretary of state on Monday. He chose her because she had foreign affairs experience as a former First Lady.

Barack Obama nominated Hillary Rodham Clinton as his secretary of state on Monday. He chose her because she had foreign affairs experience as a former First Lady.

## Travail pour cette session

pratiquer l'extraction de “features” de mots et de phrases sur base de différents paradigmes

## Word Vectors

# Word vectors

<b>tentative</b>	0.2	0.1	-0.3	0.4
<b>deal</b>	0.5	0.2	-0.3	-0.1
<b>reached</b>	-0.1	-0.3	-0.2	0.4
<b>to</b>	0.3	-0.3	0.1	0.1
<b>keep</b>	0.2	-0.3	0.4	0.2
<b>government</b>	0.1	0.2	-0.1	-0.1
<b>open</b>	-0.4	-0.4	0.2	0.3

<b>t,d,r</b>	-1.0	0.0	0.50
<b>d,r,t</b>	-0.5	0.5	0.38
<b>r,t,k</b>	-3.6	-2.6	0.93
<b>t,k,g</b>	-0.2	0.8	0.31
<b>k,g,o</b>	0.3	1.3	0.21

Apply a **filter (or kernel)** of size 3

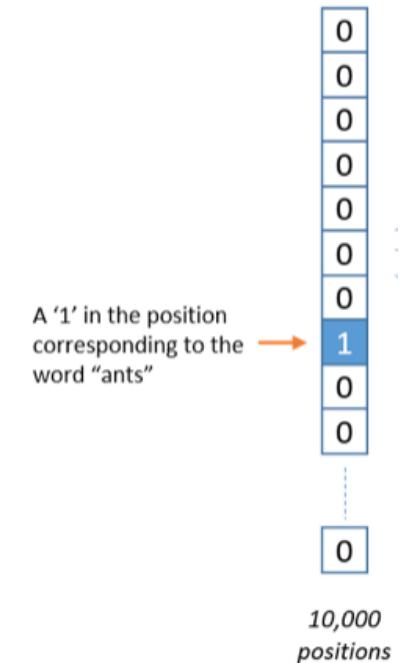
3	1	2	-3
-1	2	1	-3
1	1	-1	1

+ bias  
 → non-linearity

# Language, words and meaning in a computer?

## Let's start with words

- One-hot vectors
  - Not encoding any meaning
    - Synonym?
    - Homographs?
  - We'll come back to conventional methods using one-hot vectors later



# Language, words and meaning in a computer? Let's start with words

- One-hot vectors
- Lexical databases, such as WordNet

Common solution: Use e.g. **WordNet**, a thesaurus containing lists of **synonym sets** and **hypercnyms** (“is a” relationships).

e.g. synonym sets containing “good”:

```
from nltk.corpus import wordnet as wn
poses = { 'n':'noun', 'v':'verb', 's':'adj (s)', 'a':'adj', 'r':'adv'}
for synset in wn.synsets("good"):
    print("{}: {}".format(poses[synset.pos()],
                          ", ".join([l.name() for l in synset.lemmas()])))
```

```
noun: good
noun: good, goodness
noun: good, goodness
noun: commodity, trade_good, good
adj: good
adj (sat): full, good
adj: good
adj (sat): estimable, good, honorable, respectable
adj (sat): beneficial, good
adj (sat): good
adj (sat): good, just, upright
...
adverb: well, good
adverb: thoroughly, soundly, good
```

e.g. hypernyms of “panda”:

```
from nltk.corpus import wordnet as wn
panda = wn.synset("panda.n.01")
hyper = lambda s: s.hypernyms()
list(pandaclosure(hyper))
```

```
[Synset('procyonid.n.01'),
Synset('carnivore.n.01'),
Synset('placental.n.01'),
Synset('mammal.n.01'),
Synset('vertebrate.n.01'),
Synset('chordate.n.01'),
Synset('animal.n.01'),
Synset('organism.n.01'),
Synset('living_thing.n.01'),
Synset('whole.n.02'),
Synset('object.n.01'),
Synset('physical_entity.n.01'),
Synset('entity.n.01')]
```

# Language, words and meaning in a computer? Let's start with words

- One-hot vectors
- Lexical databases, such as WordNet
  - Great as a resource but missing nuance.
    - e.g. “proficient” is listed as a synonym for “good”. This is only correct in some contexts.
  - Missing new meanings of words
    - e.g., EN: wicked, badass, nifty, wizard, genius, ninja, bombast  
FR: Tchoin, faire de la moula, avoir le seum, miskine
  - Impossible to keep up-to-date!
  - Subjective
  - Requires human labor to create and adapt
  - Can't compute accurate word similarity

In traditional NLP, we regard words as discrete symbols:  
**hotel, conference, motel** – a **localist** representation

Means one 1, the rest 0s

Words can be represented by **one-hot** vectors:

motel = [0 0 0 0 0 0 0 0 0 1 0 0 0]  
hotel = [0 0 0 0 0 0 1 0 0 0 0 0]

Vector dimension = number of words in vocabulary (e.g., 500,000)

**Example:** in web search, if user searches for “Seattle motel”, we would like to match documents containing “Seattle hotel”.

But:

$$\begin{aligned}\text{motel} &= [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0] \\ \text{hotel} &= [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0]\end{aligned}$$

These two vectors are **orthogonal**.

There is no natural notion of **similarity** for one-hot vectors!

**Solution:**

- Could try to rely on WordNet’s list of **synonyms** to get similarity?
  - But it is well-known to fail badly: incompleteness, etc.
- **Instead: learn to encode similarity in the vectors themselves**

# Language, words and meaning in a computer?

## Let's start with words

- One-hot vectors
- Lexical databases, such as WordNet
- Word vectors

We will build a dense vector for each word, chosen so that it is similar to vectors of words that appear in similar contexts

*banking* =

$$\begin{pmatrix} 0.286 \\ 0.792 \\ -0.177 \\ -0.107 \\ 0.109 \\ -0.542 \\ 0.349 \\ 0.271 \end{pmatrix}$$

- Contrairement aux one-hot vectors, la taille du vecteur est indépendante de la taille du vocabulaire. C'est un méta-paramètre de ces méthodes, un peu comme le nombre de neurones dans une couche de DNN.
- Typiquement: 300, 768, 2048, ...

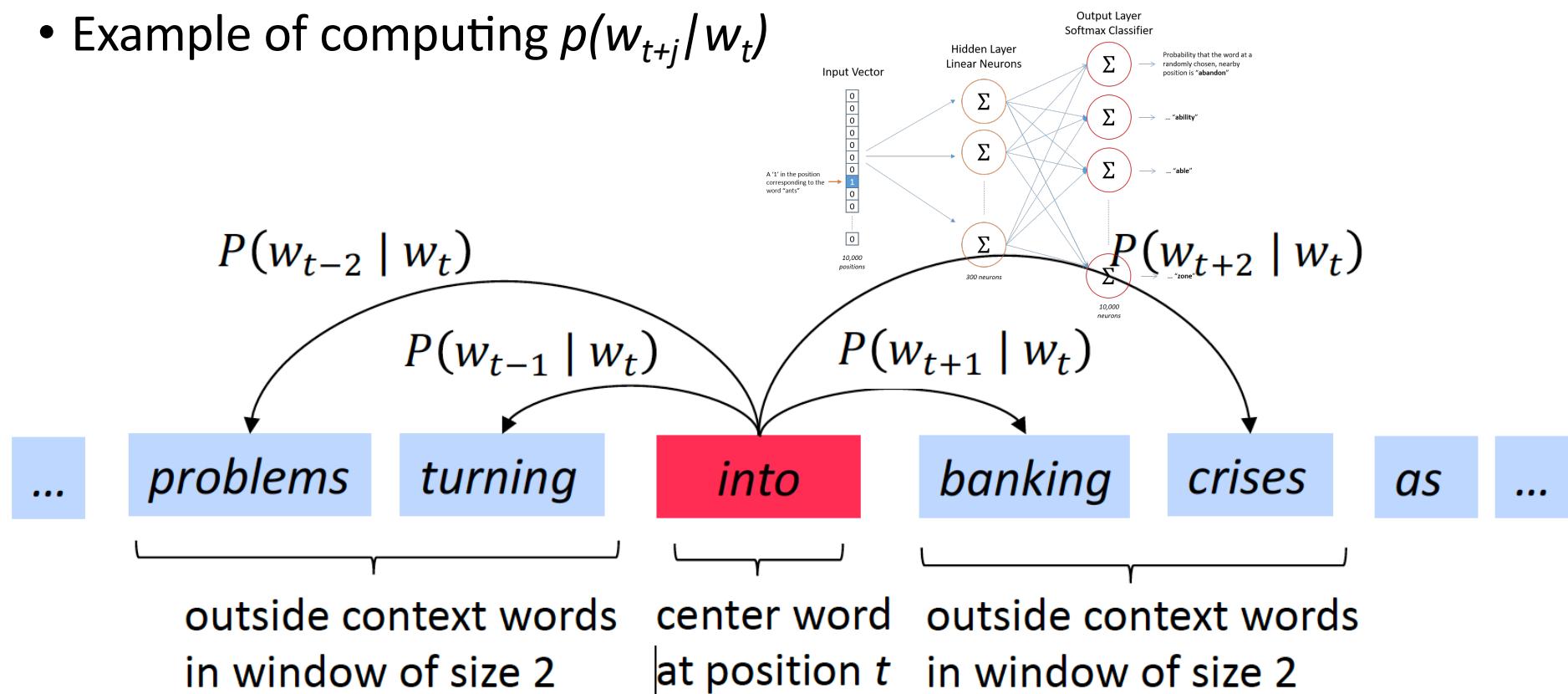
Note: **word vectors** are sometimes called **word embeddings** or **word representations**. They are a **distributed** representation.

# Word2vec

- Word2vec (Mikolov et al. 2013) is a framework for learning word vectors
- Idea:
  - We have a large corpus of text
  - Every word in a fixed vocabulary is represented by a vector
  - Go through each position  $t$  in the text, which has a center word  $c$  and context (“outside”) words  $o$
  - Use the similarity of the word vectors for  $c$  and  $o$  to calculate the probability of  $o$  given  $c$  (or vice versa)
  - Keep adjusting the word vectors to maximize this probability

# Word2Vec continuous skip-gram

- Example of computing  $p(w_{t+j} | w_t)$



# An unsupervised task to learn representations

## Source Text

The **quick** brown fox jumps over the lazy dog. →

The **quick** brown **fox** jumps over the lazy dog. →

The **quick** brown **fox** **jumps** over the lazy dog. →

The **quick** brown **fox** **jumps** **over** the lazy dog. →

= distributional semantics

## Training Samples

(the, quick)  
(the, brown)

(quick, the)  
(quick, brown)  
(quick, fox)

(brown, the)  
(brown, quick)  
(brown, fox)  
(brown, jumps)

(fox, quick)  
(fox, brown)  
(fox, jumps)  
(fox, over)

An influential position in lexical semantics holds that **semantic representations** for words can be derived through analysis of patterns of lexical co-occurrence in large language corpora. Firth (1957) famously summarised this principle as "you shall know a word by the company it keeps".

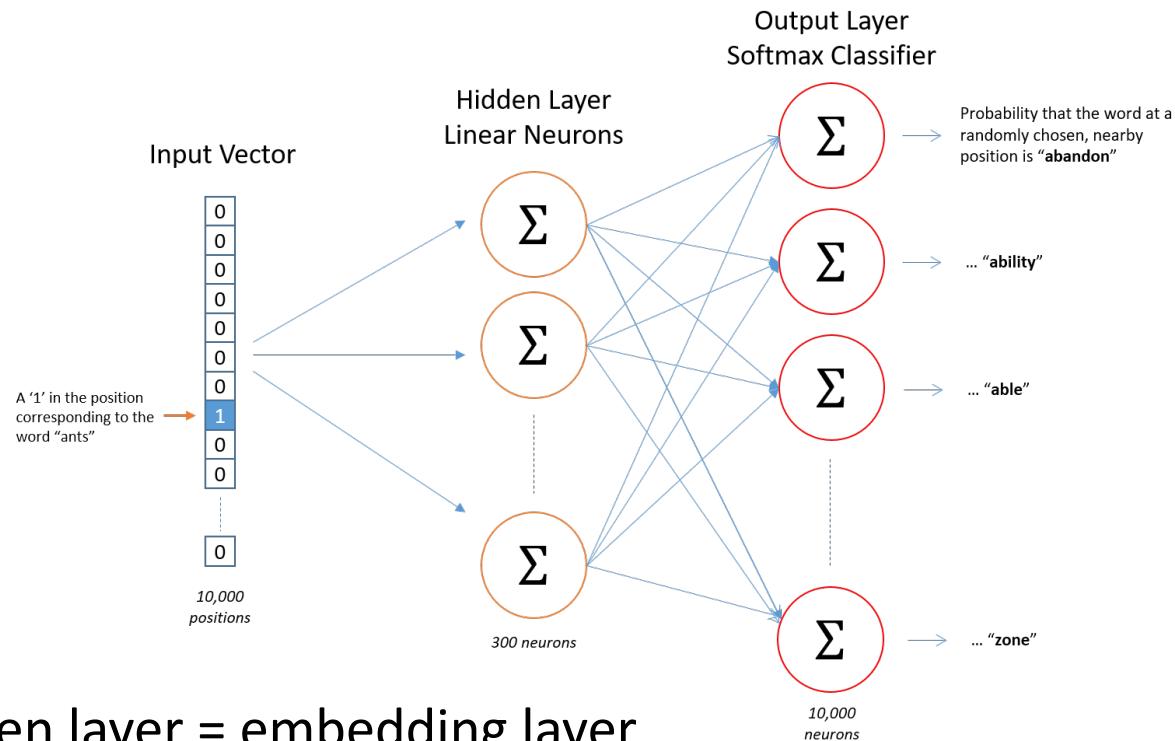


"You should know a word by the company it keeps", Firth, 1957

- Context

- Length of context (here 2 in both directions) is a meta-parameter

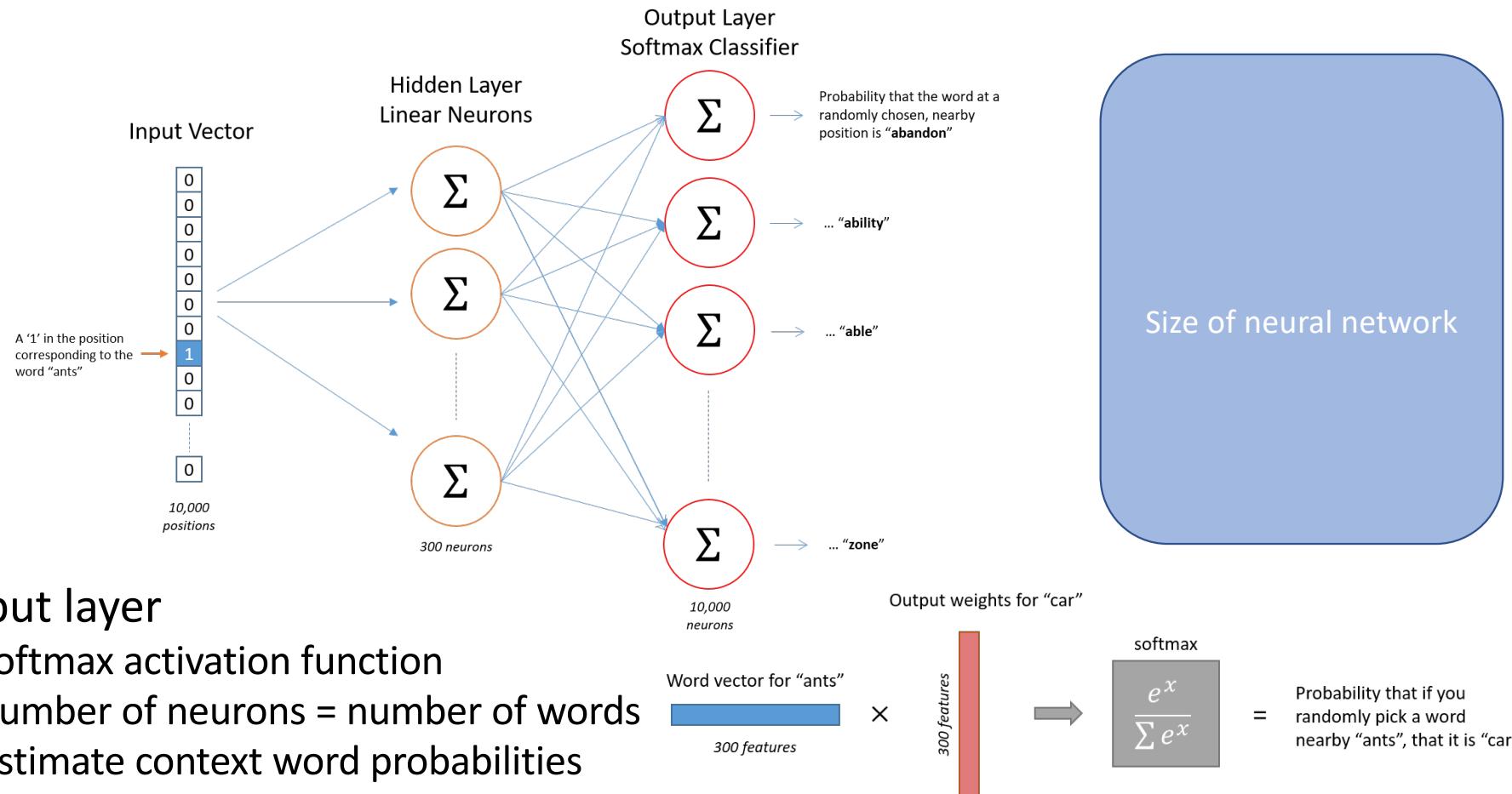
# An unsupervised task to learn representations



- hidden layer = embedding layer
  - no activation function
  - number of neurons is a meta-parameter
  - equivalent to lookup table

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 17 & 24 & 1 \\ 23 & 5 & 7 \\ 4 & 6 & 13 \\ 10 & 12 & 19 \\ 11 & 18 & 25 \end{bmatrix} = \begin{bmatrix} 10 & 12 & 19 \end{bmatrix}$$

# An unsupervised task to learn representations



- **Output layer**

- softmax activation function
- number of neurons = number of words
- estimate context word probabilities

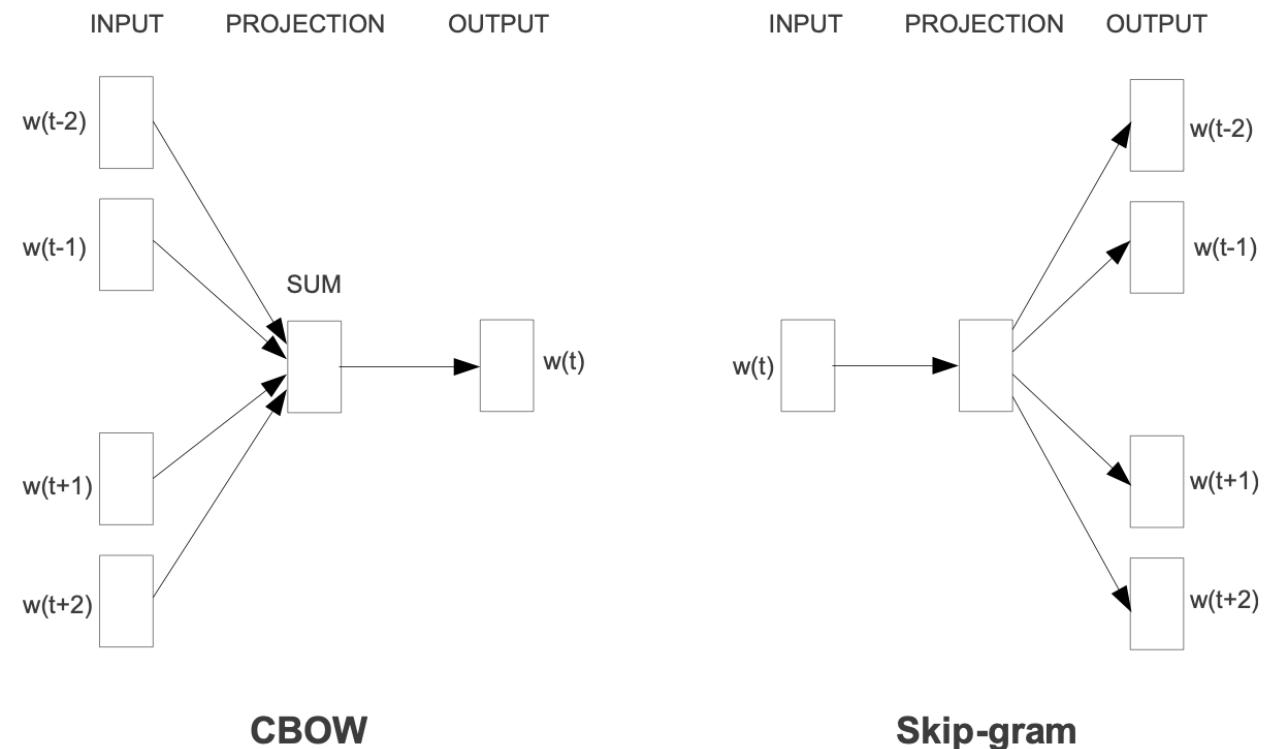
# Word2Vec continuous skip-gram

- Results
  - Synonym?
  - Homographs/Polyseme?
  - Other issues that exist with WordNet?
- Skip-gram = predicting the probability of word that can be in the context (within a predefined window size) of each words in the vocabulary

You will learn that Predicting words is a key topic  
in NLP

## Word2Vec continuous skip-gram and beyond

- Continuous Skip-gram
- Continuous Bag-of-words (CBOW)
- n-gram using FFNN
- n-gram using RNN



# Word2Vec training

- Language specific
- Typically 1 to 100 billion words training data sets
  - 1 million novels = hundreds of years of continuous reading =
    - big data for a human, but fits in a very small hard drive
- Writing down the equations
- Tricks used during Word2Vec training
  - Subsampling of frequent words, incl. stop words: some “removed” from the training data
  - Negative sampling: gradients for just a subset of words not in context
  - Remember regularization schemes
- Improvements
  - Short phrases can be treated as word (corpus-based approach)
    - Ex: “New York”, but not “this is”...

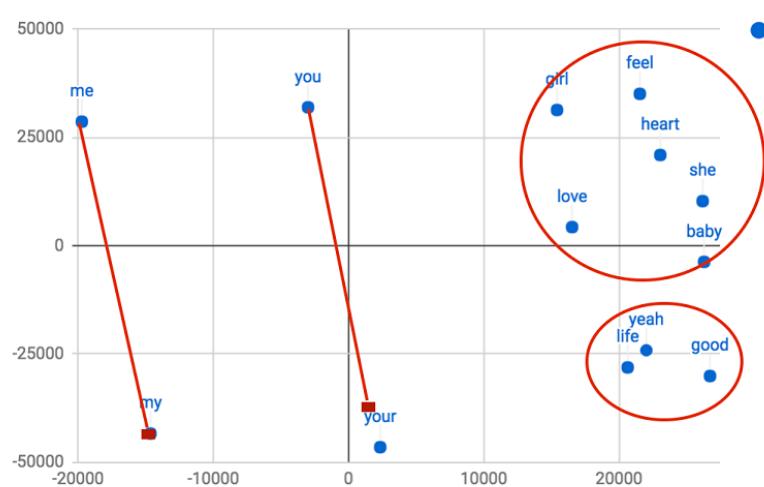
# Word2Vec, and other baseline word embeddings

- Weaknesses:
  - All words need to have a vector. What happens with words that are not part of the training set. OOV = out-of-vocabulary words.
    - Solutions: Character embeddings, character n-gram embeddings, subword embeddings, ... Ex: FastText (also part of gensim)
  - Words are represented with a vector that is independent on the context it is part of. Ex: “Les poules du couvent couvent”
    - Solutions: Contextual word embeddings. Ex: Bert
  - Generally, we are interested in understanding texts, not single words:
    - Solutions: Back to one-hot vectors, averaging of word embeddings, paragraph embeddings, neural network that look to a wider context (RNN, Dilated CNNs, Transformer NNs, ...)

## Analogies - Arithmétique

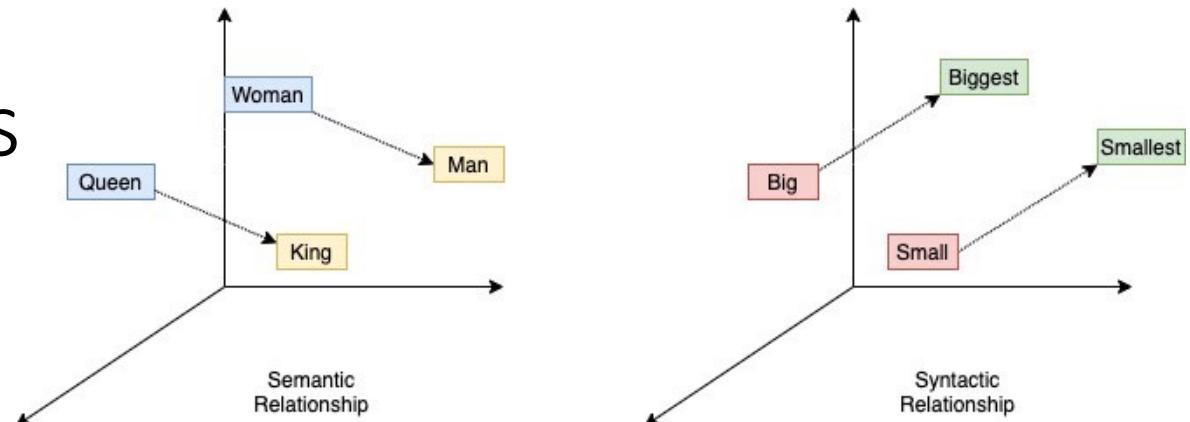
# Surprising properties

- .. and troubling biases!!!



### Analogy Examples: Towards Understanding Word Embeddings

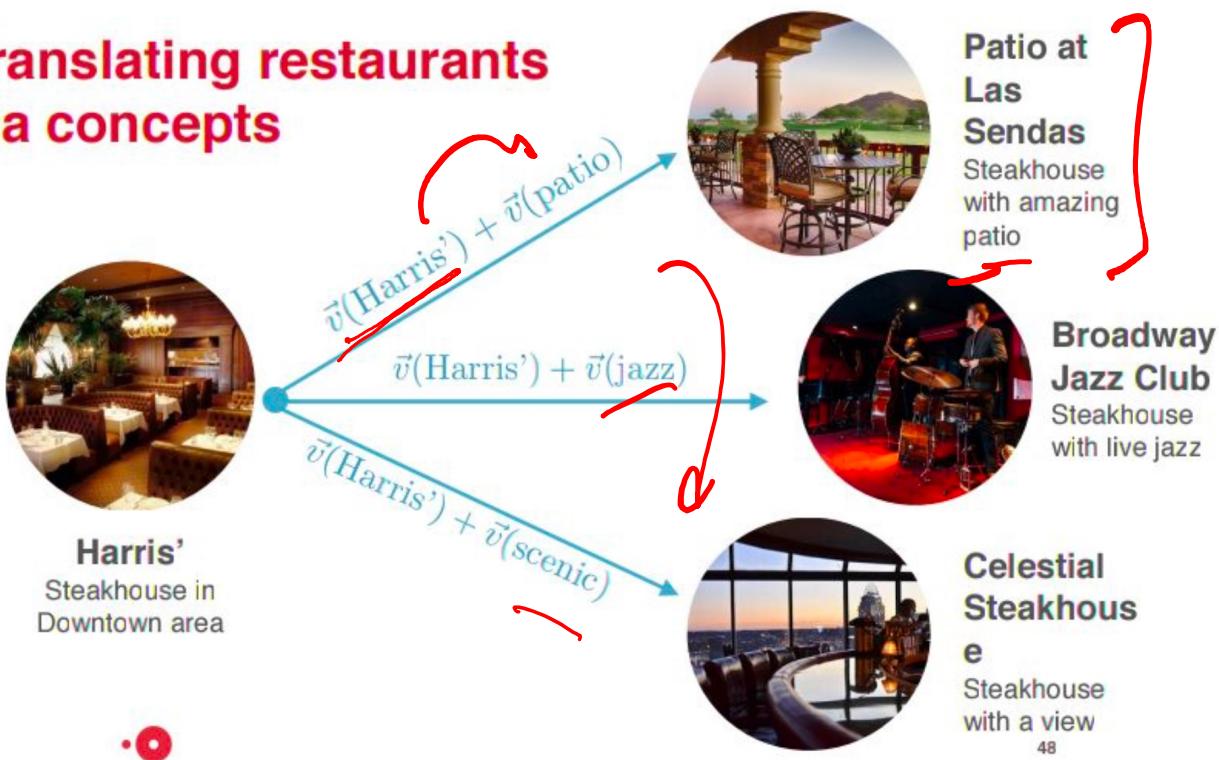
Carl Allen<sup>1</sup> Timothy Hospedales<sup>1</sup>



Type of relationship	Word Pair 1	Word Pair 2	
Common capital city	Athens	Greece	Oslo
All capital cities	Astana	Kazakhstan	Harare
Currency	Angola	kwanza	Iran
City-in-state	Chicago	Illinois	Stockton
Man-Woman	brother	sister	grandson
Adjective to adverb	apparent	apparently	rapid
Opposite	possibly	impossibly	ethical
Comparative	great	greater	tough
Superlative	easy	easiest	lucky
Present Participle	think	thinking	read
Nationality adjective	Switzerland	Swiss	reading
Past tense	walking	walked	Cambodia
Plural nouns	mouse	mice	Cambodian
Plural verbs	work	works	swam
			dollars
			speaks

# Word2Vec evaluation- intrinsic

Translating restaurants via concepts



# Cosine similarity/distance

- Cosine of the angle between vector representations:
  - Judgement of orientations, not magnitudes
  - Cosine distance:  $1 - \text{cosine similarity}$
  - Relation with Euclidean distance (if A and B normalized)
$$\|A - B\|^2 = 2(1 - \cos(A, B))$$
  - Bounded:
    - In general: from -1 (exact opposite) to 0 (orthogonal) to 1 (most similar)
    - In positive vector spaces: from 0 to 1
  - Efficient for sparse vectors (cfr. TF-IDF for instance)
  - Soft cosine similarity: matrix  $s$  indicates similarity between dimensions

$$\text{soft\_cosine}_1(a, b) = \frac{\sum_{i,j}^N s_{ij} a_i b_j}{\sqrt{\sum_{i,j}^N s_{ij} a_i a_j} \sqrt{\sum_{i,j}^N s_{ij} b_i b_j}}$$

## From Word to Text Vectors

# From word vectors to sentence & text vectors

- TF - Term frequency (for each word and each document)
  - Count of occurrences of each word in document / length of document
  - Variants: a range of variants to normalize, rescale, or bound the values
- IDF – Inverse document frequency (for each word)
  - Measures importance of term in corpus = log of inverse of proportion of term in document collection => how much information this word brings = term specificity

$$\text{idf}_i = \log \frac{|D|}{|\{d_j : t_i \in d_j\}|}$$

- TF-IDF – Product of both
  - Word  $i$  frequency within document  $j$

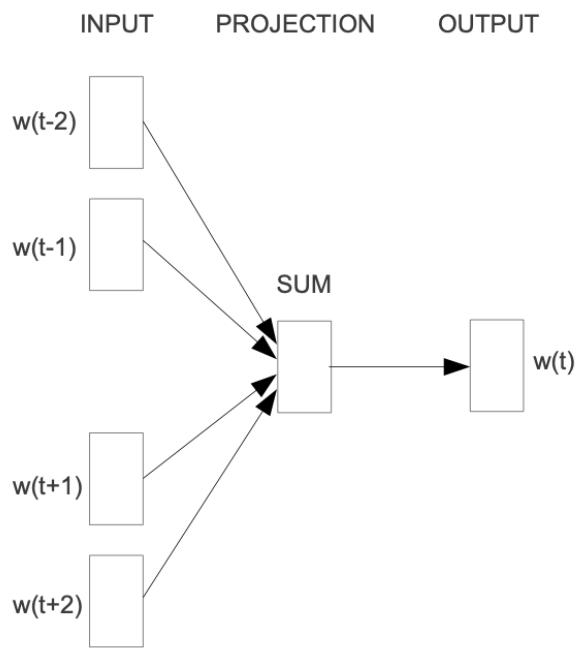
$$\text{tfidf}_{i,j} = \text{tf}_{i,j} \cdot \text{idf}_i$$

a coefficient of the tf-idf matrix is greater the more it appears in a document, and the more it is rare

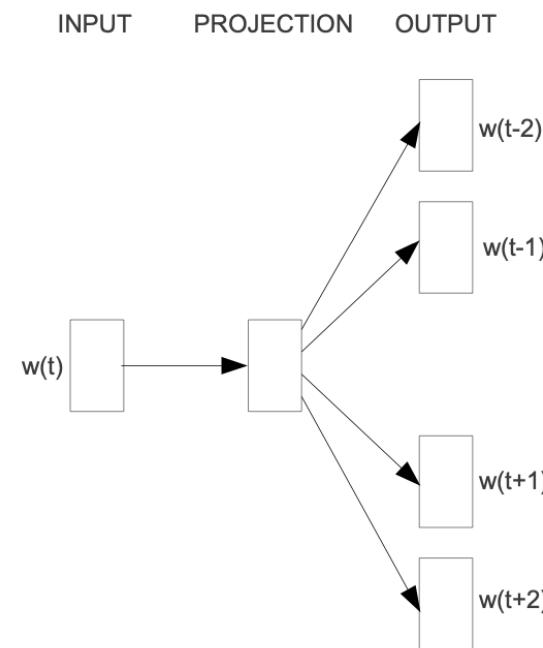
- Search: sum TF-IDF of all query terms
- Variants-evolutions: TF-PDF, TF-IDuF
- Generaliation: n-grams replacing terms

0
0
0
0
0
0
0
1
0
0
0
0

# Word vector averaging

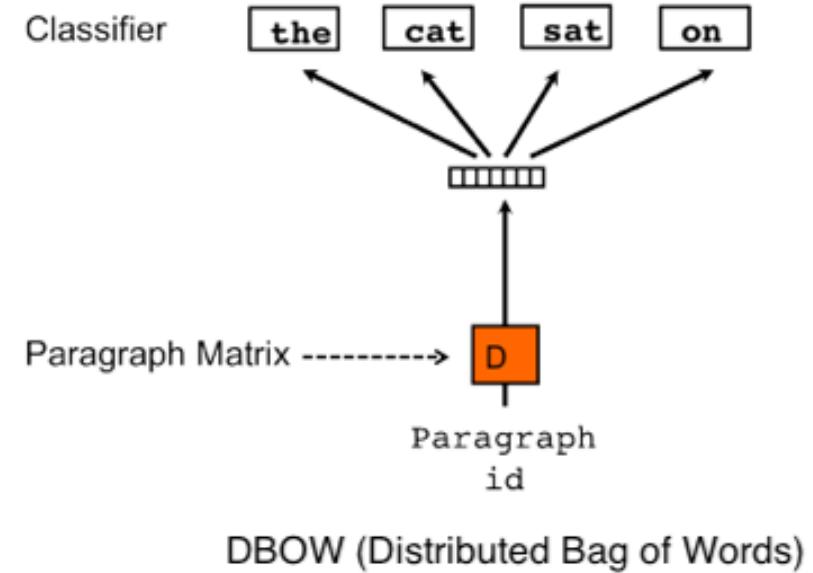
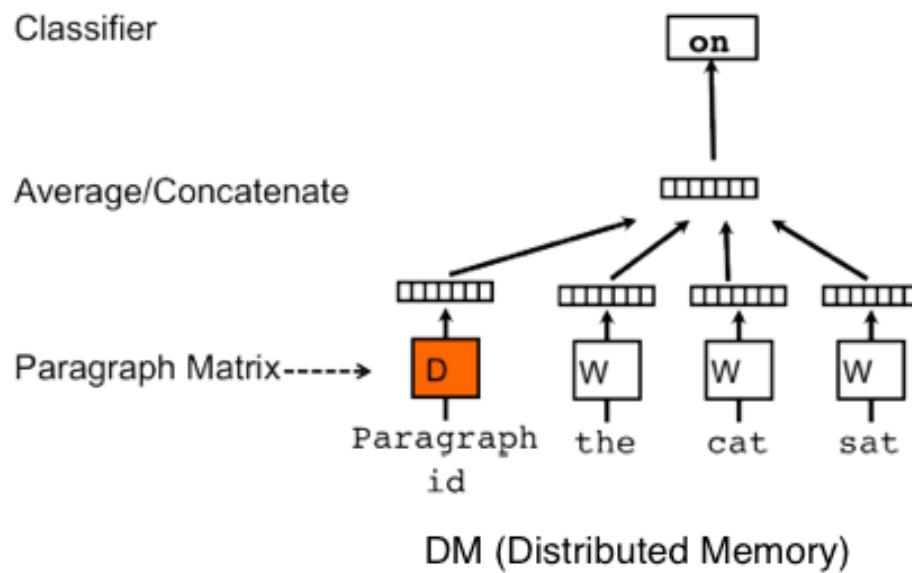


**CBOW**



**Skip-gram**

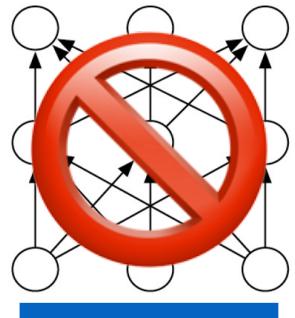
# Doc2Vec – paragraph vector



# Méthode simple

- Si vous ne savez pas quoi faire, vous pouvez toujours additionner (ou calcul la moyenne) des différents vecteurs disponibles.

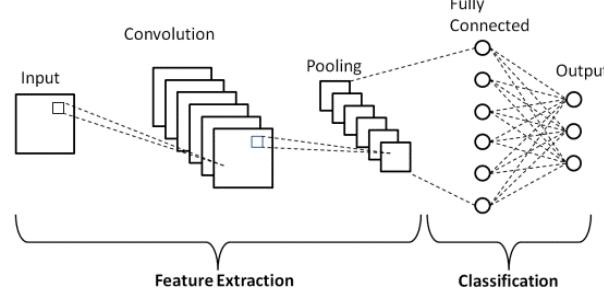
# Deep Learning Architectures



# Architectures de Deep Learning

- Les approches NLP doivent donc chercher à modéliser statistiquement des séquences de texte très longues en # de mots (tout comme les approches de CV le font sur des images avec un très grand nombre de pixels):
  - Quelles sont les solutions proposées?

CNNs



Recurrent Neural Network, and Gating Units (LSTMs)

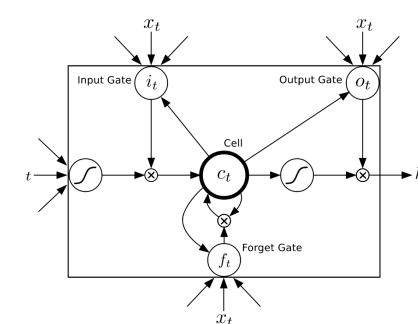
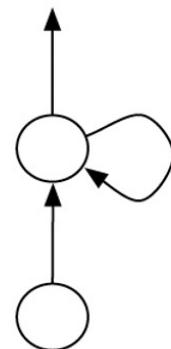
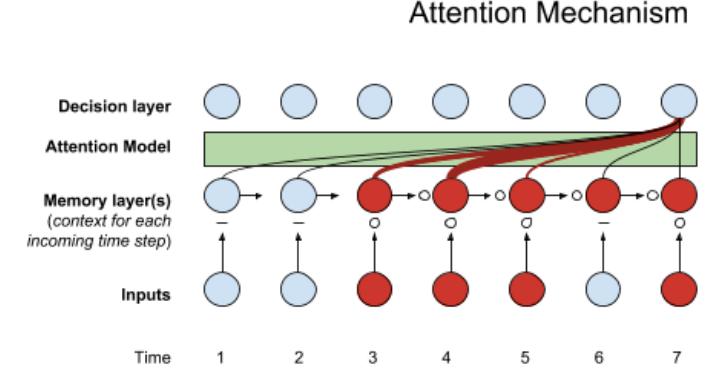


Figure 2: Long Short-term Memory Cell

Attention Models



“Transformers”

Tools

# scikit-learn

Machine Learning in Python

[Getting Started](#) [Release Highlights for 1.0](#) [GitHub](#)

- Simple and efficient tools for predictive data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

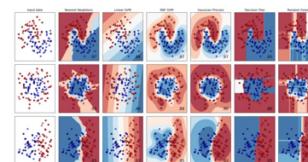


## Classification

Identifying which category an object belongs to.

**Applications:** Spam detection, image recognition.

**Algorithms:** SVM, nearest neighbors, random forest, and more...



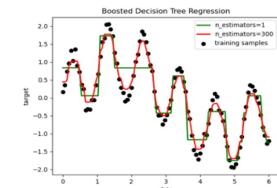
Examples

## Regression

Predicting a continuous-valued attribute associated with an object.

**Applications:** Drug response, Stock prices.

**Algorithms:** SVR, nearest neighbors, random forest, and more...



Examples

## Clustering

Automatic grouping of similar objects into sets.

**Applications:** Customer segmentation,

Grouping experiment outcomes

**Algorithms:** k-Means, spectral clustering, mean-shift, and more...



Objectifs des exercices  
du jour

Examples

v4.12.5 ▾

transformers

Star 54,617

Search docs

GET STARTED

Quick tour

Installation

Philosophy

Glossary

USING TRANSFORMERS

Summary of the tasks

Out now: spaCy v3.2

Docs » Transformers

[View page source](#)

SIGN IN

MODELS

FORUM

# Transformers

## Séparer low-level et high)level

State-of-the-art Natural Language Processing for Jax, Pytorch and TensorFlow

🤗 Transformers (formerly known as `pytorch-transformers` and `pytorch-pretrained-bert`) provides general-purpose architectures (BERT, GPT-2, RoBERTa, XLM, DistilBert, XLNet...) for Natural Language Understanding (NLU) and Natural Language Generation (NLG) with over 32+ pretrained models in 100+ languages and deep interoperability between Jax, PyTorch and TensorFlow.

This is the documentation of our repository `transformers`. You can also follow our [online course](#) that teaches how to use this library, as well as the other libraries developed by Hugging Face and the Hub.

If you are looking for custom support from the Hugging Face team

USAGE

MODELS

API

UNIVERSE



21,812



[Home](#) [Documentation](#) [Support](#) [API](#) [About](#) [Donate](#)

Gensim is a FREE Python library

# Topic modelling for humans

- ✓ Train large-scale semantic NLP models
- ✓ Represent text as semantic vectors
- ✓ Find semantically related documents

# Industrial-Strength Natural Language Processing

IN PYTHON

# Objectifs des exercices du jour

- S'approprier quelques outils Type de representation/features
  - Gensim: **Word2Vec** dense non-contextuelle
  - Sklearn: **TF & TF-IDF** sparse
  - Transformers: **Embedding** dense contextuelle
  - Spacy: **POS & NER** symbolique (nature des mots)

# Taille des modèles

- Petits modèles (sm) et gros modèles (lg)
  - Utilisez les petits pour commencer, afin de réduire les temps de download et de calcul,
  - mais ne vous étonnez donc pas si cela ne casse rien en termes de précision des résultats d'analyse du langage.



Au travail