

# ***Bases de l'IA***

## **Traitement automatique du langage naturel**

**Elena CABRIO**

[elena.cabrio@univ-cotedazur.fr](mailto:elena.cabrio@univ-cotedazur.fr)

# Plan pour cette séance

- Le TALN, c'est quoi?
- Applications
- Approches symboliques vs statistiques
- Evaluation
- Pre-traitement (tokenization, lemmatization, stemming...)
- Analyse morpho-syntaxique
- Analyse syntaxique

# Qu'est-ce que le TALN?

## Introduction

# Qu'est-ce que le TALN?

- Le **traitement automatique du langage naturel** (abr. **TALN**), ou **traitement automatique de la langue naturelle**, est une discipline qui se trouve à l'intersection de plusieurs autres branches de la science comme l'informatique, l'intelligence artificielle, la linguistique et la psychologie cognitive.
- *Linguistique informatique* ou *linguistique computationnelle* : modèles ou les formalismes linguistiques développés dans le but d'une implantation informatique.

# Parmi les applications du TALN

- *Interroger* une base de données en langage naturel, par écrit ou oralement
- *Traduire automatiquement* la langue, à la fois parlée et écrite
- *Indexer, résumer automatiquement* et ensuite effectuer des *recherches* sur une base sémantique à partir de texte non structuré
- Développer des filtres qui reconnaissent les messages dont le contenu est inapproprié (par exemple, *anti-spam*)
- Identifier automatiquement les cas de plagiat
- Créer des technologies *d'aide* aux personnes handicapées (p. ex. Analyse de la lisibilité du texte)
- *Analyser les opinions*, prévoir les tendances en collectant des informations disponibles en ligne



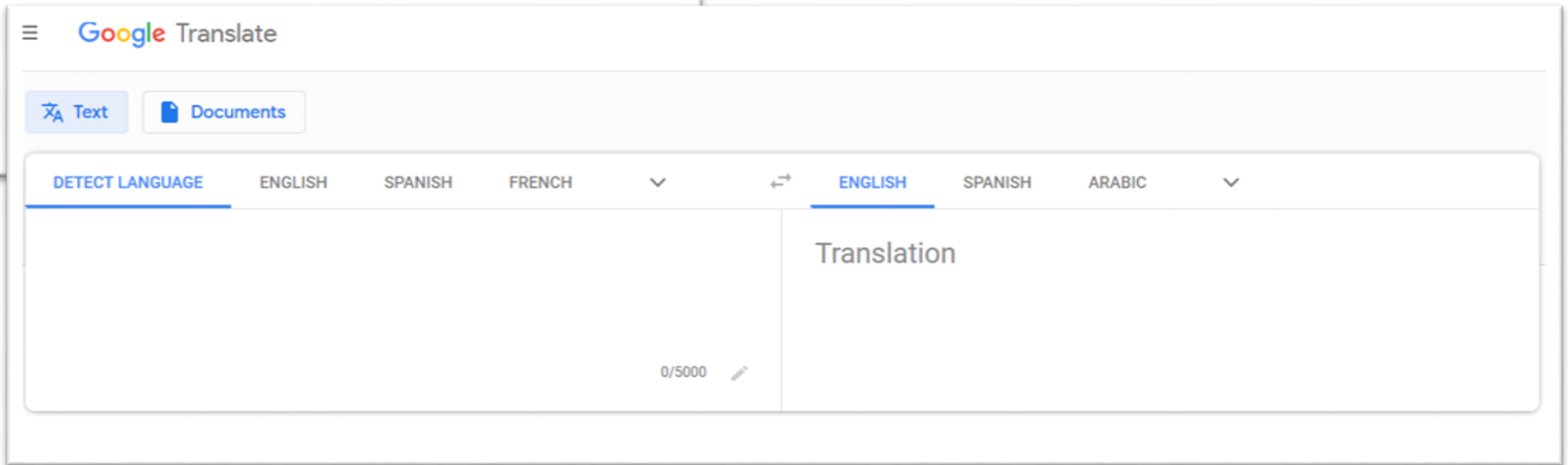
# Exemples d'applications commerciales

Google

Google Search

I'm Feeling Lucky

# Exemples d'applications commerciales



# Exemples d'applications commerciales



Google Translate

Text

Documents

DETECT LANGUAGE

ENGLISH

SPANISH

FRENCH

0/5000





# Exemples d'applications commerciales





# Exemples d'applications commerciales



# Les niveaux de compréhension du langage

**Traiter le langage naturel** nécessite l'analyse de différents niveaux de compréhension / compétence:

- **Niveau lexical:** concerne les conventions sur les mots simples
  - *savoureux, \*reuxsavou*
- **Niveau syntaxique:** concerne l'ordre correct des mots et son impact sur le sens de la phrase
  - *le chien a mordu l'enfant, l'enfant a mordu le chien*
  - *les idées vertes incolores dorment furieusement*
  - *\* mordu a enfant chien le l'*

# Les niveaux de compréhension du langage

Traiter le langage naturel nécessite l'analyse de différents niveaux de compréhension / compétence:

- **Niveau sémantique:** concerne la signification des mots et des phrases
  - *la gorge brûle, la maison brûle, la soupe brûle*
  - *\*les idées vertes incolores dorment furieusement*
- **Niveau pragmatique:** il concerne le contexte communicatif et social et son impact sur l'interprétation
  - *c'est sympa, le sandwich demande une autre bière*
  - *quelle est ma couverture ?*

# L'ambiguïté est omniprésente

- Reconnaissance vocale
  - ‘Cet homme a beaucoup de *vis*’ / ‘Cet homme a beaucoup de *vices*’
- Analyse lexicale
  - ‘Nous avons des *jumelles* à la maison’
  - ‘Je *suis* mon maître’
- Analyse syntaxique
  - ‘Elle emporte *les clefs de la maison au garage*’
- Analyse sémantique
  - ‘Comment savoir si *un avocat* est mûr’ / ‘Comment savoir si *un avocat* est ambitieux’
- Interprétation sémantique
  - ‘Chaque homme aime une femme’

# L'ambiguïté est omniprésente (cont.)

- Analyse du discours
  - *'Il a mis l'artichaut dans son assiette et il l'a mangé'*
  - *'Les sages parlent parce qu'ils ont quelque chose à dire; des imbéciles parce qu'ils doivent dire quelque chose'*
- Analyse pragmatique
  - *'Si tu arrêtes de fumer je te paye un verre' (promesse)*
  - *'Si tu sautes des leçons, je te punis' (menace)*

# Un peu d'histoire...

## Naissance

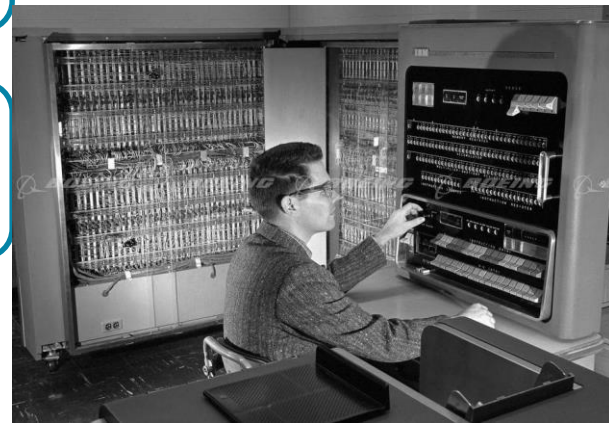
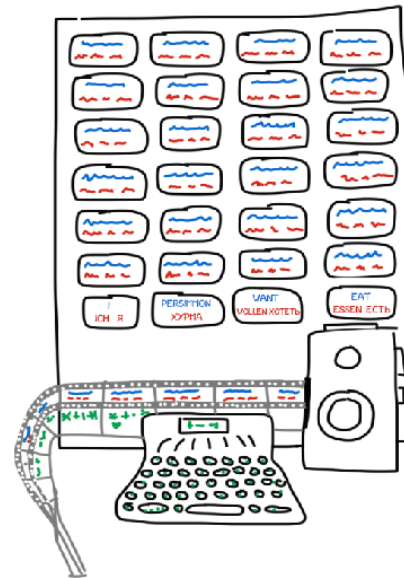
- Association of Computational Linguistics (ACL) en 1962
- Association savante ATALA en France depuis 1959
- Premiers travaux en TAL commencent dans les années 1950, thématique de la **traduction automatique**.

## Déscriptif

- pluralité de programmes de recherche et de méthodologies
- interdisciplinarité et pluridisciplinarité

## Objectifs

- applications destinées aux professionnels du langage
- applications informatiques d'usage courant



# Le comportement de la machine

## Input

- L'input (entrée) peut être considérée comme le stimulus (sensoriel, linguistique, etc.) ou les données fournies à la machine.

## Output

- L'output (la sortie) correspond à ce que la machine produit après avoir reçu l'entrée: production d'une réponse, d'un son, d'une action, d'un mouvement, etc.

## Modèle

- Le modèle filtre l'entrée, l'analyse et lui associe, selon ses caractéristiques, et à travers une série d'algorithmes, une sortie



# Quelles données en entrée?

- **Données structurées:**

- **Bases de données:** les informations sont encodées dans des tables et sont accessibles via un langage de requête spécial. Il existe un "schéma" qui permet une interprétation non ambiguë des données.
- **Bases de connaissances:** elles permettent également d'exécuter des inférences (raisonnements).

- **Données semi-structurées**

- Tableaux insérés dans des documents ou sur le Web
- Répertoires de portails Web (Google et Yahoo!, par exemple)
- Documents XML (Extensible Markup Language)
- Les données sont partiellement interprétables

# Quelles données en entrée? (cont.)

- **Données non structurées:**
  - Textes écrits dans divers formats
  - Documents Word, pdf, Power Point
  - Journaux en ligne
  - Pages Web en HTML
  - SMS
  - Champs de texte dans les bases de données
  - Messages électroniques
  - Messages sur les groupes de discussion
  - Foire Aux Questions (FAQ)
  - Nouvelles de l'agence
  - Transcriptions automatiques de journaux radio

# Quelles données en entrée? (cont.)

- **Données multilingues et multimédia:**
  - **Données multilingues** (formats différents)
    - Sites multilingues avec le même texte disponible dans différentes pages
    - Textes à l'intérieur desquels apparaissent des sections dans différentes langues
    - Traductions, par exemple des manuels d'utilisation de produits.
  - **Information multimédia**
    - Images insérées dans un texte, éventuellement avec une légende
    - Films
    - Fichiers audio, avec messages vocaux

# Traitement des données: quels besoins?

Des exemples ...

- **Trouver des informations** contenues dans des sources textuelles.
- **Extraire les informations** contenues au format texte.
- **Organiser les documents** au format texte.
- **Construire des réseaux d'utilisateurs** en fonction de leur intérêt pour certains documents (voir des études récentes sur les médias sociaux et les réseaux sociaux)

# Trouver l'information

- **Récupération d'informations** (*information retrieval*): l'utilisateur soumet une requête (*query*) et obtient des documents pertinents pour cette demande.
- **Récupération multilingue** (*cross-language retrieval*): la requête est dans une langue autre que celle des documents.
- **Réponse aux questions** (*question answering*): la requête est une question en langage naturel, la réponse est un morceau de texte.
- **Traduction** des documents d'une langue à une autre.

# Extraire l'information

- **Résumer** le contenu d'un document en utilisant quelques phrases significatives.
- **Remplir des modèles préfixés** (*template*), avec des informations telles que qui, où, quand, ...
- **Sélectionner les termes pertinents** d'un ensemble de documents, par exemple pour créer l'index thématique d'un livre.

# Organiser les informations

- **Catégorisation** des textes: attribuer une certaine catégorie à chaque document d'une collection.
- **Grouper** (*clustering*) des documents en groupes homogènes par contenu. Par exemple, pour extraire des opinions et des jugements concernant un certain produit.
- **Recherchez le sujet** (*topic*) d'un document, tel qu'un message électronique, pour l'envoyer à un destinataire approprié.
- **Classer** des documents dans une hiérarchie de concepts.

# Construire des réseaux d'utilisateurs

- Les utilisateurs sont classés en fonction de leur intérêt pour certains documents.
- **Modélisation de l'utilisateur**: un profil personnalisé est créé, qui est ensuite utilisé pour proposer de nouveaux documents.
- **Systèmes de recommandation** de documents.



# Extraction d'information

Texte :

*San Salvador, 19 avril 1989 (ACANEFE)*

*Le président du San Salvador Alfredo Cristani a condamné l'assassinat d'origine terroriste du ministre de la justice Roberto Garcia Alvarado et a accusé du meurtre le Front de Libération National Farabundo Marti.*

Cadre:

## INCIDENT

**date :** 19 avril 1989

**lieu :** El Salvador : San Salvador (CITY)

**auteur :** Front de Libération National Farabundo

**victime :** Marti Roberto Garcia Alvarado

# Question - réponse

- Trouver la réponse à une question dans une collection de textes

*‘Quelle est l'**étoile** la plus **brillante** visible de la Terre?’*

1. **Sirius** est l'**étoile** la plus **brillante** visible de la Terre en dépit d'être une ....
- 2. **Nicolas Le Riche**, l'**étoile** plus **brillante** du ballet de l'Opéra national de Paris...

## Question – réponse (cont.)

- Découvrir les relations implicites entre question et réponse

*‘Qui est **l'auteur de** ‘Le Bourgeois gentilhomme’?’*

... Molière **a écrit** ‘Le Bourgeois gentilhomme’ en 1670.

... Jean-Laurent Cochet **a mis en scène** la représentation du ‘Le Bourgeois gentilhomme’ en 1980 ...

## Question – réponse (cont.)

Découvrir les relations implicites entre question et réponse

*‘Quel est la date de naissance de Mozart?’*

.... Mozart (1751 – 1791) ....

# Question – réponse (cont.)

Découvrir les relations implicites entre question et réponse

*‘Quelle est la distance entre Naples et Ravello?’*

‘De l'aéroport de **Naples**, suivez les panneaux ‘Autostrade’ (panneaux verts). Continuer en direction de Salerno (A3). Faire environ 6 km, payer le péage (1,20 euro). Continuer sur environ 25 km. Quitter l'autoroute à Angri (sortie Angri). Tourner à gauche, suivre les indications pour Ravello. Continuer pendant environ deux kilomètres, tourner à droite et suivre les indications pour "Costiera Amalfitana". Après 100 mètres, vous arrivez à un feu devant un pont très étroit. Attention à ne pas perdre le prochain panneau "Ravello" à environ 1 Km. Au feu. Maintenant, vous pouvez vous détendre et profiter de la vue (suivez cette route pendant 22 km). Arrivé à **Ravello** ....’

# Paramètres des applications TALN

## Paramètres d'évaluation

### **Robustesse**

C'est la capacité de l'application à gérer du matériel linguistique en entrée contenant du bruit et à accepter et analyser des entrées partielles ou incomplètes.

### **Puissance**

- Décrit la capacité à couvrir la langue de l'application, son champ d'action
- Considère "combien" de la langue est traitée correctement par l'application

### **Portabilité**

C'est la possibilité de l'appliquer à de nouveaux domaines (d'autres langages, d'autres langages sectoriels, d'autres types de textes), en modifiant la structure au minimum.

### **Généralisable**

C'est la capacité du modèle informatique à rendre compte de nouveaux phénomènes linguistiques, en appliquant des modèles dérivés du matériel linguistique relativement réduit.



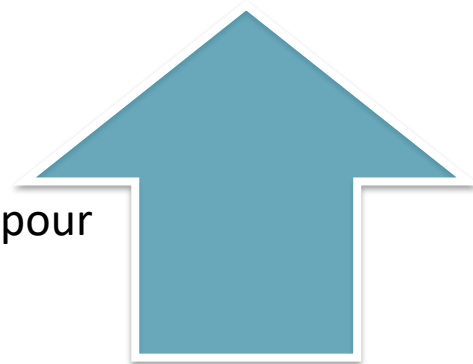
## Approches statistiques

- données extraites de textes réels
- méthode inductive
- reproduction de comportements simulant les tendances de la production linguistique réelle



## Approches basées sur des règles

- de type «grammatical»
- ensemble de conditions nécessaires et suffisantes pour spécifier un certain phénomène
- méthode déductive



- **Approches basées sur des règles**, riches en connaissances
  - La représentation du domaine est explicite, exprimée sous la forme de règles.
  - Cette représentation correspond à la connaissance d'un expert en la matière



- Approches basées sur des règles: **problèmes**
  - Il est très difficile de concevoir l'ensemble du système de règles nécessaire pour fournir à un ordinateur des connaissances linguistiques pour le traitement du langage.
  - Il est également très difficile de gérer la complexité et les interactions du système de règles.

- Approches basées sur des règles: **problèmes**
  - Il est très difficile de concevoir l'ensemble du système de règles nécessaire pour fournir à un ordinateur des connaissances linguistiques pour le traitement du langage.
  - Il est également très difficile de gérer la complexité et les interactions du système de règles.

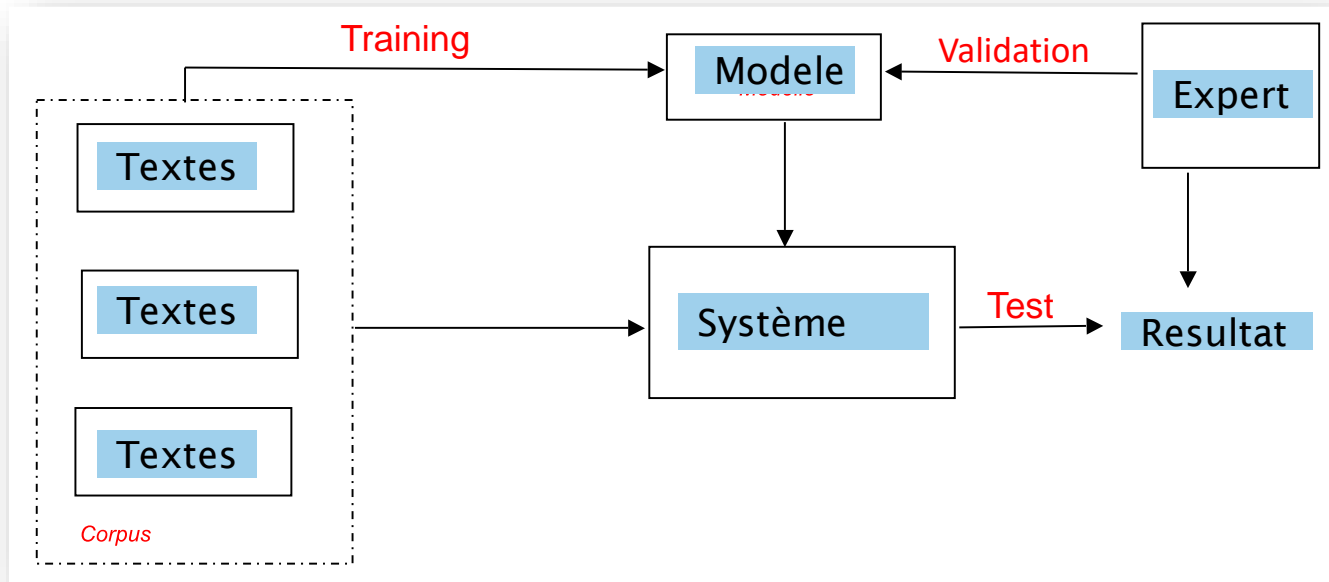
**Solution:** Au lieu d'un expert qui fournit à l'ordinateur des informations linguistiques sous forme de règles, l'expert note un texte avec des informations linguistiques et le programme apprend par lui-même les règles et leur utilisation.

# Approches principales à l'analyse du texte

- **Approches statistiques** (axées sur les données, pauvres en connaissances)
  - La représentation de la connaissance du domaine est implicite, exprimée en forme d'annotation d'un texte ou d'un corpus.
  - Un programme apprend automatiquement les règles et leur fréquence d'utilisation dans le texte
  - Les **modèles probabilistes** utilisés décrivent le comportement des mots dans les textes (GoogleTranslate)



- Approches statistiques



- **Approches statistiques: problèmes**
  - Il est très difficile et coûteux de construire des ressources linguistiques représentatives en quantité suffisante.
  - Nous n'essayons plus de reproduire la compétence linguistique avec des modèles qui formalisent nos facultés de compréhension linguistique, mais nous essayons de reproduire, pour une classe d'applications donnée, les performances linguistiques associées.
  - Cela se fait avec des modèles extraits automatiquement des données, qui doivent être en grande quantité et caractéristiques de l'application souhaitée.

- **Approches statistiques: avantages**
  - Ils permettent de rassembler des régularités présentes dans des collections de grands textes.
  - On observe des phénomènes "objectifs" vis-à-vis d'un langage, qui peuvent échapper à l'analyse "subjective" pratiquée par les linguistes.

Définition: Un programme *apprend* à partir d'une expérience de formation  $F$  à exécuter la tâche  $T$  évaluée par une mesure de performance  $P$ , si la performance  $P$  à la tâche  $T$  s'améliore après l'exposition  $F$ .

Exemple

**Tâche  $T$ :** classer les verbes dans des classes prédéfinies

**Expérience de formation  $F$ :** base de données de paires de verbes avec leurs attributs et réponses correctes

**Mesure de performance  $P$ :** % de nouveaux verbes correctement classés (par rapport à une classification établie par un expert)

La tâche la plus étudiée en apprentissage automatique (*machine learning*) consiste à déduire une fonction qui attribue les exemples représentés comme vecteurs de traits distinctifs à une classe parmi un ensemble fini de catégories données.

Exemple

Etant donné un ensemble de verbes.

**Tâche:** classification binaire: verbes de mouvement (par exemple, *courir*, *sauter*, *marcher*) et verbes de changement d'état (par exemple, *fondre*, *cuire*, *devenir*).

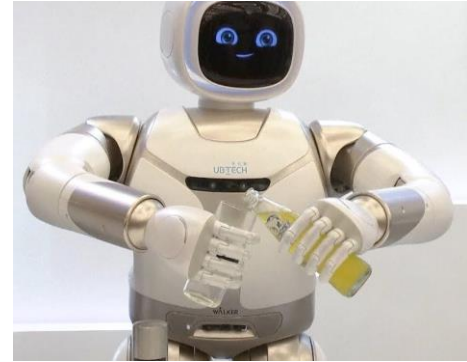
**Propriétés:** chaque fois que nous trouvons le verbe dans le corpus est transitif? est-ce passif? Votre sujet est-il animé?



# Apprentissage par classification: un exemple



<u>Exemple</u>	<u>Trans?</u>	<u>Pass?</u>	<u>Anim?</u>	<u>Classe</u>
Courir	5%	3%	90%	MoM
Sauter	55%	5%	77%	MoM
Cuire	10%	9%	20%	CoS
Devenir	80%	69%	88%	CoS



- Si  $\text{Pass} < 9\%$  et  $\text{Anim} > 20\%$  alors le verbe est MoM, sinon CoS
- Comment classifier un nouveau verbe?
- “grimper”: Trans 2%, Pass 1%, Anim 90%     $\rightarrow$     MoM

- Il est important de pouvoir évaluer de façon expérimentale (**réplicable**) les résultats obtenus sur une tâche donnée.
- La performance d'un algorithme est vérifiée par rapport au comportement humain (**gold standard**).
- Les algorithmes sont améliorés jusqu'à ce qu'ils approchent les jugements des humains.
- L'étude des mesures d'évaluation et d'évaluation des systèmes de traitement automatique des langues est un élément fondamental de la linguistique computationnelle.
- **Chaque système est évalué en le comparant à 'l'état de l'art'**= la performance du système qui a obtenu jusqu'alors de meilleurs résultats sur un gold standard.

- (*autre exemple*): à partir d'un ensemble de textes, trouver ceux et uniquement ceux **pertinents** par rapport à une **classe** considérée
- Evaluer la capacité du système à trouver les textes **pertinents** et **uniquement** ceux là
- Lorsque le système retourne une réponse par rapport à un texte et une classe, deux choix s'offrent à lui :
  - Le message **appartient** selon lui à la classe
  - Le message **n'appartient pas** selon lui à la classe
- En face de ces deux possibilités de réponses, deux cas où :
  - Le message **appartient** à la classe
  - Le message **n'appartient pas** à la classe

# Evaluation de la classification

Nom du cas	Abréviation	Description
Vrai positif	VP	Le système trouve <b>à raison</b> le message comme <b>appartenant</b> à la classe
Faux positif	FP	Le système trouve <b>à tort</b> le message comme <b>appartenant</b> à la classe
Vrai négatif	VN	Le système trouve <b>à raison</b> le message comme <b>n'appartenant pas</b> à la classe
Faux négatif	FN	Le système trouve <b>à tort</b> le message comme <b>n'appartenant pas</b> à la classe

# Evaluation de la classificati

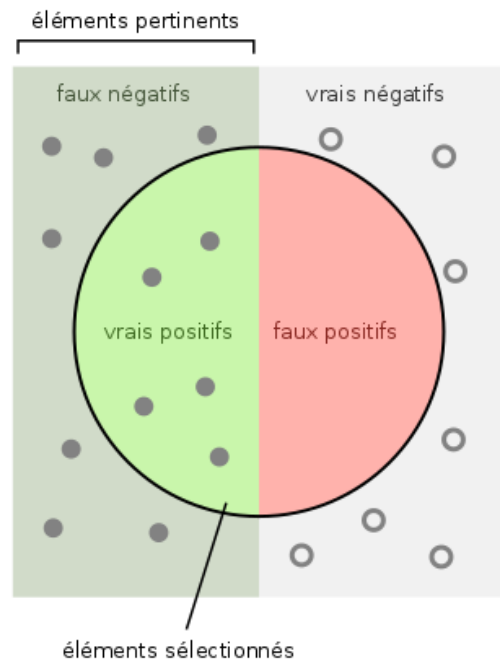
Mesure des performances: le **rappel** et la **précision**

**Rappel:** nombre de documents pertinents retrouvés au regard du nombre de documents pertinents dans l'ensemble de textes


$$rappel = \frac{\text{nb de documents correctement attribués à la classe } i}{\text{nb de documents appartenant à la classe } i}$$

**Précision:** le nombre de documents pertinents retrouvés rapporté au nombre de documents total proposé par le moteur de recherche pour une requête donnée

$$precision = \frac{\text{nb de documents correctement attribués à la classe } i}{\text{nb de documents attribué à la classe } i}$$



Combien de candidats sélectionnés sont pertinents ?

Précision = 

Combien d'éléments pertinents sont sélectionnés ?

Rappel = 

- Mesure populaire qui combine la précision et le rappel est leur moyenne harmonique, nommée F-mesure ou F-score

$$\text{F-mesure} = 2 \cdot \frac{\textit{precision} \cdot \textit{rappel}}{\textit{precision} + \textit{rappel}}$$

# Traitement automatique de base

## Tokenisation en mots

# Mots, tokens, formes, lemmes...

- Unités ``logiques'' pour le traitement de textes :  
Document  $\supset$  paragraphe  $\supset$  phrase  $\supset$  ``mot''  $\supset$  ``caractère''
- Mais un « mot » n'est pas un unité bien définie :
  - Exemples : avion, mangée, très, Robert, SNCF, 42...



# Mots, tokens, formes, lemmes...

- **Forme** : notion graphique du mot (Igor Mel'čuk)
- **Lemme** : intersection entre une forme (graphique) et un sens, parfois par composition de morphèmes
- **Token** (jeton, identificateur) : unité minimale d'information détectée lors de l' « analyse lexicale » ou « tokenization » – En français, souvent nommée « lexème »

- **Segmenter un texte en « unités minimales » pour le traiter**
- Ensemble d'automates qui reconnaissent les tokens en acceptant des chaînes, éventuellement en les typant
  - Lexème :  $-?[A-Z] ?[a-z]^*$
  - Ponctuation :  $.|...|,|!|?$
  - Nombre :  $-?[0-9]^*(,|.)[0-9]^*$
  - ...

## Example:

Les étudiants, ceux du BUT2, n'ont-ils pas tous 15,3 de moyenne ?

Les | étudiants | , | ceux | du | BUT2 | , | n' | ont | -ils | pas | tous | 15,3 | de | moyenne | ?

- **Lemme** : unité autonome (composée de morphèmes) permettant de constituer le lexique d'une langue
  - **Morphèmes** : les « parties » du lemme
  - **Autonome** : peut-être utilisé tel quel dans une phrase
- **Lemmatisation, trouver les lemmes pour chaque token au sein d'une phrase**

## Exemple:

Je porte des pommes de terre

je | porter | une | pomme de terre

Importance des mécanismes lemmatisation / racinisation pour la représentation du langage :

- Eviter des dictionnaires trop volumineux comportant toutes les formes possibles (en français, ~60 000 lemmes « courants », mais >500 000 formes fléchies « courantes »)
  - Gain en espace de stockage
  - Moindre complexité pour l'encodage du lexique

Importance des mécanismes lemmatisation / racinisation pour la représentation du langage :

- Parvenir à une représentation structurée du texte, à partir de laquelle on peut faire des traitements :
  - Le « sens » est plutôt lié au lemme qu'à la forme (par exemple : temps verbaux, masculin / féminin, etc.)
  - Lier les catégories grammaticales aux lemmes plutôt qu'aux formes : désambiguïsation morpho-syntaxique (à venir)

# Expressions composées

- **Expression composée** : lemmes juxtaposés dont le sens a une **signification différente** des lemmes qui le composent, ``expression figée’’
- Divers niveaux d'**agglutination** :
  - Locutions (tokens séparés) : « pomme de terre », « cordon bleu », « garde fou », « petit pois »...
  - Tokens séparés par un symbole de ponctuation : « coupe-gorge », « abat-jour », « aujourd'hui », « presque-île »
  - Tokens unifiés (collés) : « gentilhomme », « monsieur », « lorsque », « toutefois », « vinaigre », « autobus »
- Peuvent être fléchies (``il se rendit compte’’) et parfois dérivés (``mise en oeuvre’’, ``avant-gardisme’’)

- Les expressions composées représentent un défi :
  - Les lemmes doivent être groupés lors de la tokenisation
  - Elles introduisent de l'**ambiguïté** (plus ou moins figées)
  - De nouvelles apparaissent tous les jours
  - Contrairement à la dérivation, par définition le sens ne peut être déduit des lemmes dont elles sont composées
- Elles sont généralement traitées comme des **unités particulières du lexique**
- Elles exigent la plupart du temps un prétraitement lors de la tokenization

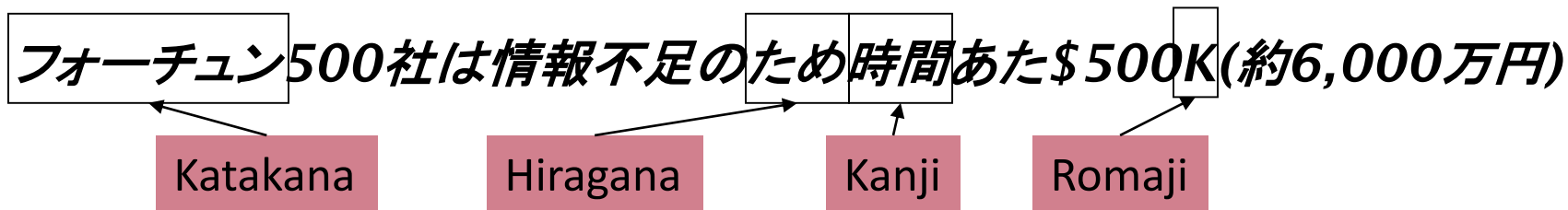
# Tokenization: problèmes

- Français
  - *L'ensemble* → un token ou deux?
    - *L ? L' ? Le ?*
    - On veut que *l'ensemble* puisse correspondre avec *un ensemble*
- Les expressions composés en Allemands ne sont pas segmentés
- *Lebensversicherungsgesellschaftsangestellter*
  - 'employé de la compagnie d'assurance-vie'



# Tokenization: problèmes

- Chinois et Japonais, pas d'espace entre les mots:
  - 莎拉波娃现在居住在美国东南部的佛罗里达。
  - 莎拉波娃 现在 居住 在 美国 东南部 的 佛罗里达
  - Sharapova vit maintenant dans le sud-est des États-Unis, en Floride
- Plus compliqué en japonais, avec plusieurs alphabets mélangés
- Dates/amounts in multiple formats



L'utilisateur final peut exprimer la requête entièrement en hiragana!

# Traitement automatique de base

Normalisation et racinisation (*stemming*)

# Normalisation

- Besoin de «normaliser» les termes
  - Recherche d'informations: le texte indexé et les termes de requête doivent avoir la même forme (ex. PV et P.V.)
- Nous définissons implicitement des classes d'équivalence de termes
  - par exemple, en supprimant des points dans un terme (ex. M.)
- Alternative: expansion asymétrique:
  - Entrée: **window** -> Rechercher: **window, windows**
  - Entrée : **windows** -> Rechercher: **Windows, windows, window**
  - Entrée : **Windows** -> Rechercher: **Windows**
- Potentiellement plus puissant, mais moins efficace

# Modification de la casse

- Applications comme Recherche d'informations: réduire toutes les lettres en minuscules
  - Puisque les utilisateurs ont tendance à utiliser des minuscules
  - Exception possible: majuscules au milieu de la phrase?
    - par exemple, *General Motors*
    - Cigales (***C**lub d'*investisseurs* pour une *gestion alternative* et *locale* de l'*épargne solidaire**) vs cigales
    - FLOT (*Éducation: Formation en Ligne Ouverte à Tous*) vs flot
- Pour l'analyse des sentiments, MT, extraction d'informations la casse est utile.

# Racinisation/Stemming

- Réduire les termes à leurs racines lors de la récupération d'informations
- *Stemming* est un hachage brut des affixes
  - Depend du langage
  - e.g., ***automate(s), automatique, automatisa****tion* sont reduit a ***automat***.

*par exemple compressé  
et la compression sont les deux  
acceptés comme équivalent de  
compresser.*



*par exempl compress  
et la compress sont le deux  
accept comm équivalent à  
compress*

# Algorithme de Porter

## Implementations (Français)

Description:

<http://snowballstem.org/algorithms/french/stemmer.html>

Demo:

[snowballstem.org/demo.html](http://snowballstem.org/demo.html)

# Analyse morpho-syntactique

**BUT:** analyser chaque mot pour lui associer divers types d'informations telles que **la catégorie grammaticale** (parts-of-speech), **des traits morphologiques** ainsi que le **lemme correspondant**

## Classes ouvertes

Noms	Verbes	Adjectifs <i>gros petite</i>
Propres <i>IBM</i> <i>Italie</i>	Communs <i>chat/chats</i> <i>neige</i>	Adverbes <i>lentement</i>
	<i>voir</i> <i>enregistré</i>	
		Nombres <i>122,312</i> <i>un</i>

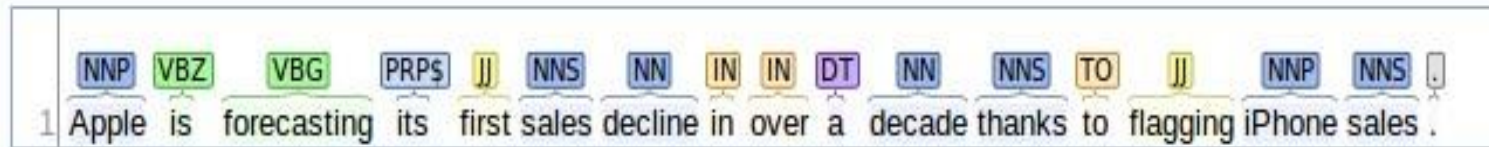
## Classes fermées

Déterminants <i>le du</i>	Prépositions <i>de avec</i>
Conjonctions <i>et car</i>	Particules <i>off up</i>
Pronoms <i>il celui-ci</i>	Interjections <i>Oh Hé</i>

# Etiquetage morpho-syntaxiques

- Les mots ont généralement plus d'une étiquette possible
  - Le bois vient de France. → le=det, bois=nom
  - Je le bois. → le = pronom, bois = verbe
- Objectif: déterminer l'étiquette pour une instance d'un mot

## Part-of-Speech:





# Exemples d'étiquetage et difficultés

- *Entrée:* Le débat est relancé.
  - ambiguïtés: le=det/pro débat=verbe/nom est=verbe/nom
- *Sortie:* Le/DET débat/NOM est/VER relancé/VER
- Applications:
  - synthèse vocale: comment prononcer *est* ?
  - recherche dans un corpus: *est* en tant que nom
  - entrée d'un analyseur syntaxique

- Combien d'étiquettes sont correctes ? **Précision**
- étiqueteurs sur l'anglais autour de 97%
- mais baseline simple = 90%
  - chaque mot du lexique → étiquette la plus fréquente
  - mots inconnus → noms
- beaucoup de mots ne sont pas ambigus
  - déterminants, prépositions, ponctuation...

# Désambiguïsation des parties du discours

Elle le fait.



Elle	PRON
le	PC
fait	VERB_P3SG
.	SENT

Elle montre le fait.



Elle	PRON
montre	V_P3SG
le	DET_SG
fait	NOUN_SG
.	SENT

- Contexte des mots
- Le bois vient de France
  - DET NOM VER PREP NAM
  - PRO VER VER PREP NAM
- Connaissance des probabilités d'étiquettes des mots

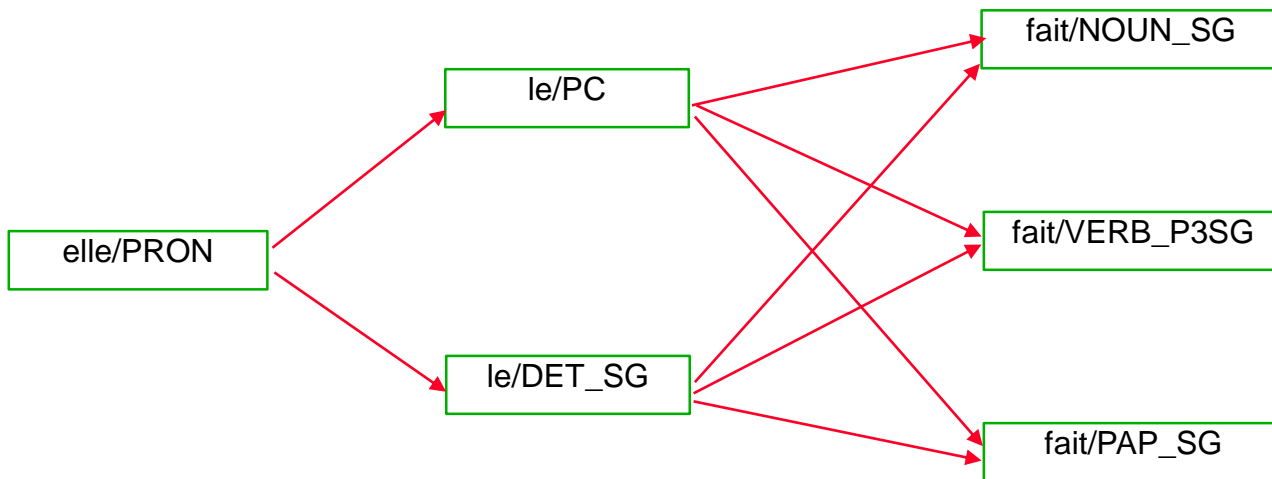
# Corpus French TreeBank

- Projet initié en 1997
- <http://ftb.linguist.univ-paris-diderot.fr/>
- Corpus journalistique (Le Monde) 1 million de mots
- Annotations
  - Morphosyntaxique
    - POS
    - Sous-catégorisation
    - Inflection
    - Lemme
    - Parties pour mots composés
  - Constituants
  - Fonctions

- Calcul des probabilités à partir d'un corpus d'apprentissage
  - probabilités lexicales
    - $\text{prob}(\text{tag} \mid \text{mot}) = \text{freq}(\text{mot}, \text{tag}) / \text{freq}(\text{mot})$
  - probabilités contextuelles
    - **bigrammes** :
      - $\text{prob}(\text{tag}_2 \mid \text{tag}_1) = \text{freq}(\text{tag}_1 \text{ tag}_2) / \text{freq}(\text{tag}_1)$
    - **trigrammes** :
      - $\text{prob}(\text{tag}_3 \mid \text{tag}_1 \text{ tag}_2) = \text{freq}(\text{tag}_1 \text{ tag}_2 \text{ tag}_3) / \text{freq}(\text{tag}_1 \text{ tag}_2)$

# Exemple

elle le fait



# Fréquences des mots et des étiquettes

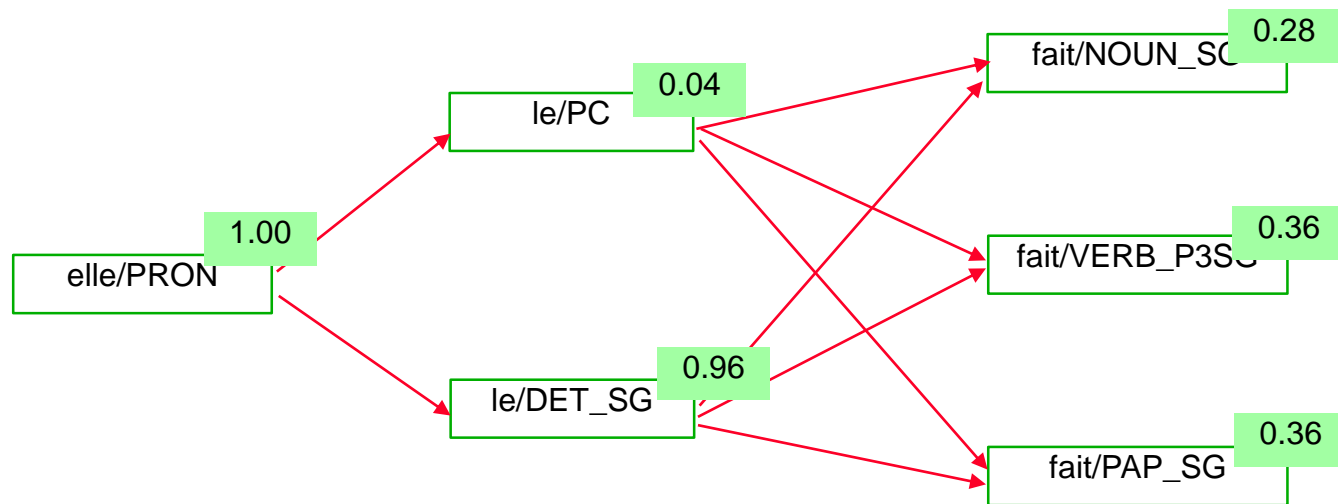
Corpus d'apprentissage: extrait "Le Monde"

freq	elle	le	fait	montre	Tot.
PRON	17	--	--	--	320
DET_SG	--	239	--	--	1329
PC	--	11	--	--	179
VERB_P3SG	--	--	5	2	371
NOUN_SG	--	--	4	0	1931
PAP_SG	--	--	5	--	207
...	...	...	...	...	...
Tot.	17	250	14	2	15.000



# Calcul des probabilités lexicales

prob ( PRON   elle )	= 17 / 17 = 1.00
prob ( DET_SG   le )	= 239 / 250 = 0.96
prob ( PC   le)	= 11 / 250 = 0.04
prob ( NOUN_SG   fait )	= 4 / 14 = 0.28
prob ( PAP_SG   fait )	= 5 / 14 = 0.36
prob ( VERB_P3SG   fait )	= 5 / 14 = 0.36



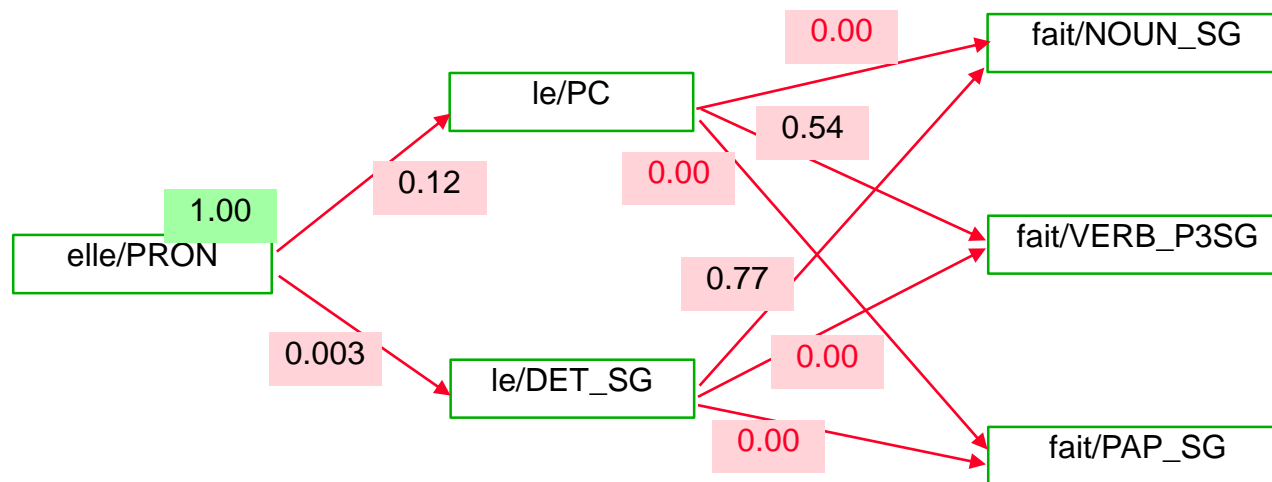
$\text{prob} ( \text{elle/PRON le/PC fait/VERB\_P3SG} )$	$= 1.00 * 0.04 * 0.36$	$= 0.014$
$\text{prob} ( \text{elle/PRON le/DET\_SG fait/NOUN\_SG} )$	$= 1.00 * 0.96 * 0.28$	$= 0.269$
$\text{prob} ( \text{elle/PRON le/DET\_SG fait/VERB\_P3SG} )$	$= 1.00 * 0.96 * 0.36$	$= 0.346$

# Fréquences des séquences d'étiquettes

		tag <sub>2</sub>						Tot.
		PRON	DET_SG	PC	VERB_P3SG	N_SG	PAP_SG	
tag <sub>1</sub>	PRON	--	1	38	82	--	32	320
	DET_SG	4	5	--	--	1033	--	1329
	PC	--	--	3	59	--	--	179
	VERB_P3SG	17	53	10	--	9	--	371
	NOUN_SG	3	29	12	46	13	1	1931
	PAP_SG	1	42	--	1	10	--	207
	...	...	...	...	...	...	...	...
Tot.		320	1329	179	564	1931	207	15.000

...

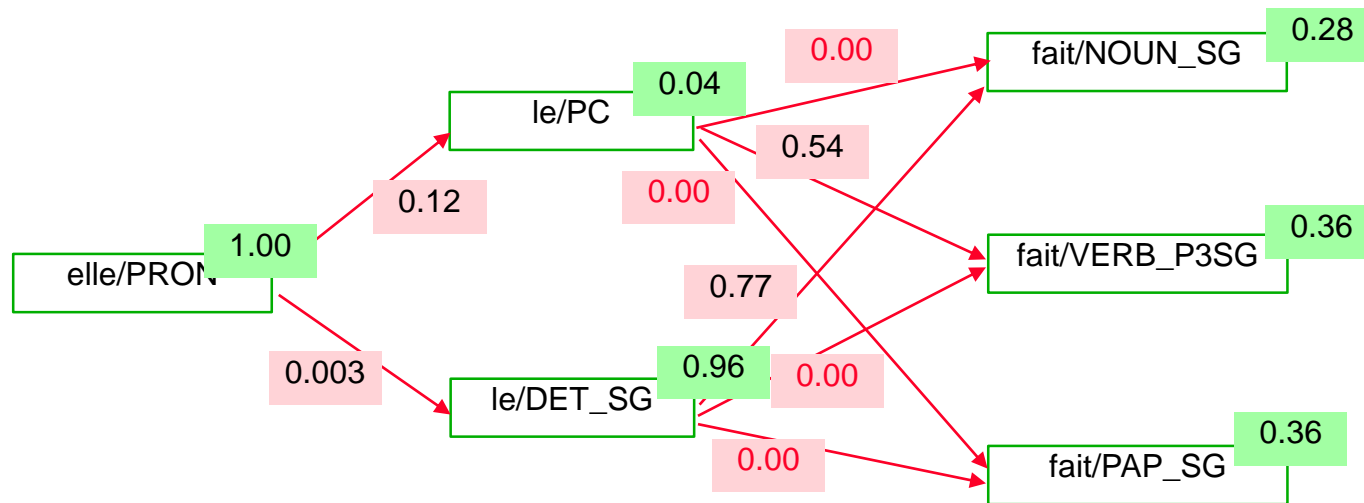
$\text{prob} ( \text{PC}   \text{PRON} )$	$= 38 / 320 = 0.12$
$\text{prob} ( \text{DET\_SG}   \text{PRON} )$	$= 1 / 320 = 0.003$
$\text{prob} ( \text{VERB\_P3SG}   \text{PC} )$	$= 97 / 179 = 0.54$
$\text{prob} ( \text{PAP\_SG}   \text{PC} )$	$= 0 / 179 = 0.00$
$\text{prob} ( \text{NOUN\_SG}   \text{PC} )$	$= 0 / 179 = 0.00$
$\text{prob} ( \text{NOUN\_SG}   \text{DET\_SG} )$	$= 1033 / 1329 = 0.77$
$\text{prob} ( \text{VERB\_P3SG}   \text{DET\_SG} )$	$= 0 / 1329 = 0.00$
$\text{prob} ( \text{PAP\_SG}   \text{DET\_SG} )$	$= 0 / 1329 = 0.00$
...	



$$\text{prob} ( \text{elle/PRON le/PC fait/VERB\_P3SG} ) = 0.12 * 0.54 = 0.0648$$

$$\text{prob} ( \text{elle/PRON le/DET\_SG fait/NOUN\_SG} ) = 0.003 * 0.77 = 0.0231$$

$$\text{prob} ( \text{elle/PRON le/DET\_SG fait/VERB\_P3SG} ) = 0.003 * 0.00 = 0$$

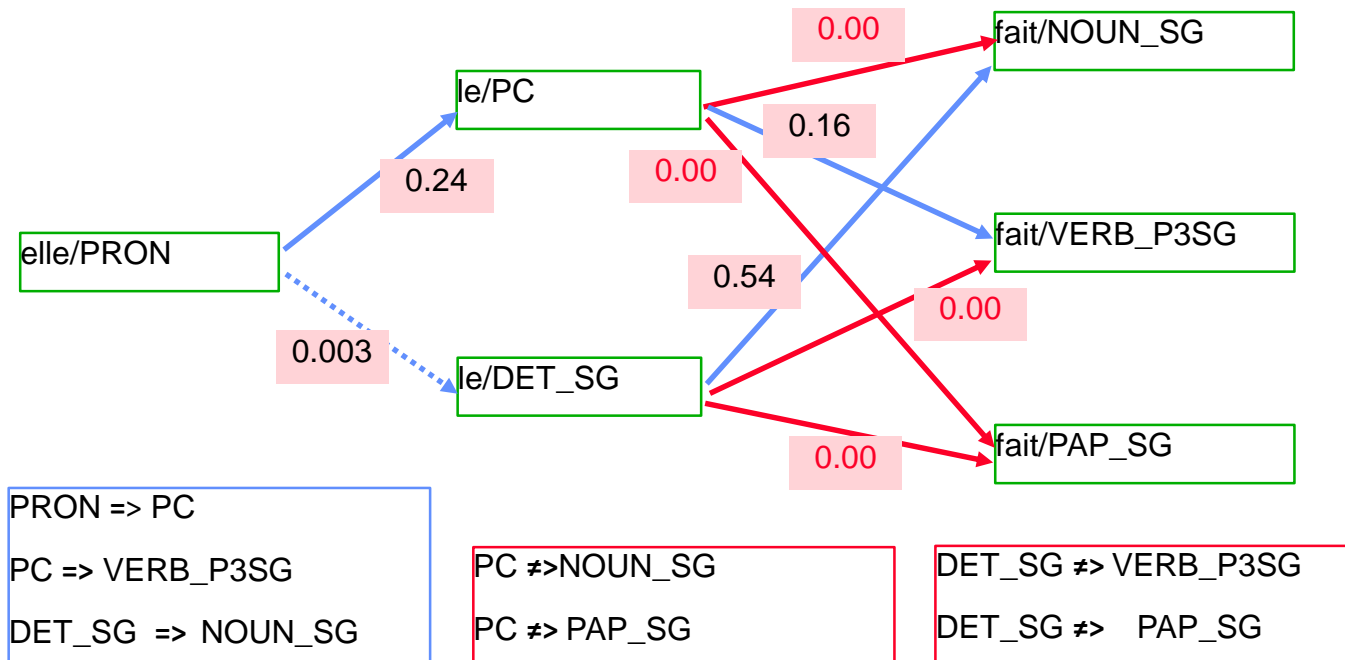


$$\text{prob ( elle/PRON le/PC fait/VERB_P3SG )} = 1.00 * 0.12 * 0.04 * 0.54 * 0.36 = 0.00093$$

$$\text{prob ( elle/PRON le/DET_SG fait/NOUN_SG )} = 1.00 * 0.003 * 0.96 * 0.77 * 0.28 = 0.00058$$

$$\text{prob ( elle/PRON le/DET_SG fait/VERB_P3SG )} = 1.00 * 0.003 * 0.96 * 0.00 * 0.36 = 0$$

- règles “positives”
  - pour définir les séquences possibles
  - exemple:
    - un pronom personnel est suivi d'un verbe
    - un déterminant est suivi d'un nom
- règles “négatives”
  - pour exclure des séquences impossibles
  - exemple:
    - un pronom enclitique ne précède pas un nom
    - un déterminant ne précède pas un verbe





- Brill tagger.
- L'idée générale très simple: deviner l'étiquette de chaque mot, puis revenir en arrière et corriger les erreurs. De cette façon, un tagger Brill transforme successivement un mauvais marquage d'un texte en un meilleur (**méthode d'apprentissage supervisée**).
- Contrairement au marquage n-gram, il ne compte pas les observations mais compile une liste de règles de ``correction transformationnelle''.
- Les règles sont linguistiquement interprétables
- <https://www.nltk.org/api/nltk.tag.html#module-nltk.tag.brill>

# Modèles de Markov cachés

- Les modèles de Markov cachés (HMM) sont largement utilisés pour attribuer la séquence d'étiquettes correcte à des données séquentielles ou pour évaluer la probabilité d'une étiquette et d'une séquence de données données.
- Ces modèles sont des machines à états finis caractérisés par un certain nombre d'**états**, des **transitions entre ces états** et des **symboles de sortie** émis dans chaque état.
- Le HMM est une extension de la chaîne de Markov, où chaque état correspond de manière déterministe à un événement donné. Dans le HMM, l'observation est une fonction probabiliste de l'état.
- Les HMM partagent l'hypothèse de la chaîne de Markov, à savoir que la probabilité de transition d'un état à un autre ne dépend que de l'état actuel - c'est-à-dire que la série d'états ayant conduit à l'état actuel n'est pas utilisée.

# Modèles de Markov cachés

- Le HMM est un graphe orienté, avec des arêtes pondérées en fonction de la probabilité (représentant la probabilité d'une transition entre les états source et récepteur), où chaque sommet émet un symbole de sortie lorsqu'il est entré. Le symbole (ou observation) est généré de manière non déterministe.
- Pour cette raison, le fait de savoir qu'une séquence d'observations en sortie a été générée par un HMM donné ne signifie pas que la séquence d'états correspondante (et ce qu'est l'état actuel) est connue. C'est le "caché" dans le modèle de Markov caché.
- Un HMM est souhaitable pour la tâche d' étiquetage morpho-syntaxique car la séquence d'étiquettes présentant la plus grande probabilité peut être calculée pour une séquence donnée de mots. Pour tenir compte de la combinaison optimale des tags pour une unité plus grande, telle qu'une phrase, le HMM exploite l'algorithme de Viterbi, qui calcule efficacement le chemin optimal à travers le graphe étant donné la séquence de mots.

Les étiqueteurs grammaticaux sont très nombreux pour les langues saxonnes mais plus rares pour le français. Des étiqueteurs sont accessibles avec un modèle pour le français prêt à l'emploi, des autres peuvent fonctionner pour le français mais doivent être entraînés sur un corpus français pré-étiqueté.

## **NLTK (Natural Language Toolkit)**

<http://www.nltk.org/>

## **Stanford Parser et CoreNLP (méthodes statistiques)**

<https://stanfordnlp.github.io/CoreNLP/>

<http://corenlp.run/>

<http://nlp.stanford.edu:8080/parser/index.jsp>

## **TreeTagger**

<https://cis.uni-muenchen.de/~schmid/tools/TreeTagger/>

## **spaCy (méthodes deep)**

<https://spacy.io/>

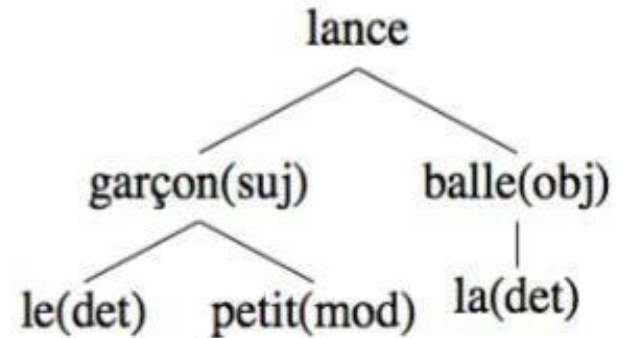
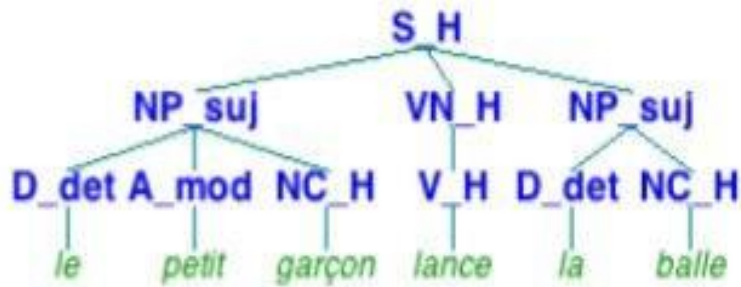
<https://explosion.ai/demos/>

# Traitement automatique de base

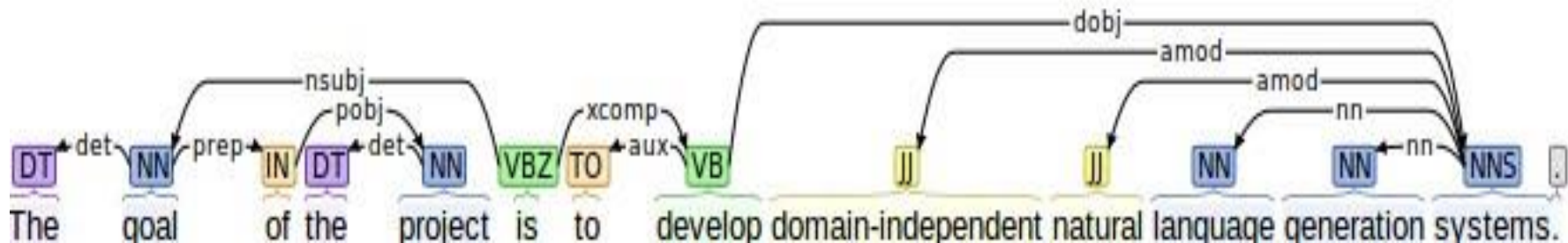
## Analyse syntaxique

- Analyse syntaxique traditionnelle
  - Généralement fondée sur le paradigme génératif de Chomsky
  - Objet = générer tous et seulement les énoncés possibles dans une langue (énoncés grammaticaux)
  - En analyse = associer à un énoncé (phrase) grammatical(e) de la langue sa structure syntaxique
    - arbre des séquences de réécritures permettant d'obtenir la phrase à partir de l'axiome S de la grammaire

# Exemple de sortie attendue



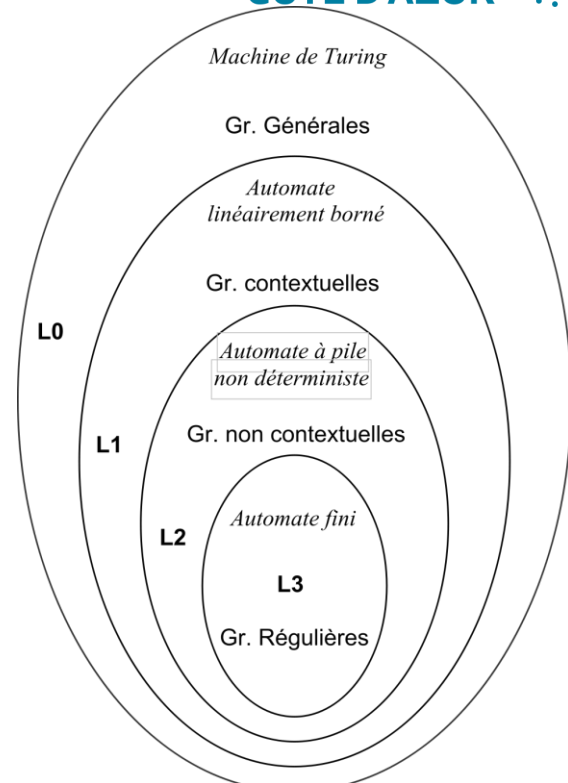
# Exemple d'analyse en dépendances





# Grammaires

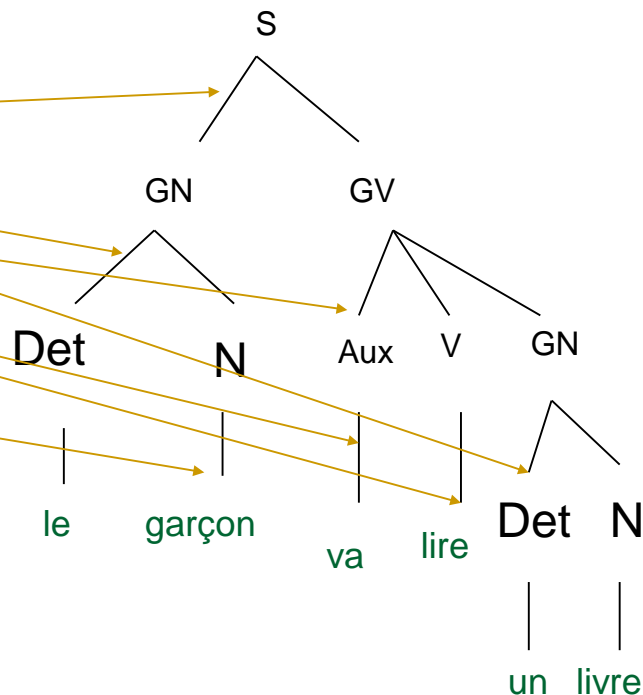
- $G=(V_n, V_t, R, S)$ 
  - $V_n$  : vocabulaire non terminal
  - $V_t$  : vocabulaire terminal
  - $R$  : ensemble de règles de réécriture
  - $X \rightarrow Y S$  : axiome de la grammaire
- Suivant les règles de  $R$ 
  - Grammaire non contrainte  $\rightarrow$  trop « lâche »
  - Grammaire en contexte :
    - «  $X$  se réécrit  $Y$  dans le contexte  $u v$  »
    - $uXv \rightarrow uYv$
  - Grammaire hors contexte :  $X \rightarrow Y$
  - Grammaire régulière (trop figée)
    - $A \rightarrow a$  ou  $A \rightarrow aB$



# Grammaires hors-contexte

- Exemple :

- $S \rightarrow GN\ GV$
- $GN \rightarrow Det\ N$
- $GV \rightarrow (Aux)\ V\ GN$
- $Aux \rightarrow va$
- $V \rightarrow lire\ |\ bat\ |\ mange\ |\dots$
- $Det \rightarrow le\ |\ la\ |\ les\ |\ un\ |\dots$
- $N \rightarrow garçon\ |\ livre\ |\ pomme\ |\dots$

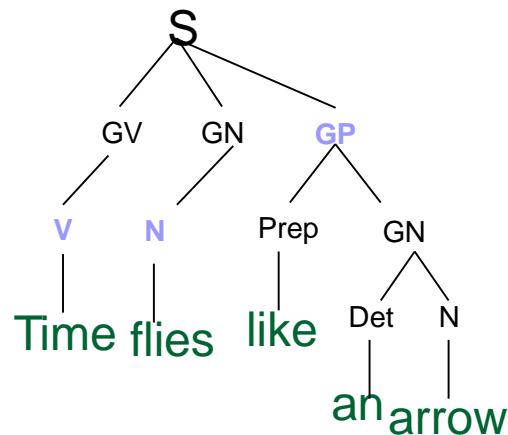
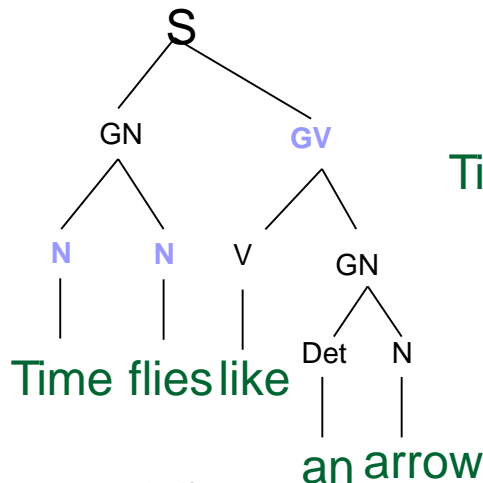
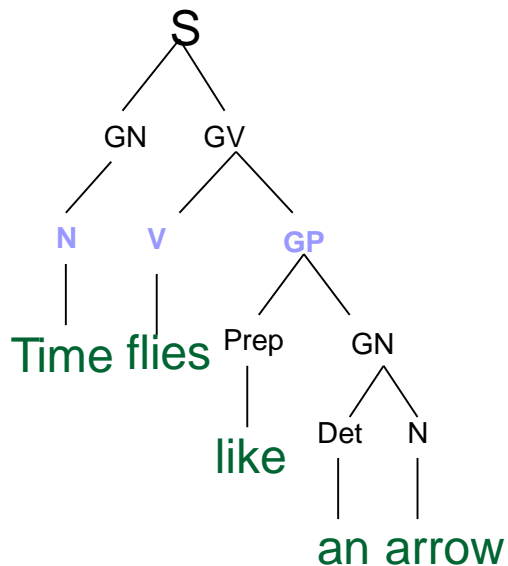


Le garçon va lire un livre

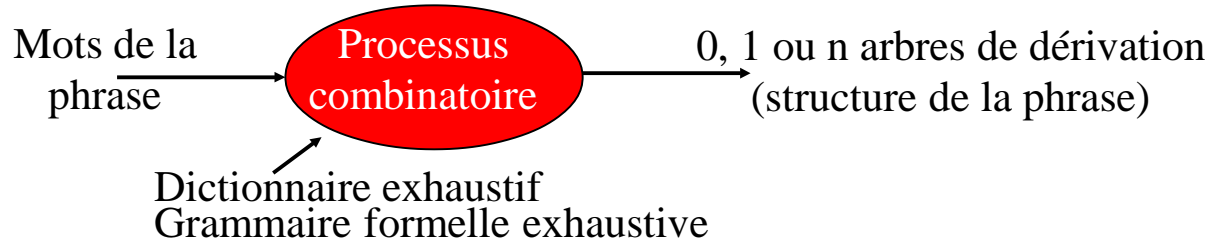
Mais aussi : *le pomme va mange la livre*

# Grammaires hors-contexte

- Différences entre structure de surface et structures profondes
- Exemple « chomskyen » : Time flies like an arrow:

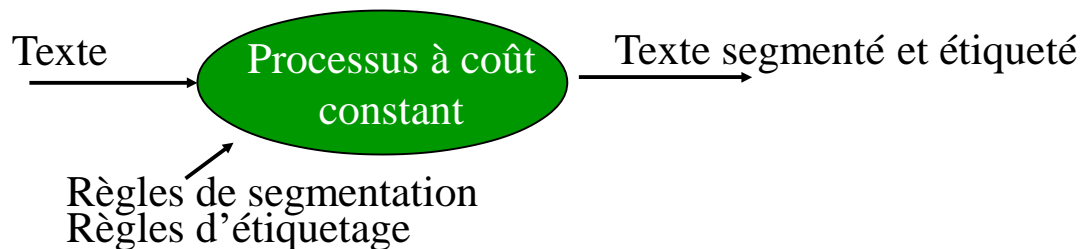


- Théorie des langages formels de Chomsky
  - Formalisation mathématique pas une théorie linguistique
  - La langue n'est pas un langage indépendant du contexte
    - Les accords
- Grammaires contextuelles insuffisantes
  - Constituants discontinus : Combien cette salle a-t-elle de fenêtres ?



- Caractéristiques (HPSG, LFG, TAG, ...) :
  - Règles de grammaire de type hors-contexte
  - Structures de traits
  - Unification
- Problème : manque de robustesse

Analyse robuste, analyse partielle, analyse de surface (shallow parsing)



- Approche empirique : héritage de la reconnaissance de la parole
- Travail sur texte réel, but opérationnel d'abord
- Analyse vue comme un processus informatique
- Principalement des méthodes statistiques

- Robustesse : plusieurs définitions dans la littérature du TAL
- Idée commune :
  - Capacité d'un système de TAL à traiter des données linguistiques réelles (produites par des locuteurs indépendamment du système)
- Définition (pour un analyseur)
  - Capacité d'un système à produire des analyses utiles pour des textes réels
  - Analyses utiles : analyses (au moins partiellement) correctes et utilisables dans une tâche automatique (application)

- Une analyse au moins pour chaque entrée
  - Situations d'absence d'analyses fréquentes dans les analyseurs traditionnels
  - Enoncés agrammaticaux dans les textes réels
  - Mais, plus fréquemment : constructions grammaticales non prédites par le modèle ou les descriptions linguistiques de l'analyseur
- Nombre d'analyses concurrentes limité
  - Les analyseurs traditionnels produisent souvent de trop nombreuses analyses (parfois des milliers pour une longue phrase), dont des analyses redondantes (ambiguïtés artificielles)



- Emergence de méthodes d'analyse robuste
- Trois tendances générales
  - Ajout de mécanismes ad hoc spécifiques pour rendre les analyseurs traditionnels robustes
  - Analyse à base de modèles statistiques
  - Analyse de surface à base de règles (rule-based *shallow parsing*)

- Idée de base
  - Limiter la « profondeur » et la richesse de l'analyse syntaxique
  - Prévoir la possibilité d'analyses partielles
- But
  - Obtenir des structures syntaxiques minimales, sous- spécifiées mais linguistiquement motivées (syntagme noyau = *chunk*)
  - Des structures utiles en tant que telles dans des applications
  - Première phase d'une analyse syntaxique plus complète

# Exemple d'analyse

- [ Bill NP] [vit V] [ l'homme NP] [ sur la colline PP] [ avec un  
téléscope PP]
- Chunks : NP, V, PP
- Ambiguïté de rattachement implicite

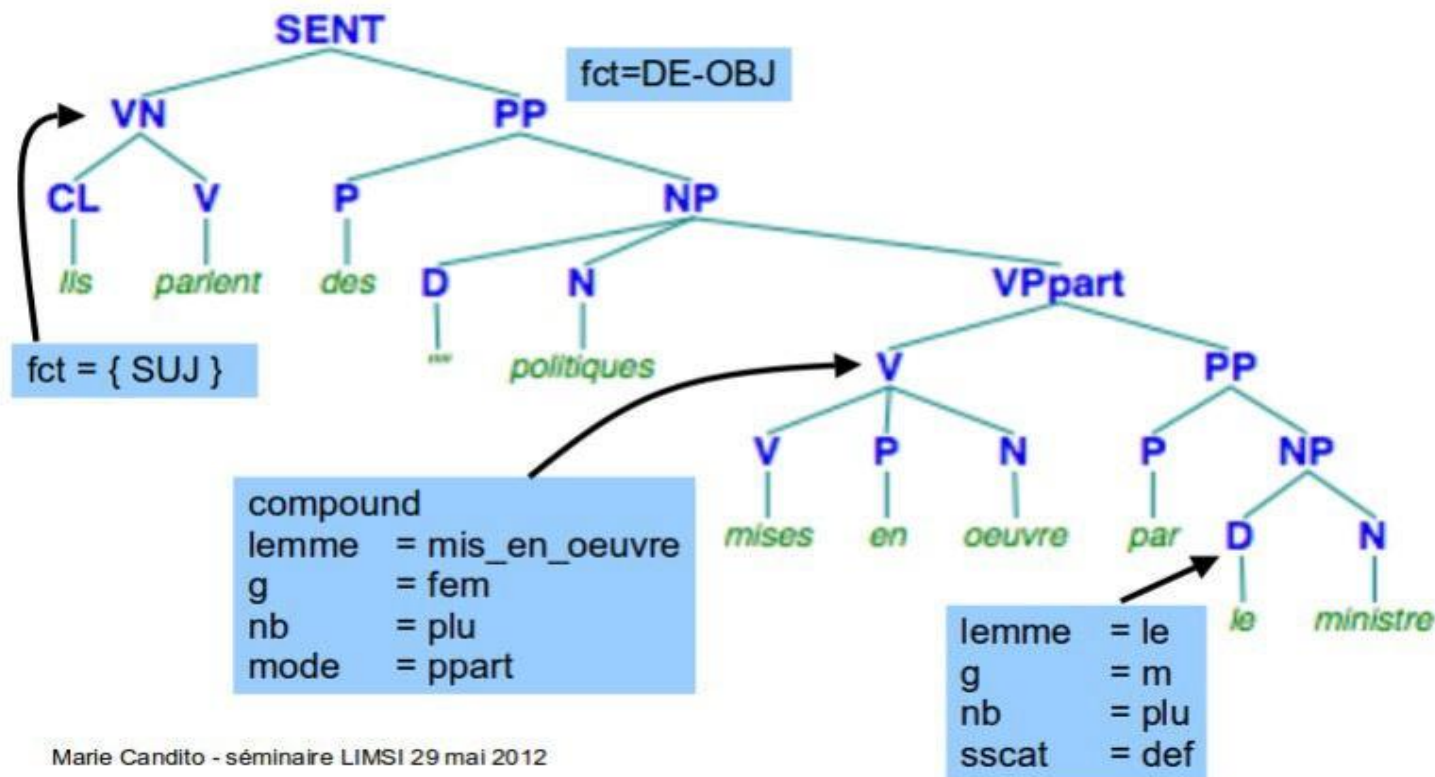
# Analyse de surface: étapes de traitements

- Prétraitement
  - Etiquetage morpho-syntaxique (segmentation, analyse morphologique, désambiguïsation)
- Analyse syntaxique de surface
  - Reconnaissance des syntagmes noyaux (chunks) : SN, SP, SV
  - Groupes complexes et propositions
  - Attribution de fonctions syntaxiques (Sujet, Objet, etc.)
- Analyse incrémentale

# Analyse par apprentissage supervisé

- Nécessité de grands corpus annotés
  - Penn TreeBank pour l'anglais
  - French TreeBank pour le français

# Représentation dans le FTB



# Principe de l'analyse probabiliste en constituants

- Probabilistic context-free grammar (PCFG)
  - dès (Booth, 69)
  - une CFG + probabilités:
    - chaque règle est associée à une probabilité
- Probabilité d'un arbre
  - = probabilité conjointe de toutes les applications de règles sous-jacente à l'arbre
  - «grammaires hors-contexte»
  - => hypothèse d'indépendance entre chaque règle
- Extraire une PCFG d'un treebank
  - CFG = règles rencontrés dans les arbres du corpus
  - probabilités associées aux règles = estimées par fréquence relative (max de vraisemblance)

# Inconvénients des PCFG

- Hypothèses d'indépendance trop fortes
  - ne tiennent pas compte du lexique
  - il existe des dépendances structurelles entre les règles
    - Syntagme adjectival : jamais prénominal si contient complément postadjectival  
un [ très charmant ] garçon  
\*un [ très charmant envers tous ] garçon
- Plusieurs solutions proposées
  - algorithme lexicalisé
  - division de symbole/automatique