Séance 1 — Du texte au résultat : comprendre le fonctionnement du TAL

Objectifs de la séance

À la fin de la séance, l'étudiant doit être capable de :

- définir le Traitement Automatique des Langues ;
- distinguer **texte**, **donnée linguistique**, **traitement** et **résultat** ;
- comprendre quelques tâches fondamentales du TAL ;
- voir concrètement comment un programme traite un texte ;
- comprendre la différence entre une approche à règles et une approche fondée sur les données ;
- expliquer l'impact du Big Data sur le TAL ;
- comprendre intuitivement le principe des Transformers et de l'attention ;
- situer les LLM dans l'évolution du TAL ;
- interpréter et questionner un résultat produit automatiquement.

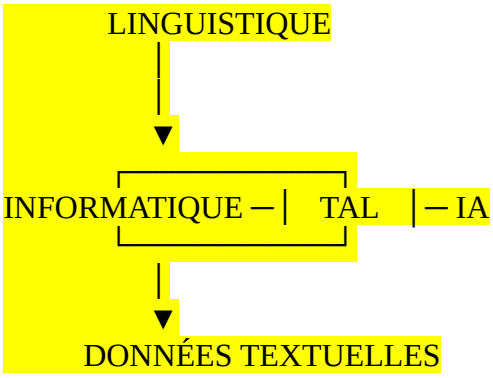
1. Introduction : qu'est-ce que le TAL ?

Le **Traitement Automatique des Langues (TAL)**, appelé également **Natural Language Processing (NLP)** en anglais, désigne l'ensemble des méthodes informatiques permettant de **traiter, analyser, comprendre ou générer des données produites dans une langue naturelle**.

Une langue naturelle est une langue utilisée par les êtres humains :

- français ;
- arabe ;
- anglais ;
- espagnol ;
- etc.

Le TAL se situe donc à l'intersection de plusieurs domaines :



Le TAL cherche notamment à répondre à des questions comme :

- Où sont les mots dans un texte ?
- Quelle est la catégorie grammaticale de chaque mot ?
- Quel est le lemme d'un mot ?
- Quelle est la structure syntaxique d'une phrase ?
- Qui sont les personnes mentionnées ?
- Quels sont les lieux ?
- De quoi parle le texte ?
- Deux textes parlent-ils du même sujet ?
- Quel est le sentiment exprimé ?
- Peut-on résumer automatiquement un document ?
- Peut-on traduire automatiquement un texte ?
- Peut-on générer du texte ?

2. Pourquoi le TAL est-il nécessaire ?

Un ordinateur ne reçoit pas spontanément un texte comme un être humain.

Pour nous :

> **« Victor Hugo est né à Besançon. »**

nous reconnaissons immédiatement :

- Victor Hugo → personne ;
- est né → événement de naissance ;
- Besançon → lieu.

Pour un ordinateur, au départ, il s'agit essentiellement d'une **suite de symboles**.

V i c t o r H u g o e s t n é à B e s a n ç o n

Il faut donc transformer progressivement cette séquence en représentations exploitables.

On peut représenter le processus ainsi :

Texte humain
↓
Données informatiques
↓
Analyse linguistique
↓
Représentation structurée
↓

Résultat

C'est cette transformation qui constitue le cœur du TAL.

3. Un premier exemple très simple avec Python

Commençons sans aucune bibliothèque de TAL.

```
texte = "Victor Hugo est né à Besançon en 1802."

mots = texte.split()

print(mots)
```

Résultat :

```
['Victor', 'Hugo', 'est', 'né', 'à', 'Besançon', 'en', '1802.']
```

Le programme a effectué une première opération de TAL :

La tokenisation

La **tokenisation** consiste à découper un texte en unités appelées **tokens**.

Un token peut être :

- un mot ;
- un signe de ponctuation ;
- parfois une sous-partie de mot, selon le système utilisé.

Par exemple :

"Le chat dort."
peut devenir :
Le
chat
dort
.

Point important

Notre première méthode :

```
texte.split()
```

Elle produit par exemple :
1802.
comme un seul élément.

Un véritable système de TAL doit donc utiliser des méthodes plus sophistiquées.

4. Utiliser une véritable bibliothèque de TAL : spaCy

Nous allons maintenant utiliser **spaCy**, une bibliothèque Python spécialisée dans le traitement automatique des langues.

Nous allons d'abord installé le modèle français :

fr_core_news_sm

On commence par charger le modèle :

```
import spacy

nlp = spacy.load("fr_core_news_sm")
```

Puis :

```
texte = "Victor Hugo est né à Besançon en 1802."

doc = nlp(texte)

for token in doc:

    print(token.text)
```

Résultat :

```
Victor

Hugo

est

né

à

Besançon

en

1802

.
```

Nous avons donc obtenu une tokenisation plus adaptée au TAL.

5. La lemmatisation

Une autre tâche importante du TAL est la **lemmatisation**.

Le lemme est la forme de référence d'un mot.

Exemples :

Forme dans le texte	Lemme
mange	manger
mangeait	manger
mangeront	manger
maisons	maison
enfants	enfant
était	être

Avec spaCy :

```
for token in doc:
    print(token.text, "→", token.lemma_)
```

On peut obtenir :

```
Victor → Victor
Hugo → Hugo
est → être
né → naître
à → à
Besançon → Besançon
en → en
1802 → 1802
. → .
```

Pourquoi est-ce utile ?

Supposons que nous cherchions combien de fois le verbe **manger** apparaît dans un roman.

Nous pourrions rencontrer :

```
mange
mangeait
mangé
manger
mangeront
```

Une recherche exacte du mot `"manger"` ne trouverait pas toutes ces occurrences.

La lemmatisation permet de regrouper ces formes autour de leur lemme.

6. L'étiquetage morphosyntaxique — POS tagging

Le TAL peut également déterminer la catégorie grammaticale d'un mot.

C'est le **Part-of-Speech tagging (POS tagging)** ou **étiquetage morphosyntaxique**.

Exemple :

Le → déterminant
chat → nom
mange → verbe
une → déterminant
souris → nom

Avec spaCy :

```
for token in doc:
    print(token.text, "→", token.pos_)
```

Résultat possible :

Victor → PROPN
Hugo → PROPN
est → AUX
né → VERB
à → ADP
Besançon → PROPN
en → ADP
1802 → NUM
. → PUNCT

Quelques étiquettes :

Étiquette	Signification
NOUN	nom
VERB	verbe
ADJ	adjectif
ADV	adverbe
PRON	pronom
DET	déterminant

Étiquette	Signification
ADP	préposition
PROPN	nom propre
NUM	nombre
PUNCT	ponctuation

7. Une question importante : comment le programme sait-il tout cela ?

Nous avons demandé :

token.lemma_
token.pos_

Mais comment spaCy sait-il que :

> **né → naître**

et que :

> **Victor → PROPN**

?

La réponse est :

> Le programme utilise un **modèle de TAL entraîné sur des données linguistiques**.

Le programme ne « réfléchit » pas comme un linguiste humain.

Il applique un modèle qui a appris des régularités à partir de données.

Cela nous conduit à l'évolution historique du TAL.

8. Première époque : le TAL à base de règles

Les premiers systèmes de TAL reposaient largement sur des **règles linguistiques explicites**.

Un expert pouvait écrire :

SI un mot possède telle caractéristique
ALORS lui attribuer telle catégorie.

Par exemple, de manière très simplifiée :

SI le mot commence par une majuscule

ET n'est pas en début de phrase
ALORS il pourrait être un nom propre.

Mais cette règle produit immédiatement des problèmes.

Dans :

> **Paris est une grande ville.**

`Paris` est probablement un nom propre.

En revanche :

> **Paris est une personne que je connais.**

pose d'autres problèmes.

Et surtout :

> **Le Président est arrivé.**

`Président` commence par une majuscule mais ce n'est pas nécessairement un nom propre.

Limite

La langue est extrêmement complexe.

Il est impossible de décrire toutes les situations avec quelques règles simples.

9. Deuxième étape : les méthodes statistiques

Une autre idée apparaît :

> **Au lieu d'écrire toutes les règles manuellement, pourquoi ne pas apprendre à partir de nombreux exemples ?**

Prenons :

Le chat mange.
Le chien mange.
Le garçon mange.
Le chat dort.
Le chien dort.

Le système observe des régularités.

Il peut apprendre que :

chat → probablement NOM
chien → probablement NOM
mange → probablement VERBE

dort → probablement VERBE

Il ne reçoit donc plus uniquement des règles.

Il reçoit des **données**.

CORPUS
↓
observations
↓
statistiques
↓
modèle
↓
prédiction

10. Qu'est-ce qu'un corpus ?

Un **corpus** est un ensemble organisé de données linguistiques utilisé pour étudier ou traiter une langue.

Il peut contenir :

- des romans ;
- des articles de presse ;
- des conversations ;
- des pages Web ;
- des transcriptions ;
- des messages ;
- des documents scientifiques.

Exemple :

CORPUS
│
├── roman1.txt
├── roman2.txt
├── roman3.txt
├── article1.txt
├── article2.txt
└── …

Les corpus deviennent progressivement beaucoup plus importants avec la numérisation.

11. Le Big Data : un changement majeur pour le TAL

L'apparition du **Big Data** transforme profondément le TAL.

Avant

Le linguiste ou l'informaticien devait souvent déterminer manuellement les caractéristiques importantes.

Texte
↓
règles
↓
caractéristiques conçues par l'humain
↓
modèle
↓
résultat

Après

Les systèmes peuvent apprendre automatiquement de nombreuses représentations à partir de grandes quantités de données.

Énorme corpus
↓
Apprentissage
↓
représentations apprises
↓
modèle
↓
prédiction

Le TAL dispose alors d'une quantité considérable de données. Cela permet d'entraîner des modèles beaucoup plus complexes. Il s'agit d'une avancée fondamentale, car elle a conduit à **l'apprentissage automatique.**

12. Du TAL traditionnel aux Transformers

L'évolution du TAL peut être résumée en trois grandes étapes :

1. Le Machine Learning (l’apprentissage automatique) : apprendre à partir des données

Avec le **Machine Learning (apprentissage automatique)**, on change d’approche.

> Au lieu de programmer toutes les règles linguistiques, on donne au système des **exemples** afin qu’il puisse apprendre des régularités à partir des données.

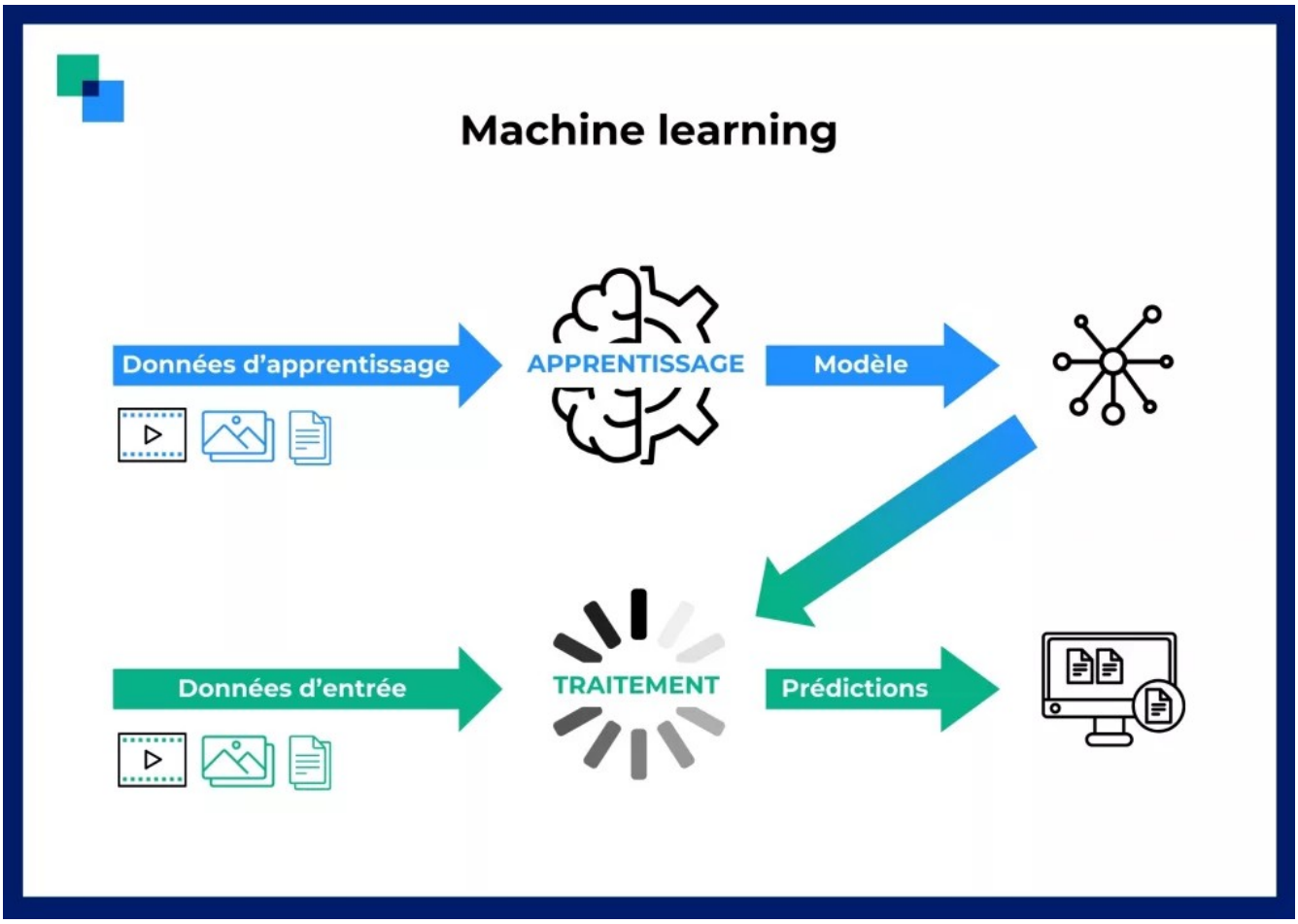


Figure 1: Fonctionnement du Machine Learning

Mais, dans le **Machine Learning traditionnel**, l'être humain intervient encore de manière importante : il doit généralement déterminer **quelles caractéristiques des données sont pertinentes** pour permettre au modèle d'apprendre.

Par exemple, pour déterminer si un texte est **positif ou négatif**, l'humain peut décider d'utiliser comme caractéristiques :

- la présence de certains mots ;
- la fréquence des mots ;
- la présence d'adjectifs positifs ou négatifs ;
- la longueur du texte ;
- certains marqueurs linguistiques.

Le modèle apprend ensuite à partir de ces caractéristiques quelles régularités permettent de distinguer les différentes catégories.

On peut donc résumer le processus ainsi :

Données → caractéristiques définies par l'humain → apprentissage du modèle → résultat

Cette intervention humaine constitue une limite importante : **la qualité du modèle dépend en partie de la pertinence des caractéristiques choisies par l'humain**.

C'est précisément sur ce point que le **Deep Learning** va apporter une évolution majeure : au lieu de demander à l'humain de définir manuellement toutes les caractéristiques pertinentes, les réseaux neuronaux profonds peuvent **apprendre automatiquement des représentations à partir des données (Big data)**.

On passe ainsi progressivement de :

« L'humain indique au modèle quoi observer » (Machine learning)

à :

« Le modèle apprend lui-même quoi observer dans les données. » (Deep learning)

2. Le Deep Learning : apprendre des représentations plus complexes

Le **Deep Learning (apprentissage profond)** constitue une évolution du Machine Learning. Il utilise des **réseaux de neurones artificiels comportant plusieurs couches**, capables d'apprendre automatiquement des représentations plus complexes des données.

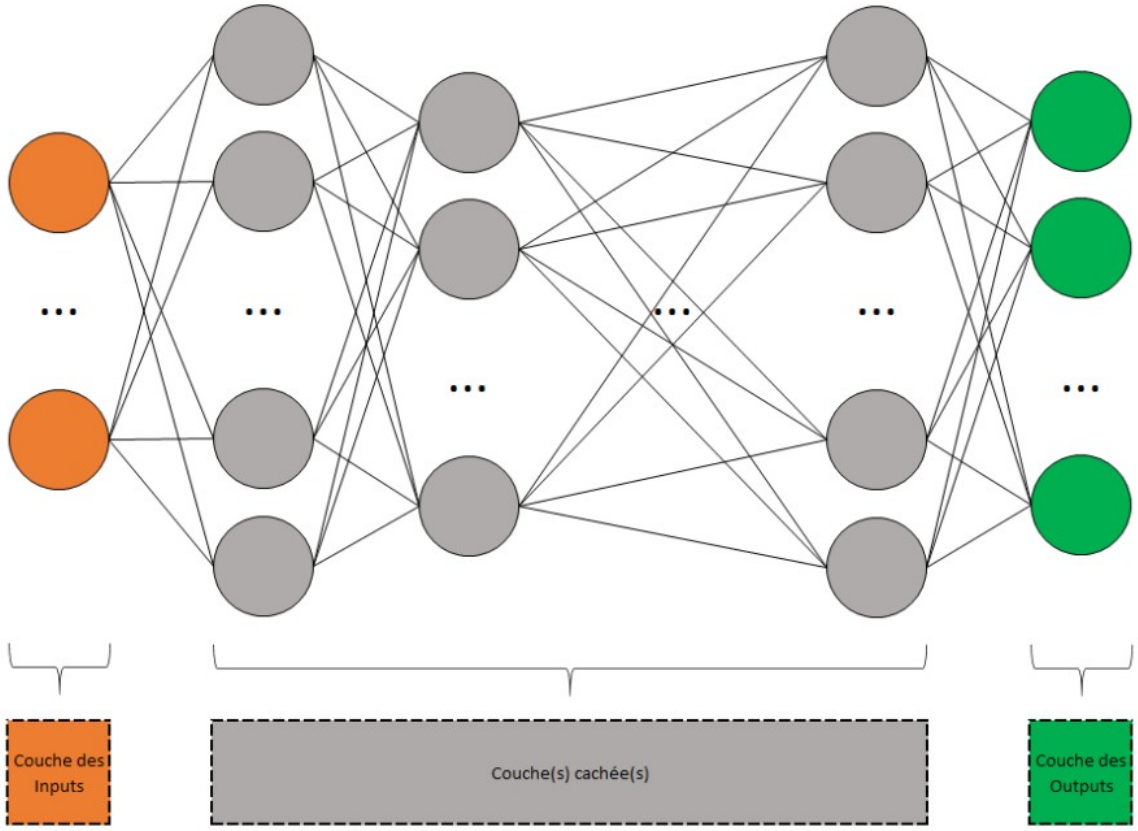


Figure 2: réseau de neurones informatique

Cette évolution est particulièrement importante pour le langage.

Au lieu de devoir déterminer manuellement toutes les caractéristiques importantes d'un texte, le modèle peut apprendre progressivement des représentations à partir de grandes quantités de données.

On passe donc progressivement de :

> **« Quelles caractéristiques devons-nous donner au système ? » (Machine learning)**

à :

> **« Donnons suffisamment de données au système pour qu'il apprenne lui-même des représentations utiles. » (Deep learning)**

Le développement du **Big Data**, c'est-à-dire la disponibilité de très grandes quantités de données numériques, a fortement favorisé cette évolution.

3. Les Transformers : mieux prendre en compte le contexte

Une difficulté fondamentale du langage est qu'un mot peut avoir plusieurs interprétations selon son contexte.

Exemple : un même mot, plusieurs interprétations

Comparez :

> **« L'avocat défend son client devant le tribunal. »**
>
> **« L'avocat est mûr et délicieux. »**

Dans les deux phrases, le mot **avocat** est identique.

Pourtant :

- dans la première phrase, *avocat* désigne une **personne exerçant le métier d'avocat** ;
- dans la deuxième, *avocat* désigne un **fruit**.

Le sens d'un mot dépend donc fortement du **contexte** dans lequel il apparaît.

Les modèles modernes de TAL cherchent à produire des **représentations contextuelles** : la représentation d'un mot peut dépendre des autres mots qui l'entourent.

C'est l'une des grandes avancées apportées par l'architecture **Transformer**.

Les Transformers utilisent notamment un mécanisme appelé **attention**, qui permet au modèle de prendre en compte les relations entre les différents éléments d'une séquence lorsqu'il construit leurs représentations.

On peut représenter très simplement cette évolution :

TAL traditionnel

↓

Machine Learning

↓

Deep Learning

↓

Transformers

↓

Modèles de langage modernes

Pourquoi cette évolution est-elle importante ?

Elle a permis de traiter des textes de plus en plus complexes et, combinée à :

- de très grandes quantités de données ;
- des capacités de calcul importantes ;
- des modèles comportant beaucoup de paramètres ;

elle a conduit au développement des **grands modèles de langage (Large Language Model - LLM)** , notamment les modèles **GPT (Generative Pre-trained Transformer)** utilisés dans des systèmes comme **ChatGPT**.

À retenir : l'évolution du TAL correspond notamment au passage de systèmes fondés principalement sur des règles vers des modèles capables d'apprendre à partir de grandes quantités de données et de mieux prendre en compte le contexte linguistique. Les Transformers constituent une étape majeure de cette évolution.

13. GPT : Generative Pre-trained Transformer

GPT signifie :

Generative Pre-trained Transformer

Décomposons le terme.

Generative

Le modèle peut **générer du texte**.

Par exemple :

Question

↓

GPT

↓

réponse générée

Pre-trained

Le modèle est d'abord **pré-entraîné** sur de grandes quantités de données textuelles.

Il apprend notamment à modéliser les régularités du langage.

Transformer

Il utilise une architecture basée sur les **Transformers**.

14. Si les LLM peuvent faire du TAL, pourquoi étudier le TAL ?

Nous disposons aujourd'hui de **grands modèles de langage (LLM)** capables d'effectuer de nombreuses tâches de traitement automatique des langues : traduction, résumé, classification, extraction d'informations, analyse de sentiment, identification d'entités, génération de textes, etc.

Ces modèles sont développés par de grandes entreprises technologiques et entraînés sur des **quantités considérables de données**. Des outils comme ChatGPT peuvent ainsi produire, en quelques secondes, des résultats qui auraient auparavant nécessité le développement d'un programme TAL spécifique.

Une question se pose alors :

> **Si les LLM sont capables de faire du TAL et disposent de beaucoup plus de données et de ressources que nous, pourquoi devons-nous encore étudier le TAL et développer nos propres programmes ?**

La réponse ne consiste pas à dire que les LLM sont inutiles. Au contraire, ils constituent aujourd'hui des **outils puissants du TAL moderne**.

Le problème est plutôt de comprendre **peut-on accepter leur résultat sans l'analyser, sans connaître la méthode utilisée et sans le vérifier ?**

14.1. Un LLM ne produit pas une vérité absolue

Un LLM produit une réponse à partir de régularités apprises dans les données et de calculs probabilistes.

Il ne fonctionne donc pas comme une **base de données de vérités** qui consulterait une information certaine et la restituerait nécessairement correctement.

De manière simplifiée, lorsqu'on lui demande de produire une réponse, le modèle estime quelles suites de mots sont les plus plausibles compte tenu de la demande et du contexte.

Cela explique pourquoi un LLM peut produire une réponse :

- très bien formulée ;
- cohérente ;
- convaincante ;
- mais néanmoins **fausse**.

C'est ce que l'on appelle une **hallucination** : le modèle génère une information qui paraît plausible mais qui ne correspond pas nécessairement à la réalité.

Même ChatGPT indique explicitement dans son interface :

> **« ChatGPT can make mistakes. Check important info. »**

Cette simple indication rappelle une règle fondamentale :

> **Une réponse produite par une IA n'est pas automatiquement une information vérifiée.**

14.2. Une réponse correcte ne suffit pas dans une recherche scientifique

Dans un contexte de recherche, nous ne cherchons pas seulement à obtenir une réponse qui semble correcte.

Nous devons pouvoir répondre à plusieurs questions :

- **Comment le résultat a-t-il été obtenu ?**
- **Quelle méthode a été utilisée ?**
- **Quelles données ont été utilisées ?**
- **Quel traitement a été appliqué aux données ?**
- **Quels paramètres ou critères ont été utilisés ?**
- **Le résultat peut-il être vérifié ?**
- **Un autre chercheur pourrait-il reproduire ou contrôler l'expérience ?**
- **Quelles sont les limites et les erreurs possibles ?**

Prenons un exemple.

Supposons qu'un chercheur demande à un LLM :

> *« Analyse les adjectifs utilisés dans ce roman et indique ce qu'ils révèlent sur la représentation de la nature. »*

Le modèle peut fournir une analyse intéressante. Mais si le chercheur présente simplement cette réponse comme **le résultat de sa recherche**, plusieurs problèmes apparaissent.

Comment savons-nous :

- quels passages ont réellement été analysés ?
- comment les adjectifs ont-ils été identifiés ?
- certains adjectifs ont-ils été oubliés ?
- certaines catégories grammaticales ont-elles été mal identifiées ?
- comment le modèle a-t-il interprété les mots ambigus ?
- la même analyse donnerait-elle exactement le même résultat avec une autre configuration ?

- le résultat peut-il être reproduit ou contrôlé ?

Le chercheur doit donc pouvoir **expliquer et justifier sa méthode**.

14.3. Le TAL permet de rendre la méthode explicite

C'est précisément ici que l'étude du TAL reste importante.

Connaître le TAL permet de ne pas considérer le modèle comme une simple **boîte noire** à laquelle on pose une question et dont on accepte automatiquement la réponse.

Par exemple, au lieu de demander simplement à un LLM :

> « Quels sont les adjectifs de ce texte ? »

on peut utiliser une méthode TAL pour :

1. segmenter le texte ;
2. identifier les tokens ;
3. effectuer une analyse morphosyntaxique ;
4. identifier les adjectifs ;
5. compter leurs occurrences ;
6. examiner leur contexte ;
7. vérifier manuellement les erreurs ;
8. interpréter les résultats.

Le chercheur dispose alors d'une **procédure explicite et contrôlable**.

14.4. L'objectif n'est donc pas de remplacer les LLM

Étudier le TAL ne signifie pas qu'il faut refuser les LLM ou développer systématiquement un programme à partir de zéro.

Au contraire, le chercheur peut utiliser **les deux approches**.

L'objectif du cours n'est donc pas de former uniquement des programmeurs capables de reproduire ce que font les grandes entreprises.

Il s'agit surtout de développer une **culture méthodologique et critique du traitement automatique des langues**.

L'étudiant doit être capable de :

Utiliser → un outil TAL ou un LLM.

Comprendre → les principes généraux de la méthode utilisée.

Évaluer → la qualité et les erreurs du résultat.

Vérifier → les informations produites.

Comparer → plusieurs méthodes ou outils.

Justifier → les choix méthodologiques.

Interpréter → les résultats dans le cadre de sa recherche.

Ainsi, la question n'est pas :

> **« L'IA peut-elle faire du TAL à notre place ? »**

mais plutôt :

> **« Comment utiliser les outils d'IA et de TAL de manière rigoureuse, vérifiable et scientifiquement justifiable ? »**