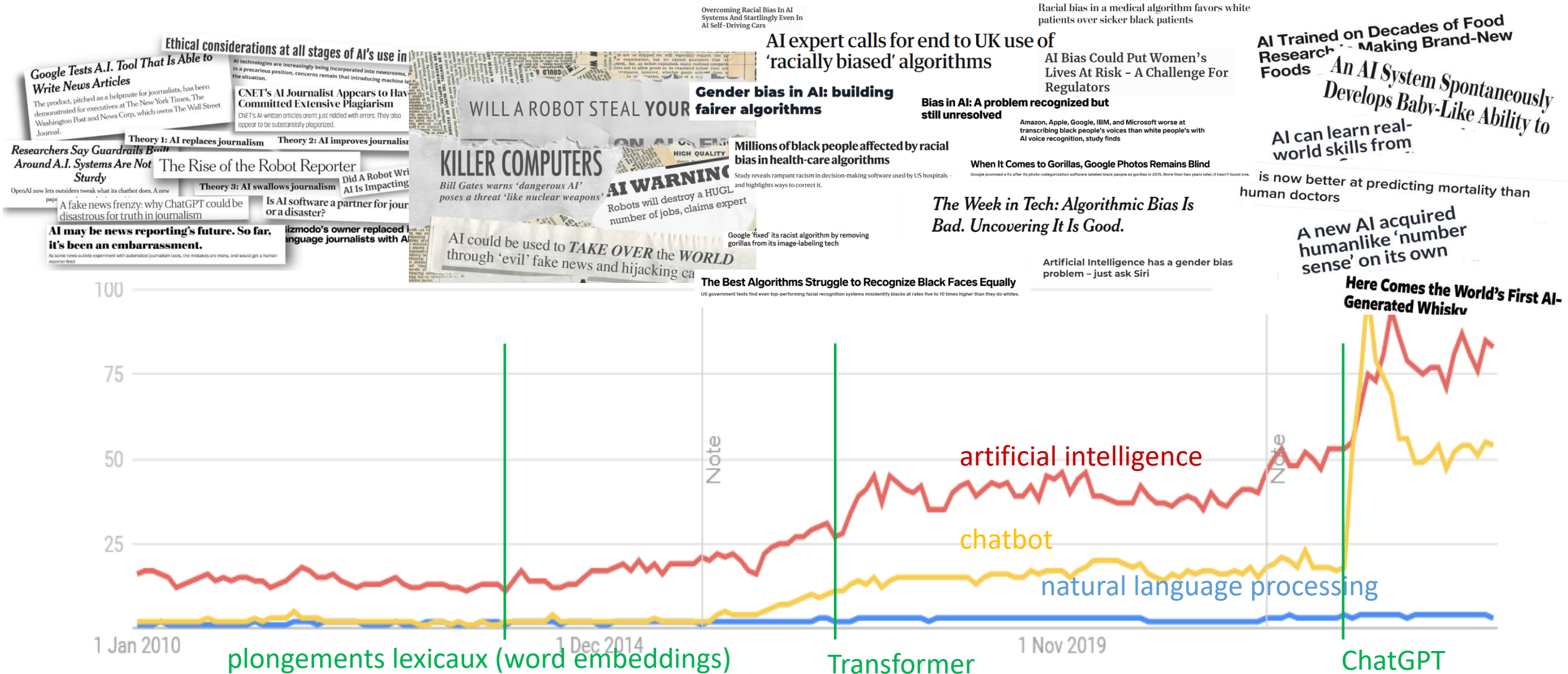


# Traitement automatique de la langue

## Panorama de deux heures

Gábor Bella  
IMT Atlantique  
Juin 2024

# Le TAL (ou plutôt l'IA) dans la conscience publique



Tendances de recherche sur Google, 2010–2024

# Qu'est-ce que le TAL ?

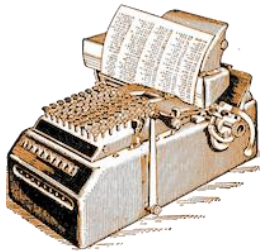
- **Langage** : un système de communication structuré composé de sons, de signes écrits ou de gestes.
- **Langue** : un langage qui a évolué naturellement chez les humains par l'utilisation et la répétition, sans planification consciente ni préméditation.
- **Linguistique computationnelle** : une discipline à cheval entre linguistique et informatique, comprenant :
  - « linguistique assistée par l'ordinateur » : l'étude des langues à l'aide de l'informatique,
  - le traitement automatique de la langue (TAL, en anglais NLP) : permettre à l'ordinateur de résoudre des problèmes sur des données exprimées à travers la langue.

# Pourquoi le TAL ?



Langage naturel

Pas besoin de TAL.



Langage formel

On emploie des logiciels, des logiques, des structures de données.

Pas besoin de TAL.

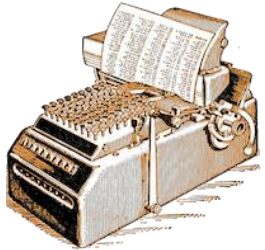
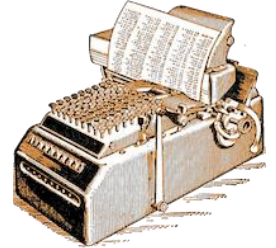


# Pourquoi le TAL ?



## Compréhension du langage naturel (*Natural Language Understanding*)

OBJECTIF : faire travailler l'ordinateur sur des données exprimées en langage naturel.  
EXEMPLES : reconnaissance de l'écriture et de la parole, classification, recherche de l'information, extraction de l'information, analyse de sentiments, etc.



## Génération de langage naturel (*Natural Language Generation*)

OBJECTIF : « traduire » des informations formelles en langage naturel, destiné aux humains.  
EXEMPLES : production de rapports, synthèse de la parole, résumé automatique, etc.



## Compréhension



## Génération

OBJECTIF : faciliter la communication entre humains.  
EXEMPLES : chatbot, système de dialogue bout-à-bout (centrale téléphonique), traduction automatique, résumé automatique, sous-titres automatiques, etc.



# Historique : les débuts

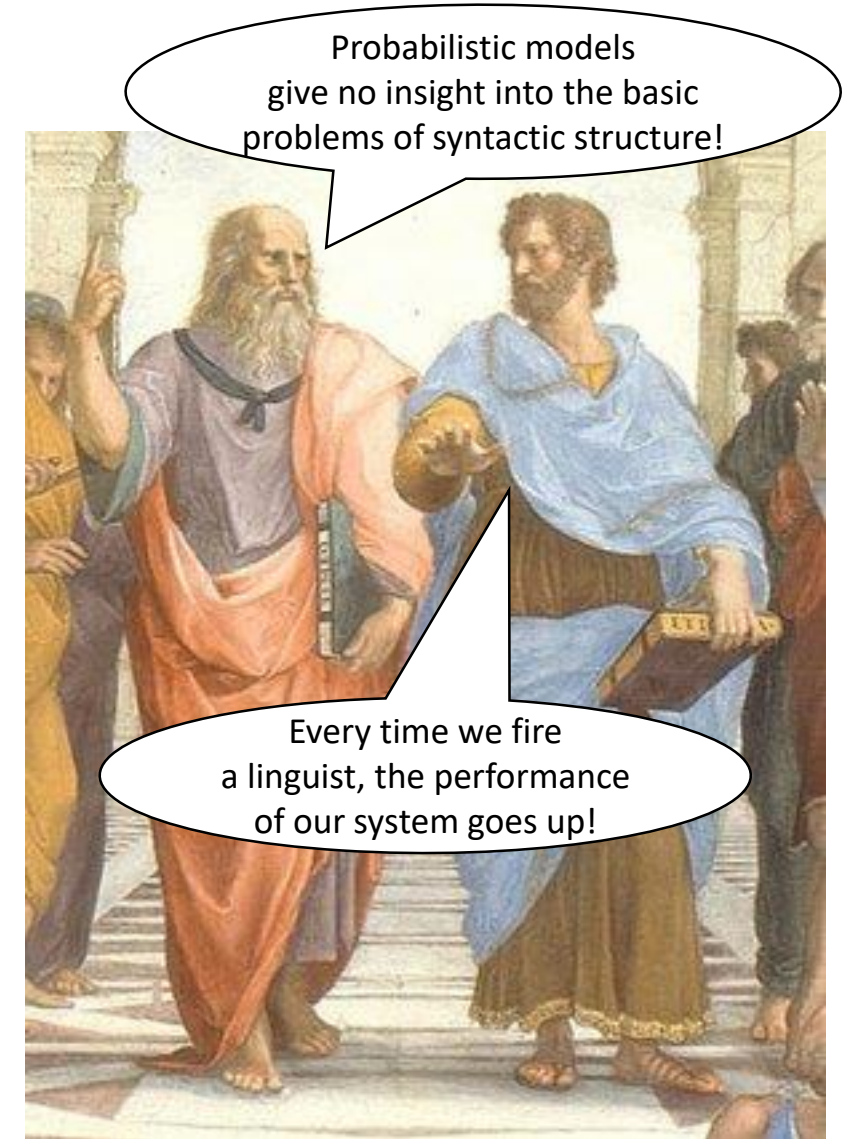
## Motivation initiale : traduction automatique

- Contexte : la guerre froide et la mondialisation ;
- création de la discipline de l'IA autour de 1956 ;
- financement généreux de la part de l'État américain ;
- premiers essais sur des centaines de phrases, limitées à des domaines précis.

## L'hiver de l'IA (années 1960-80)

- rapport ALPAC (1964) : bilan négatif => les fonds étatiques se réduisent ;
- la puissance des ordinateurs n'est pas à la hauteur des idées ;
- crise pétrolière...

## Deux courants opposés : rationalisme et empirisme.



Noam Chomsky, *Syntactic Structures* (2002), p. 17.  
Frederick Jelinek (father of speech recognition), 1985.



# Historique

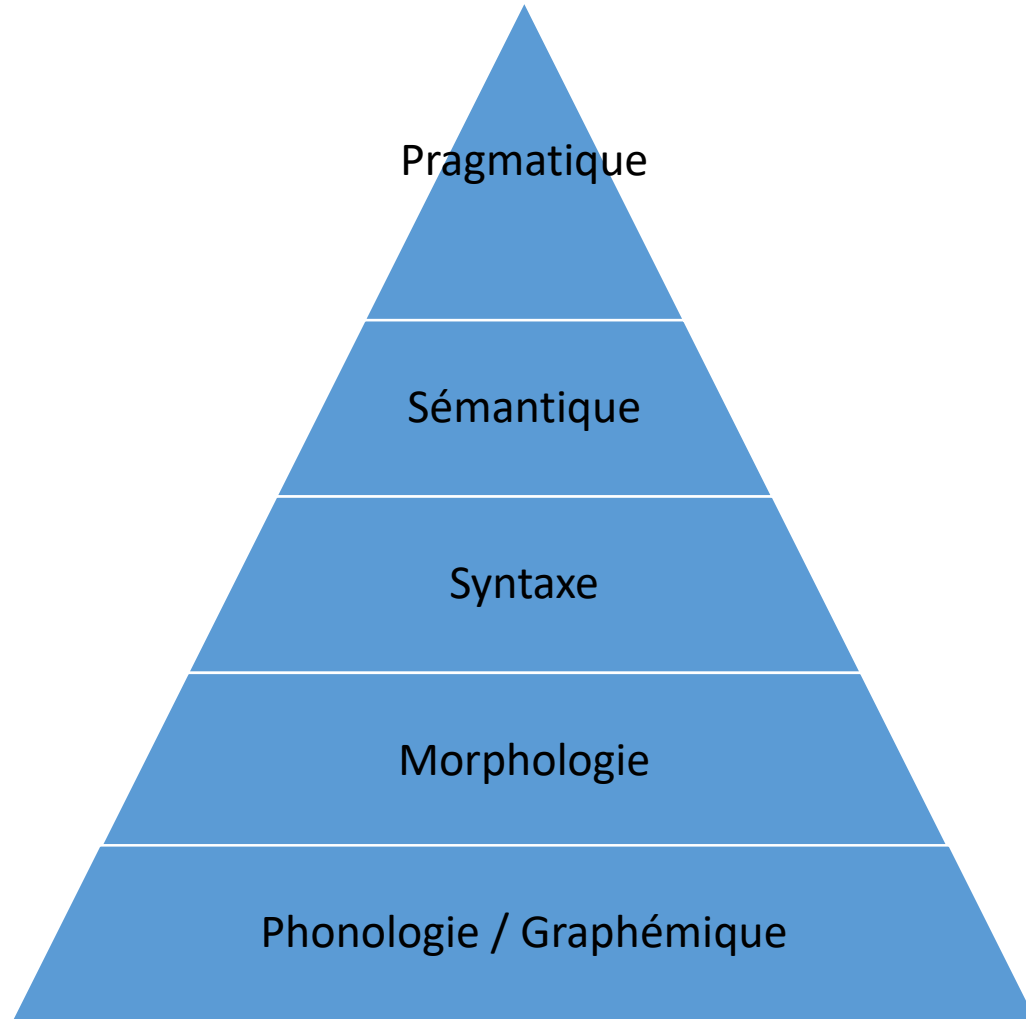
## Rationalisme et structuralisme : approches symboliques

- Le langage possède une structure interne complexe avec des unités fondamentales et des règles de composition (« **compositionnalité** ») : *vert + pomme = pomme verte*.
- Chomsky : langages formels, grammaires génératives => l'analyse de la langue est automatisable, à condition de connaître sa grammaire en profondeur.
- Les limites du structuralisme : **les exceptions et l'irrégularité** sont omniprésentes.

## Empirisme et holisme : approches statistiques et apprentissage automatique

- Une vision « de statisticien » de la langue qui fait émerger la régularité à partir des usages linguistiques : des corpus de textes. C'est le contexte (et non la structure) qui joue le rôle déterminant dans la construction du sens.
- Les réseaux de neurones artificiels ont déjà été définis dans les années 50.  
Les réseaux profonds ont été étudiés dès les années 60.  
Cependant, la puissance de calcul nécessaire n'était là qu'à partir des années 2000–2010.

# Approche structuraliste



Langage parlé / Langage écrit

Comprendre le sens global d'un discours  
en fonction de son contexte

La composition du sens au niveau de la phrase

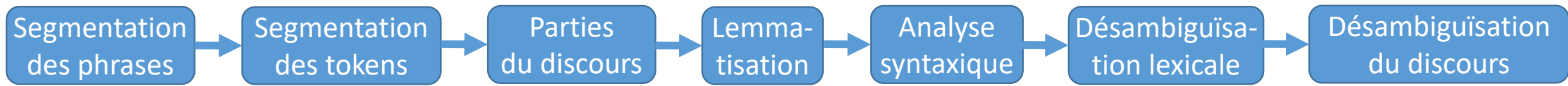
La structure grammaticale des phrases

Les morphèmes (unités du sens) et leur composition :  
inflexion, dérivation, mots composés, etc.

Phonèmes, signes d'écriture, ponctuation



# Exemple : une chaîne de traitement TAL classique



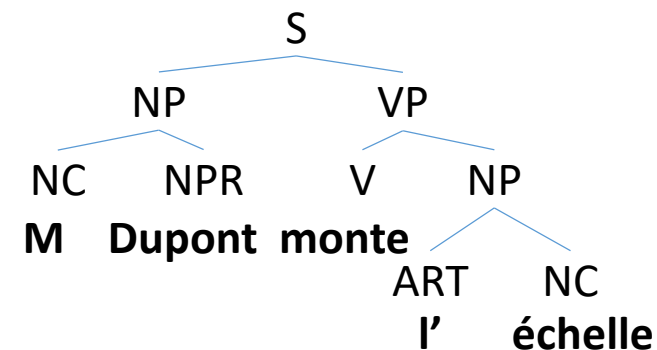
*M. Dupont monte l'échelle. Soudain il tombe.*

*M. Dupont monte l'échelle. | Soudain il tombe.*

*M . Dupont monte l' échelle . | Soudain il tombe .*

*M<sup>NC</sup> .PUNCT Dupont<sup>NPR</sup> monte<sup>V\_PR\_IND</sup> l'<sup>ART</sup> échelle<sup>NC</sup> .PUNCT*

*M<sup>NC</sup> .PUNCT Dupont<sup>NPR</sup> (monter+e)<sup>V\_PR\_IND</sup> la<sup>ART</sup> échelle<sup>NC</sup> .PUNCT*



## monter (verbe)

1. aller en haut
2. porter en haut
3. (cuisine) rendre mousseux
4. ...

## échelle (nom commun)

1. dispositif pour se déplacer en hauteur
2. (musique) série de fréquences
3. (géométrie) proportion de taille
4. ...

$\text{monsieur}(\text{Dupont}) \wedge \text{échelle}^1(x) \wedge \text{monter}^1(\text{Dupont}, x) \wedge \text{tomber}^2(\text{Dupont})$

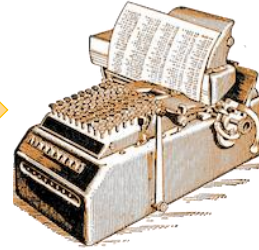


# Avantages de l'approche structuraliste

- **Compositionnalité** : une tâche très complexe se décompose en sous-tâches « moins complexes » ;
- chaque sous-tâche est bien comprise et maîtrisée par l'humain ;
- les sous-tâches sont en général peu onéreuses ;
- passage depuis l'informel vers le formel ;



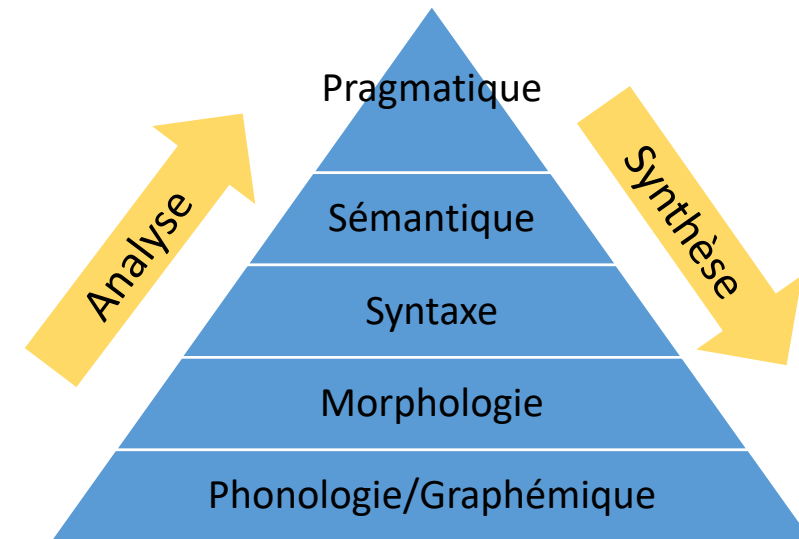
Compréhension du langage naturel



- en principe, le même processus peut se rejouer à l'envers !  
(par exemple, en anglais)



Génération du langage naturel



# Les limites de l'approche structuraliste

- Les composants individuels restent complexes.

*J'ai parlé au prof. Dupont. Puis je suis parti.*

*J'ai parlé au prof. | Dupont. | Puis je suis parti.*

*J'ai parlé au prof. Dupont. | Puis je suis parti.*

*J'ai parlé au prof. Puis je suis parti.*

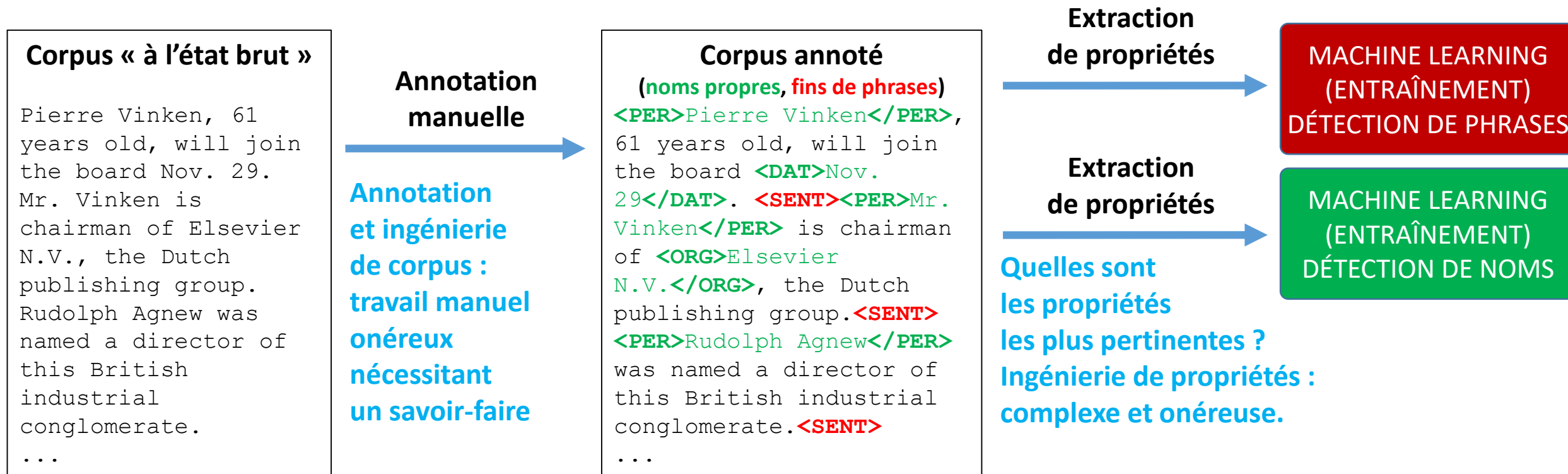
## RÈGLES POUR LA SÉPARATION DES PHRASES

1. POINT+ESPACE+MAJ => nouvelle phrase.
2. Sauf si précédé par "prof".
3. ???!!! ☹☹☹

- Une seule erreur en amont de la chaîne se propage en aval.
- Une phrase consiste en 20 mots en moyenne. Un seul mot mal-analysé peut corrompre l'analyse de la phrase entière.
- Dans le langage naturel, les exceptions et l'irrégularité sont omniprésentes.
- Pas toujours besoin d'une analyse exhaustive, p. ex. pour décider si c'est du *spam*.

# Méthodes statistiques fondées sur les corpus

- Idée : obtenir des connaissances linguistiques à travers l'analyse automatique de corpus.
- Verrous technologiques :
  - besoin de corpus de grande taille (milliers, millions, milliards de phrases) ;
  - besoin de puissance de calcul suffisante ;
  - avant Internet (années 2000), les corpus de grande taille étaient difficiles d'accès.

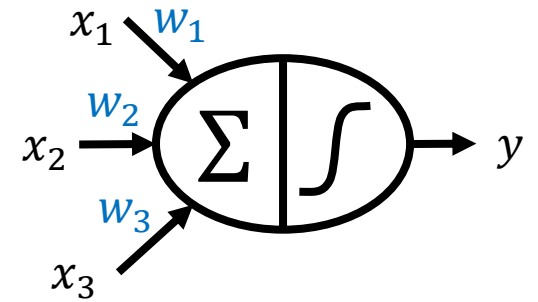
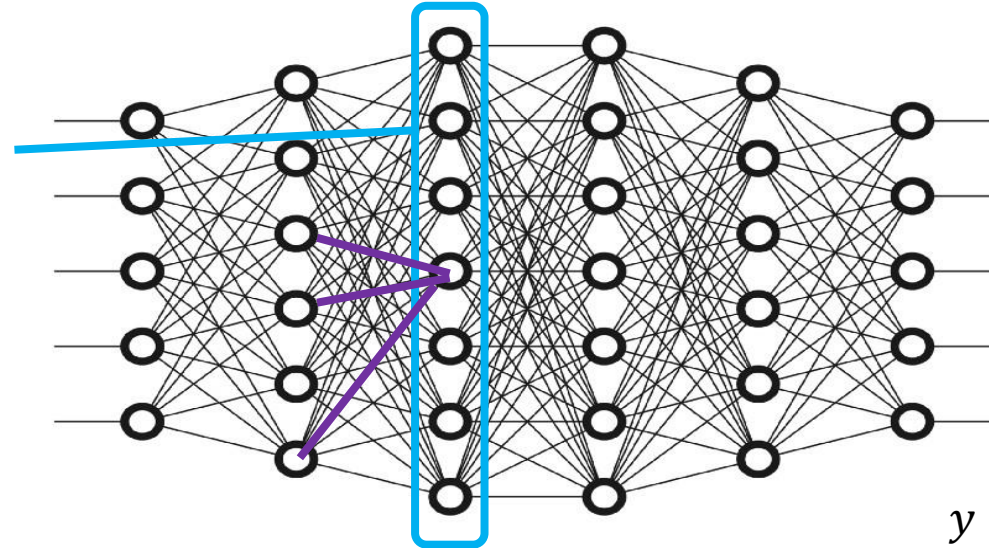


# TAL neuronal « profond » : deux paradigmes

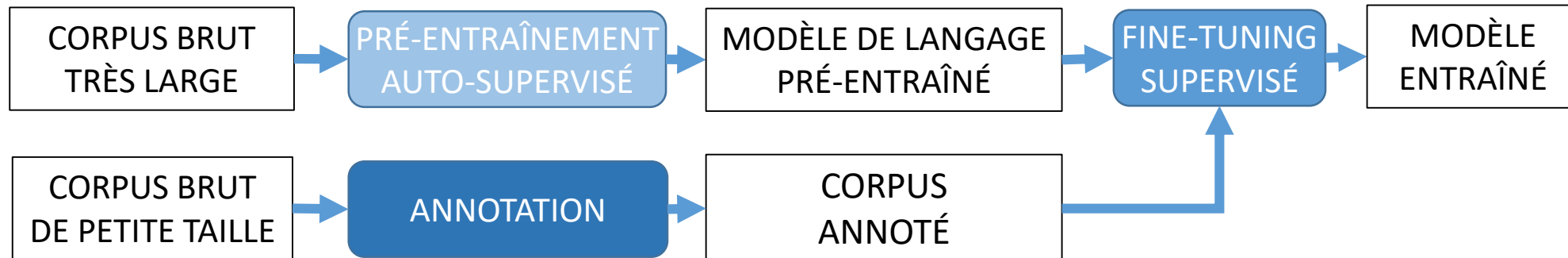


Couche  $\approx$  composant de chaîne de traitement

Poids & biais  $\approx$  sélection de propriétés



$$y = \sigma(w_1x_1 + w_2x_2 + w_3x_3 + b)$$



Le pré-entraînement est 100 % automatique. La supervision d'un corpus plus petit reste nécessaire pour adapter le modèle aux tâches spécifiques.

# Comparaison du TAL symbolique et neuronal

TAL symbolique	TAL neuronal
L'ordinateur suit des instructions précises définies par des experts	L'ordinateur apprend ce qu'il faut faire à partir d'un grand nombre d'exemples, (théoriquement) aucune expertise linguistique n'est nécessaire
Fonctionne avec précision dans des domaines limités, se généralise mal à de nouveaux problèmes/domaines	Se généralise très bien, mais il y a toujours une part aléatoire dans les résultats
Nécessite peu de données	Nécessite une très grande quantité de données d'entraînement
Requiert très peu de ressources de calcul	Requiert énormément de ressources
On comprend bien son fonctionnement	Le réseau de neurones est une boîte noire
Se généralise très mal à de nouveaux problèmes/domaines	Se généralise bien

# Sémantique distributionnelle

“La signification d’un mot est son usage dans le langage.”

—Ludwig Wittgenstein, Recherches philosophiques §43, 1953

- **Hypothèse distributionnelle** : *contexte similaire => signification similaire* ;

*J’ai mis du **gasoil** dans ma voiture. / J’ai mis du **carburant** dans ma voiture.*

- **idée** : on part d’une table rase, on « fait émerger » le sens des mots statistiquement, à partir d’un large corpus de texte ;
- on calcule les « co-occurrences » de mots à l’aide d’une **fenêtre glissante**.

*The cat chased the mouse in the barn.  
A cat chased a rat in the attic.*

mouse	2		1		1			
rat	1		1		1		1	
the		2	2	2	3	1		1
a		2	2		1			1
barn	1				1			
attic	1				1			



	the	cat	chased	mouse	in	barn	a	rat	attic
the		2	2	2	3	1		1	1
cat	2		2				2		
chased	2	2		1			2	1	
mouse	2		1		1				
in	3			1		1	1	1	1
barn	1				1				
a		2	2		1			1	
rat	1		1		1		1		
attic	1				1				

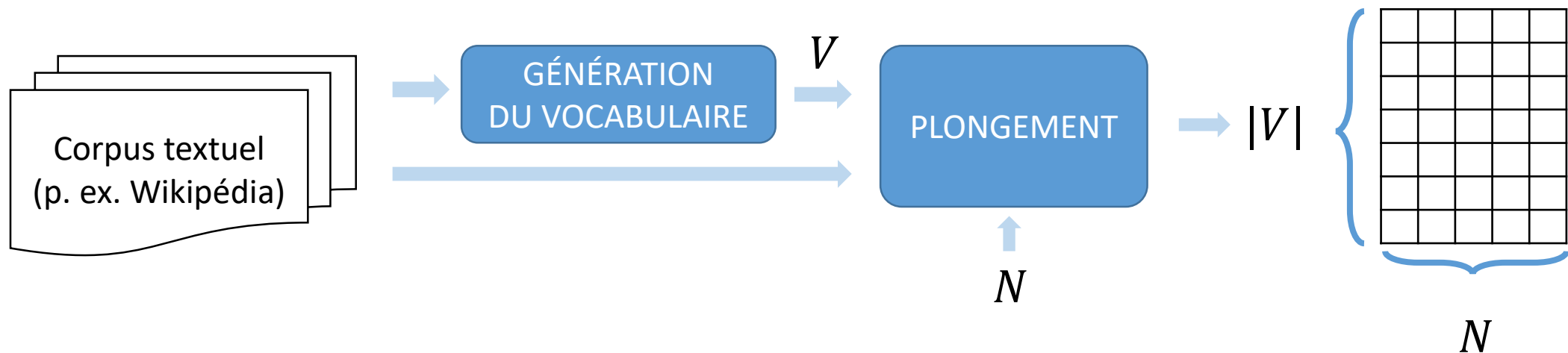
Taille de la fenêtre glissante = +/- 2.



# Les plongements lexicaux (*word embeddings*)

Un plongement lexical est une optimisation (plus efficace, plus compacte) de la matrice de co-occurrences.

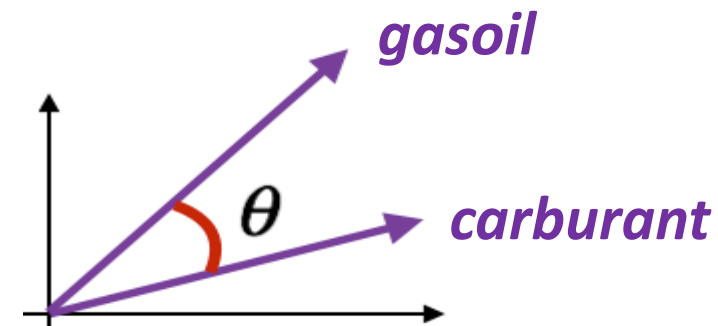
1. À partir d'un large corpus, on génère un **vocabulaire de base**  $V$  en prenant les mots les plus fréquents du corpus (par exemple,  $|V| = 50\,000$ ).
2. Pour chaque mot unique du vocabulaire  $V$ , on calcule sa signification comme un **vecteur de mot**  $\mathbb{R}^N$  avec une dimension  $N$  prédéfinie ("propriétés sémantiques").
3. La **matrice de plongement** est composée d'un nombre  $|V|$  de vecteurs.



# Propriétés des vecteurs de mots

- La similarité des vecteurs est calculée à l'aide de la **similarité cosinus** :

$$\text{sim}(\vec{v}_1, \vec{v}_2) = \frac{\vec{v}_1 \cdot \vec{v}_2}{\|\vec{v}_1\| \|\vec{v}_2\|} = \cos(\theta)$$



- Qu'est-ce que cela signifie ? Rien d'autre que les deux mots apparaissent dans des contextes similaires :

*gasoil* « est similaire à » *carburant* mais aussi *gasoil* « est similaire à » *voiture*

- La similarité cosinus N'EST PAS une similarité sémantique !  
Elle mesure plutôt l'existence d'une certaine relation entre les sens.
- La **polysémie** n'est pas représentée au sein des plongements :

“Il **est** né à l'**est**.”

Les deux significations d'« **est** » sont confondues au sein d'un seul vecteur.

- Des **opérations arithmétiques** (+, −) sont possibles avec les vecteurs de mots :

*king* − *man* + *woman* ≈ *queen*

# 'Using Word Embeddings...'

>10,000 articles sur Google Scholar:

- **classification :**
  - twitter election classification
  - classifying mobile applications
  - hierarchical text classification
  - topic discovery for short texts
  - computing personality traits from tweets
  - fake review detection
  - aggressive language identification
  - context-aware misinformation detection
  - sentiment analysis
- **extraction de l'information :**
  - enhance keyword identification for scientific publications
  - unsupervised acronym disambiguation
  - automatic keyphrase extraction
  - automated disease cohort selection from Electronic Health Records
- **recherche de l'information :**
  - enhancing question retrieval in community question answering
  - automatic query expansion
  - full text search for legal document collections
- **apprentissage de connaissances automatique :**
  - learn a better food ontology
  - ontology enrichment
- **éthique :**
  - investigate cultural biases
  - examine gender bias in Dutch newspapers, 1950-1990
  - improve the privacy of clinical notes
  - analyze how universities conceptualize "diversity" in their online institutional presence
  - measure ethnic stereotypes in news coverage
- **linguistique :**
  - unsupervised morphology induction
  - predict the literal or sarcastic meaning of words
- **autres applications :**
  - visualization of medical concepts
  - exploring semantic representation in brain activity
  - network intrusion detection
  - predict stock price movements
  - legal assistance
  - 3D lithological mapping of borehole descriptions

# Modèles de langage

- L'ordre des mots est important. *L'homme a tué le lion. / Le lion a tué l'homme.*
- Or, les plongements de mots ne tiennent pas compte de la syntaxe.

- La « **modélisation du langage** » consiste à prédire le token suivant à partir de  $n$  tokens précédents :

$$\hat{w}_{n+1} = \operatorname{argmax}_{w_{n+1}} P(w_{n+1} | w_n, \dots, w_1)$$

- Plus  $n$  est grand, plus il faut de l'évidence statistique  $\Rightarrow n$  est limité.
- De nombreux usages : texte prédictif sur portable, generation d'histoires, etc.

*Je bois mon café avec du lait et du ... [?]*

*Longtemps, je me suis couché de bonne ... [?]*

- C'est une très bonne base de « connaissances générales » sur le langage naturel.
- Il est très utile de générer des corpus auto-supervisés d'une manière automatique :

*I drink my coffee with milk and ~~sugar~~*



# Dépendances lointaines au sein du langage naturel

Modélisation de langage

Puisque j ' habite en France , je parle très bien ...

Résolution d'anaphores,  
traduction automatique

The animal did not cross the street as it was too tired.

L'animal n'a pas traversé la rue car il était trop fatigué.

The animal did not cross the street as it was too wide.

L'animal n'a pas traversé la rue car elle était trop large.

**Les schémas de Winograd :**  
la résolution de l'anaphore dépend du sens de la phrase entière. C'est un test classique, considéré très difficile, pour évaluer l'intelligence de l'IA.

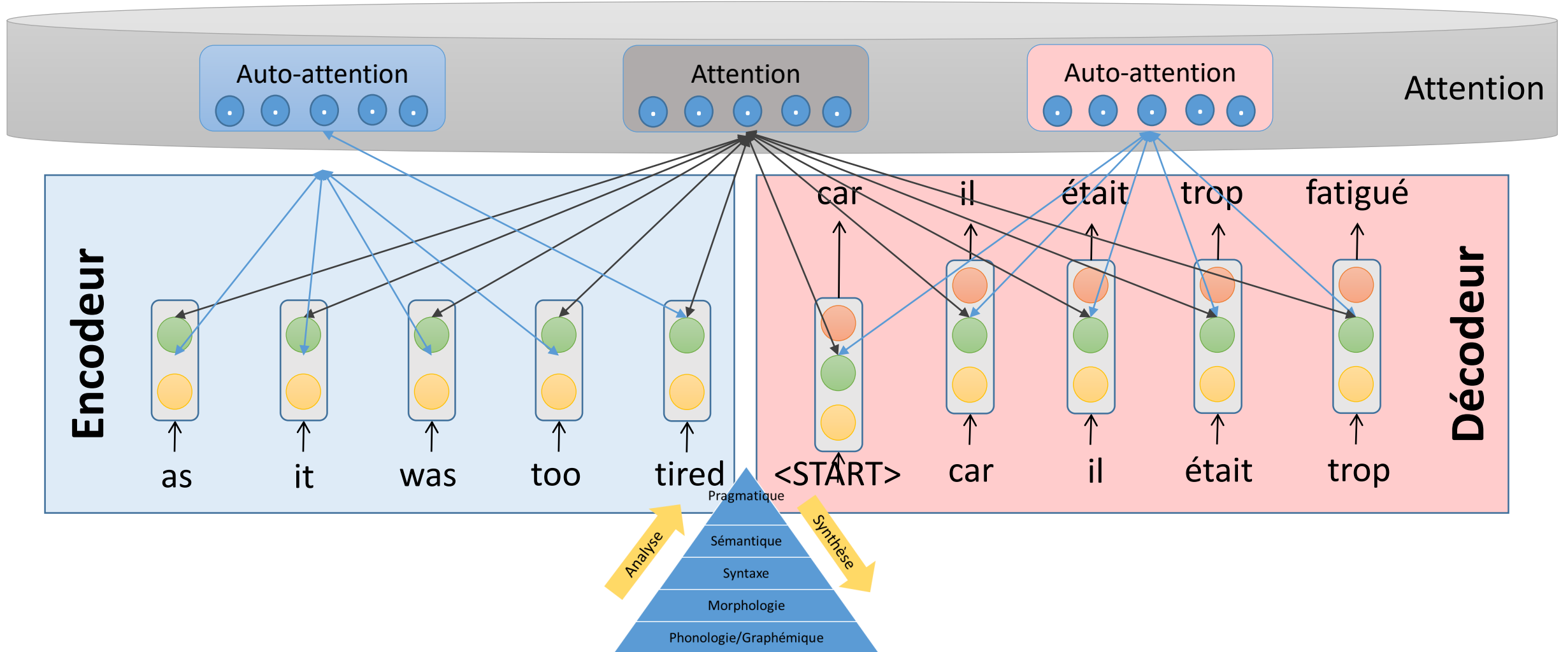
Bien résolu avec des performances proches de l'humain depuis 2019, à l'aide du *Transformer*.

[Kocijan et al., 2023: The defeat of the Winograd Schema Challenge.](#)

[Comment Google se débrouille-t-il ?](#)

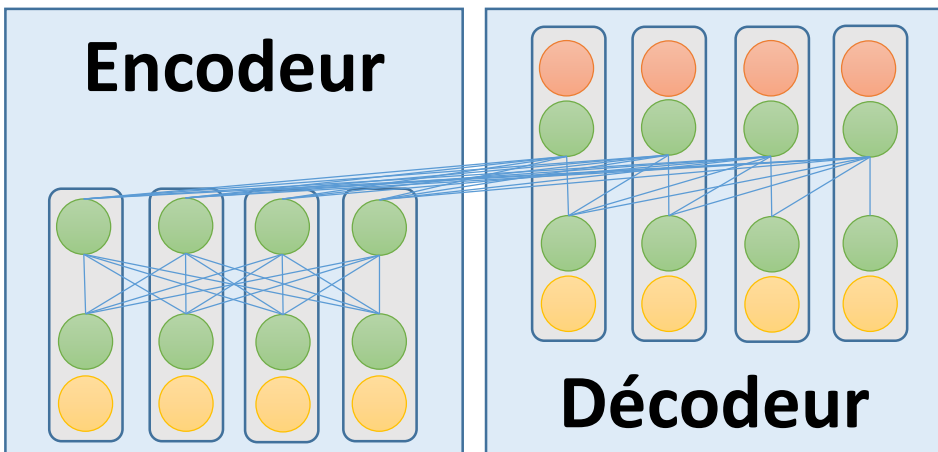
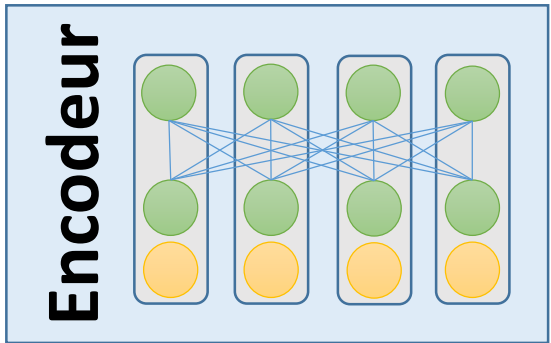
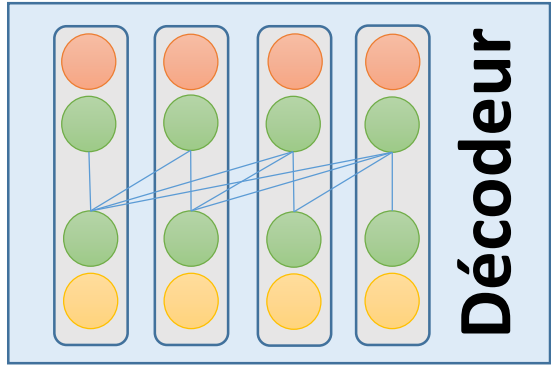
(Cliquer pour essayer.)

# Le mécanisme d'attention dans le Transformer



- On retrouve l'analyse (= encodeur) et la synthèse (= décodeur) du TAL symbolique.
- L'encodeur « comprend » le texte d'entrée, le décodeur « génère » le texte de sortie.
- Le mécanisme d'attention permet de retenir des dépendances au sein de longues textes.

# Configurations Encodeur-Décodeur



- Tâche d'entraînement classique : modélisation de langage ;  
Quel âge as - <?>
- applications : génération de langage (text prédictif, histoires, inférence en langage naturel, etc.) ;
- modèles « auto-régressifs », comme le Generative Pretrained Transformer (Radford et al., 2018).
- Tâche d'entraînement classique : modélisation masquée ;  
C ' est <MASK> irr## ési## sti## ble## <MASK> bon
- démo : <https://demo.allennlp.org/masked-lm>
- applications : tâches de compréhension variées (analyse de sentiment, extraction de l'information, recherche de l'information, etc.) ;
- modèles « auto-encodeurs », come BERT, RoBERTa, etc.
- Tâche d'entraînement classique : corruption de texte  
Thank you ~~for inviting~~ me to your party ~~last~~ week.  
Thank you me to your party week.
- applications : traduction automatique, résumé, questions/réponses, etc. ;
- modèles de langages hybrides, comme T5.



# « Prompting » : principes de base

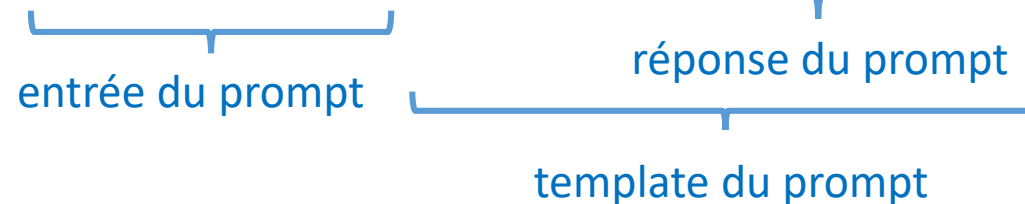
- Pour GPT 2/3/4, fine-tuning est exécuté à travers « l'ingénierie de prompt » : **les tâches en aval – classification de textes, questions/réponses, traduction, etc. – sont reformulés en tant que tâches de modélisation de langage.**
- Exemple : analyse de sentiments.

Fine-tuning conventionnel :

Prompting :

*I love this movie.* => `class = [+ : 1.0, 0 : 0.0, - : 0.0]`

*I love this movie. Overall, it was a **good** movie.*

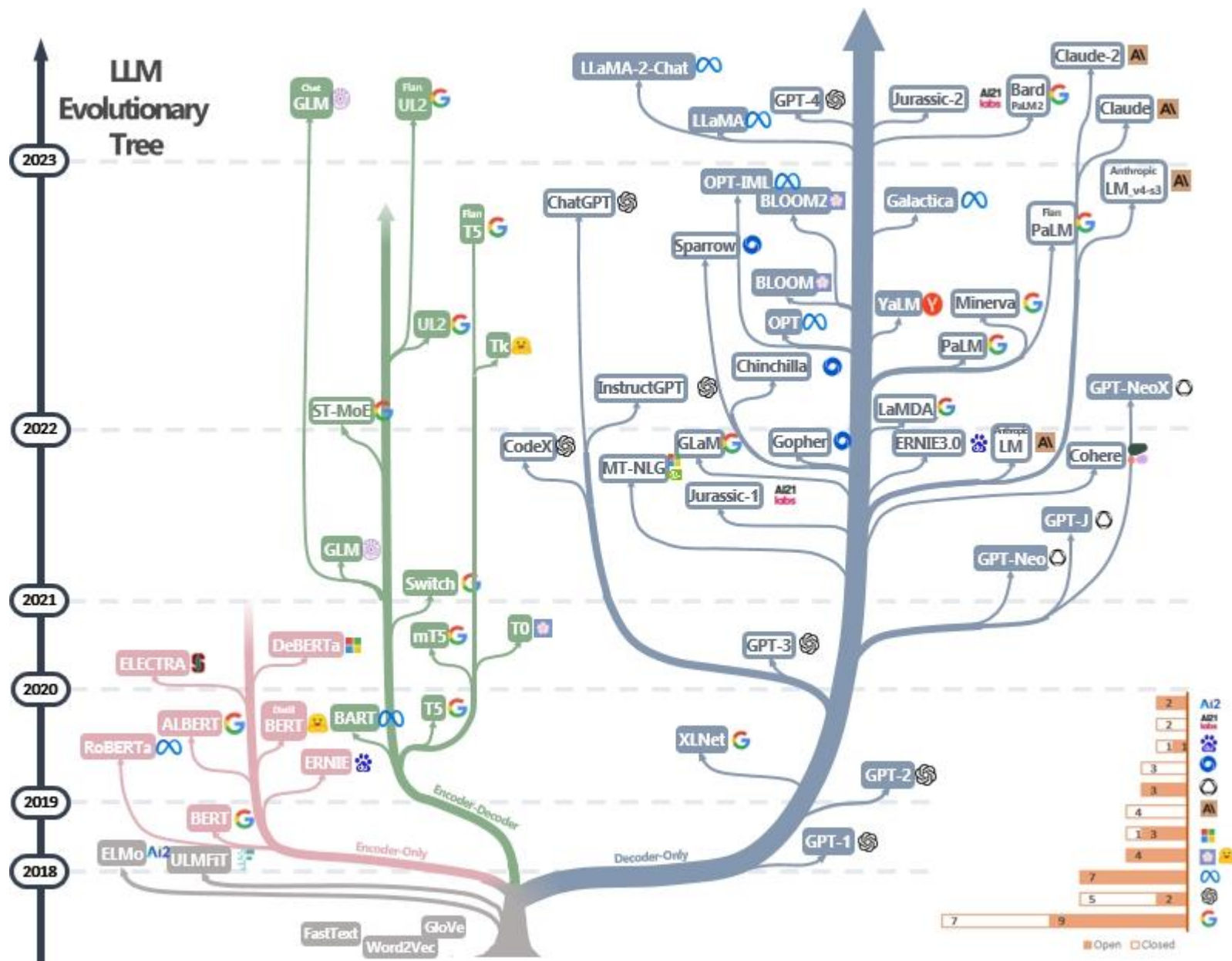
  
entrée du prompt      réponse du prompt  
template du prompt

- La tâche consiste à prédire la réponse ('good'), en tant que prediction de texte masqué.
- Pourquoi ?
  - Plus besoin de changer d'architecture neuronale pour le fine-tuning.
  - La tâche est très similaire à la tâche de pré-entraînement (modélisation de langage), on exploite directement les connaissances déjà pré-entraînées.
  - Bien adapté à l'interaction homme-machine : questions/réponses, chatbot, etc.

# Examples de « Prompting »

Tâche	Entrée ([X])	Template	Réponse ([Z])
Analyse de sentiments	I love this movie.	[X] The movie is [Z].	great fantastic
Classification de sujets	He prompted the LM.	[X] The text is about [Z].	sports science
Classification d'intentions	What is the taxi fare to Denver?	[X] The question is about [Z].	quantity city
Extraction de sentiments	Poor service but good food.	[X] What about service? [Z].	Bad Terrible
Inférence en langage naturel	[X1]: An old man with ... [X2]: A man walks ...	[X1]? [Z], [X2]	Yes No
Reconnaissance d'entités nommées	[X1]: Mike went to Paris. [X2]: Paris	[X1][X2] is a [Z] entity.	organization location
Résumé automatique	Las Vegas police ...	[X] TL;DR: [Z]	The victim ... A woman ...
Traduction automatique	Je vous aime.	French: [X] English: [Z]	I love you. I fancy you.
Similarité textuelle	[X1]: A man is smoking. [X2]: A man is skating.	[X1][Z], [X2]	Yes No

# L'évolution des LLM



# Consommation en énergie et émission de CO<sub>2</sub>

Opération	Architecture neuronale	Temps	kWh × PUE	CO <sub>2</sub> eq	Équiv. émission voiture
entraînement unique	Transf. Base (Vaswani et al., 2017)	12 h	27 kWh	12 kg (US)	80 km
entraînement unique	Transf. Big (Vaswani et al., 2017)	84 h	201 kWh	87 kg (US)	580 km
entraînement unique	BERT Base (Devlin et al., 2019)	79 h	1507 kWh	652 kg (US)	4,347 km
NAS (algo génétique)	Evolved Transf. (So et al., 2019)		7,500 kWh	3,200 kg (US)	21,300 km
entraînement unique	GPT-3 (Patterson et al., 2021)		1,287,000 kWh	552,100 kg (US)	durée de vie de 10 voitures
entraînement unique	BLOOM (Luccioni et al., 2023)		433,196 kWh	24,690 kg (FR)	½ durée de vie d'une voiture
entraînement +expérimentation	BLOOM (Luccioni et al., 2023)		1,163,088 kWh	66,290 kg (FR)	> durée de vie d'une voiture
inférence, par jour	BLOOM (Luccioni et al., 2023)	1 jour		19 kg (US)	127 km

- PUE : Power Usage Effectiveness, on inclut la perte liée à l'usage du *datacentre*.
- CO<sub>2</sub>eq : émissions équivalentes en CO<sub>2</sub>.
- BLOOM est considéré similaire en performances à GPT-3.

# Questions ouvertes, sujets pressants

- « **Hallucinations** » : l'IA prétend de connaître la réponse mais elle affabule  
=> problèmes de confiance, d'explicabilité, de responsabilité.
- **Équité et biais** : l'IA privilégie certains groupes sociaux (genres, races, langues, classes sociales, etc.)  
=> est-ce la faute de la société (les données d'entrée) ? de l'ingénieur ?
- **Intelligence** : est-ce que ChatGPT (GPT-4, etc.) est « vraiment intelligent » ? mais que signifie l'intelligence ?
- **Enseignement** : est-ce que ChatGPT peut devenir le couteau suisse des élèves ?  
Trouver sa place dans l'éducation ?  
=> Le risque de ne jamais acquérir des compétences clés ? De produire des employés dépendants de l'IA et donc remplaçables par celle-ci ?
- **Plagiat** : l'IA réutilise massivement des œuvres d'art, du code source, sans attention aux droits d'auteur ;  
=> pourtant, l'usage des modèles IA les plus performants n'est pas gratuit.
- **Dangers sociétaux** : est-ce que l'IA va nous remplacer, nous rendre inutiles ? C'est déjà un peu le cas.  
=> Quelle société construire ? En termes de protection des travailleurs, du niveau de vie, du sens de la vie ?
- **Guerre et terrorisme** : la facilité de connecter une IA puissante avec des engins destructeurs simples (par exemple : drone + explosifs).

# TF-IDF

- Le **TF-IDF (Term Frequency–Inverse Document Frequency)** est une mesure statistique de l'importance d'un terme au sein d'un corpus de documents.
- Usages :
  - **extraction de mots clés** : trouver les termes qui caractérisent le mieux un document ;
  - **recherche de l'information** : quelles sont les pages Web qui correspondent le mieux aux termes de la requête ?
  - **détection de spam** : quelles sont les termes les plus souvent utilisés dans les messages indésirables ?
- Idée #1 : quels sont les mots qui apparaissent le plus souvent dans le document ?
  - on suppose avoir segmenté les mots préalablement ;
  - dans chaque document  $d$  de notre corpus  $D$ , on compte le nombre d'occurrences  $f$  de chaque mot unique (on laisse tomber les déclinaisons, les conjugaisons, etc.) et on retient les mots les plus pertinents ;
  - pour ne pas pénaliser les documents courts, on normalise par le nombre de mots.



# TF-IDF

Si  $f$  signifie le nombre d'occurrences d'un terme  $t$  au sein d'un document  $d$  au sein de notre corpus  $D$ , alors le Term Frequency se calcule comme suit :

$$\text{tf}(t, d) = \frac{f(t, d)}{\sum_{t' \in d} f(t', d)}$$

- Problème :

Liste des 45 mots les plus fréquents<sup>1</sup> :

1 le/la/les (déterminants)	16 dans (préposition)	31 plus (adverbe)
2 de (préposition)	17 en (préposition)	32 dire (verbe)
3 un/une (déterminants)	18 du (déterminant)	33 me (pronom)
4 être (verbe)	19 elle (pronom)	34 on (pronom)
5 et (conjonction)	20 au (déterminant)	35 mon (déterminant)
6 à (préposition)	21 de/des (déterminants)	36 lui (pronom)
7 il (pronom)	22 ce (pronom)	37 nous (pronom)
8 avoir (verbe)	23 le (pronom)	38 comme (conjonction)
9 ne (adverbe)	24 pour (préposition)	39 mais (conjonction)
10 je (pronom)	25 pas (adverbe)	40 pouvoir (verbe)
11 son (déterminant)	26 que (pronom)	41 avec (préposition)
12 que (conjonction)	27 vous (pronom)	42 tout (adjectif)
13 se (pronom)	28 par (préposition)	43 y (pronom)
14 qui (pronom)	29 sur (préposition)	44 aller (verbe)
15 ce (déterminant)	30 faire (verbe)	45 voir (verbe)



# TF-IDF

Idée #2 : donner préférence aux termes spécifiques au document.

Si  $D$  signifie l'ensemble de documents dans notre corpus, alors l'*Inverse Document Frequency* d'un terme  $t$  se calcule comme suit :

$$\text{idf}(t, D) = \log \frac{1 + |D|}{1 + |\{\forall d_i \in D : t \in d_i\}|}$$

En supposant un logarithme à base 10 :

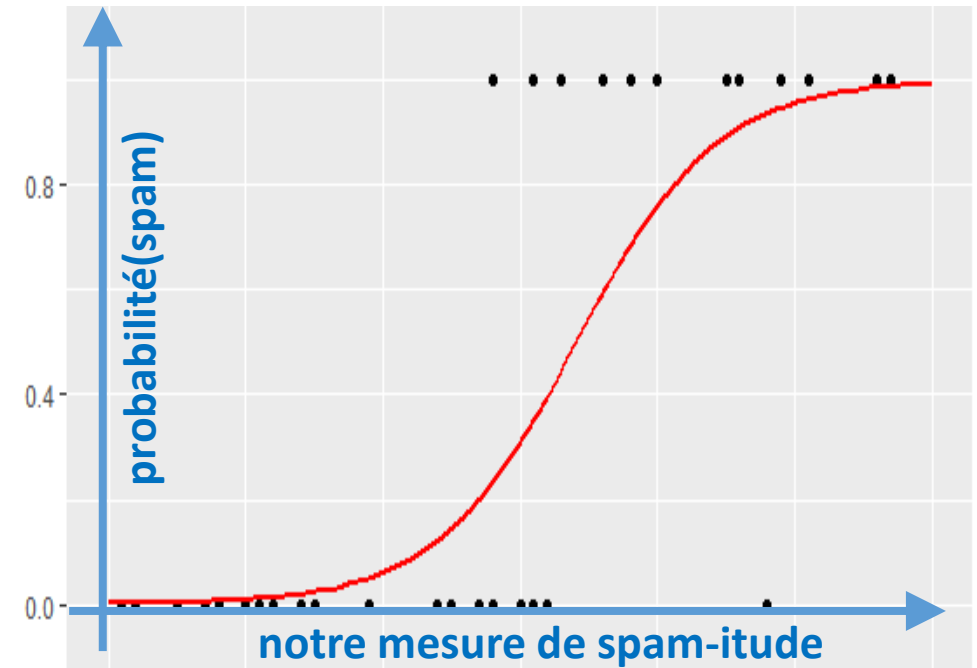
- Si  $t$  apparaît dans tous les documents :  $\text{idf}(t) = \log 1 = 0$ .
- Si  $t$  apparaît dans 50% des documents :  $\text{idf}(t) \approx \log 2 = 0,301$ .
- Si  $t$  apparaît dans 10% des documents :  $\text{idf}(t) \approx \log 10 = 1$ .
- Si  $t$  apparaît dans 1% des documents :  $\text{idf}(t) \approx \log 100 = 2$ .

$$\text{tfidf}(t, d, D) = \text{tf}(t, d) \cdot \text{idf}(t, D)$$

Démonstrateur (anglais) : <https://demonstrations.wolfram.com/TermWeightingWithTFIDF/>

# Régression logistique

- Pour détecter les spam, une estimation linéaire n'est pas très pratique.
- On préfère une décision binaire : oui ou non.
- Solution : on transforme la fonction linéaire en fonction qui retourne des valeurs entre 0 et 1.
- La **régression logistique** « explique » les données à l'aide d'une fonction dite logistique, et non linéaire.



$$y = ax + b \quad p = \frac{1}{1+e^{-y}} \quad \text{fonction logistique}$$

Si  $y = (-\infty; +\infty)$ ,  $p = (0; 1)$ , ce qui convient parfaitement pour modéliser des probabilités.