Cours de Traitement Automatique du Langage

Chapitre 05 : Analyse syntaxique

DI5 – 2021-22

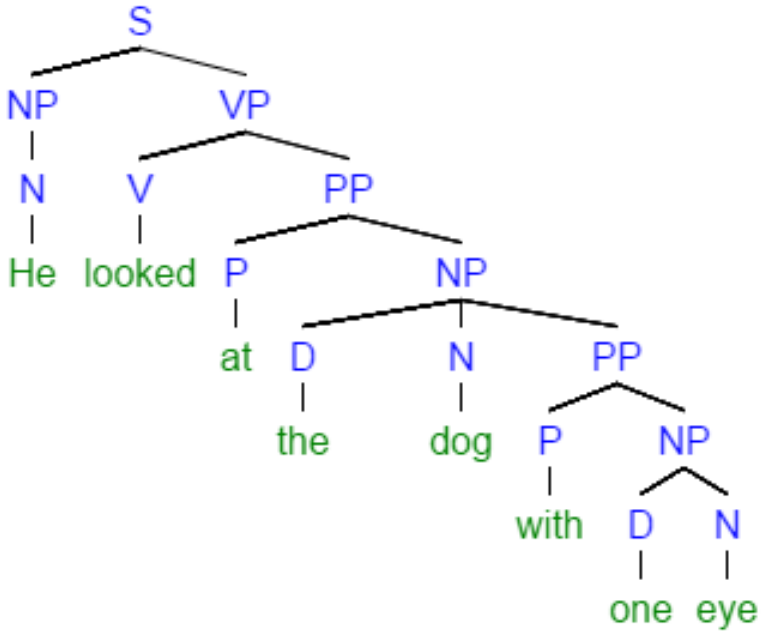
Analyse syntaxique

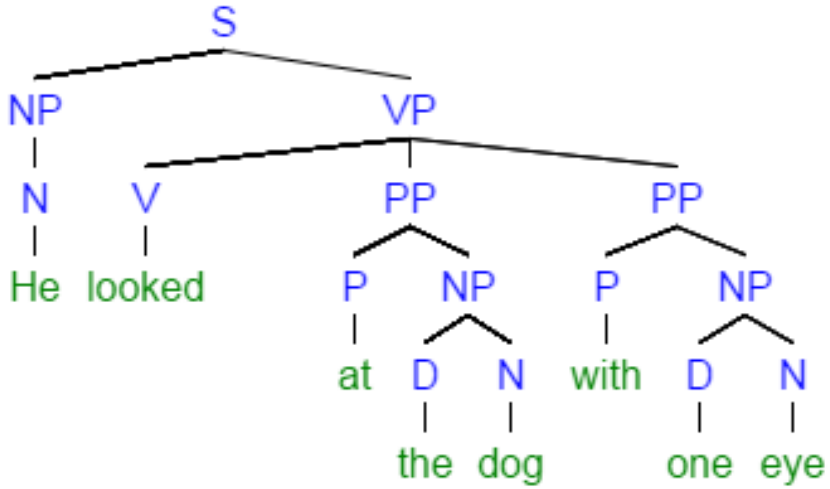
Objectifs

- Construire la structure syntaxique d'un énoncé donné → Arbre
- Les arbres d'analyse peuvent être utilisés dans des applications telles que la vérification grammaticale, l'analyse sémantique, les chatbots,…

Mais

- **Ambiguïté syntaxique** : analyse non déterministe





Analyse Syntaxique

2 types de langages…

Modélisation des règles de construction des énoncés d'un langage à partir des éléments du vocabulaire (**grammaire du langage)** plus ou moins difficile**…**

- **langages naturels**
- **langages artificiels** programmation informatique, mathématiques, logique, calcul symbolique, etc...

Jugement de grammaticalité

Enoncé appartenant ou non au langage.

- **programmation** vérification du programme par le compilateur
- **mathématiques** contrôle de parenthésage des expressions arithmétiques ou symboliques, correction d 'un raisonnement logique...
- **langage naturel** grammaticalité d'un énoncé difficile à circonscrire

  grammaire de la langue reflétant bien souvent une volonté, de normalisation (*Bescherelle*, *Grevisse* dans une moindre mesure).

Syntaxe et grammaticalité

- **Qu'est qu'un énoncé grammaticalement correct ?**

jugement normatif ← ? → usage fréquent attesté

- **Les agrammaticalités : indices d'évolution diachronique** [Gadet 1992]

Des exemples jugés non grammaticaux à un moment donné sont acceptés progressivement du fait de leur usage de plus en plus fréquent.

- **morpho-syntaxe**
  - *mouler le café* 👎
  - *s'assir* 👎
  - *becter* 👍 à partir de *becqueter*
  - *agoniser d'injures* 👎
  - *magne-toi* 👍 à partir de *manier*

QUIZZ !!

- **syntaxe** :
  - *un pote que j ai passé mon enfance avec lui* 👎
  - *une ville où il y fait bon vivre* 👎

Evolution contrainte : certains principes linguistiques semblent cependant toujours devoir être respectés

Syntaxe et grammaticalité

Les agrammaticalités : marqueurs sociaux

QUIZZ !!

- **syntaxe** : accord du COD. Règle de lettrés ?

ils se sont plus tout de suite	*ils se sont plu tout de suite*
les pommes que j'ai lavées	*les pommes que j'ai lavé*
les efforts que cela m'a coûtés	*les efforts que cela m'a coûté*
les raclées que je me suis prises	*les raclées que je me suis pris*

- **prosodie** :

  - accents anglais marqueurs sociaux

  - prononciations erronées : *gageure*

Les Grammaires…

Définition [Chomsky 1956]

Grammaire G : quadruplet $<V_t,V_n,S,P>$ où :

- V_t ensemble des **symboles terminaux** encore appelé **vocabulaire** (on parle alors des mots), représente un ensemble non vide de symboles.
- V_n ensemble des symboles non terminaux (ou **catégories syntaxiques**) représente un ensemble non vide de symboles tel que l'intersection entre V_t et V_n est vide.
- S **symbole initial** (ou **axiome**) est un symbole de Vn et sert de point de départ.
- P ensemble des **règles de production** de la forme $\alpha \rightarrow \beta$ avec α (tête) et β (corps) toute séquence de symboles construite sur $V = V_t \cup V_n$ (β pouvant être éventuellement vide). Ces productions sont encore appelées **règles de réécritures** car elles précisent que la séquence de symbole α peut être remplacée par la séquence de symboles β.

⇨ **s'applique aussi bien au traitement du langage naturel que des langages artificiels (compilateurs langage informatique)**

Les Grammaires…

Notations

- **Terminaux** chaîne de caractères minuscules

  exemples : *Paul, le, chien*
- **Non terminaux** chaîne de caractères entre chevron

  exemples : *<S>, <GN>, <GV>*

Langage généré par une grammaire

- **Définition :** L(G) langage généré par la grammaire G = ensemble des séquences de symboles obtenues en partant du symbole initial S et en appliquant les productions de G par réécritures successives jusqu'à ce que la chaîne résultante ne présente plus que des symboles terminaux.
- **Notation** L(G) ou L<S>

Les Grammaires …

Exemple de grammaire (utilisée par la suite)

V_t = {*aime,beaucoup,passionnément,un_peu,il,elle,m,t*}

V_n = { <S>,<SN>,<SV>,<GV>,<GADV>,<pronom>,<verbe>,<clitique>,<adv> }

P =

<S>	→ <SN><SV>	(règle 1)
<SN>	→ <pronom>	(règle 2)
<SV>	→ <GV>	(règle 3)
<SV>	→ <GV><GADV>	(règle 4)
<GV>	→ <verbe>	(règle 5)
<GV>	→ <clitique><verbe>	(règle 6)
<GADV>	→ <adv>	(règle 7)
<verbe>	→ *aime*	(règle 8)
<pronom>	→ *il* | *elle*	(règle 9)
<clitique>	→ *m* | *t*	(règle 10)
<adv>	→ *un_peu* | *beaucoup* | *passionnément*	

Langage généré par la grammaire (extrait) :

L(G) = {*il aime, elle aime, il aime beaucoup, il m aime beaucoup, elle m aime … }*

Hiérarchie de Chomsky

Grammaire d'ordre 0 aucune restriction

Ordre 1 : grammaire contextuelle (*context-sensitive grammar*)

γ <X> $\alpha \rightarrow \gamma \ \beta \ \alpha$

avec α, β et γ séquences (éventuellement nulle) de symboles (terminaux ou non) et <X> non terminal

contextuelle X se réécrit en β dans le contexte ($\gamma \ \alpha$)

Ordre 2 : grammaire hors contexte (CFG : *context free grammar*)

<X> $\rightarrow \beta$

avec <X> symbole non terminal et β séquence de symboles quelconques (terminaux ou non).

exemples <verbe> $\rightarrow$ *aimer* <GN> $\rightarrow$ <det><nom>

Ordre 3 : grammaire régulière (*regular grammar*)

<X> $\rightarrow \beta$ <Y>

avec <X> et <Y> symboles non terminaux unique (ou nul pour <Y>) et β symbole terminal unique.

exemples <verbe> $\rightarrow$ *aimer* <GV> $\rightarrow$ se <Vreflex>

Quel est le type de la grammaire exemple de la diapositive précédente ?

Réponse : Hors contexte (CFG)

Hierarchie de Chomsky

Forme normale de Chomsky

[Hopcroft et al. 2007]

- **Grammaire hors contexte** $<V_t,V_n,S,P>$ pour laquelle les règles sont exclusivement de la forme :

  $<A> \rightarrow <B><C>$ ou $<A> \rightarrow a$ avec $<A>,<B>,<C> \in V_n$, $a \in V_t$

- Toute CFG peut se réécrire sous forme normale de Chomsky

1. START : suppression de l'axiome S dans le corps des règles
2. TERM : suppression des terminaux dans les corps de règle de longueur ≥ 2

   exemple si `<A> → <B> a` on introduit `<Na> → a` et `<A> → <B><Na>`
3. BIN : suppression des corps de règle de plus de 2 sympboles

   exemple `<A>→<B><C><D>` remplacé par `<A>→ <B><CD>` et `<CD>→<C><D>`
4. DEL : suppression des règles avec membre droit vide (ε)
5. UNIT : suppression des règles unité de la forme $<X> \rightarrow <Y>$

   exemple `<A>→<B>` : `<A>` réécrit partout avec le corps des règles `<B> → ...`

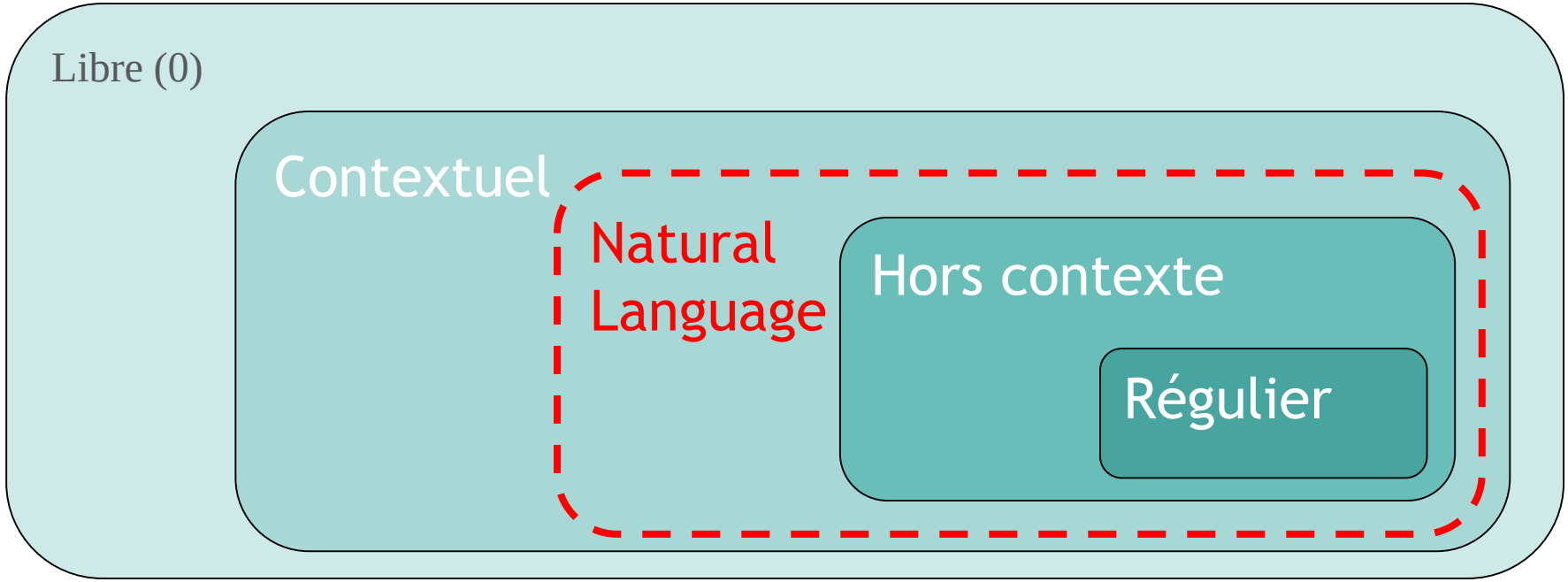
- Certains algorithmes d'analyse imposent que la grammaire soit mise sous forme normale de Chomsky.

Exemple algorithme CYK (Cocke-Younger-Kasami)

Hiérarchie de Chomsky

Hiérarchie de Chomsky et pouvoir de génération

Langages générés par une grammaire : **régulière < CFG < contextuelle < 0**



Les Grammaires…

Grammaire probabiliste

- G < Σ;N; P; S > est une grammaire.
- Les regles sont de la forme A → β *[p] avec* $A \in N$; $\beta \in (\Sigma \cup N)^*$
- p est la probabilite d'occurrence de la regle
- Cette probabilité est estimée à partir d'un corpus annoté
- Ces corpus appelés TreeBank font l'objet de nbreux projets

Estimation des probabilité des règles

$$P(A \rightarrow \beta | A) = \frac{C(A \rightarrow \beta)}{C(A)}$$

Σ est le vocabulaire : ensemble des symboles terminaux - Σ = {le, petit, chat, mange, un, poisson, ...}
N est l'ensemble des variables : symboles non terminaux N = {S, NP, VP, DET, N, ADJ, ..}
S ∈ N est l'axiome. - P est l'ensemble des regles de production.

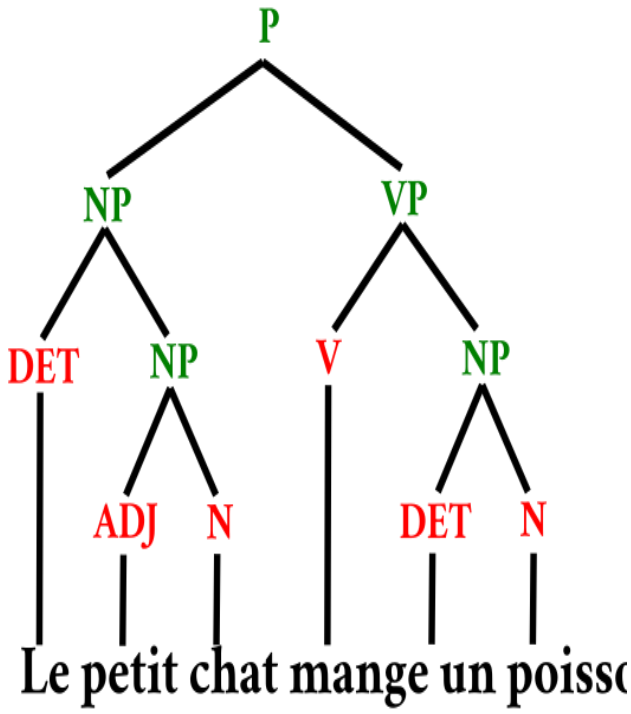
Les Parsers…

Parsers pour le TAL

- Les Grammaires de Langues sont produites manuellement
- Les analyses de phrases sont réalisées par des algorithmes spécifiques

- En sortie = Arbre binaire correspondant a une phrase
- En entrée, on a 1 grammaire + 1 phrase
  - Le petit chat mange un poisson
  - P→NP VP
  - NP→DET NP'
  - NP→DET N
  - NP'→ADJ N
  - VP→V NP

Rem : La deuxième règle n'ai pas écrite (NP→DET NP) sinon on peut avoir plusieurs déterminants pour un nom



Les Parsers…

Analyse

- Analyse top-down (mise en **pile des règles activables**) ou bottom-up (**shift-reduce**)
- Le plus connu → Algorithme bottom-up CKY (**Cocke-Kasami-Younger)**
- Construction de la structure de la séquence (ou rejet si elle n'appartient pas au langage) telle qu'établie indirectement au cours de **la dérivation**.

Arbre d'analyse (arbre syntaxique)

- Illustre comment l'axiome de la grammaire **se dérive** successivement en non-terminaux et terminaux pour former la phrase.

Principe de construction

- A chaque dérivation, on adjoint l'arbre élémentaire correspondant à la règle de production utilisée. Noeud initial : <S>

Exemples

→ **CF cours compilation**

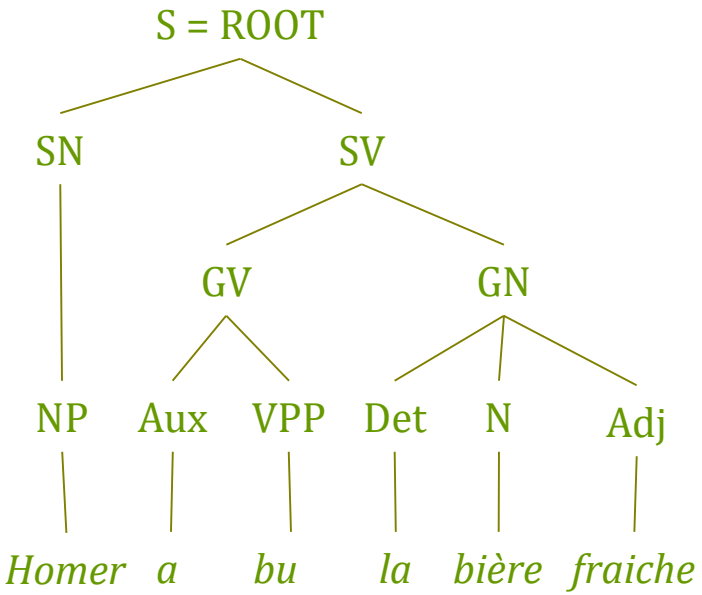
Analyse syntaxique - 2 approches

Grammaires de constituants

[Chomsky 1956]

Constituant (syntagme) – Groupe de mots formant une unité syntaxiques cohérente dont la structure interne est indépendante du reste de l'énoncé

Structure syntaxique – Enchâssement **récursif** des syntagmes

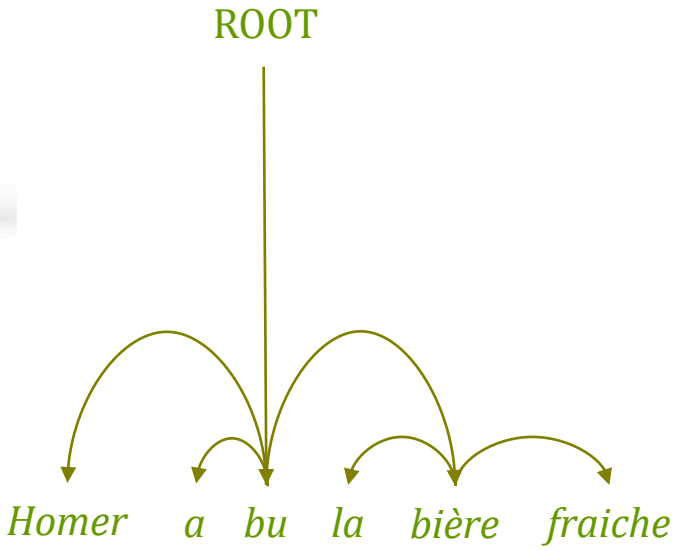


Grammaires de dépendances

[Tesnière 1959]

Gouvernance remplacé par la notion de **dépendances** orientées directement **entre les mots**

Structure syntaxique – Graphe de relations entre mots



Analyse syntaxique – 2 approches

Grammaires de constituants *(phrase-structure grammars)*

– Théorie des langages : adapté au traitement sur ordinateur

– Analogie grammaires scolaires en première approche

– Approche liée à l'origine à l'anglais et adapté aux langues à ordre fixe

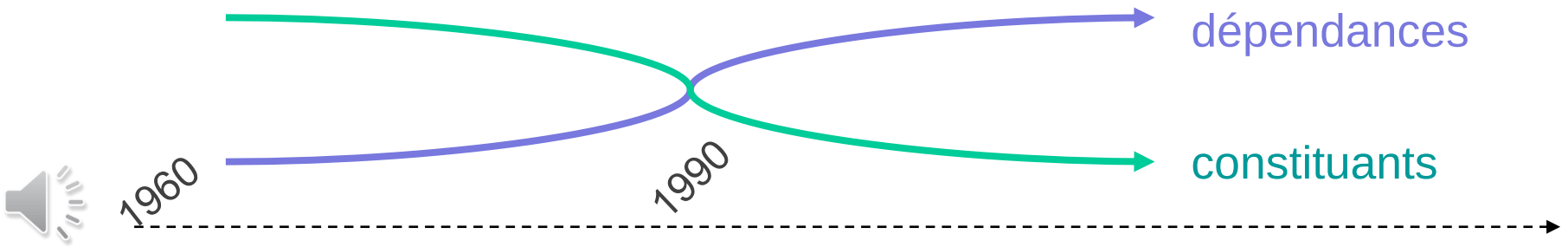
Grammaires de dépendances *(dependency grammars)*

– Visée linguistique : décrire les langues dans toute leur diversité

– Impact TAL initial limité : langues à ordre variable (russe, finnois …)

– Approche adaptée à l'apprentissage automatique : paires de mots

Influence en TALN



1/ Grammaire de constituants

- Une phrase est constituée de plusieurs syntagmes
- Un syntagme contient un noyau (head) qui est l'élément central
- Selon le noyau, le syntagme peut être : nominal (NP), adjectival (AP),verbal (VP) ou prépositionnel (PP)

- Le système formel utilisé pour modéliser la structure des constituants d'une phrase sont **les grammaires** :
  - régulière,
  - contextuelle,
  - à contexte libre,
  - probabilistes

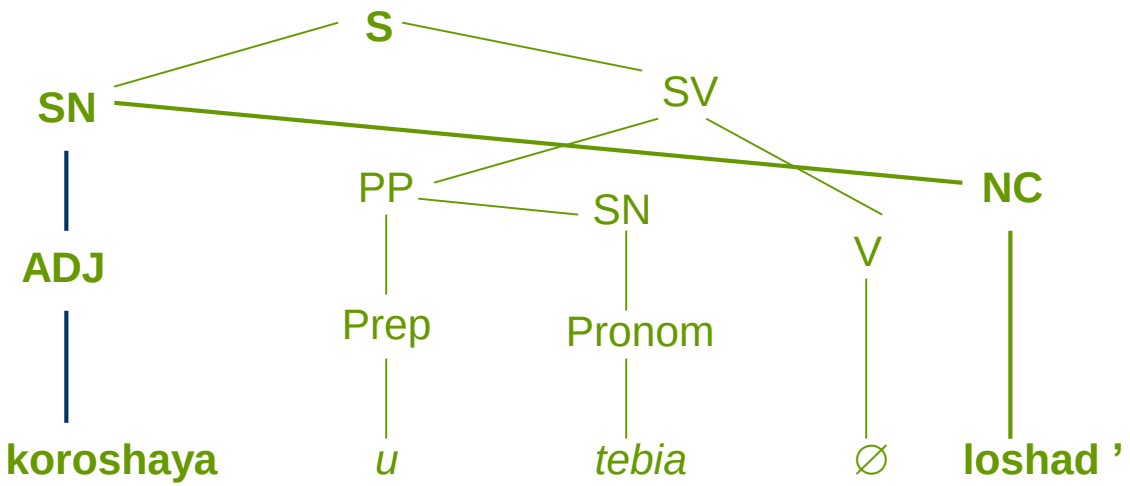
1/ Grammaires de constituants

Limitations

- syntagmes discontinus
- structure syntaxique **non projective** (croisement d'arcs)

Exemple : langues à ordre variable (russe, finnois, coréen, hongrois…)

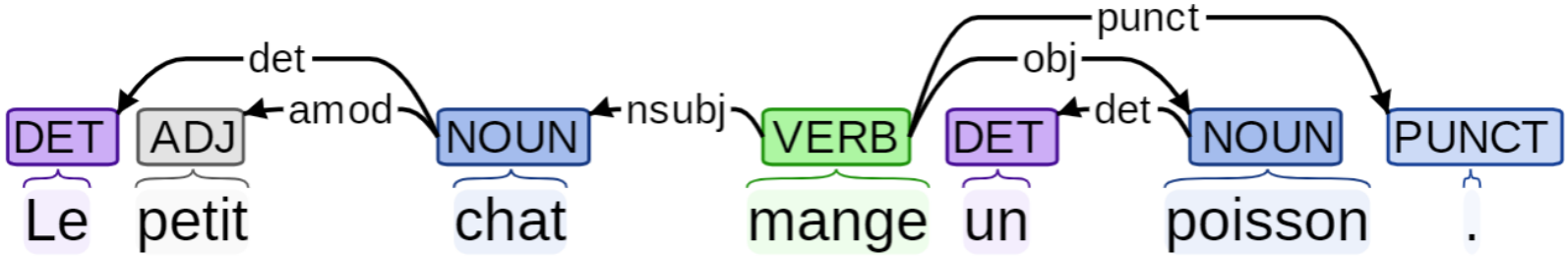
khoroshaya u tebia loshad '
bon avec toi cheval (*tu as un bon cheval*)



Alternative : grammaires de dépendances

2/ Grammaires de Dépendances

- Grammaire de dépendances: un ensemble des relations binaires entre les mots de la phrase
- La structure syntaxique est décrite en terme de mots (et pas des syntagmes)
- Les relations peuvent être : un sujet nominal (nsubj), un objet (obj), un modificateur d'adjectif (amod), déterminant (det), etc



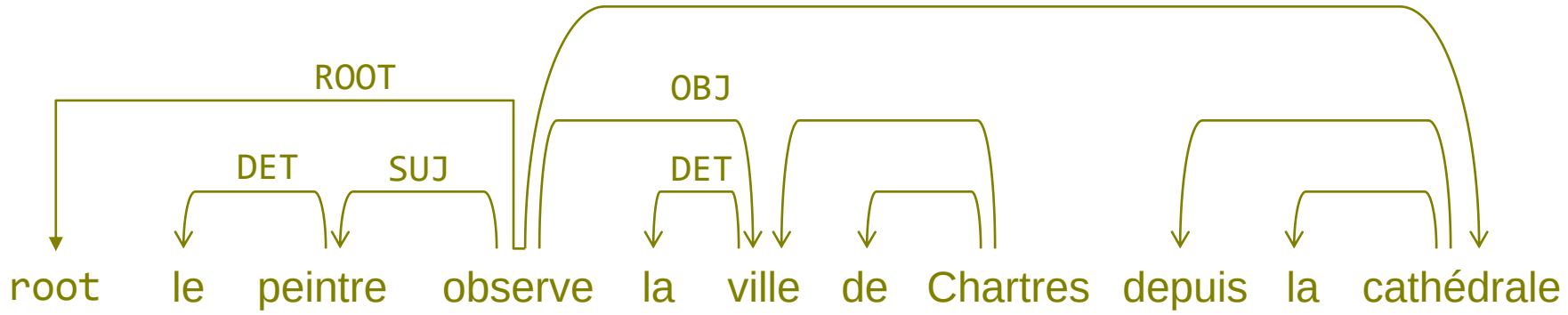
Un exemple de dépendances générée par `https://corenlp.run/`

2/ Grammaires de Dépendances

Représentation

- formalisme informatique pour une analyse en dépendances
- arbre ancré sur un nœud `root` pointant sur un prédicat dominant tout le stemma (verbe de la principale le plus souvent)
- relations étiquetées par des fonctions syntaxiques

exemple *le peintre observe la ville de Chartres depuis la cathédrale*

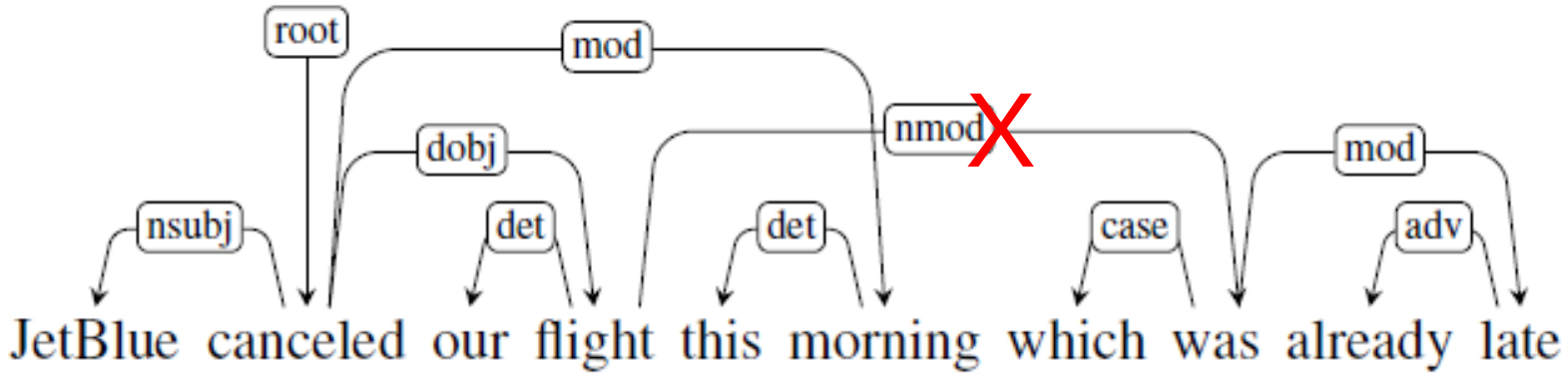


2/ Grammaires de Dépendances

Méthodes d'analyse

- Détection des « Head » (**tète lexicale**) → Le mot dans le syntagme qui est grammaticalement le plus important
- Les relations de dépendances correspondent a des **arcs typés dirigés et étiquetés** allant des **têtes** aux **dépendants**
- Mise en place de **Taxonomies** des relations → **Universal Dependencies – UD** (par des linguistes)
- **Classification** des relations entre chaque non-terminal et sa tète au travers d'**arbres de dépendances**

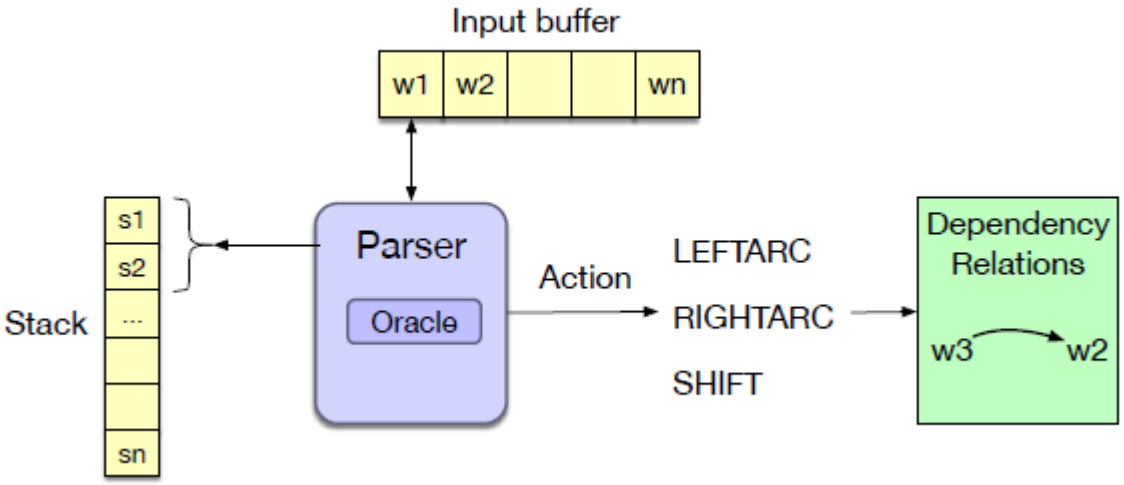
- Applicable que sur certain types d'arbres :

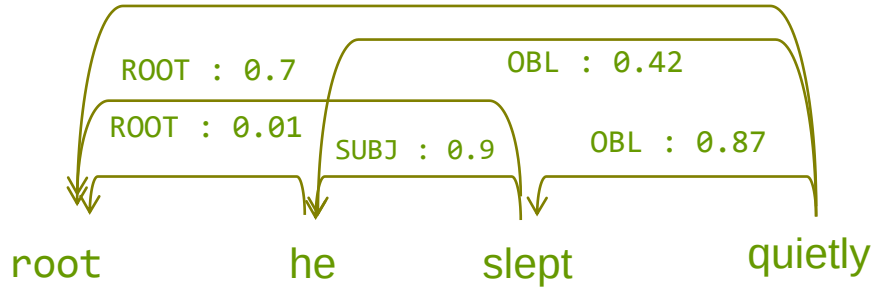


2/ Grammaires de dépendances

Classification des dépendances – 2 approches :

- Les méthodes basés sur les **transitions** emploient l'algorithme **« shift-reduce parsing »** basé sur une pile pour classifier les structures de dépendance.
- Les méthodes basées sur les **graphes** : **scoring des arcs** (proba de dépendence entre mots) couplés à un algo de **Maximum Spanning Tree** sur les scores
- Les 2 types de méthodes emploient des techniques **d'apprentissage supervisé**.
- Des **Treebanks de dépendances** fournissent les données nécessaires à l'apprentissage de ces systèmes





2/ Grammaires de dépendances

one example: Stanford's dependency parser [Dozat et al. 2017]

- **Pre-processing**
  - *Tokenizer + sentence segmentation* → 2-layers BiLSTM + 1-D CNN
    - Output classes : end of token, end of sentence, multi-word token, multi-word end of sentence, other
  - *Lemmatizer* →Hybrid system : dictionaries + neural network (to deal with rare, long words).
  - *Embedding* → *POS/UFeats Tagger* – BiLSTM on pre-trained word2vec,
- **Dependency classifier : word status**
  - **Input :** Word embeddings, lemma embeddings, character-level embeddings, summed XPOS & UFEAT embeddings
  - **Outputs** : 4 specialized vectors: word as a dependent, word as a head, word as a dependent deciding on its label; word as head deciding on the labels of its dependents
- **Dependency arc scoring**
  - Multi-layered perceptron + summation to find the best score
- **Graph based parser**
  - CLE algorithm on the dependency arc scores

Analyse sémantique : linguistique

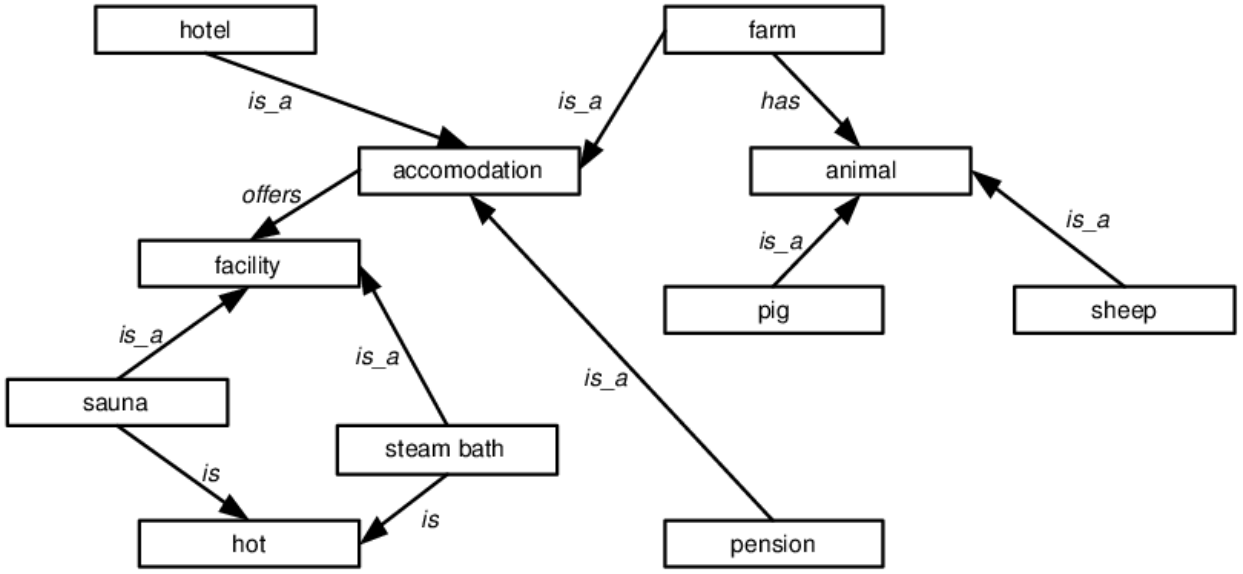
Donner un sens aux mots…

- Étude **hors-contexte** du sens des mots
  - Caractérisation de classes sémantiques à l'aide de traits (*sèmes*)

    chat, matou, minou : /animé/ + /animal/ + /félin/ + /domestique/

- Liens entre la significations des mots : réseau lexical

  - **Wordnet** *:* synonymie/antinomie, hyponymie/hyperonimie (*is_a*), partie_de (*has*)
  - **JeuxdeMots** : http://www.jeuxdemots.org/jdm-accueil.php



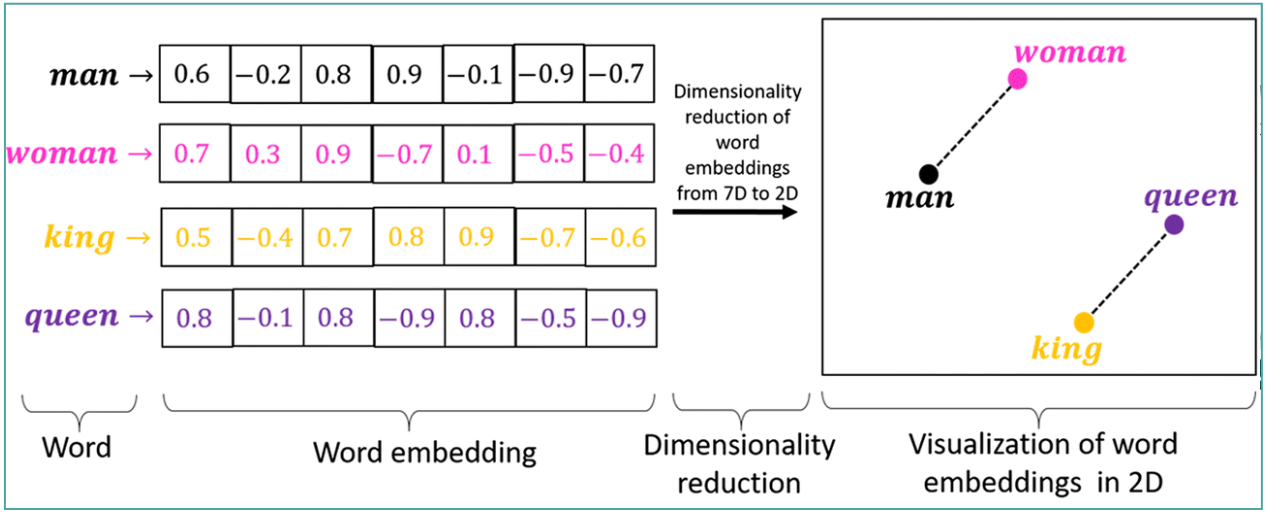
Sémantique lexicale

Réseaux lexicaux sémantiques

- Wordnet https://wordnet.princeton.edu/
- Conceptnet https://wordnet.princeton.edu/
- Wordnet https://wordnet.princeton.edu/

Sémantique distributionnelle : sens du mot décrit par un vecteur multi-dimensionnel construit par apprentissage automatique

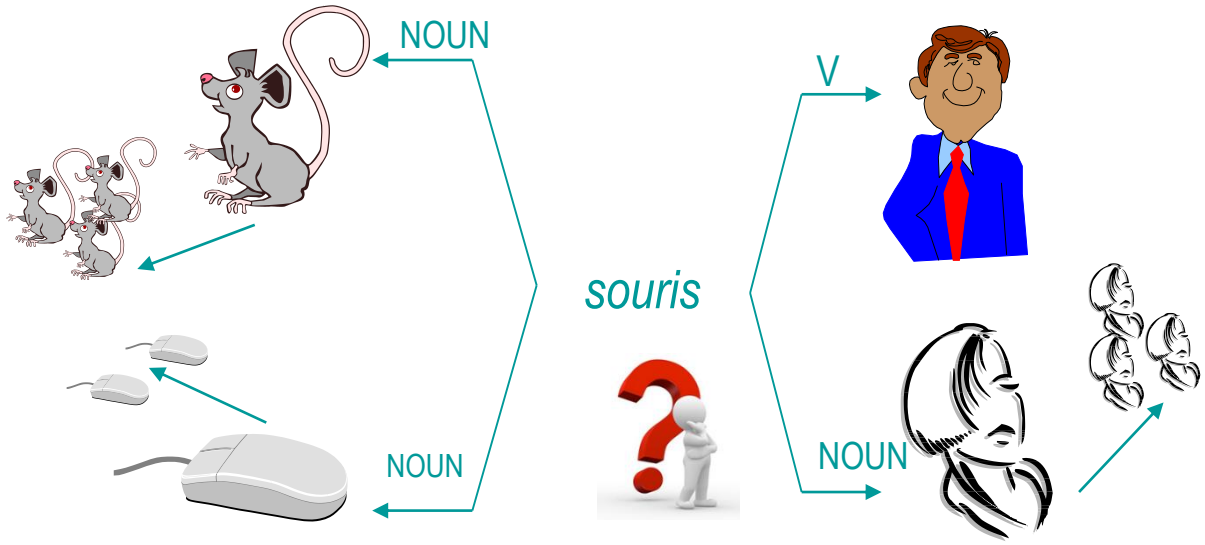
- LSA (Latent Semantic Analysis) https://wordnet.princeton.edu/
- Plongements de mots (*Word embeddings*) : Word2Vec



https://medium.com/@hari4om/word-embedding-d816f643140

Cas réels : ambiguïté sémantique

Difficulté : une même forme peut avoir plusieurs sens



Solutions

- Désambiguïsation sémantique en contexte
- Plongements de mots contextuels (réseaux récurrents, modèles d'attention)

→ ELMo https://allennlp.org/elmo

→ BERT [Devlin et al. 2019]

Analyse sémantique : linguistique

Sémantique de l'énoncé

Structure sémantique : relations sémantiques entre les éléments de l'énoncé : formalisme logique (λ-calcul)

Tous les restaurants ont une autorisation préfectorale

$\forall x$ restaurant(x) $\Rightarrow$ ($\exists y$ autorisation(y) $\wedge$ avoir(x,y))

Cas réels : ambiguïté structurelle et résolution des anaphores

Anaphore : reprise anaphorique = terme dont le sens dépend d'un antécédent

Exemple : anaphore pronominale — ***Jean*** *arrive.* ***Il*** *est en retard*

Ambiguïté de rattachement des anaphores

Jacques *tient* ***le sac*** *de* ***Bernadette****.* ***Elle*** *le regarde*

Analyse pragmatique

Pragmatique

Interprétation en contexte de discours : calcul de la référence

- **référence : contexte de la tâche** — déterminer l'objet de la tâche associé à un élément du discours

  *Pouvez-vous me sortir **le dossier de Monsieur Durand***

  *Je lui cherche un hôtel **au sud de la rocade*** → référence spatiale

- **co-référence :** plusieurs expressions réfèrent au même objet du discours

  ***Emmanuel Macron** a gagné l'élection présidentielle. **Le président élu** prendra ses fonctions dans une semaine. **Il** a déclaré… .*

  anaphore : l'anaphore ne donne pas nécessairement lieu à coréférence, mais il faut connaître un autre objet du discours pour résoudre la référence de l'expression considérée

  *Je préfère **l'Italie** à **l'Espagne**. **Sa cuisine** fait une plus grande part aux légumes frais surtout dans **le Nord*** ⇨ anaphores associatives

- **contexte de l 'univers du discours**(*world knowledge*)

  J 'ai réservé deux classes affaires sur le AF2031 d 'aujourd'hui ⇨ellipses

Analyse pragmatique

Pragmatique : analyse du dialogue

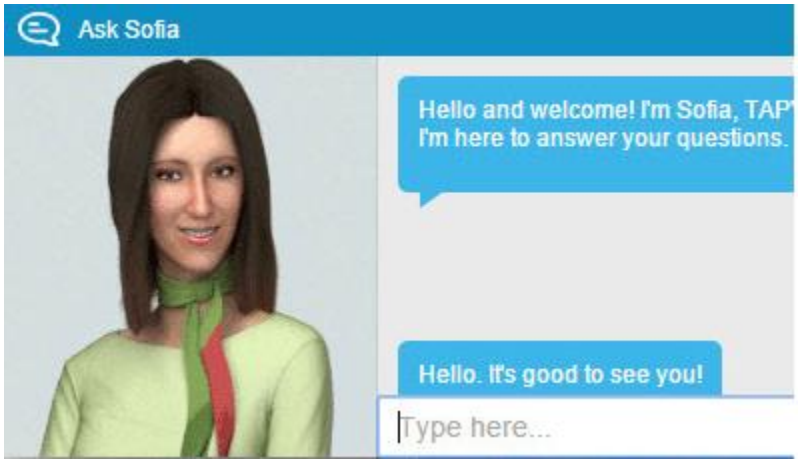
Actes de dialogue (*speech acts*) : intention de l'interlocuteur et rapport des énoncés langagiers au monde [Searle 1969/2009]

- assertifs (assertion, affirmation…) : les mots s'ajustent au monde
- directifs (ordre, demande, conseil…) : le monde s'ajuste aux mots
- promissifs (promesse, offre, invitation…) : le monde s'ajuste aux mots
- expressifs (félicitation, remerciement…) : pas de direction d'ajustement
- déclaratifs (nomination, baptême…) : direction d'ajustement double

Gestion du dialogue : grammaires de dialogue, gestion par plan (liens avec l'Intelligence Artificielle)

Exemples

- serveurs vocaux
- chatbots (agents conversationnels)



Tools : available parsers

French parsers (open source)

- **FRMG+DyALog** — TAG Grammar
  http://alpage.inria.fr/frmgwiki/wiki/frmg-une-grammaire-du-fran%C3%A7ais
  online : http://alpage.inria.fr/frmgwiki/frmg_main/frmg_server

- **BONSAI** — Automatically-induced PCFG, based on Berkeley [Candito & al. 2009]
  http://alpage.inria.fr/statgram/frdep/fr_stat_dep_bky.html

- **MALT Parser** — Statistical dependancy parser [Candito & al 2011]
  http://www.maltparser.org/mco/french_parser/fremalt.html

- **Stanford parsers** — PCFG or – French language model [Green & al. 2011]
  http://nlp.stanford.edu/software/lex-parser.shtml
  Neural Network Dependency parser [Chen & Manning 2014]
  https://nlp.stanford.edu/software/nndep.html

- **Stanza** — Python Library, including neural dependency parser [Qi & al. 2020]
  https://stanfordnlp.github.io/stanza/

Tools : Word Embedding

Plongements de mots

- **Word2Vec** https://radimrehurek.com/gensim/models/word2vec.html
  Non contextuels
- **FastText** https://fasttext.cc/docs/en/crawl-vectors.html
  Non contextuels – 157 langues dont français
- **ELMo** https://allennlp.org/elmo
  Contextuels – Pas de français dans les modèles pré-entrainés
- **BERT** https://github.com/google-research/bert
  Contextuels – Pas de français dans les modèles pré-entrainés
- **FlauBERT** https://huggingface.co/flaubert/flaubert_base_cased
  Contextuels – Français
- **CamemBERT** https://huggingface.co/camembert-base
  Contextuels – Français

Treebank

- **French Treebank** [Abeillé & al. 2003]
  www.llf.cnrs.fr/fr/Gens/Abeille/French-Treebank-fr.php

Bibliographie

Références & Ressources

- **Abeillé A., Toussenel F., Chéradame M.** (1999), dernière révision 2015) Corpus arboré pour le français (TFB) : annotation en constituants.
- **Marcus M., Marcinkiewics M.A., Santorini B**. (1993) Building a large annotated corpus of English : the Penn Treebank. *Computational Linguistics*, 19(2), 313-330.
- **Mc Donald R. and al.** (2005) Non projective dependency parsing using Span Tree Algorithms. Proc*. EMNLP'2005*.
- **Nivre J., Nilssson J.** (2005) Pseudo-projective dependency parsing. Proc. *ACL'2005*, Ann Arbor. Pp. 99-106.
- **Mel'cuk I.** (1988) Dependency syntax : theory and practice. Suny Press. Albany, NY.
- **Tesnière L.** (1959) Éléments de syntaxe structurale. Klincksiek, Paris.

- **Universal Dependencies** – http://universaldependencies.org/introduction.html