MINES PARIS | PSL | Centre de recherche en informatique

# Traitement Automatique des Langues
# *Natural Language Processing*
# Notes de cours

Georges-André Silber, CRI, Mines Paris, Université PSL
georges-andre.silber@minesparis.psl.eu
https://mines.paris/nlp

ES3A_MES-07, 2025–2026
Version du 11 décembre 2025

## Table des matières

**1 Logistique du cours** **1**

**2 Introduction** **2**
2.1 L'IA dans la pop culture . . . 2
2.2 Qu'est-ce que le TAL ? . . . 2
2.3 Histoire sélective du TAL . . . 2
2.4 Apprentissage automatique statistique . . . 3
2.4.1 Analyse syntaxique statistique . . . 4
2.4.2 Modèle de langue : distribution de probabilité . . . 4
2.4.3 Création d'un modèle de langue par comptage : *n*-grams . . . 4
2.4.4 Approches par prédiction . . . 5
2.5 Apprentissage automatique neuronal . . . 5
2.6 4e révolution de l'accès à l'information . . . 9

**3 Grammaires formelles** **9**
3.1 Informatique et langages . . . 9
3.2 Langage formels, Hiérarchie de Chomsky . . . 10
3.3 Langages rationnels . . . 10
3.4 Langages algébriques . . . 10
3.5 Langages contextuels . . . 10
3.6 Langages récursifs . . . 10
3.7 La machine de Turing . . . 10
3.8 Expressions régulières . . . 11

## 1 Logistique du cours

Le cours de Traitement Automatique des Langues (TAL), en anglais *Natural Language Processing* (NLP), se compose de 16 séances d'1h30. Ces séances seront partagées entre cours magistraux et travaux pratiques, principalement en Python 3.

Votre note finale au cours sera la moyenne pondérée des notes obtenues pour chaque TP que vous me rendrez et du projet final, ce dernier ayant une pondération de 1/2. Un principe général est que vous me rendiez tous

les TP, avec un joker possible. Les TP seront à rendre en fin de séance, avec la possibilité de le rendre plus tard, mais impérativement avant la séance suivante.

Les moyens de calcul que vous pourrez utiliser pour les travaux pratiques et le projets seront composés de vos propres machines, de machines des mines et de moyens de calculs loués chez Scaleway pour ce cours.

Vous aurez le choix entre plusieurs projets (6 dans l'édition 2024–2025 du cours), avec un niveau technique différent entre chaque projet.

# 2 Introduction

## 2.1 L'IA dans la pop culture

Le développement récent de l'IA et du NLP est une révolution culturelle, notamment depuis le "choc" chatGPT de 2022. L'IA a cependant une longue histoire, notamment dans le domaine de la langue, que l'on peut illustrer par des œuvres des la "pop culture" :

— Le "*turc mécanique*" (https://fr.wikipedia.org/wiki/Turc_mÃľcanique) ;
— Le *test de Turing* (Computing Machinery and Intelligence) [**?**] ;
— *Blade Runner* (Ridley Scott, 1982). Adaptation du livre "*Do Androids Dream of Electric Sheeps ?*" de Philip K. Dick (1966). Terre dévastée en 2019, à Los Angeles, il reste des humains qui n'ont pas pu ou pas choisi d'aller sur mars. Test Voight-Kampff pour détecter les *réplicants*, adaptation du test de Turing ;
— "*2001 l'odyssée de l'espace*" de Stanley Kubrick (1968) et son IA HAL 9000 : https://youtu.be/ARJ8cAGm6JE.
— "*Terminator*" de James Cameron (1984). IA militaire Skynet qui a détruit la planète. Androïde T-800 qui est renvoyé dans le passé pour détruire la mère du futur leader de la resistance (Sarah Connor). https://www.youtube.com/watch?v=QaagRs5pX_E
— "*Wargames*" de John Badham (1984). IA militaire WOPR (War Operation Plan Response), conçue pour pallier la défaillance des humains dans la décision de déclenchement du feu nucléaire.https://youtu.be/7R0mD3uWk5c, https://youtu.be/tGNBdjVO04Y, https://youtu.be/F7qOV8xonfY.
— "*Her*" de Spike Jonze (2013).

## 2.2 Qu'est-ce que le TAL ?

Fondamentalement, il s'agit d'apprendre les langues aux machines.

Applications du TAL (transparents MVA 2024, de 8 à 27) (transparents originaux).

Challenges (transparents MVA 2024, de 63 à 87).

Les grands modèles de langues réalisent très bien la plupart de ces tâches (les LLM sont à l'état-de-l'art, souvent abrégé SoTA comme *State of The Art*).

## 2.3 Histoire sélective du TAL

La langue humaine est cœur de l'*Intelligence Artificielle* : "*reproduire (imiter) informatiquement des comportements qui font traditionnellement appel à l'intelligence humaine*". L'utilisation du langage est l'un de ces comportements.

— 1933, les machines à traduire de Georges Artsrouni
— 1940–1949, progrès en théorie des automates, langages formels, probabilités, théorie de l'information. Travaux de Booth, Weaver, Richens
— 1949, Mémorandum "Translation" de Warren Weaver
— 1950, *Computing Machinery and Intelligence* (A. Turing)
— 1954, expérience Georgetown-IBM, traduction du russe vers l'anglais
— 1955, introduction du terme "*Artificial Intelligence*" par John Mac Carthy à la conférence de Dartmouth

— 1958, "The History and Recent Progress of Machine Translation" par A.D. Booth
— 1966, ELIZA (Joseph Weizenbaum)
— 1968, SHRDLU (PhD de Terry Winograd au MIT)
— 1970–2000, « ontologies conceptuelles », approches symboliques
— 1988, approches par apprentissage automatique statistique
— 2010, approches par apprentissage automatique neuronal
— 2018, BERT (Google)
— 2020, GPT-3 (OpenAI)
— 2022, ChatGPT (OpenAI)
— 2023, Poids et code ouvert : llama (Meta)
— 2025, Multimodalité : Gemini (Google), Mistral AI, Claude (Anthropic), GPT 5 (OpenAI)

**Memorandum "translation" (1949)** Grand impact scientifique et politique :

1. l'ambiguïté peut-être résolue grâce au contexte ;
2. la traduction a une solution formelle, ou solution mécanique, car le langage a une structure logique ;
3. les méthodes cryptographiques s'appliquent (par exemple, l'anglais peut-être vu comme du russe chiffré) ;
4. le sens peut-être représenté indépendamment de la langue.

**Expérience Georgetown-IBM (1954)** 60 phrases traduites du russe en anglais. 250 mots et 6 règles syntaxiques. Enthousiasme considérable : "*five, perhaps three years*" (IBM), "*dans 15 ans on estime que des traducteurs électroniques pourront être utilisés dans les assemblées internationales comme les nations unies*" (Le Monde). "The Brain" dans les articles de presse. Quelques retours plus nuancés : "*a vast amount of work is still needed*" (Neil Macdonald, 1954), "*a* kitty hawk *flight* (J. Hutchins, 2006).

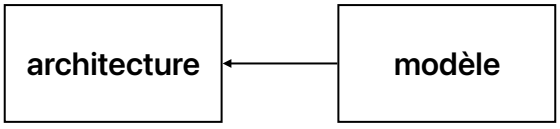
**Conférence de Dartmouth (1955)** John McCarthy est également l'inventeur du temps partagé, du langage fonctionnel LISP (LISt Processing), prix Turing 1971. L'un des pères fondateurs de la discipline. Le terme "*Artificial Intelligence*" a été utilisé pour la première fois dans le cadre de la session de travail de Dartmouth, connue comme le *Darthmouth Workshop*, où pendant huit semaines se sont réunis McCarthy, Minsky, Rochester et Shannon : "*the study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it.*"

## 2.4 Apprentissage automatique statistique

Deux familles d'approches pour faire faire un calcul à une machine :

— règles (approche symbolique) : on écrit un programme explicite, fixe, décrivant la manière exacte d'effectuer les opérations ;
— apprentissage automatique supervisé : on montre des exemples à la machine du résultat à obtenir, et la machine apprend à reproduire le résultat (trouve elle-meme un "programme" réalisant le calcul).

En apprentissage automatique, on a une architecture fixe et un modèle variable qui comporte des paramètres, tel que décrit dans le schéma ci-dessous.




"Apprendre" → Trouver les "meilleures" valeurs possibles pour les paramètres. Les paramètres du modèles sont chargés dans l'architecture pour reproduire les résultats appris.

### 2.4.1 Analyse syntaxique statistique

Supposons que l'on cherche à traduire du français vers l'anglais : on veut trouver la meilleure traduction $y$ en anglais d'une phrase $x$ en français. On modélise la qualité d'une traduction par une probabilité $\mathrm{P}(y|x)$. On cherche la meilleure traduction $y$, que l'on modélise par $\mathrm{argmax}_y \mathrm{P}(y|x)$. L'application du théorème de Bayes, en faisant l'hypothèse raisonnable que $\mathrm{P}(x)$ n'est pas nul puisque c'est la phrase de départ :

$$\mathrm{P}(y \mid x) = \frac{\mathrm{P}(x \mid y).\mathrm{P}(y)}{\mathrm{P}(x)}$$

nous permet de décomposer cette probabilité en deux composantes séparées :

$$\mathrm{argmax}_y \mathrm{P}(x \mid y).\mathrm{P}(y)$$

en ne s'intéressant qu'aux composantes où intervient $y$.

On a deux modèles, un *modèle de traduction* $\mathrm{P}(x \mid y)$, qui modélise comment les mots et séquences de mots doivent être traduits pour préserver le sens (appris à partir de corpus parallèles), et un *modèle de langue* ($\mathrm{P}(x)$) qui modélise comment produire des phrases en anglais correct (appris à partir de corpus monolingues).

### 2.4.2 Modèle de langue : distribution de probabilité

Un modèle de langue est une distribution de probabilité sur les séquences de mots, plus la probabilité est élevée, plus la séquence de mots est correcte par rapport à la langue considérée.

Par exemple, en français, il faut que dans notre modèle l'inégalité suivante soit vraie :

$$\mathrm{P}(\text{Je mange une pomme verte}) > \mathrm{P}(\text{est llkdef bla topaz})$$

### 2.4.3 Création d'un modèle de langue par comptage : $n$-grams

Pour plus de simplicité, nous allons introduire ici le terme *token* par lequel nous allons désigner indifféremment un caractère, un groupe de caractères ou un mot.

Un $n$-gram est une séquence de $n$ tokens : un 2-gram ou *bigram* est une séquences de 2 tokens, un 3-gram ou *trigram* une séquence de trois caractères, etc. L'un des modèles de langue les plus simple est le *modèle n-gram* qui est un modèle probabiliste qui peut estimer la probabilité d'un token étant donné les $n-1$ tokens précédents.

Considérons la tâche de déterminer la probabilité $\mathrm{P}(t \mid c)$ d'un token $t$ étant donné un *contexte* $c$, où $c$ est une suite de tokens. Par exemple, avec des tokens équivalents à des mots, considérons que $c$ est égal à La plus belle ville du monde est et qu'on veille connaître la probabilité que $t$ soit Périgueux :

$$\mathrm{P}(\text{Périgueux} \mid \text{La plus belle ville du monde est})$$

.

Une manière d'estimer cette probabilité est de compter dans un très grand corpus de textes la fréquence d'apparition $f_c$ de la phrase de contexte "La plus belle ville du monde est" et la fréquence d'apparition $f_{ct}$ de la de la phrase complète "La plus belle ville du monde est Périgueux". On pourrait ensuite déterminer $\mathrm{P}(t \mid c) = f_{ct}/f_c$.

Outre que cette approche nécessite une grande quantité de mémoire pour stocker les phrases associées à leurs fréquences, le langage est très divers et une phrase qui n'a pas été vue dans le corpus d'apprentissage donnera une probabilité nulle.

En partant de l'hypothèse que l'on peut approximer le contexte par uniquement quelques tokens précédents, on peut utiliser une chaîne de Markov, permettant par exemple de construire un modèle 3-gram permettant d'écrire, avec le caractère □ indiquant le mot vide :

P(La plus belle ville du monde est Périgueux)
$= P(\text{La} \mid \square\,\square) \times P(\text{plus} \mid \square\,\text{La}) \times P(\text{belle} \mid \text{La plus}) \times P(\text{ville} \mid \text{plus belle}) \times \cdots \times P(\text{Périgueux} \mid \text{monde est})$

Le comptage se simplifie car il revient à ne plus compter que des 3-gram pour obtenir des probabilités. Par contre cette approche a des limites dans les contextes long. Comment par exemple déterminer avec le "long" contexte suivant :

"Marie, pour réfléchir, à l'habitude de se parler à"

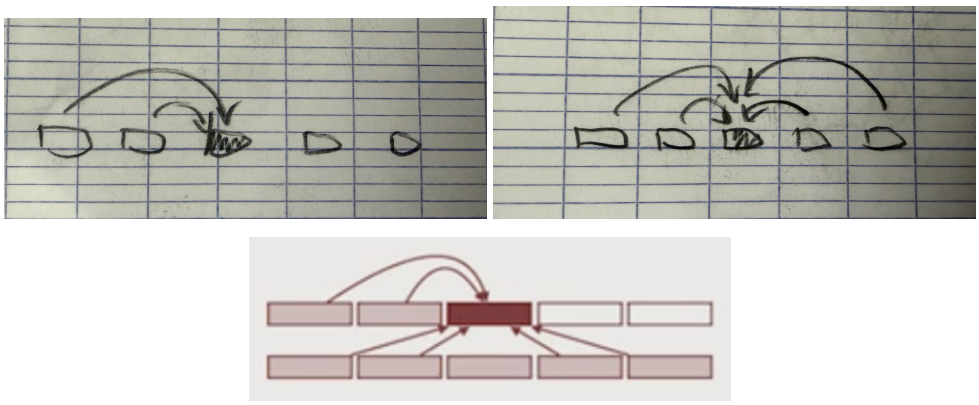
si le token suivant doit être "elle-même" ou "lui-même" ?

**Note : fin de la séance du 9/12/2025.**

#### 2.4.4 Approches par prédiction

Apprendre à prédire le token le plus probable étant donné un certain contexte :

— contexte gauche : modèle génératif (GPT en apprentissage neuronal)
— contexte complet : modèle par masquage (BERT en apprentissage neuronal)



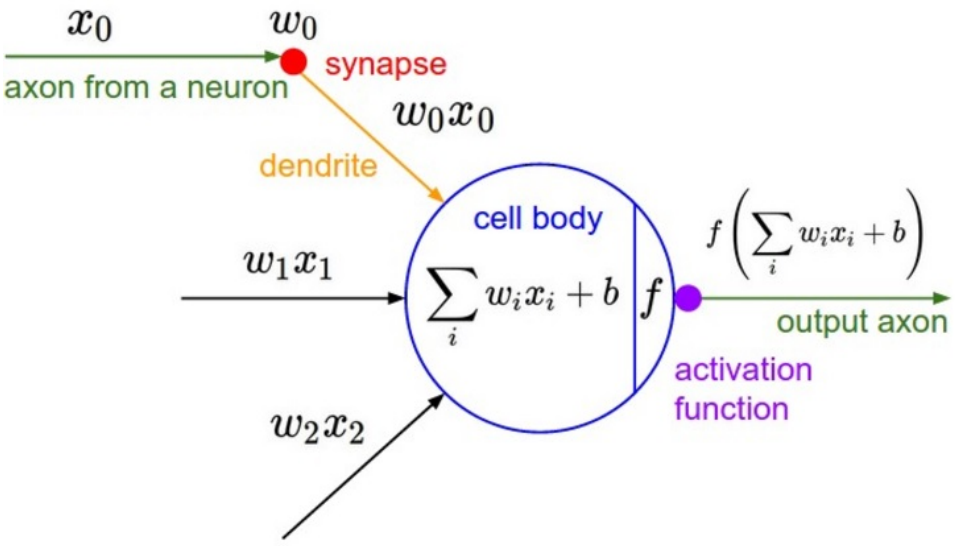
## 2.5 Apprentissage automatique neuronal

**Aujourd'hui : domination des approches neuronales.** Pourquoi ? Progression de la recherche (deep learning) ; technologies de la puissance de calcul (cpu, gpu, mémoire, réseaux rapides) ; données massives disponibles sur étagère (internet) ; grands corpus arborés créés à la main (héritage de la linguistique "classique" et statistique).

**Histoire sélective et rapide des réseaux neuronaux**

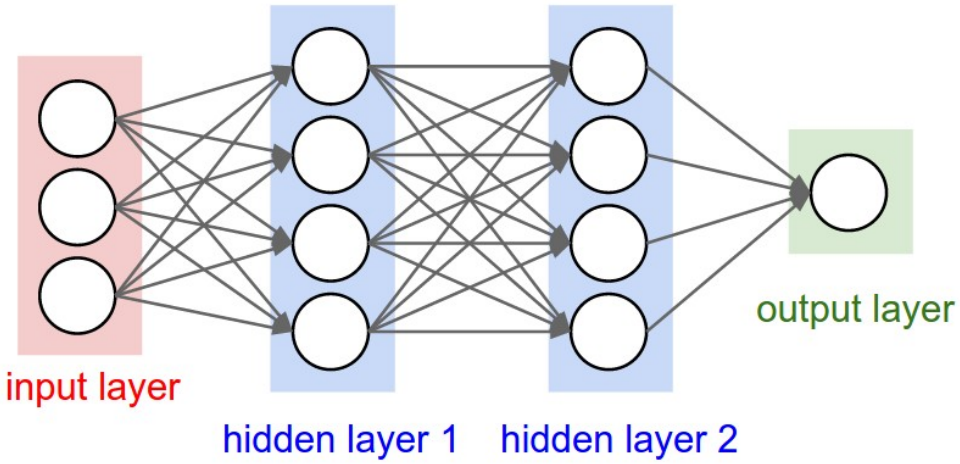
— 1943, Notion de neurone artificiel (McCulloch & Pitts)
— 1957/1958, Apprentissage supervisé, Perceptron (Rosenblatt, 1957, 1958)
— 1962, Plusieurs couches en propagation avant (Rosenblatt)
— 1986, Rétropropagation du gradient (Rumelhart, Hinton, Williams)
— 1989, Réseaux convolutifs (Le Cun *et al.*)
— 1990, Réseaux récurrents (Elman)
— 1997, LSTM (Hochreiter)
— 2006, *Deep Learning*, $c \geq 3$ (Hinton, Bengio)
— 2017, Architecture *Transformer* (Vaswani *et al.*)

**Neurone artificiel**



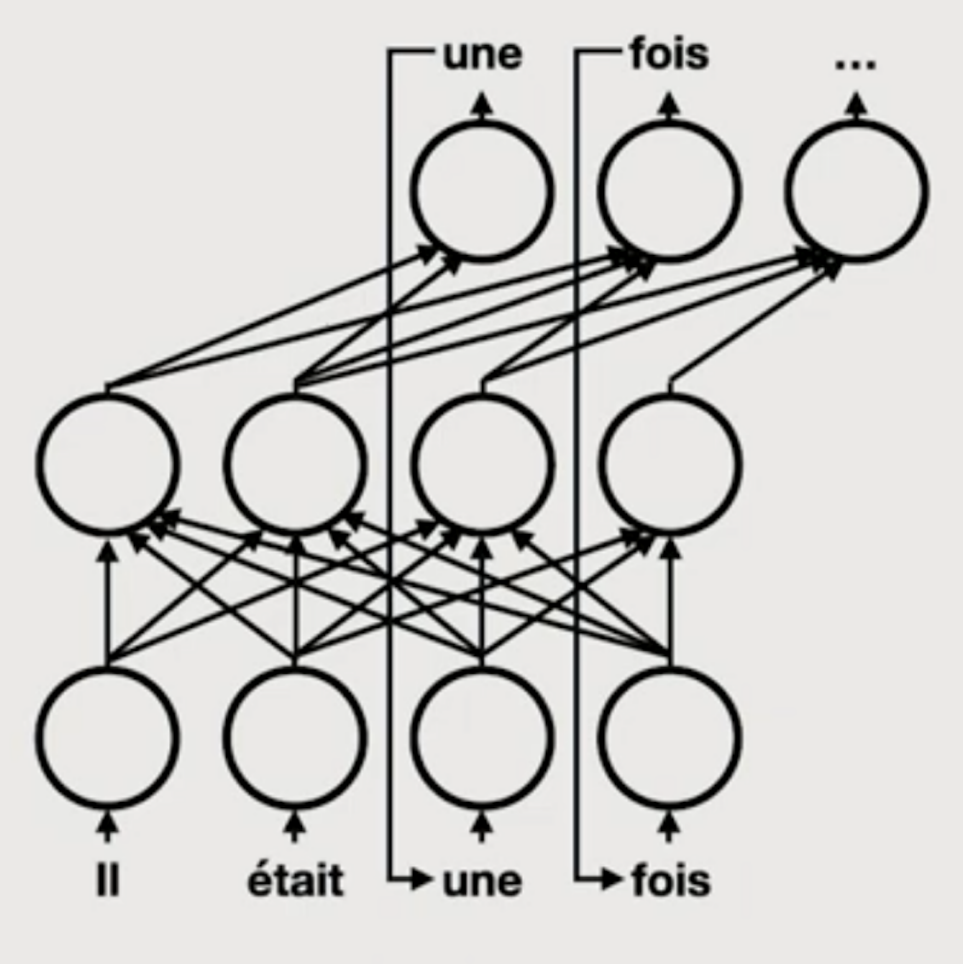

Taken from : https://www.jeremyjordan.me/intro-to-neural-networks/

**Réseau multi-couches**



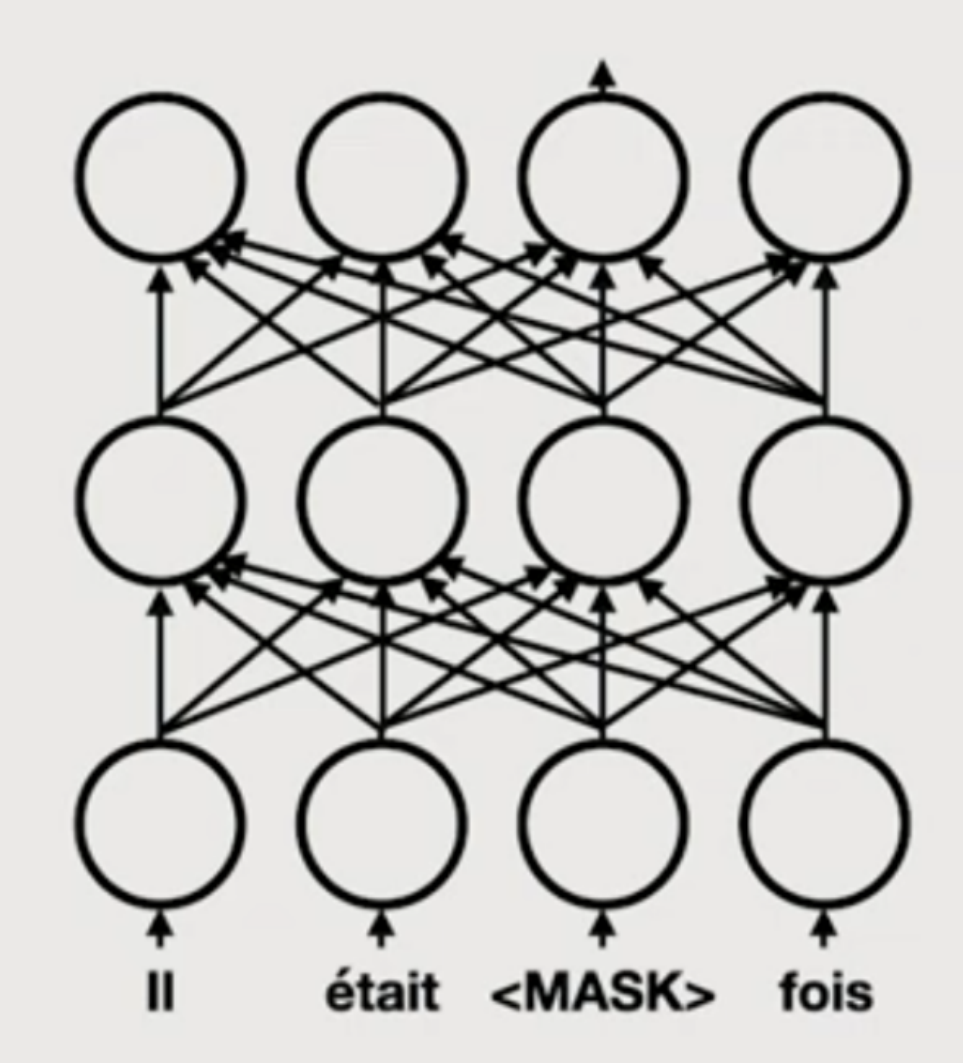

Taken from : https://cs231n.github.io/neural-networks-1/

**Réseau causal**



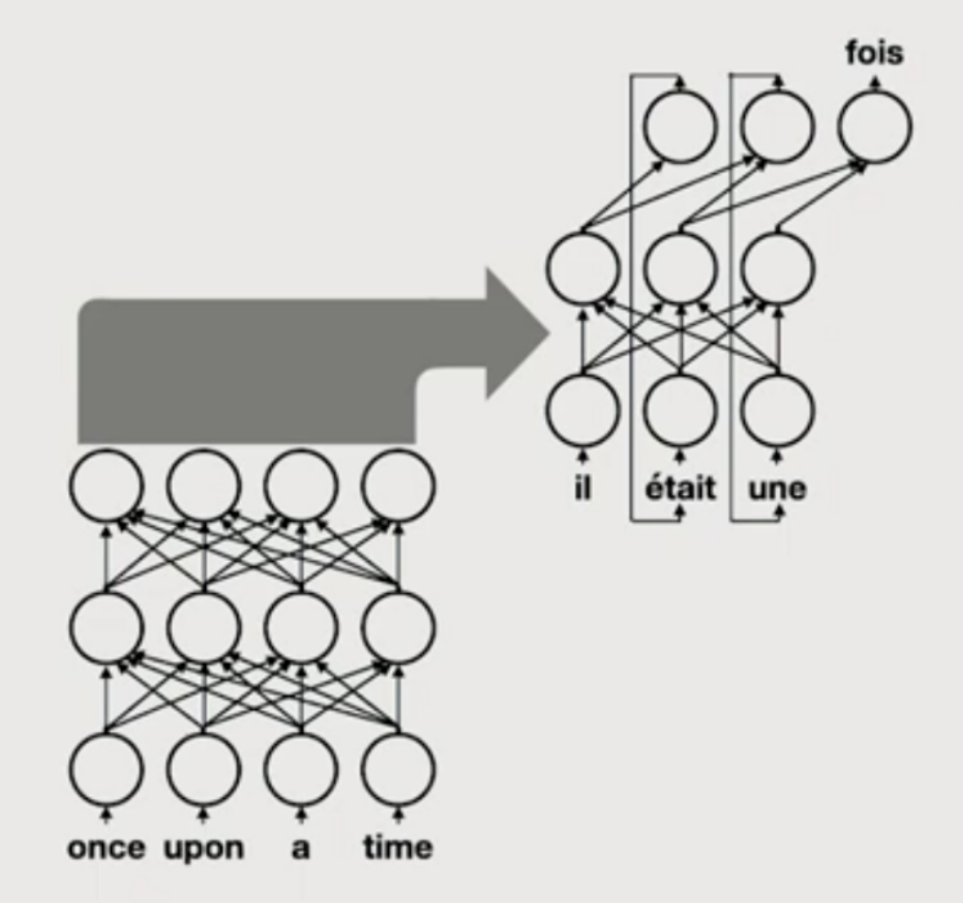

Taken from : B. Sagot

**Réseau par masquage**




Taken from : B. Sagot

**Réseau conditionnel**




Taken from : B. Sagot

**Perplexité** Généralisation : donner une probabilité élevée à des séquences de test jamais vues lors de l'apprentissage.

Plus la perplexité est basse, "meilleur" est le modèle.

Limité par le fait que le données de test doivent être similaires aux données d'apprentissage.

### 2.6 4e révolution de l'accès à l'information

Extrait de la leçon inaugurale de Benoît Sagot au Collège de France (11/2023) :

1. Écriture : stockage des informations de manière externe et pérenne. Outil d'accès à l'information ;
2. Imprimerie : externalisation et diffusion facilités ;
3. Web : numérisation massive, moteurs de recherche. Automatisation de l'identification des sources ;
4. IA : restitution des informations et capacité externe de raisonnement.

## 3 Grammaires formelles

### 3.1 Informatique et langages

Relation forte depuis l'origine de l'informatique en tant que science.

## 3.2 Langage formels, Hiérarchie de Chomsky

https://fr.wikipedia.org/wiki/HiÃľrarchie_de_Chomsky

## 3.3 Langages rationnels

Langages réguliers, expressions régulières.

## 3.4 Langages algébriques

Langages hors-contexte.

Langages de programmation.

## 3.5 Langages contextuels

Langages sensibles au contexte.

Langues naturelles se situent entre langages contextuels et algébriques, "langages légèrement sensibles au contexte".

## 3.6 Langages récursifs

Langages récursivement énumérables.

Programmes.

## 3.7 La machine de Turing

Objet mathématique abstrait composé :

- d'une bande infinie découpé en cases pouvant contenir un symbole ;
- d'une tête de lecture pouvant à chaque étape lire un symbole, écrire un symbole, puis se déplacer sur la bande d'une case à gauche ou à droite ;
- un registre fini d'états dans lesquels peut se trouver la machine ;
- une table d'action indiquant pour un état et un symbole l'action à effectuer.

Une machine de Turing déterministe est un septuplet $M = \langle Q, \Gamma, b, \Sigma, \delta, q_0, F \rangle$ où :

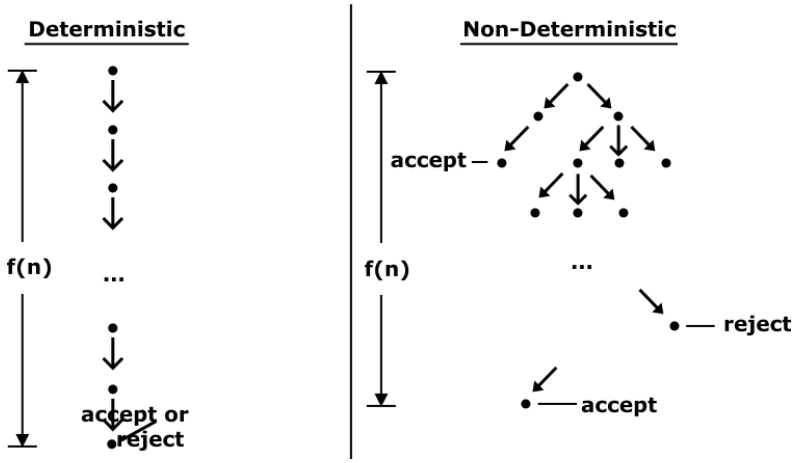
- $Q$ est l'ensemble fini non vide des *états* ;
- $\Gamma$ est l'ensemble fini non vide des *symboles de la bande* ;
- $b \in \Gamma$ est le symbole *blanc* ;
- $\Sigma \subseteq \Gamma \setminus \{b\}$ est l'ensemble des *symboles d'entrée*, les seuls symboles autorisés initialement sur la bande ;
- $\delta : (Q \setminus F) \times \Gamma \not\to Q \times \Gamma \times \{\leftarrow, \rightarrow\}$ est la fonction partielle de *transition*. Si $\delta$ n'est pas définie sur l'état courant et le symbole courant, la machine s'arrête ;
- $q_0 \in Q$ est l'état initial ;
- $F \subseteq Q$ est l'ensemble des *états acceptants* : le contenu initial de la bande est *accepté* par $M$ si elle s'arrête dans un état de $F$.

Exemple de partie de $\delta$ : $\delta(q_1, x) = (q_2, y, \leftarrow)$ indique que dans l'état $q_1$ quand $x$ est lu sur la bande, on passe en état $q_2$, on écrit $y$ et on se déplace à $\leftarrow$.

Une machine de Turing non déterministe est un septuplet $M = \langle Q, \Gamma, b, \Sigma, \delta, q_0, F \rangle$ où :

- $Q$ est l'ensemble fini non vide des *états* ;
- $\Gamma$ est l'ensemble fini non vide des *symboles de la bande* ;
- $b \in \Gamma$ est le symbole *blanc* ;

— $\Sigma \subseteq \Gamma \setminus \{b\}$ est l'ensemble des *symboles d'entrée*, les seuls symboles autorisés initialement sur la bande ;
— $\delta \subseteq (Q \setminus F \times \Gamma) \times (Q \times \Gamma \times \{\leftarrow, \rightarrow\})$ est la relation de *transition* ;
— $q_0 \in Q$ est l'état initial ;
— $F \subseteq Q$ est l'ensemble des *états acceptants* : le contenu initial de la bande est *accepté* par $M$ si *une branche* s'arrête dans un état de $F$.




## 3.8 Expressions régulières

— Expressions régulières par génération d'un automate fini (Ken Thompson).
— `grep`, `lex`, analyseur lexical
— https://regexcrossword.com
— Python : `import re`
— hyperscan

**Exemple 1 : utilisation dans un IDE**

```
#define MAX_URI_COUNTRY 3
#define MAX_URI_CORPUS 5
#define MAX_URI_NATURE 70
#define MAX_URI_YEAR 5
#define MAX_URI_MONTH 3
#define MAX_URI_DAY 3
#define MAX_URI_NUMBER 30
#define MAX_URI_VERSION 9
#define MAX_URI 256
```

```
MAX_(\w+)
$1_MAX
```

**Exemple 2 : découpage d'un arrêt de cour d'appel d'Agen**

```
intro_re = re.compile(
  r'^(?P<intro>.*?)(?=('
  r'<p>\s*A\s+rendu\s+l.arrêt\s+((réputé\s+)?
  r'contradictoire|par\s+défaut)'
  r'|<p>\s*EXPOS(É|E)\s*DU\s*LITIGE'
  r'|<p>A rendu réputé l.arrêt réputé contradictoire'
  r'))',
  re.UNICODE|re.DOTALL|re.MULTILINE|re.IGNORECASE)
decision_re = re.compile(
  r'(?P<decision>(<p>par\s*ces\s*motifs).*)$',
```

```
  re.U|re.DOTALL|re.MULTILINE|re.IGNORECASE)
```

**Exemple 3 : numéros d'alinéas (Droit Quotidien)**

```
alinea_number = (
  r"("
  r"\w\)(?=\s+)"
  r"|\d{1,2}°(\s+bis)?(?=\.?\s+)"
  r"|\d{1,2}(\s+bis)?(?=\.?\s+)"
  r"|[IVX]+(?=\.\s+)"
  r")"
)
```