République Algérienne Démocratique & Populaire

Ministère de l'Enseignement Supérieur & de la Recherche Scientifique

Université Dr Moulay Tahar de Saida

Faculté de Technologie

Département d'informatique

Polycopié de cours

Ressources Lexicales (RLs) :

Cours et exercices

DR Hadj Ahmed BOUARARA

Année universitaire

2021-2022

Table des matières

Table de Figures

Table des Tableaux

[1] Pour plus d'informations sur leur téléchargement et leur utilisation, veuillez consulter le site Web de NLTK

Chapitre 0 :

Avant cours

« La vie c'est comme une bicyclette,

Il faut avancer pour ne pas perdre l'équilibre »

ALBERT Einstein

0.1. Bienvenue au cours du Ressources lexicales :

Je vous souhaite la bienvenue au cours Ressources Lexicales (RLs). Je m'appelle BOUARARA Hadj Ahmed, enseignant à l'université de DR Molay Tahar, SAIDA Algérie, et je serai votre hôte pour ce cours. Je suis heureux de vous avoir parmi nous et j'espère que vous apprécierez ce cours au fur et à mesure que vous commencerez votre voyage vers les Ressources Lexicales. Ce cours est destiné aux étudiant du master 2 MICR (modélisation informatique de la connaissance et du raisonnement).

0.2. Avant-propos :

L'application des technologies linguistiques à la construction et à l'extension automatiques de ressources lexicales s'est avérée fructueuse en ce qu'elle a fourni divers outils pour optimiser ce processus souvent prohibitif et coûteux. Les techniques de TAL fournissent des technologies de bout en bout qui peuvent relever tous les défis dans le pipeline de création et de maintenance des ressources linguistiques. Dans ce polycopié, nous résumerons les efforts existants dans cette direction, y compris l'extraction à partir du texte de phénomènes linguistiques tels que la terminologie, les définitions et les gloses, les exemples et les relations, ainsi que les techniques de regroupement pour les sens et les sujets. Nous résumerons en outre les travaux récents sur l'intégration automatique de connaissances issues de ressources hétérogènes telles que BabelNet, ConceptNet, wordnet ou ontology.

À la fin de ce cours, vous devez acquérir suffisamment de connaissances de base pour pouvoir approfondir vos compétences, que ce soit en tant que carrière de chercheur ou en appliquant les RLs dans d'autres domaines intéressants comme la médecine, la biologie, l'éducation..…etc

Hadj Ahmed BOUARARA

0.3. Objectif du cours RL :

Ce polycopié traite des ressources informatiques pour les données lexicales et de leurs utilisations. Premièrement, les types de données lexicales disponibles sont décrits, y compris ceux liés à la forme (orthographe, prononciation, flexion, classe de mots), au sens (définition/équivalent, synonymes/antonymes/hyperonymes, classification thésaurus), au contexte (collocations grammaticales, collocations lexicales, idiomes) et pragmatique (distribution, fréquence). Différentes formes sous lesquelles les données lexicales sont collectées sont examinées, notamment : les listes de fréquence des mots ; dictionnaires imprimés sous forme lisible par machine, avec et sans codes linguistiques et classification ; lexiques de machines; bases de données lexicales; et les lexiques de linguistique informatique du sens comme wordnet, sentiwordnet, framenet, et ontology.

On note aussi que l'ajout de connaissances explicites dans les systèmes du TAL est actuellement un défi important en raison des gains qui peuvent être obtenus dans de nombreuses applications. En même temps, traiter et stocker ces connaissances dans des ressources lexicales n'est pas une tâche simple. Dans ce contexte, la principale motivation de ce cours est de montrer comment le traitement automatique du langage naturel et les ressources lexicales ont interagi jusqu'à présent, et une vue sur des scénarios potentiels dans un avenir proche. Le polycopié est divisé en deux chemins principaux. Tout d'abord, nous nous penchons sur le TAL pour la création et l'enrichissement des RLs, où nous abordons une gamme de méthodes du TAL visant spécifiquement à améliorer les référentiels de connaissances linguistiquement exprimables. Deuxièmement, nous couvrons différents cas d'utilisation dans lesquels les ressources lexicales pour le TAL ont été exploitées avec succès y compris la recherche lexicale elle-même, la recherche spécifique au domaine, la lexicographie, la création d'aides à l'écriture (correcteurs orthographiques et de style), la traduction assistée par ordinateur, l'enseignement des langues, la recherche d'information, reconnaissance vocale, analyse de sentiment, similarité sémantique et l'intelligence artificielle.

0.4. Annotations

- **TALN** : traitement automatique du language natural
- **TAL** : traitement automatique des langues
- **NLTK** : natural language toolkit
- **RLs** : ressources lexicales
- **TA** : traduction automatique
- **RDF** : ressources descreption framework
- **RDFs** : ressource descreption framework schema
- **Xml** : extensible markup language
- **DTD** : data type definition
- **OWL** : ontology web language

- **TEI** : text encoding initiative
- **LMF** : lexical markup framework
- **NLP** : natural language preprocessing
- **MRD** : machie readable dictionary
- **PPDB** : ParaPhrase data base (PPDB)
- **RI** : recherche d'information

0.5. Organisation du cours :

Ce cours est organisé comme suit :

- **Chapitre 1 : Introduction générale**

Cette section donne une vue générale sur l'objectif principale du cours et ça relation avec d'autres domaines comme le TAL, l'apprentissage automatique et l'intelligence artificielle d'une façon globale.

- **Chapitre 2 : les RLs morhplogiques et syntaxiques**

Cette section explique les différents RLs morphologiques et syntaxiques (text corpora, concordance, dictionaires, glossaire, prebank, PPDB….ect) , leurs avantages, utilisation et structures avec citations de leurs défis dans des applications réelles.

- **Chapitre 3 : l'encodage des RLs morphologiques et syntaxiques**

Dans ce chapitre nous allons voir les languages les plus importants utiliser pour le codage et la numérisation des RLs syntaxiques et morphologiques comme (XML, DTD, TEI5 et LMF).

- **Chapitre 4 : les RLs sémantiques**

Cette partie permet l'illustration des différents RLs utiliser pour lever l'ambiguité sémantiques comme wordnet, ontology, thesaurus, framenet, conceptnet…..ect

- **Chapitre 5 : l'encodage des RLs sémantiques**

Ce chapitre est consacré à l'explication du RDF/RDFs et OWL qui sont les languages principales dans la construction des RLs sémantiques.

- **Chapitre 6 : Applications des RLs**

Dans ce chapitre nous allons détaillés quelques applications connus qui utilisent les RLs dans leurs traitement comme la traduction automatique, correction d'orthographe, étiquetage morphosyntaxique, lemmatisation, stemming, désambiguisation, indexation, analyse de sentiment…..ect.

- **Chapitre 7 : les exercices d'application**

Les exercices avec solution sur les différents points et RLs vus dans les chapitres précédents.

- **Chapitre 8 : les travaux pratiques**

Pour permettre aux étudiants de mieux pratiqué l'utilisation des RLs un ensemble de labs (comme reconnaissance vocale, calcule de similarité sémantique, recherche multilingue ….ect) ont été préparer utilisant le language de programmation python.

0.6. Quelles sont les conditions pour terminer le cours ?

Le cours est conçu pour l'étude auto-rythmée d'environ 4-8 heures par semaine pendant 14 semaines, y compris des quiz / vérifications des connaissances, des tutoriels/ activités d'apprentissage pratiques, des devoirs, des tests pour vos aider à vous préparer aux évaluations finals et d'autres lectures.

Chapitre 1 :

Les différents types d'ambiguïtés

1.1. Introduction

Dans ce chapitre l'objectif est de voir le besoin des RLs et les différents problems qui nécessitent la presence des RLs. Pour cela nous allons d'abord conaitre quelques definitions clés du module.

- Définition du Lexique : c'est l'ensemble de tous les mots d'une langue ou d'un langage (plus généralement l'ensemble des unités lexicales) **[1]**.
- Lexicographie **:** L'objectif est de recenser les mots, les classer, les définir et les illustrer, par des exemples ou des expressions, pour rendre compte de l'ensemble de leurs significations. Nous pouvons distinguer deux types de langages : langage standard comme le français et l'anglais (qui a un grammaire, un dictionnaire, vocabulaire, il peut être formalisé et régulariser) et le langage vernaculaire comme « darija » (le contraire du précédent) **[2]**.
- La linguistique **:** La science qu'étudie une langue ou un langage et elle peut être divisée en trois branches : 1) Tous ce qui concerne le son, le son de la parole et les éléments sonore en générale dans un langage particulier. 2) Structure et morphologie du mot et de la phrase. 3) Concernant la signification des mots et des phrases et la conversation dans le contexte **[1]**.

1.2. **Ambiguïté linguistique VS imprécision :** L'ambiguïté est une idée ou une situation qui peut être comprise de plusieurs façons (plusieurs interprétations). L'ambiguïté est similaire à l'imprécision, sauf que l'ambiguïté fait référence à quelque chose ayant plusieurs significations possibles, tandis que l'imprécision renvoie à un manque général de claret **[3]**.

1.2.1. Phonétique et Phonologie : La phonétique a pour objectif d'Étudier les ondes sonores émises par les organes humain pour rassembler la matière 1er (Le sons de la parole et la Super-segmentation

de ces informations). Par contre la phonologie utilise la matière 1ère pour découvrir des motifs et formuler des règles **[1]**.

- Ambiguité phonologique : l'ambiguïté phonologique peut être décomposée en 4 parties :
- Ambiguïté de Prononciation (homophone): La prononciation d'un mot ou une expression est la manière de l'articuler et elle est liée à l'environnement et la société dans lequel on l'a appris. La prononciation d'un mot peut être écrite à travers la phonétique. L'Ambiguïté dans la prononciation provient lorsque deux mots différents se prononcent de la même manière ex : right, write et rite **[1]**.
- Tonalité : lorsque la tonalité de la prononciation d'un mot peut changer le sens du mot ex : le mot chinois 母 peut-être scold, hemp ou mother dépend du tonalité de la prononciation**[3]**.
- Réduplication : spécialement dans les langues afro-asiatique comme la langue indonésienne ex buke signifier un livre et buke buke signifier livre(s) **[1]**.
- Jointure : c'est lorsque lors du prononciation d'un mot on les colles ex : A Name / An aim , j'étouffais / j'ai tout fait **[1]**.

1.2.2. Morphologique : C'est étudier la formation et la forme du mot (unité lexicale) avec sa structure interne, son utilisation et l'objectif d'identifier les morphèmes et analysé leur sens. Composé de deux parties :

- Morph : signifier la forme
- Ologique : l'étude de quelque chose
- Morphème **:** La plus petite unité dans un langage porteur de sens ex : books a deux morphèmes book et s Unladylike a trois morphèmes un, lady, like **[1]**.

1.2.2.1. **Les relations lexico-sémantiques :** Dans un lexique on peut distinguer cinq différents types de relations lexico sémantique :

- Polysémie : correspond à la propriété que certaines unités lexicales auront plusieurs sens. Ex :
    - Ferme : 1- le verbe fermer conjuguer avec je ou il/elle **[4]**.
      2- la ferme.
    - Avocat fruit / avocat (métier).
    - Porte : verbe porter ou nom porte de maison**[1]**.
- Homonymie : C'est la relation entre plusieurs mots ayant le même signifiant graphique ou phonique mais des signifiés différents **[1]**. Les Homophones : les mots qui ont la même prononciation mais des orthographes différentes. Ex : 1- peau, pot. 2- Mail/male**[1].** Les homographes : les mots qu'ont les mêmes orthographes mais des prononciations différentes **[1]**. Ex :
    - Ils président/ un président
    - Ils violent la loi/ Un vert violent.

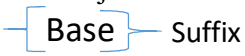
- Une vis/il vis.

- Hyperonymie / hyponymie : L'hyperonyme est un terme dont le sens inclut d'autres termes : ses hyponymes par exemple : insecte : mouche, abeille et fourmi **[4]**.
- Synonymie : La synonymie est un rapport de similarité sémantique entre des mots ou des expressions d'une même langue en d'autre terme, Des unités lexicales différentes désignant le même concept**[5].** Ex :
  - Aperçu, modèle, spécimen
  - Aimer, adorer.
- Antonymie : Relation entre les opposés où le positif d'un terme ne nécessite pas d'être le négatif de l'autre [5]. Antonymie gradué : Dans ce cas l'opposition est une question de degré et non absolue. Ce type est généralement associer aux adjectifs. Ex : cold, cool, warm, hot [5]. Non Gradable antonomy : Des opposés binaire comme : mort / vivant, Male / female ou Pass/ fall. Incompatible : Comme par exemple les couleurs Bleu vs noir sont incompatible [5].
- Méronymie : Décrit la relation de part of (partie de). Il y'a le type fonctionnel (détachable) Ex Nez et la bouche sont des meronyme Du visage et le visage est un holonym du nez et bouche. Par contre il y'a le type continu (non détachable) ex feu et flam [5].

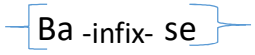
1.2.2.2. **Les opérations morphologiques :** Un ensemble de traitement text utilisés pour changer la forme morphologique du mot **[6]**.

- Abréviation **:** appelé aussi acronymes qui représente le nommage d'un mot ou expression généralement composée par la prise de ses premières lettres. Ex : ressources lexicale (RL) ou Radio detection and ranging (RADAR) **[6].**
- Probléme abbréviation : lors du processus de recherche d'information un problème sera posé parce que RADAR sera pris différemment que Radio detection and ranging. **[6]** [2]
- Les affix : Prefix : le processus d'ajouter un morphéme prefix au début d'une forme de base d'un mot **[1]**.

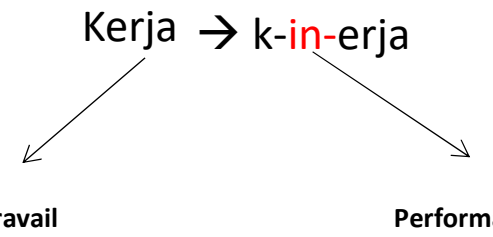
- Suffix : le processus d'ajouter un morphéme suffix à la fin d'une forme de base d'un mot **[1]**.



- Infix : processus d'ajouter un morphéme au milieu d'une forme de base d'un mot **[1]**.



vue pour les langues affroasiatique comme indonisie. Ex Kerja c'est travail



- Circumfix : Situation généralement pour la langue allmande Comme ge-leg-t (vers le bas

- Problème **:** dans les problèmes de recherche d'information on doit automatiquement calculer l'importance de chaque terme dans les textes en utilisant le terme frequency (TF). Malheureusement, les mots comme play et played sont les mêmes mais lors de l'indexation c'est deux termes vont être pris chacun indépendamment ce qui poser des problèmes et imprécision lors de calcule de similarité entre la requête et les documents [6]. La question qui se pose comment résoudre ce problème ?
- Suppletion **:** La suppléance, telle que définie par les linguistes, est l'utilisation d'une autre tige pour une forme infléchie qui n'est pas apparentée à sa forme de base. Si vous avez déjà eu une grammaire d'une langue mentionnez "verbe irrégulier (conjugaisons)", "adjectifs irréguliers", ou "nom irrégulier (déclinaisons)" alors vous avez probablement rencontré le phénomène. Ex : verb be (am, is, are, was, were, be) [3].
- Probléme **:** un grand probléme sera posé dans cette situation dans le processus de recherche d'information vu que am, is, was représentent le méme verbe be [5].
- Mot composée **:** C'est la combinaison de plusieurs formes de base ex : pomme-de-terre, text-mining…..ect [4].

1.2.2.3. Ambiguïté lexical et morphologique :

Se produit Lorsque la forme morphologique d'un mot a deux significations différente ou plus. Généralement cela se produira pour trois raison :

- Lexicosémantique : les mots polysémiques et hormophones.
- les mots composés ex : river edge, financial institution, a pile or mass of clouds
- lorsque un mot a deux ou plusieurs catégories grammaticales (ex : le mot ferme en français peut étre le verbe fermer ou le nom ferme, le mot porte N/V et le mot Avocate NM/Nf).

1.3. Ambiguïté Syntaxique :

Etudier la structure de la phrase ainsi que l'ordre et la combinaison des mots dans une phrase. Généralement, L'ambiguïté structurelle apparaît si deux structures syntaxiques différentes ou plus peuvent être affectées à une phrase **[8]**. ex :

1- There are a lot of man and women around**[2]**.
    - There are a lot of old men and women of any age around.
    - There are a lot of men and women who are all old.
2- We need more intelligent administrators**[2]**.
    - We need a larger number of intelligent administrator
    - We need more administrators who are intelligent.

En d'autres termes, un ordre des mots peut être associé à deux ou plusieurs significations différentes suivante **[11]**:

S→ NP VP

NP → det N / N / NP PP/ pro N/ adj N

PP→ P NP

VP → V NP/ V / Vcompo / VP PP

Compo → S

Où :

S : sentence

NP : phrase nominal

VP : phrase verbal

P : preposition

PP : phrase prepositionnel

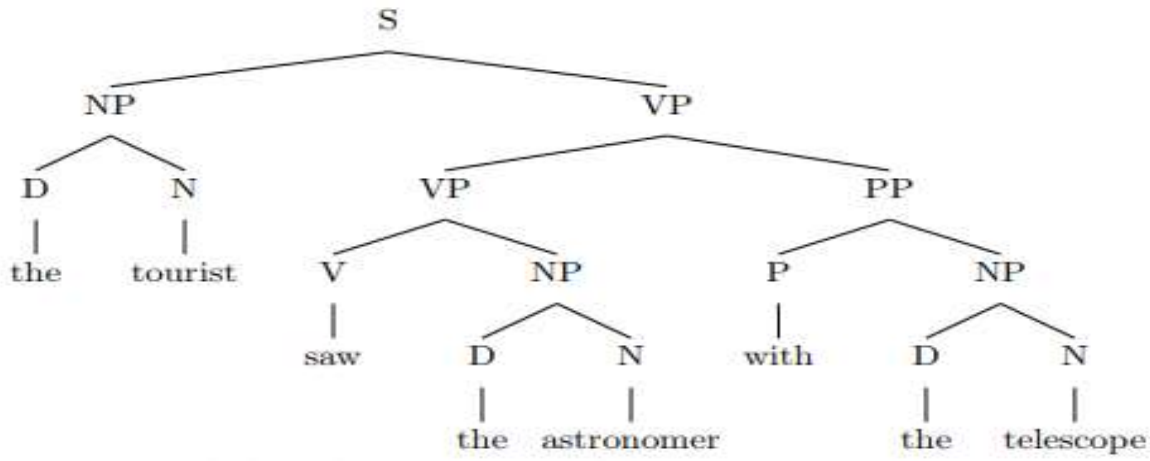
N : nom

V : verbe

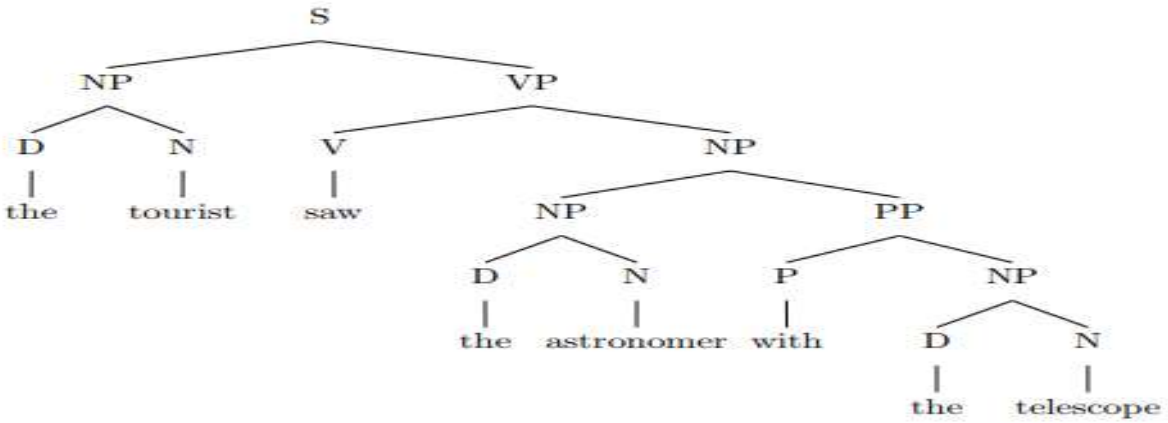
Pro : pronom

Ex : la phrase "The tourist saw the astronomer with the telescope" a deux interpretation [13].

1) Le touriste a vu l'astronome avec le télescope



2) The astronomer that the tourist saw had a telescope.



1.4. **Ambiguïté référentielle :** L'ambiguïté référentiel concerne la relation entre les objets et leurs expressions. Elle est présente lorsqu'un mot, dans le contexte d'une phrase particulière, peut se référer à deux ou plusieurs propriétés ou choses. Ex : « Le garçon a déclaré à son père le vol. **Il** était très en colère » **[10]**.

il: est ambiguë, car il peut se référer à la fois au garçon et au père.

1.5. **Ambiguïté du nombre :** Elle survient lorsqu'une phrase peut être comprise de différentes façons parce qu'elle contient deux quantificateurs ou plus **[15]**. Ex: Some man loves every woman.

On peut interpréter cette phrase qu'il y'a :

- Un homme qui aime toutes les femmes **[16]**.
- Chaque femme est aimée par au moins un homme **[17]**.

1.5.1. Ambiguïté sémantique : étude du sens d'un mot ou d'une phrase en analysant la signification du mot et la relation sémantique entre les mots afin de construire le sens global de la phrase parce que Le sens d'un discours est la somme des significations des mots individuels et de la manière dont ils sont organisés en une structure. Lorsque les mots d'une phrase lorsqu'ils sont pris individuellement n'ont pas une ambiguïté lexicale mais la relation entre eux peut donner deux sens à la phrase **[18]**.

1.6. Pragmatique

- l'étude de l'utilisation des significations linguistiques, des mots et des phrases, dans des situations réelles et dans un contexte interactionnel. Cela dépasse le sens littéral d'un énoncé (explicite) et se concentre sur les significations implicites. Par conséquent, sans la pragmatique, il y aurait très peu de compréhension de l'intention et du sens **[18]**. Prend en considération :
- la négociation de sens entre locuteur et auditeur.
- le contexte de l'énoncé.
- le sens potentiel implicite d'un énoncé.
- **Ambiguïté pragmatique :** Se produit si dans une conversation et dans un contexte spécifique un énoncé à une interprétation et dans un autre contexte à une autre interprétation. Généralement quand plusieurs situations de communication peuvent être possibles d'une phrase**[18]**.

1.7. Définition ressources lexicales : Pour lever l'ambiguïté et facilité les taches aux machines afin d'avoir un traitement facile et efficace du langage naturel, des ressources linguistiques sont utilisées.

- Déf 1 : les ressources lexicales rassemblent des connaissances sur les mots, leurs sens et leurs usages. Le but est d'aider la machine à comprendre et traiter automatiquement le langage naturel **[19]**.
- Déf2 : Réservoir d'entrées (mots, concepts) structurées sous support informatique, avec des informations associées **[19]**.

1.8. Types de données lexicales

Comme point de départ sur les ressources lexicales, nous allons jeter un rapide coup d'œil aux données lexicales impliquées. Les faits lexicaux suivants, bien connus de tous les lexicographes, reflètent différents aspects de l'information lexicale nécessaire pour décrire l'usage d'un mot. Les lemmes d'une RL pourraient être décrits selon les principaux aspects suivants :

Morphologique

- Orthographe
- Prononciation
- Inflexion
- classe de mots

Sens	- Définition / équivalence - Synonyme, antonyme, hyperonyme …ect - Thésaurus, classification
Contexte	- collocations grammaticales - collocations lexicales - expressions idiomatiques
Pragmatique	- Distribution (domaine, style, register) - Frequency

Chapitre 2 :

Les Ressources Lexicales morphologiques et syntaxiques

2.1. Introduction :

L'objectif de ce chapitre est de répondre aux questions suivantes :

- Comment extraire automatiquement les mots et phrases clés qui résument le style et le contenu d'un texte ?
- Quels outils et techniques le langage de programmation Python fournit-il pour un tel travail ?
- Quels sont les différents RLs morphologiques et syntaxiques, leurs avantages, utilisation et structures?
- Quels sont les défis intéressants des RLs morphologiques et syntaxiques dans le traitement du langage naturel ?

2.2. Corpora Textes bruts

Un texte brute représente une resource lexicale simple puisque nous pouvons en extraire de nombreuses informations :

- Liste de formes de mots
- Leur usage diachronique => dérive lexicale
- Cooccurrence de mots => sémantique lexicale
- Calculer les plongements de mots => synonymes, antonymes, analogies lexicales
- leurs relations => relations syntagmatiques, paradigmatiques
- leur combinaison => sémantique compositionnelle (phrase)
- découvrir des expressions à plusieurs mots
- les décomposer => informations morphologiques

Dans le domaine du TALN, en particulier TALN statistique, il est nécessaire de former le modèle ou l'algorithme avec beaucoup de données. À cette fin, les chercheurs ont rassemblé de nombreux corpus de textes. En règle générale, chaque corpus de texte est une collection de sources de textes. Il existe des dizaines de corpus de ce type pour une variété de tâches TALN. Alors que l'anglais a de nombreux corpus, d'autres langues naturelles ont aussi leurs propres corpus, mais pas aussi étendus que ceux de l'anglais. Les corpus les plus populaires sont détaillés par la suite :

2.2.1. Gutenberg Corpus :

Le corpus anglais du projet Gutenberg est composé de tous les livres électroniques en anglais disponibles dans la base de données Gutenberg en octobre 2014. Il contient 25 000 livres électroniques gratuits, hébergés sur http://www.gutenberg.org/. Ce projet est une initiative volontaire pour numériser et archiver des œuvres culturelles, afin d'encourager la création et la distribution de livres. Michael S. hart[2] a démarré ce projet en 1971 et c'est la plus vieille bibliothèque numérique à ce jour. La plupart des œuvres contenues dans cette collection sont les textes complets de livres dans le domaine public. Ce projet tente de rendre son contenu le plus accessible possible, sur du long terme, en format ouvert que n'importe quel ordinateur peut comprendre. Le 3 Octobre 2015, le Projet a atteint un total de 50.000 œuvres collectées. Ce corpus a été utiliser dans différente applications comme :

- l'analyse statistique du langage
- étudier la variabilité linguistique dans le temps
- la recherche d'information
- quantification du contenu de l'information
- détection du domaine du texte

L'exemple en python suivant montre les texts diponibles dans le corpus gutenberg dans les ressources de l'API NLTK.

```
>>> import nltk
>>> nltk.corpus.gutenberg.fileids()
['austen-emma.txt', 'austen-persuasion.txt', 'austen-sense.txt', 'bible-
kjv.txt','blake-poems.txt', 'bryant-stories.txt', 'burgess-busterbrown.txt',
'carroll-alice.txt', 'chesterton-ball.txt', 'chesterton-brown.txt',
'chesterton-thursday.txt', 'edgeworth-parents.txt', 'melville-
moby_dick.txt','milton-paradise.txt', 'shakespeare-caesar.txt', 'shakespeare-
hamlet.txt', 'shakespeare-macbeth.txt', 'whitman-leaves.txt']
```

2.2.2. Web and Chat Text

Le corpus de chat par exemple a les caractéristiques suivantes :

- Collectés pour la recherche sur la détection des prédateurs sur Internet.
- Contient plus de 10 000 messages.
- Organisé en 15 fichiers.
- Chaque fichier contient plusieurs centaines de messages collectés à une date donnée.
- Chaque fichier représente également une salle de discussion spécifique à l'âge (adolescents, 20 ans, 30 ans, 40 ans, plus une salle de discussion générique pour adultes).

[2] Michael Stern Hart est le créateur et l'animateur du projet Gutenberg, qui met sur l'Internet des livres libres de droit.

- Le nom du fichier contient la date, la salle de discussion et le nombre de messages.

```
>>> from nltk.corpus import webtext
>>> for fileid in webtext.fileids():
...     print(fileid, webtext.raw(fileid)[:65], '...')

firefox.txt Cookie Manager: "Don't allow sites that set removed cookies to se...
grail.txt SCENE 1: [wind] [clop clop clop] KING ARTHUR: Whoa there!  [clop...
overheard.txt White guy: So, do you have any plans for this evening? Asian girl...
pirates.txt PIRATES OF THE CARRIBEAN: DEAD MAN'S CHEST, by Ted Elliott & Terr...
singles.txt 25 SEXY MALE, seeks attrac older single lady, for discreet encoun...
wine.txt Lovely delicate, fragrant Rhone wine. Polished leather and strawb...
```

2.2.3. Brown Corpus

Le Brown Corpus a été le premier corpus général lisible par ordinateur, compose d'un ensemble de textes préparés pour la recherche linguistique sur l'anglais moderne. Ce corpus est composer d'un million de mots en anglais créé en 1961 à l'Université Brown. Il contient du texte provenant de 500 sources où les sources ont été classées par genre, news, editoriels…ect.

ID	Fichier	Genre	Description
A16	Ca16	News	Chicago Tribune: Society Reportage
B02	Cb02	Editorial	Christian Science Monitor: Editorials
C17	Cc17	Reviews	Time Magazine: Reviews
D12	Cd12	Religion	Underwood: Probing the Ethics of Realtors
E36	Ce36	Hobbies	Norling: Renting a Car in Europe
F25	Cf25	Lore	Boroff: Jewish Teenage Culture
G22	Cg22	Belle_lettres	Reiner: Coping with Runaway Technology
H15	Ch15	Government	US Office of Civil and Defence Mobilization: The Family Fallout Shelter
J17	Cj19	Learned	Mosteller: Probability with Statistical Applications

K04	Ck04	Fliction	W.E.B. Du Bois: Worlds of Color
L13	Cl13	Mystery	Hitchens: Footsteps in the Night
M01	Cm01	Science_Fiction	Heinlein: Stranger in a Strange Land
N14	Cn15	Adventure	Field: Rattlesnake Ridge
P12	Cp12	Romance	Callaghan: A Passion in Rome
R06	Cr06	Humor	Thurber: The Future, If Any, of Comedy

Tableau 1: Document pour chaque section du corpus brown

On peut accéder au corpus sous la forme d'une liste de mots, ou d'une liste de phrases (où chaque phrase n'est elle-même qu'une liste de mots). Nous pouvons éventuellement spécifier des catégories ou des fichiers particuliers à lire.

```
>>> from nltk.corpus import brown
>>> brown.categories()
['adventure', 'belles_lettres', 'editorial', 'fiction', 'government',
'hobbies','humor', 'learned', 'lore', 'mystery', 'news', 'religion',
'reviews', 'romance','science_fiction']
```

Ce corpus peut aider les chercheurs dans de nombreux applications :

- Calculer la similarité entre deux textes
- Détection de plagiat
- Etiquetage morphosyntaxique
- Classification supervisée
- Classification non supervisée

2.2.4. Reuters Corpus

Le corpus Reuters contient 10 788 documents d'information totalisant 1,3 million de mots. Les documents ont été classés en 90 thèmes, et regroupés en deux ensembles, appelés « apprentissage » et « test ». Contrairement au Brown Corpus, les catégories du corpus Reuters se chevauchent, simplement parce qu'un reportage couvre souvent plusieurs sujets. Nous pouvons demander les sujets couverts par un ou plusieurs documents, ou les documents inclus dans une ou plusieurs catégories. Pour plus de commodité, les méthodes de corpus acceptent un seul id de fichier ou une liste d'id de fichier.

2.2.5. Corpus d'adresses inaugural :

Des textes sont inclus ici comme « discours inauguraux » les discours prononcés par les présidents élus à la suite d'une cérémonie publique au cours de laquelle ils prêtent serment. Les présidents peuvent également avoir prononcé un discours important après leur entrée en fonction, mais nous ne les classons pas comme discours inauguraux. Cependant, le corpus est en réalité une collection de 55 textes, un pour chaque discours présidentiel. Une propriété intéressante de cette collection est sa dimension temporelle. Notez que l'année de chaque texte apparaît dans son nom de fichier. Ce corpus est utilizer pour détecter la similarité entre deux discours et aussi pour la detection des entités nommés. Il existe d'autre corpus comme le montre le tableau suivant

Corpus	Contents
CESS Treebanks	1M words, tagged and parsed (Catalan, Spanish)
Chat-80 Data Files	World Geographic Database
CoNLL 2000 Chunking Data	words, tagged and chunked
Gazetteer Lists	Lists of cities and countries
MacMorpho Corpu	1M words, tagged (Brazilian Portuguese)
Movie Reviews	movie reviews with sentiment polarity classification
Names Corpus	male and female names
Question Classification	questions, categorized
RTE Textual Entailment	sentence pairs, categorized
SEMCOR	words, part-of-speech and sense tagged
Stopwords Corpus	2,400 stopwords for 11 languages
Swadesh Corpus	comparative wordlists in 24 languages

Tableau 2 : différents corpus pour différente applications NLTK[3] **[24]**

2.3. Vocabulaire d'un text :

Le fait le plus évident concernant les textes qui ressort des exemples précédents est qu'ils diffèrent par le vocabulaire qu'ils utilisent. Dans cette section, nous verrons comment utiliser l'ordinateur pour compter les mots d'un texte de diverses manières utiles. La faite d'obtenir le vocabulaire et les statistiques d'un corpus peut aider plusieurs applications dans le doamine du TAL. Nous commençons par connaître la longueur d'un texte du début à la fin, en fonction des mots et des symboles de ponctuation qui apparaissent. Nous utilisons le terme **len** pour obtenir la longueur de quelque chose :

[3] Pour plus d'informations sur leur téléchargement et leur utilisation, veuillez consulter le site Web de NLTK

```
>>> len(text3)
```

Le vocabulaire d'un texte n'est que l'ensemble de tokens qu'il utilise, puisque dans un ensemble, tous les doublons sont regroupés. En Python, nous pouvons obtenir les éléments de vocabulaire d'un text avec la commande : set(text3). Lorsque vous faites cela, de nombreux écrans de mots survoleront.

```
>>> sorted(set(text3))
['!', "'", '(', ')', ',', ',)', '.', '.)', ':', ';', ';)', '?', '?)',
'A', 'Abel', 'Abelmizraim', 'Abidah', 'Abide', 'Abimael', 'Abimelech',
'Abr', 'Abrah', 'Abraham', 'Abram', 'Accad', 'Achbor', 'Adah', ...]
>>> len(set(text3))
2789
```

Calculons maintenant une mesure de la richesse lexicale du texte. L'exemple suivant nous montre que le nombre de mots distincts n'est que de 6% du nombre total de mots, ou de manière équivalente que chaque mot est utilisé 16 fois en moyenne ce qui peut nous aider dans l'application de detection d'auteurs.

```
>>> len(set(text3)) / len(text3)
0.06230453042623537
```

Ensuite, concentrons-nous sur des mots particuliers. Nous pouvons compter la fréquence à laquelle un mot apparaît dans un texte et calculer quel pourcentage du texte est occupé par un mot spécifique :

```
>>> text3.count("smote")
5
>>> 100 * text4.count('a') / len(text4)
1.4643016433938312
```

2.4. Concordance :

La concordance est l'outil le plus puissant avec une variété d'options de recherche. Il peut rechercher des mots, des phrases, des balises, des documents, des types de texte ou des structures de corpus et affiche les résultats en contexte sous la forme d'une concordance. Elle peut être triée, filtrée, comptée et traitée davantage pour obtenir le résultat souhaité. En dépit d'être un outil puissant, la concordance utilisée avec de grands corpus peut trouver tellement de résultats qu'il peut être fastidieux de les analyser et de les interpréter. Les options d'affichage permettent d'afficher des

informations supplémentaires telles que des lemmes. Des recherches complexes peuvent être appliquées avec des critères non spécifiques ou des critères facultatifs **[24]**.

2.4.1. Concordance en python :

Le NLTK fournit une fonction de concordance pour donner un contexte à un mot donné. Dans les trois exemples ci-dessous, nous allons montrer le contexte autour d'un terme populaire pour les critiques de films. En TAL, les utilisateurs souhaitent parfois rechercher des séries de phrases contenant un mot-clé particulier dans un passage ou une page Web **[24]**. Voici un exemple :

```
>>  text1.concordance("monstrous")
Displaying 11 of 11 matches:
ong the former , one was of a most monstrous size . ..
. This came towards us
ON OF THE PSALMS . " Touching that monstrous bulk of the whale or ork we have
ll over with a heathenish array of monstrous clubs and spears . Some were
d as you gazed , and wondered what monstrous cannibal and savage could ever
that has survived the flood ; most monstrous and most mountainous ! That H
they might scout at Moby Dick as a monstrous fable , or still worse and more
th of Radney .'" CHAPTER 55 Of the monstrous Pictures of Whales . I shall ere
ing Scenes . In connexion with the monstrous pictures of whales , I am str
ere to enter upon those still more monstrous stories of them which are to be
ght have been rummaged out of this monstrous cabinet there is no telling .
```

Cette RL est utiliser pour :

- Rechercher les phrases contenant un mot spécifique
- Trouver les différents contextes d'un mot
- Utilisé pour construire des corpus parelel
- Comment un mot est utiliser à travers l temps dans un texte
- Trouver les mots contextuellement similaire
- les parties d'un discours qui entourent un mot ou un passage et peuvent éclairer sa signification
- aide à développer la connaissance du vocabulaire, de la grammaire et du genre dans des contextes authentiques et significatifs

2.4.2. Concordance pour calculer la similarité contextuelle :

Une concordance nous permet de voir les mots dans leur contexte. Par exemple, nous avons vu que monstrueux se produisait dans des contextes tels que “les images” et “une taille”. Quels autres mots apparaissent dans une gamme similaire de contextes ? On peut le savoir en ajoutant le terme similaire au nom du texte en question, puis en insérant le mot pertinent entre parenthèses. Dans l’exemple suivant l’objectif est d’extraire les mots similaires au mot « monstrous ».

```
>>> text1.similar("monstrous")
mean part maddens doleful gamesome subtly uncommon careful untoward
exasperate loving passing mouldy christian few true mystifying
imperial modifies contemptible
>>> text2.similar("monstrous")
very heartily so exceedingly remarkably as vast a great amazingly
extremely good sweet
```

NB : vous pouvez observer clairement que nous obtenons des résultats différents pour différents textes.

La différence entre corpus et concordance :

la différence entre corpus et concordance est que corpus est le corps tandis que la concordance est l'accord, la conformité et la consonance [24].

2.5. Dictionnaire (machine readable dictionary (MRD)) :

Le dictionnaire lisible par machine est un dictionnaire stocké sous forme de données machine (ordinateur) au lieu d'être imprimé sur papier. Il s'agit d'un dictionnaire électronique et d'une base de données lexicale. En raison de la tyrannie de l'ordre alphabétique, les éditeurs de dictionnaires en profitent souvent pour répartir les différents types d'informations lexicales dans des dictionnaires plus petits, spécialisés dans un ou deux aspects lexicaux : dictionnaires d'orthographe, de prononciation, de définition, synonymes, idiomes, etc. cette RL peut étre utilizer pour :

- Lemmatization et steming
- Indexation
- Améliorer les moteurs de recherché
- Régler le probléme des mots compose
- Étiquetage morphosyntaxique
- Traduction automatique
- Correction d’orthographe

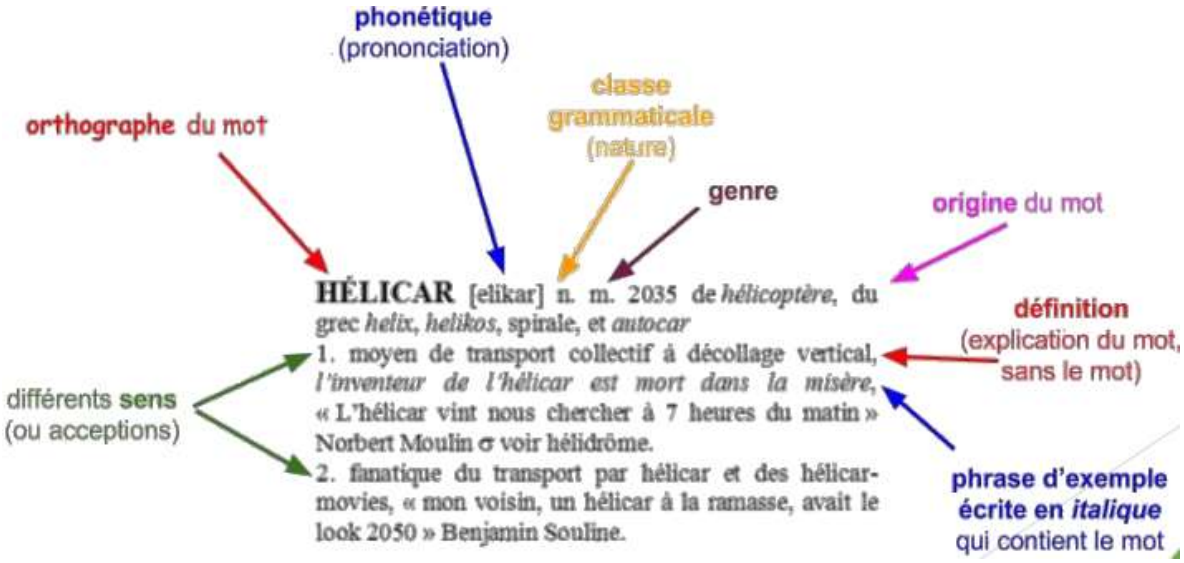


Figure 1: les différentes informations disponibles dans un dictionnaire [24]

2.5.1. Les types de dictionnaires :

- **dictionnaire morphologique :** Dans les domaines de la linguistique informatique et de la linguistique appliquée, un dictionnaire morphologique est une ressource linguistique qui contient des correspondances entre la forme de surface et les formes lexicales des mots. Les formes de surface des mots sont celles que l'on trouve dans le texte en langage naturel. La forme lexicale correspondante d'une forme de surface est le lemme suivi d'informations grammaticales (par exemple la partie du discours, le genre et le nombre). En anglais, donner, donner, donner, donner et donner sont des formes superficielles du verbe donner. La forme lexicale serait "donner", verbe. Il existe deux types de dictionnaires morphologiques : les dictionnaires alignés sur les morphèmes et les dictionnaires complets (non alignés) **[24]**.
- **Un dictionnaire de prononciation :** Un type de ressource lexicale légèrement plus riche est un tableau, contenant un mot plus quelques propriétés dans chaque ligne. Le dictionnaire de prononciation a été conçu pour être utilisé par les synthétiseurs vocaux **[24]**. Pour chaque mot, ce lexique fournit une liste de codes phonétiques - des étiquettes distinctes pour chaque son contrasté - appelés « phones » comme le montre le code suivant :

```
>>> entries = nltk.corpus.cmudict.entries()
>>> len(entries)
133737
>>> for entry in entries[42371:42379]:
...     print(entry)
...
('fir', ['F', 'ER1'])
('fire', ['F', 'AY1', 'ER0'])
('fire', ['F', 'AY1', 'R'])
('firearm', ['F', 'AY1', 'ER0', 'AA2', 'R', 'M'])
('firearm', ['F', 'AY1', 'R', 'AA2', 'R', 'M'])
('firearms', ['F', 'AY1', 'ER0', 'AA2', 'R', 'M', 'Z'])
('firearms', ['F', 'AY1', 'R', 'AA2', 'R', 'M', 'Z'])
('fireball', ['F', 'AY1', 'ER0', 'B', 'AO2', 'L'])
```

- **Le MRD de synonymes:** un dictionnaire composé de bases de données lexicales qui regroupent les mots avec leurs synonymes **[24]**.
- **Le MRD de definition :** Un dictionaire qui donne des significations simples du mot en maintient une structure organisationnelle riche en vocabulaire.
- **Le MRD spécialisé conçu pour un domaine spécifique comme la médcine ou l'informatique**
- **Dictionaire bilingue :** Un dictionnaire bilingue donne des mots en deux langues. Chaque langue est regroupée par ordre alphabétique, avec des traductions dans l'autre langue **[24]**.
- **Dictinaire multilingue :** les mots sont stockés dans une langue et leurs significations sont stockées dans plusieurs langues **[24].**

2.5.2. Historique du MRD dictionaries :

Les premiers MRD largement diffusés étaient le Merriam-Webster Seventh Collegiate (W7) et le Merriam-Webster New Pocket Dictionary (MPD). À l'origine, chacun était distribué sur plusieurs bobines de bande magnétique sous forme d'images de carte avec chaque mot séparé de chaque définition sur une carte perforée séparée avec de nombreux codes spéciaux indiquant les détails de son utilisation dans le dictionnaire imprimé. Olney a esquissé un grand plan pour l'analyse des définitions dans le dictionnaire, mais son projet a expiré avant que l'analyse puisse être effectuée. Robert Amsler de l'Université du Texas à Austin a repris l'analyse et a terminé une description taxonomique du dictionnaire de poche grâce au financement de la National Science Foundation, mais son projet a expiré avant que les

données taxonomiques puissent être distribuées. Roy Byrd et al. à IBM Yorktown Heights a repris l'analyse du Webster's Seventh Collegiate à la suite des travaux d'Amsler. Enfin, dans les années 1980, en commençant par le soutien initial de Bellcore et plus tard financé par diverses agences fédérales américaines, dont NSF, ARDA, DARPA, DTO et REFLEX, George Armitage Miller et Christiane Fellbaum de l'Université de Princeton ont achevé la création et la large diffusion d'un dictionnaire et sa taxonomie dans le projet WordNet, qui est aujourd'hui la ressource de lexicologie informatique la plus largement distribuée **[25]**.

2.6. Glossaire

Un glossaire est un regroupement de termes et de leurs significations. Cette définition peut également être étendue pour dire qu'un regroupement de termes énoncés dans un glossaire donné se retrouve ou a un lien avec un sujet, un texte ou un dialecte précis. Une autre façon de définir un glossaire se trouve sous la forme d'un dictionnaire concis, classé en ordre alphabétique pour favoriser la consultation rapide. Un glossaire est une liste de mots ou de phrases utilisés dans un domaine particulier avec leurs définitions. Il peut étre utilizer dans le probléme de l'abreviation pour l'indexation ou dans les moteurs de recherché pour améliorer la pertinence **[24]**.

2.7. Probank :

La Banque de propositions appelée PropBank adopte une approche pratique de la représentation syntaxique, en ajoutant une couche d'informations prédicat-argument, ou étiquettes de rôle sémantique, aux structures syntaxiques des phrases de la PennTreebank[4]. En raison de la difficulté de définir un ensemble universel de rôles thématiques, les rôles sémantiques dans PropBank sont définis par rapport à un sens verbal individuel. La ressource résultante couvre chaque instance de chaque verbe du corpus et permet d'établir des statistiques représentatives calculé. Elle fournit un lexique qui divise chaque mot en sens affinés connus sous le nom de « frameset », décrit les rôles d'argument qui peuvent être utilisés avec chaque frameset et fournit des exemples d'utilisation dans une variété de contextes syntaxiques **[47]**.

Chaque sens de chaque verbe a donc un ensemble spécifique de rôles, auxquels ne sont attribués que des nombres plutôt que des noms : Arg0, Arg1, Arg2, etc. En général, Arg0 représente le PROTO-AGENT et Arg1, le PROTO-PATIENT. La sémantique des autres rôles est moins cohérente, étant souvent définie spécifiquement pour chaque verbe. Néanmoins, il y a quelques généralisations ; l'Arg2 est souvent l'état bénéfique, l'instrument, l'attribut ou l'état final, l'Arg3 le point de départ, l'instrument bénéfique ou l'attribut, et l'Arg4 le point final. De telles entrées PropBank sont appelées fichiers de frames ; notez que les définitions dans le fichier frame pour chaque rôle sont des gloses informelles destinées à être lues par des humains, plutôt que d'être des définitions formelles **[47]**.

Exemple détaillé :

agree.01

- **Arg0:** Agreer

[4] https://catalog.ldc.upenn.edu/LDC99T42

- **Arg1:** Proposition
- **Arg2:** Other entity agreeing

Ex1: [Arg0 The group] agreed [Arg1 it wouldn't make an offer].

Ex2: [ArgM-TMP Usually] [Arg0 John] agrees [Arg2 with Mary] [Arg1 on everything].

Fall.01

- **Arg1:** Logical subject, patient, thing falling
- **Arg2:** Extent, amount fallen
- **Arg3:** start point
- **Arg4:** end point, end state of arg1

Ex1: [Arg1 Sales] fell [Arg4 to $25 million] [Arg3 from $27 million].

Ex2: [Arg1 The average junk bond] fell [Arg2 by 4.2%].

Notez qu'il n'y a pas de rôle Arg0 pour fall, car le sujet normal de fall est un PROTO-PATIENT. Les rôles sémantiques de PropBank peuvent être utiles pour récupérer des informations sémantiques superficielles sur les arguments verbaux. Considérons le verb increase **[47]**.

increase.01 "go up incrementally"

- **Arg0:** causer of increase
- **Arg1:** thing increasing
- **Arg2:** amount increased by, EXT, or MNR
- **Arg3:** start point
- **Arg4:** end point

Un étiquetage de rôle sémantique PropBank nous permettrait de déduire la similitude dans les structures d'événements des trois exemples suivants, c'est-à-dire que dans chaque cas Big Fruit Co. est l'AGENT et le prix des bananes est le THÈME, malgré les différentes formes de surface

- [$_{Arg0}$ Big Fruit Co.] increased [$_{Arg1}$ the price of bananas].
- [$_{Arg1}$ The price of bananas] was increased again [$_{Arg0}$ by Big Fruit Co.]
- [$_{Arg1}$ The price of bananas] increased [$_{Arg2}$ 5%].

Alors que PropBank se concentre sur les verbes, un projet connexe, NomBank (Meyers et al., 2004) ajoute des annotations aux prédicats nominaux. Par exemple, l'accord de nom dans l'accord d'Apple avec IBM serait étiqueté

avec Apple comme Arg0 et IBM comme Arg2. Cela permet aux étiqueteurs de rôles sémantiques d'attribuer des étiquettes aux arguments des prédicats verbaux et nominaux **[47]**.

2.7.1. La différence entre framenet et propbank :

PropBank diffère de FrameNet, la ressource à laquelle il est le plus souvent comparé, de plusieurs manières. PropBank est une ressource orientée verbe, tandis que FrameNet est centré sur la notion plus abstraite de frames, qui généralise les descriptions à travers des verbes similaires (par exemple "décrire" et "caractériser") ainsi que des noms et d'autres mots (par exemple "description"). PropBank n'annote pas d'événements ou d'états de choses décrits à l'aide de noms. PropBank s'engage à annoter tous les verbes d'un corpus, tandis que le projet FrameNet choisit des ensembles de phrases d'exemple à partir d'un large corpus et, dans quelques cas seulement, a annoté des portions de texte continues plus longues. Les annotations de style PropBank restent souvent proches du niveau syntaxique, tandis que les annotations de style FrameNet sont parfois plus sémantiquement motivées. Dès le départ, PropBank a été développé dans l'idée de servir de données d'entraînement pour les systèmes d'étiquetage de rôles sémantiques basés sur l'apprentissage automatique. Cela exige que tous les arguments d'un verbe soient des constituants syntaxiques et que les différents sens d'un mot ne soient distingués que si les différences portent sur les arguments. En raison de ces différences, l'étiquetage sémantique des rôles par rapport à PropBank est souvent une tâche un peu plus facile que la production d'annotations de style FrameNet [47].

2.7.1. Mise à jour du propbank :

Ce projet a été mise à jour dans le but de créer des PropBanks parallèles (le Treebank/PropBank anglais-chinois) et également des PropBank de d'autres genres, tels que Broadcast News, Broadcast Conversation et WebText ..ect. Il a été aussi mappé sur VerbNet et FrameNet dans le cadre de SemLink [47].

2.8. PRAPHRASE DATA BASE (PPDB) :

Les paraphrases, c'est-à-dire différentes réalisations textuelles du même sens, se sont avérées utiles pour une grande variété d'applications de traitement du langage naturel. Cette RL contient plus de 220 millions de paires de paraphrases[5], comprenant 73 millions de paraphrases phrastiques et 8 millions de paraphrases lexicales, ainsi que 140 millions de modèles de paraphrases, qui capturent de nombreuses transformations syntaxiques en préservant le sens. Les paraphrases sont extraites de corpus parallèles bilingues totalisant plus de 100 millions de paires de phrases et plus de 2 milliards de mots anglais. Chaque paire de paraphrases dans PPDB contient un ensemble de scores associés, y compris des probabilités de paraphrases dérivées des données bitextes et une variété de scores de similarité distributionnelle monolingue calculés à partir des n-grammes de Google.

[5] http://paraphrase.org.

Chapitre 3 :

L'encodage des ressources lexicales morphologiques et syntaxiques

---

3. Introduction :

Ce chapitre définit un module d'encodage de ressources lexicales morphologiques et syntaxique, en particulier des dictionnaires, glossaires monolingues et multilingues.

3.1. Definition du XML :

Extensible markup language est un language Standarizer par le W3C en 1998. Il a comme version le XML 1.0 et le XML 1.1 mais généralement la version 1.0 est la plus utilisé. Il permet le stockage l'echange, la description et la struction des données. Dans ce module le XML est utiliser pour la description et la construction des ressources lexicales syntaxique

3.1.1. La difference entre XML et base de données (BDD) :

- Le XML présente une vitesse de lecture très rapide, si vous devez accéder à une base de données, il faut d'abord exécuter le script PHP, ouvrir la base de données, rechercher dans la table, ça prend du temps, alors qu'avec un fichier XML vous le lisez très rapidement avec le langage PHP, et vous avez accès directement à l'information,
- La connexion avec une base de données différent d'un language à un autre tout dépend de l'environnement de développement utiliser.
- Le xml est Compatible avec plusieurs langages de programmation.
- Fonctionne avec la majorité des systémes d'exploitation.
- La facilité de convertir un document xml au format pdf html excel…ect
- Le xml permet de copier et sauvgarder facilement
- Facile à comprendre y'a pas de modèle de conception ni notion d'évènement ni contrainte.

- Le XML veut également compatible avec le web afin que les échanges de données puissent se faire facilement à travers le réseau Internet.
- Le XML se veut donc standardisé, simple, mais surtout extensible et configurable afin que n'importe quel type de données puisse être décrit.

3.1.2. Les caractéristiques du XML :

- La declaration : un document XML doit toujours commencer par un prologue :

  <?xml version="1.0" encoding="UTF-8"? Standalone= yes >

- **Les balises et les attributs (corps du document) :** Le corps du document XML constitué de l'ensemble des balises qui décrivent les données. Bien sur les noms des balises sont générique et non pas prédéfinie comme le HTML. Nous avons aussi la possibilité d'ajouter les attributs qui représente une information cachée supplémentaire.
- **Un document XML bien formé :** on dit qu'un document est bien formé si il respecte les critéres suivants :
  - Respecter les régles de nommage des balises et des attributs
  - La présence d'une balise racine
  - La présence de la déclaration
  - Chaque balise ouvrante a une balise fermante et la hiérarchie des balises est respectée où il y'a pas de chevauchement.
- **Un document xml valide :** un documentest valide si il suit son DTD et il est bien formé.

3.2. C'est quoi DTD (data type definition) :

Le DTD est un grammaire pour la structuration et la conception des documents XML. Il n'est pas obigatoire il peut étre facultative, interne ou externe au document XML. IL contient des déclarations pour les éléments, attributs, entités, notations utilizes. On peut distingué deux avantages principals du DTD :

- Partage d'une même structure entre plusieurs documents, structures « standard » pour une communauté
- Vérification stricte et automatisable de la correction des documents

Voici un exemple d'un document XML et de son DTD :

```xml
<?xml version="1.0" encoding="ISO-8859-1" standalone="no" ?>
<!DOCTYPE document SYSTEM "accueil.dtd">
<document type='exemple'>
  <salutation> Bonjour! </salutation>
</document>
```

XML

```
<!-- fichier accueil.dtd. Exemple de DTD simple -->

<!-- Définition de l'élément racine -->
<!ELEMENT document (salutation)>
<!-- Définition de l'attribut type pour l'élément document -->
<!ATTLIST document type CDATA #IMPLIED>

<!-- Un élément salutation ne contient que du texte -->
<!ELEMENT salutation (#PCDATA)>
```

DTD

Les propreités des éléments du DTD : les elements, balise et attributs en xml sont représentés comme suite en DTD :

Les Sous-éléments : plusieurs façons de les combiner

- séquence : une balise chapitre est représenté end td comme suit :

  <!ELEMENT chapitre (titre, intro, section)>

  • Remarque: l'ordre des éléments est important

- alternative : l'ordre des balises est trés important

  <!ELEMENT chapitre (titre, intro, (section|sections))>

  L'exemple precedent veut dire que dans le document xml on utilize section ou sections.

- indicateurs d'occurrence: * (0-n), + (1-n), ? (0-1)

  <!ELEMENT chapitre (titre*, intro?, section+)>

  *: 0 ou plusieurs. +:1 ou plusieurs. ?: optional

- Données :

  texte : <!ELEMENT chapitre (#PCDATA) >

  #PCDATA: chaine de caractéres.

Les attributs avec DTD : <ATTLIST nom-élément nom-attribut type-attribut declaration-default>

– Pour un élément donné on décrit la liste de ses attributs

– Chaque attribut: un nom, un type et une valeur par défaut

– Remarque: l'ordre des attributs n'est pas important

Exemple : <!ELEMENT ex (#PCDATA)>

<!ATTLIST cible ID #REQUIRED

Nb (1|2|3) '1'

Propriétaire CDATA #fixed 'moi' >

Valeur par défaut d'un attribut

– La valeur en question

– #REQUIRED : attribut obligatoire, valeur à être précisée dans le document

– #IMPLIED : attribut facultatif, valeur à être précisée dans le document

– #FIXED (suivi de la valeur) : valeur de l'attribut fixée pour tout élément instance

3.3. Les directives du TEI5 :

Cette partie définie l'ensemble d'instructions pour l'encodage des ressources lexicales de tous types en particulier les dictionnaires électroniques monolingue et multilingue, les glossaires, concordances et les listes des mots **[19]**. Les balises utilisées pour décrire la structure de base d'une ressource lexicale sont **[17]**:

3.3.1. Structure d'un document TEI :

Les informations concernant les mots sont contenues dans des éléments XML délimité par les balises **<TEI>** et **</TEI>**. Un document **TEI** complet se compose d'un **<TeiHeader>** contenant toutes les métadonnées qui le décrivent, la partie <body> qui contient le document lui-même. <Entry> : contient une entrée structurée de dictionnaire. Cette structure commune est obligatoire pour tous les documents TEI **[20]**.

<TEI> [21]

<teiHeader>

<!---...-->

</teiHeader> **[21]**

<body>**[21]**

<!--...-->

</body>

</TEI>

3.3.1.1. TEI Header : Il est constitué de trois composants :

- **<title>** : information concernant le titre de la ressource.
- **<authors>** : le responsable de cette ressource.
- **<**publicationStmt**>**: des détails concernant la publication de cette ressource.

3.3.1.2. Body : contient la totalité des entrées d'une ressource lexical évidemment groupé en un ou plusieurs balises div. Ces divisions peuvent correspondre à des sections pour différentes langues dans un dictionnaire bilingue **[25].**

<Entry> : L'élément entry peut contenir les éléments suivants :

- Des informations sur la forme du mot traité (orthographe, prononciation, césure, etc.)
- Information grammaticale (partie du discours, sous-catégorisation grammaticale, etc.)
- Définitions ou traductions dans une autre langue **[26]**.
- Étymologie **[15]**.
- Exemples **[14]**.
- Informations d'utilisation**[22].**

Chaque entrée comporte plusieurs balises, chacune fournissant un type d'information différent sur le mot traité **[12]**.

3.3.2. Informations sur les formes écrites et parlées :

Chaque entrée commence souvent par des informations sur la forme orthographique ou la coupure du mot (Ex : Porte-----por/te, appelé césure ou syllabification) **[11]**. Dans cette partie, d'autres informations sur la forme peuvent être données y'compris les formes variantes ou alternative, les formes fléchies, la prononciation…ect **[10]**. Les balises et les éléments suivants doivent être utilisés pour coder ces informations**[10]** :

<form >: Regroupe toutes les informations relatives à la morphologie et à la prononciation d'une entrée. Les attributs relative à cette balise sont :

- **@extent** indique si la prononciation ou orthographie se rapporte au mot entier ou seulement à une partie Les valeurs suggérées comprennent: 1] full(full form) ; 2] pref(prefix) ; 3] suff(suffix) ; 4] inf(infix) ; 5] part(partial) **[9]**
- **@type** qualifie la forme comme simple, composée, etc. Les valeurs suggérées comprennent: 1] simple; 2] lemma; 3] variant; 4] compound; 5] derivative; 6] inflected; 7] phrase

- <**orth>** donne l'orthographe d'un mot-vedette.

Exemple :

<form type**="derivative"> [10]**
<orth>Déshéritement**</orth>**
</form>

- <**pron>** (prononciation) contient la/les prononciation(s) du mot

Exemple :
<entry>
<form>
<orth>amygdale**</orth>**
<pron extent**="full">**[ami(g)dal]**</pron>**
</form>
</entry> [10]

- <**syll>** (syllabisation) contient la syllabisation du mot-vedette.

Exemple:
<form> [10]
<orth>chauve-souris**</orth>**

<syll>chau|ve|sou|ris**</syll>**

</form>

- <**stress**> contient le modèle d'accentuation d'une entrée de dictionnaire, s'il est donné à part. En règle générale les informations sur l'accentuation sont comprises dans les informations sur la pronunciation **[11]**.

example:

<form>

<orth>alternatingcurrent**</orth>**

<stress>,....'..**</stress>**

</form>

- **<lbl>** : étiquette pour la forme d'un mot, pour un exemple, pour une traduction, ou pour tout autre type d'information par exemple :

example :

<entry>

<form type**="abbrev">**

<orth>cf.**</orth>**

</form>

<form type**="full">**

<lbl>abréviationpour**</lbl>**

<orth>confer**</orth>**

</form>

</entry>

3.3.3. Information grammaticale (syntaxique) :

L'élément <gramGrp> regroupe des informations grammaticales, comme part of speach (POS), des informations de sous-catégorie (par exemple, des motifs syntaxiques pour les verbes, des distinctions de dénombrement et de masse pour les noms), etc **[22]**. Les balises et les éléments qui décrit les informations grammaticales d'un mot sont :

- **gen** (genre) : identifie le genre morphologique d'un élément lexical, tel qu'il est donné par le dictionnaire. Il Contient des caractères et des éléments du niveau expression comme *masculin*, *féminin*, *neutre*, etc**[22]**.

Exemple :

<entry> [22]

<form>

<orth>pamplemousse**</orth>**

</form>

```
<gramGrp>
 <pos>nom</pos>
 <gen>masculin</gen>
</gramGrp>
</entry>
```

- **number** (nombre) indique le nombre grammatical associé à une forme, telle qu'elle est donnée par le dictionnaire (pluriel ou masculine) **[22]**.

Example 1:

```
<entry>
 <form>
  <orth>wits</orth>
  <pron>wIts</pron>
 </form>
 <gramGrp>
  <number>pl</number>
  <pos>n</pos>
 </gramGrp>
</entry>
```

Exemple 2:

<entry> [22]

```
 <form>
  <orth>épousailles</orth>
  <pron>[epuzaj]</pron>
 </form>
 <gramGrp>
  <number>pl.</number>
  <pos>n.</pos>
 </gramGrp>
</entry>
```

- **per** (personne) : contient des indications sur la personne grammaticale (1re, 2e, 3e, etc.) liée à une forme fléchie donnée dans un dictionnaire **[13]**.
- **tns** (temps) : indique le temps grammatical lié à une forme fléchie donnée dans un dictionnaire **[22]**.

Example:

```
<entry>
 <form type="flected"> [22]
  <orth>vas </orth>
 </form>
  <gramGrp>
    <per>2</per>
    <number> S </number>
    <tns> P</tns>
    <mood> indicative </mood>
  </gramGrp>
 </entry>
```

- **pos** (partie du discours) indique la partie du discours attribuée à une entrée de dictionnaire telle que nom, verbe, adjective **[22]**.

Example:

```
<entry>
 <form>
  <orth>isotope</orth>
 </form>
 <gramGrp>
  <pos>adj</pos>
 </gramGrp>
</entry>
```

3.3.4. Des informations sémantiques :

Les dictionnaires peuvent décrire les significations des mots dans une grande variété de façons différentes au moyen de synonymes, de paraphrases, de traductions dans d'autres langues…etc**[22]**. Toutes les informations peuvent être étiquetés en utilisant l'élément <def> qui contient le texte de la définition dans une entrée de dictionnaire et **<sense> qui** regroupe toutes les informations relatives à un des sens d'un mot dans une entrée de dictionnaire (définitions, exemples, équivalents linguistiques, etc.) **[13]**.

Example

```
<entry>
 <form>
  <orth>compétiteur</orth>
  <hyph>com|péti|teur</hyph>
```

<pron>[köpetitœR]</pron>
</form>
<gramGrp>
<pos>n</pos>
</gramGrp>
<sens>
<def>Personne qui entre en compétition.</def>
<sens>
</entry>

- **Traduction : Pour les cas des ressources lexicales multilingue nous utilisons la balise** <cit type="translation"> et l'attribut xml:lang pour indiquer la langue cible et la langue source [22].

Exemple :

<entry>
<form>
<orth>to horrify</orth> **[26]**
</form>
<cit type="translation" xml:lang="en">
<quote>horrifier</quote>
<quote> to horrify </quote>
<def>elle était horrifiée par la dépense.</def>
</cit>
</cit>
</entry>

3.4. Lexicale markup framework

LMF est un Méta-modèle (Classe, attribut, association agrégation, généralisation et héritage) sous forme de spécifications qui permet de représenter le lexique avant la construction d'une ressources lexicale (RL) **[48]**.

- Conforme aux principes de modélisation du langage UML (unified modelling language) qui est pertinent pour la description linguistique.
- Les mêmes spécifications sont utilisées pour les petits et les grands lexiques, simple ou complexe.
- Les données sont représentées à l'aide de codages de caractères Unicode.
- LMF assure la simplification des lexiques (monolingue ou multilingue) par le biais d'une représentation conceptuel dans un cadre technologique **[48]**.

Objectif :

- Fournir un modèle commun pour la création, l'utilisation et la combinaison des RLs.
- Gérer l'échange de données entre les RLs.
- Permettre de fusionner un grand nombre de RLs pour former de vaste et extensible RLs mondiales.
- Il vise à créer un modèle lexicographique générique.
- Indépendant des langages de codage des RLs comme TEi.

3.4.1. La structure générale du LMF :

Le principe général du LMF est constitué d'une partie noyau (core package) qui représente le squelette structurel qui décrit la hiérarchie de base des informations d'une entrée lexicale et peut être étendu pour satisfaire certaines exigences liées au traitement de certains problèmes linguistiques pour cela des extensions doivent être ajoutés couvrant notamment, les aspects morphologique, syntaxique, sémantique et MRD (Machine Readable Dictionary) [48].

3.4.1.1. Partie noyau (core package) :

La partie noyau du LMF, spécifie les notions du lexique, de mot, de forme et de sens En décrivant la hiérarchie de base des informations qui peuvent être incluses dans une entrée lexicale comme le montre le diagramme suivant :

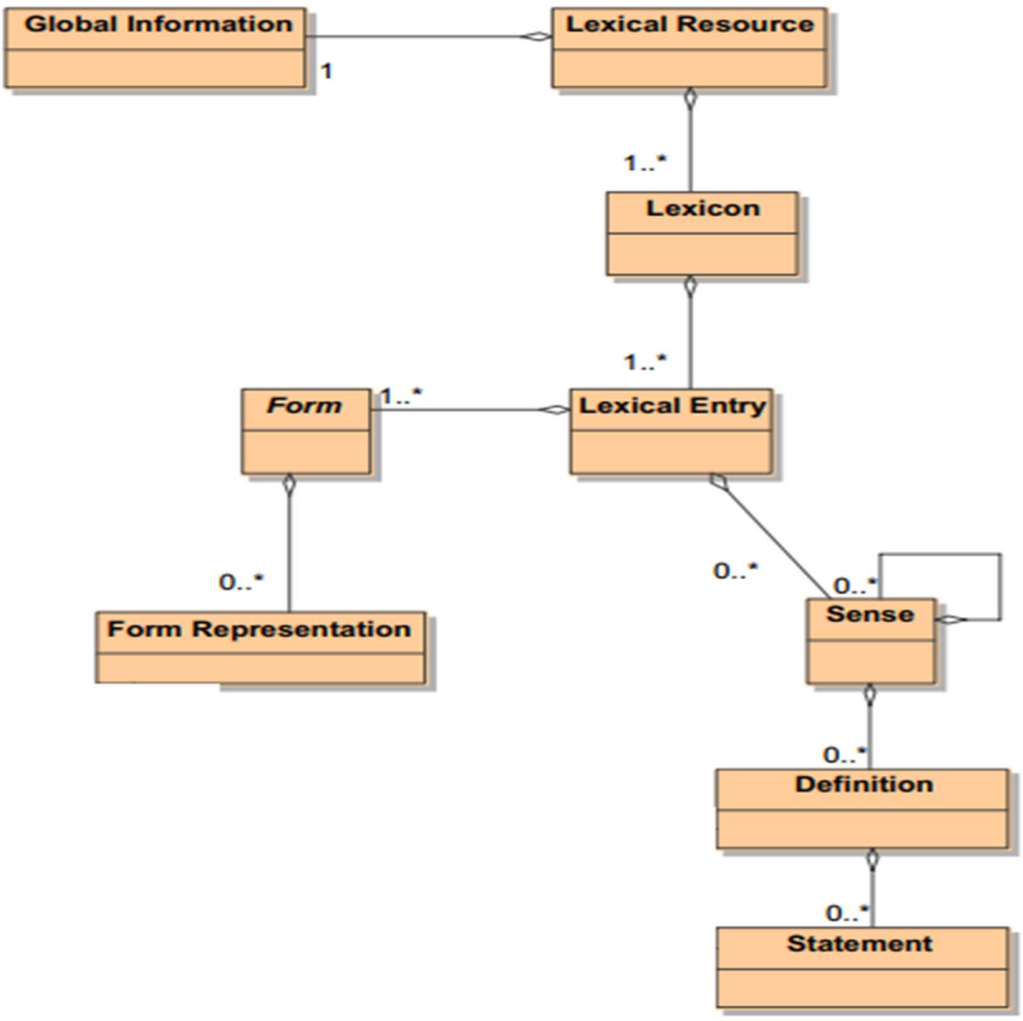


Figure 2: structure générale d'une conception LMF [48]

- **Lexical resource class :** représente la ressource en entier et apparait une et une seul fois et contient un ou plusieurs lexicons (1..*). Ce qui veut dire que la ressource lexicale peut être monolingue ou multilingue **[48]**.
- **Lexicon class**: une ressource qui contient des entrées lexicales pour un langage donné. Elle représente le récipient pour toutes les entrées lexicales d'une langue. Cette classe doit contenir au moins une entrée lexicale et n'autorise pas de sous-classes **[48]**.
- **Global information class:** représente les informations administratives et n'autorise pas de sous classes. Cette class doit contenir au minimum l'attribut « coding language» qui représente le code de la langue de cette ressource lexicale. Elle peut contenir aussi « script coding », « character coding » pour spécifier la version unicode utiliser dans les instances du RL **[48]**.
- **Lexical entry class:** représente un lexème dans un langage donné. Elle n'autorise pas de sous classe vu qu'elle est un récipient pour diriger les classes Form et Sense. Par conséquent, elle dirige le rapport entre les formes et

leurs sens apparentés. Une entrée lexicale peut avoir une ou plusieurs formes différentes, et peut avoir zéro ou plusieurs sens différents **[48]**.

- **Form class: Elle** autorise des sous-classes et permet de représenter la variation morphologique d'un lexème par le biais de Gérer une ou plusieurs variantes de la forme écrite ou parlée de l'entrée lexicale (lemma, stem ….ect).
- **Form representation class :** est une classe représentant une variante orthographique d'une form.
- **Sense class:** représente un ou plusieurs sens d'une entrée lexicale et autorise les sous classes. Elle contient les attributs qui décrivent le sens du mot. Elle peut être partagée par plusieurs entrées lexicales.
- **Definition class : elle** représente une description narrative d'un sens et elle est affichée pour les utilisateurs humains afin de leur faciliter la compréhension de la signification d'une entrée lexicale. Une instance de Sense peut avoir de zéro à plusieurs définitions **[48]**.

3.4.1.2. Les extensions LMF :

- Toutes les extensions sont conformes au package principal LMF et ancrée dans un sous-ensemble des classes du package principale **[48]**.
- Une extension ne peut pas être utilisée pour représenter des données lexicales indépendamment du package principal. Selon le type de données linguistiques impliquées, une extension peut dépendre d'une autre extension.
- une extension est un package UML.
- Rappelons que les extensions doivent être sélectionnées selon le besoin du créateur du lexique.

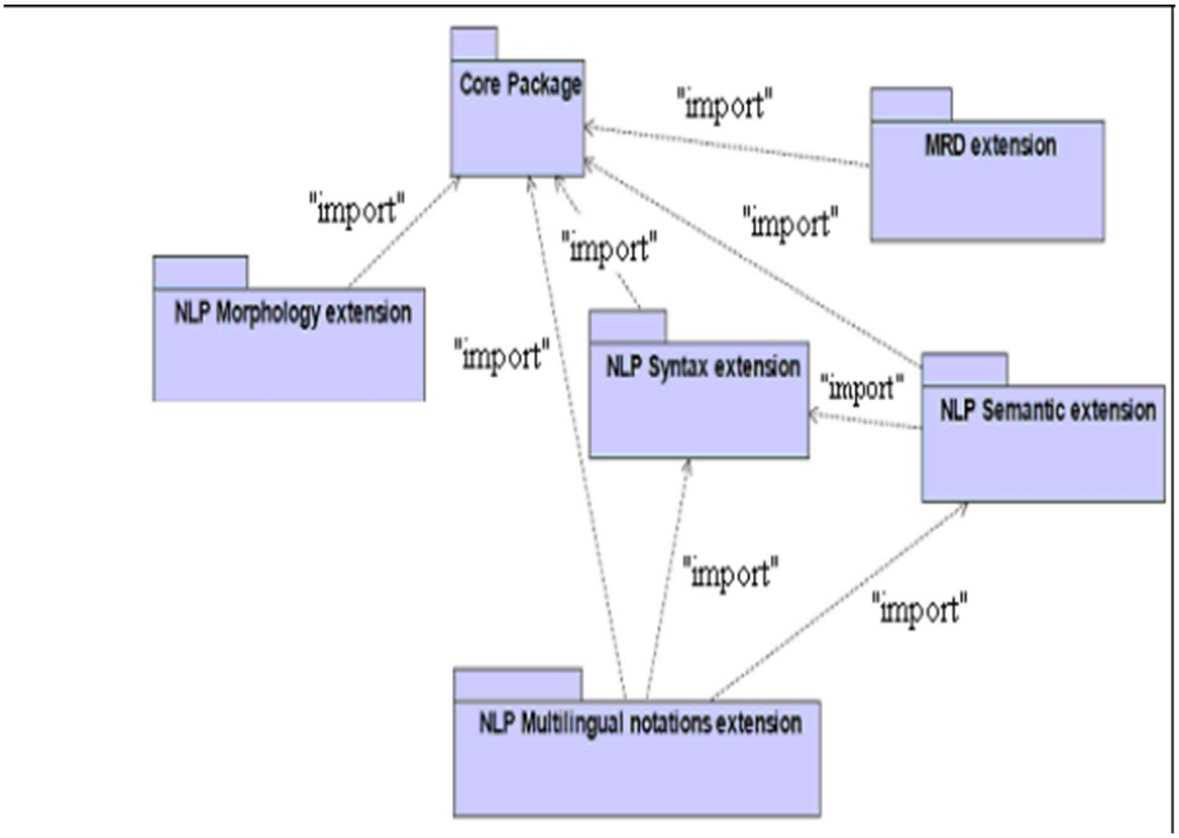


Figure 3 : représentation des packages composant le LMF et leurs relations **[48]**

- **Extension morphologique :** Le but de cette extension est de fournir des mécanismes décrivant la morphologie des entrées lexicales par:

• La description des variantes grammaticales (lemma, stemroot, wordform)

• La relation de l'entrée lexicale avec d'autres entrée lexicale (related form, derived form, refered root)

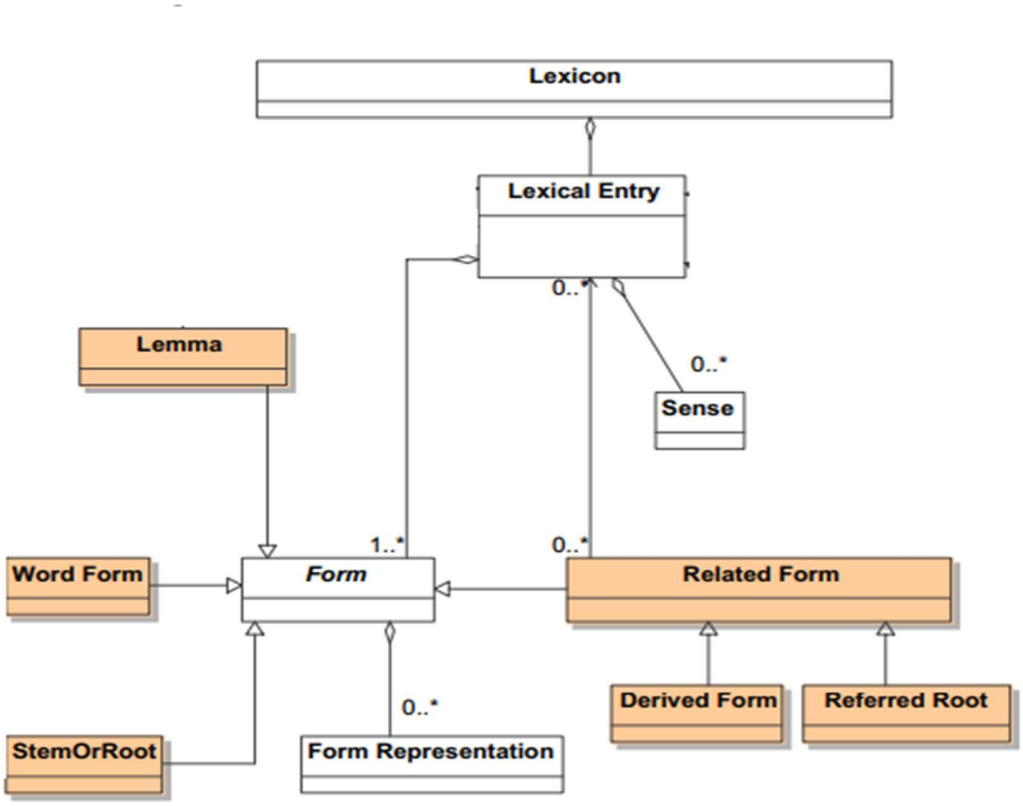


Figure 4 : les composants de l'extension morphologique du LMF **[48]**

- **Lemma class:** Permet de représenter le lemme d'un entrée lexicale et elle hérite les caractéristiques de la classe form.
- **Word form class:** Elle représente la forme qu'un lexème peut prendre lorsqu'il est utilisé dans une phrase. Cette classe peut gérer lexème simple, composée ou des expressions multi-mots
- **StemOrRoot class :** Elle est une sous-classe de form et représente un morph (la partie principale de la forme du mot).
- Related form class: sous classe de form représentant une forme de mot ou un morph qui peut être lié à l'entrée lexicale de différentes manières (par exemple, la dérivation, la racine). Elle peut étre saisi avec les sousclasses suivante: Derivated Form class and Referred Root class
- Derived form class: est une sous-classe de related form représentant une forme de mot de type dérivationnel.

Refered root class : Elle est une sous-classe de related form représentant un morph de type racine. Cette class représente spécifiquement une racine gérée par une instance de Lexical Entry class qui est différente et partagée par deux ou plusieurs autres instances de Lexical Entry class.

Nom des Classes	**Exemple des attributs**	**Commentaire**
Lemma	writtenForm phoneticForm geographicalVariant scheme	/writtenForm/ et /phoneticForm/ valeurs unicode
Word form	writtenForm phoneticForm hyphenation grammaticalNumber grammaticalGender grammaticalTense person	Lorsque /writtenForm/ a la valeur "kitten", /hyphenation/ aura la valeur "kit ten". /grammaticalNumber valeur /plural/
StemOrRoot	writtenForm phoneticForm	
Derived Form	writtenForm phoneticForm	
Referred root	writtenForm	

Tableau 3 : les informations d'une entrée lexicale dans LMF **[48]**

Exemple :

- L'entrée lexicale est le lemme clergyman associer avec deux formes fléchis clergyman et clergymen. Le codage de la langue est ISO **[48]**.

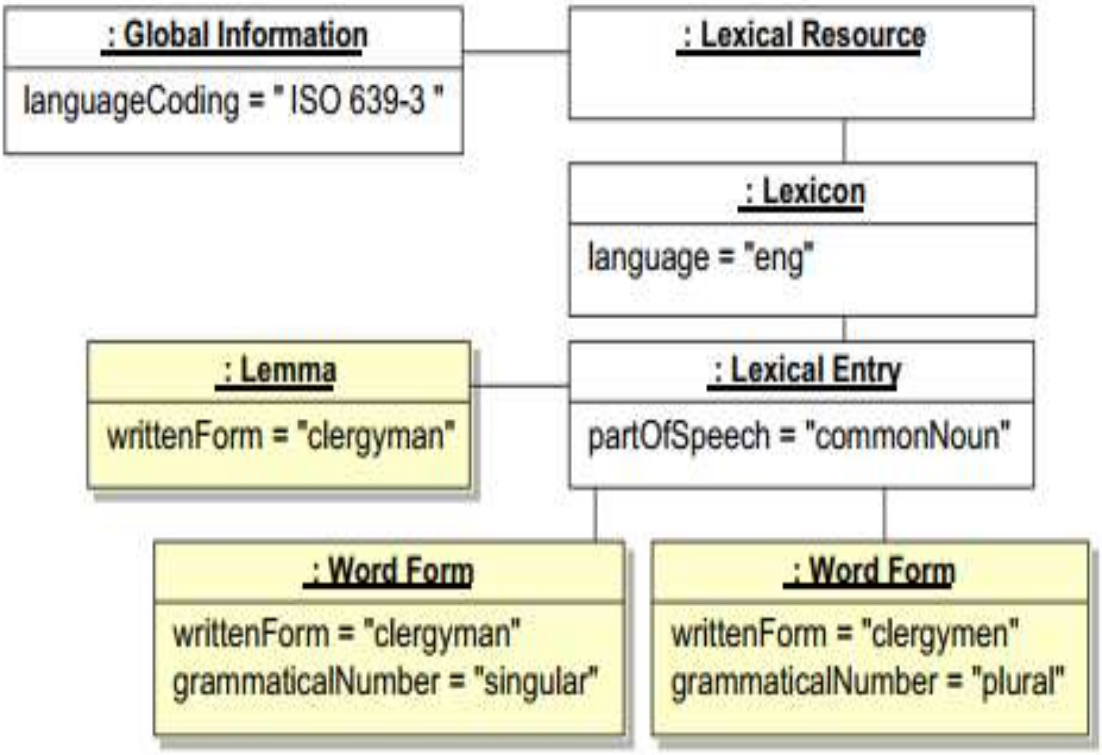


Figure 5 : exemple de représentation du mot clergymen **[48]**

Chapitre 4 :

Les ressources lexicales sémantique

4. Introduction

La sémantique lexicale est l'étude du sens des "mots" ou plutôt des morphèmes d'une langue. Pour définir le "sens" d'un mot, on recourt en général à d'autres mots et on peut dire que le sens d'un mot est une représentation discrète d'un aspect dans un contexte particulier **[25]**. Par exemple les mots « mouse » et « bank » sont ambigué puisqu'ils peuvent avoir plusieurs sens selon le contexte :

- Sense 1 du mot mouse : a mouse controlling a computer system in 1968.
- Sense 2 du mot mouse : a quiet animal like a mouse.
- Sense 1 du mot bank : a bank can hold the investments in a custodial account.
- Sense 2 du mot bank : as agriculture burgeons on the east bank, the river

Dans ce chapitre nous allons voir les différentes RLs qui peuvent étre utilisées pour résoudre les problémes d'ambiguité sémantiques :

4.1. Wordnet :

Appelé « Dictionnaire conceptuel », « un reseau de conceptes », « Dictionaire de synonymes », « Taxonomie des concepts ».cette RL est la plus couramment utilisée pour les relations sensorielles en anglais. Elle se compose de trois bases de données distinctes, une pour les noms et les verbes et une troisième pour les adjectifs et les adverbes. Chaque base de données contient un ensemble de lemmes, chacun annoté avec un ensemble de sens. La version actual de WordNet 3.0 contient 117 798 noms, 11 529 verbes, 22 479 adjectifs et 4 481 adverbes. Le nom moyen a 1,23 sens et le verbe moyen a 2,16 sens. Elle est accessible sur le Web ou pour le téléchargement locale. La figure suivante montre l'entrée pour le nom « bass » **[33][25]**.

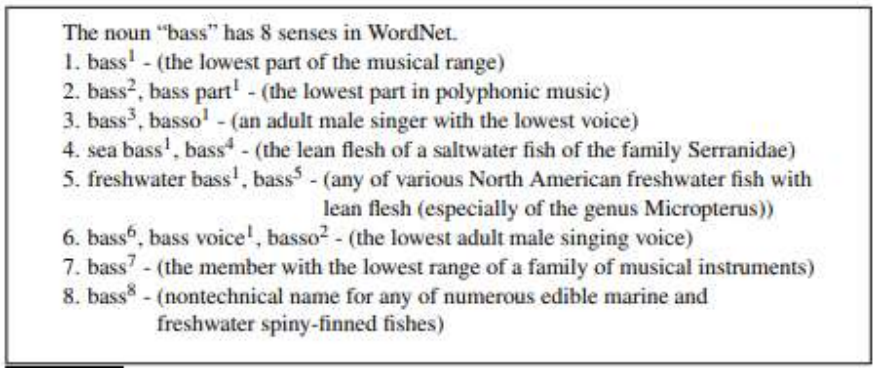


Figure 6 : Une partie de l'entrée WordNet 3.0 pour le nom « bass » **[33]**

Notez qu'il y a huit sens pour le nom, chacun a un gloss (une définition de style dictionnaire), une liste de synonymes pour le sens, et parfois aussi des exemples d'utilisation (indiqués pour le sens de l'adjectif). WordNet ne représente pas la prononciation. L'ensemble de tous les synonymes pour un sens WordNet est appelé un synset ; les synsets sont une primitive importante dans WordNet. L'entrée pour « bass » comprend des synsets comme {bass[1] , deep[6]}, or {bass[6] , bass voice[1] , basso[2]} **[25]**.

4.1.1. Les caractéristiques du wordnet

- Les noeuds proches dans le graphes sont similaire (figure suivante)
- Les conceptes sont organisé en fonciton du sens des mots
- Peut étre vu comme un graphe ou les nœuds sont les synsets et les arcs sont les liens entre les synses
- Elle couvre la grande majorité des noms, verbes, adjectifs et adverbes de la langue anglaise.
- Elle est composé de 117798 Noms, 11525 Verbes, 22479 Adjectifs, 4471 Adverbes, 115 000 synsets.
- **Mappage entre les versions de wordnet :** Il existe une correspondance des identifiants de synsets entre versions de WordNet. Ce mappage est indispensable pour assurer une traçabilité avec la version la plus récente. En effet, plusieurs ressources complémentaires à WordNet, et dignes d'intérêt, ont été définies pour la version 1.7 ou 2.0. Curieusement, le site Web de Princeton n'offre de mappage « officiel » que pour les noms et les verbes. Heureusement, d'autres sites proposent également des correspondances (construites automatiquement) pour les adjectifs et adverbes **[25][24]**.
- **Fréquence des lemmes :** WordNet donne une fréquence d'apparition pour chaque lemme définissant un synset. Ce nombre indique combien de fois un mot apparaît dans un sens spécifique. Pour un nom ou un verbe, la somme cumulée des fréquences d'un synset et de ses hyponymes au sein d'un sous-arbre de la hiérarchie permet de calculer son Contenu Informationnel **[25]**.
- Dans l'esprit des réseaux sémantiques, le Wordnet est organisée autour de «concepts lexicalisés» abstraits plutôt que de formes de mots ou de lexèmes triés par ordre alphabétique.

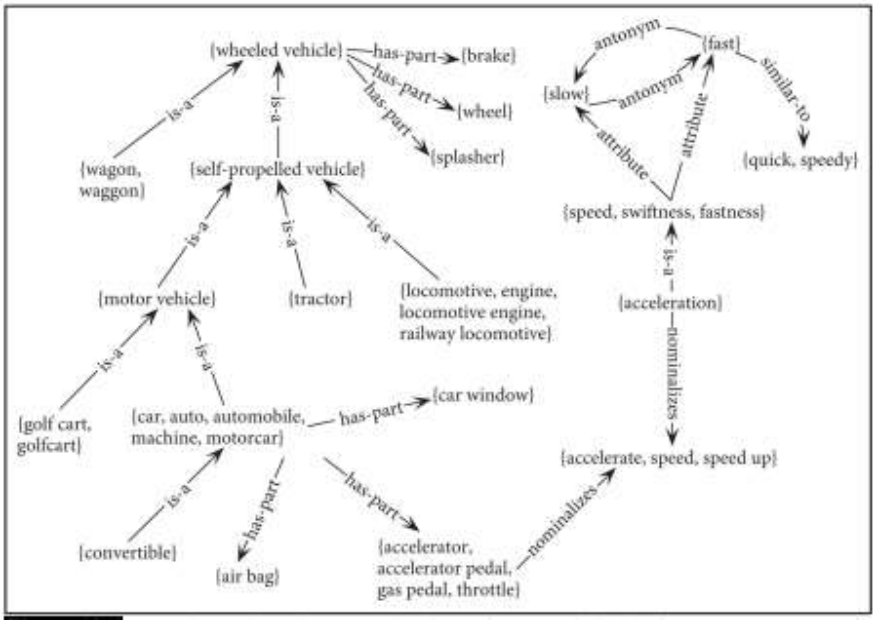


Figure 7 : WordNet vu sous forme de graphique **[25]**.

4.1.2. Historique du wordnet :

Wordnet est inspiré des théories psycholinguistiques de la mémoire lexicale humaine et l'acquisition du sens lexical par les enfants. Elle a été crée en 1985 au Laboratoire de sciences cognitives de l'Université de Princeton[6] **[24]**.

4.1.3. La structure générale du synset dans wordnet :

Chaque synset dans wordnet est constitué d'un ensemble de mots ayant le même sens avec un gloss ou définition du concept et l'identifiant comme le montre la figure suivante **[25]**.

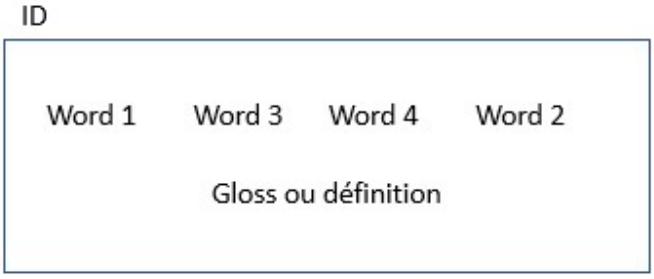


Figure 8 : la structure d'un synset dans wordnet **[25]**

[6] https://stringfixer.com/fr/Princeton_University_Department_of_Psychology

Chaque synset est caractériser par :

1- Ensemble de mots sémantiquement lié
2- Qu'ils ont même POS
3- Peuvent être échanger dans un contexte particulier
4- Ils font référence aux même concept

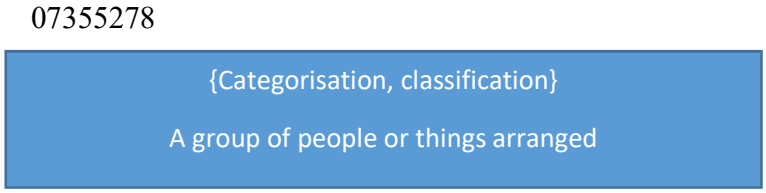


Figure 9 : exemple de synset avec ID 07355278, gloss : a group of people or things arranged **[25]**

4.1.4. Les Relations sémantique entre Synsets

Cette section explore les relations entre les sens des mots, en particulier ceux qui ont fait l'objet d'une enquête computationnelle importante comme la synonymie, l'antonymie et l'hyperonymie.

- **Synonymie** Deux mots A et B sont synonym si dans un contexte donné A peut remplacer B et B peut remplacer A sans affecter la sen générale. Les synonymes incluent des paires telles que : couch/sofa, vomit/throw up, filbert/hazelnut, car/automobile. Nous avons mentionné aussi qu'en pratique, le mot synonyme est couramment utilisé pour décrire une relation de synonymie approximative. Mais de plus, la synonymie est en fait une relation entre les sens plutôt que les mots **[38]**. Considérant les mots big et large ceux-ci peuvent sembler être des synonymes dans les phrases suivantes, car nous pourrions échanger big et large dans l'une ou l'autre phrase et conserver le même sens :
- How big is that plane?
- Would I be flying on a large or small plane?

Mais notez la phrase suivante dans laquelle nous ne pouvons pas remplacer big par large :

- Miss Nelson, for instance, became a kind of big sister to Benjamin.

- Miss Nelson, for instance, became a kind of large sister to Benjamin.

C'est parce que le mot big a un sens qui signifie être plus âgé ou grandi, alors que large n'a pas ce sens. Ainsi, nous disons que certains sens du big et large sont (presque) synonymes alors que d'autres ne le sont pas [24].

- **Antonymie :** Alors que les synonymes sont des mots ayant un sens identique ou similaire, les antonymes sont des mots ayant un sens opposé, comme :

long/short, big/little, fast/slow, cold/hot, dark/light, rise/fall, up/down, in/out

Deux sens peuvent être des antonymes s'ils définissent une opposition binaire ou sont aux extrémités opposées d'une certaine échelle. C'est le cas pour long/short, fast/slow ou big/little, qui sont des inversions aux extrémités opposées de l'échelle de longueur ou de taille. Un autre groupe d'antonymes, les réversifs, décrivent un changement ou un mouvement dans des directions opposées, tel un rise/fall ou up/down **[25]**.

- **Relations taxonomiques :** Une autre façon dont les sens des mots peuvent être liés est taxonomiquement. Un mot (ou sens) est un hyponyme d'un autre mot ou sens si le premier est plus spécifique, désignant une sous-classe de l'autre. Par exemple, voiture est un hyponyme de véhicule, chien est un hyponyme d'animal, mangue est un hyponyme de fruit. A l'inverse, on dit que véhicule est un hyperonyme de voiture, et animal est un hyperonyme de chien **[33]**.
- **Hyponymy / hypernymy :** c'est la base de la forme hiérarchique du wordnet en formant de longs chemins de sousclasses/super-classes présenté entre les noms [25].

- X is a hyponymy de y si x is a kind of y. Hyponymy est transitive et asymétrique.
- Hyperonymy est l'inverse de l'hyponymy. Ex: hypernymy – {tree, tree diagram} is a kind of {plane figure, two-dimensional figure}.
- hyponymy – {tree} peut étre {chestnut, chestnut tree},

Exemples : partant du sens le plus général du mot CAT (le « chat »), on obtient une liste ordonnée d'ancêtres et de descendants, permettant de déterminer qu'un chat est un carnivore, un mammifère, un animal, etc **[35]**.

- **Meronymy vs holonymy :** partie-de (part of) Une autre relation commune est la méronymie, la relation partie-tout. Une roue fait partie d'une voiture alors nous disons que roue est un méronyme de voiture, et voiture est un holonyme de roue. ex: branch is part of tree. X is meronymy of y si x is a part of y. Holonymy est l'inverse du meronymy. Ex Les meronyms de body sont: Arm, head, blood, tissue. Alors Head est holonym de face et face est un holonyme de eye et eye holonym de pupil. pour dire que le synset {x, x/,….} est une partie du synset {y, y/…..} si chaque mot x is part of y. Grâce à ces relations, on peut déterminer aussi qu'un chat a des pattes, un pelage, une queue [25]. les synsets sont interconnecté par des relations comme Hyponymy : is a et Meronymy : part of
- **Troponymy :** Relation lexicale, C'est une forme d'hyponymy entre les verbes, V2 est une maniére de v1 ex: Speak et stammer. Stammer est un troponym (particular ways) de speak. shamble et walk
- **Pertainymy / related to :** Pertainymy est une relation définie seulement pour les adjectifs {« sunny » pertains au nom « sun »}. Related to :le verbe paint est related to les « paints et painting », « run » related to « runny » **[24]**.
- **Entailement :** Entailement est une relation qui signifier implication. Ex: {« snooring » implique « sleeping » alors « Sleeping » entails « snoring ».

4.1.5. Récapitulation des relations qui existent dans wordnet :

WordNet représente tous les types de relations sensorielles discutés dans la section précédente. Ces relations peuvent étre classées dans des groupes (relations entre synsets, relations entre lemmes, relations entre noms et relations entre verbes). Le tableau suivant résume les différents types de relations qui existent dans wordnet :

Relation	Description	Exemple
Hypernym	Is a generalization of	Furniture is a hypernym of chai
Hyponym	Is a kind of	Chair is a hypernym of furniture
Troponym	Is a way to	Amble is a troponym of walk
Meronym	Is part/substance/member of	Wheel is a (part) meronym of a bicycle
Holonym	Contains part	Bicycle is a holonym of a wheel
Antonym	Opposite of	Ascend is an oppposite of descend
Attribute	Attribute of	Heavy is a attribute weight
Entailement	Entails	Ploughing entails digging
Cause	Cause to	To offend causes to resent
Also see	Related verb	To lodge is related to reside
Similar to	Similar to	Dead is similar to assassinated
Participle of	Is participle of	Stored (adj) is the participle of « to store »
Pertainym of	Pertains to	Radial pertains to radius

Tableau 4: les différentes relations du wordnet **[25]**

Relation	Entre	Nombre	Exemple
Hypernym / hyponym	Verbe / verbe	13124	Exhale / breathe
	Nom / nom	75134	Cat / feline
Instance hyponym	Nom / nom	8515	Eifeel tower / tower
Part	Nom / nom	8874	France / europe
Member	Nom / nom	12262	France / european union
Substance	Nom/nom	793	Serum / blood
Attribute	Adjectif / nom	643	Inaccurate / accuracy

Verb group	Verbe / verbe	1748	Gelatinize#1 / gelatinize#2
Verb entailment	Verbe / verbe	409	Dream / sleep
Verb cause	Verbe / verbe	219	Anesthetize / sleep
Adjective similar	Adjectif / adjectif	22622	Dying / moribund
Topic domain	Nom / adjectif	1108	Computer science / addressable
	Nom / nom	4146	Computer science / computer
	Nom / adverbe	37	
	Nom / verbe	1236	Computer science / cascade
Region domain	Nom / adjectif	75	
	Nom / nom	1246	French / France
Usage domain	Nom / adjectif	227	
	Nom / nom	563	Neutralization / euphemism
	Nom / adverbe	73	
	Nom / verbe	14	
Voir aussi	Adjectif/adjectif	2683	Black / DARK

Tableau 5 : les relations sémantiques entre synsets qui existent dans wordnet [24]

Relation	**Entre**	**Et**	**Nombre**	**Exemple**
Usage domaine	Nom	Nom	379	//
Voir aussi	Verbe	Verbe	582	Sleep late / sleep
Adjectif participle	Adjectif	Verbe	124	Applied / apply
Antonym	Adjectif	Adjectif	4080	Good / bad
	Adverbe	Adverebe	718	Poorly / well
	Nom	Nom	2142	Winner / looser
	Verbe	Verbe	1089	Die / be born

Pertainym	Adjectif	Nom	4814	Academic / academia
	Adverbe	Adjectif	3213	Boastdully / boastful
	Adjectif	Adjectif	38	
Derivation	Nom	Verbe	21579	Killing / killing
	Adjectif	Nom	11401	Dark / darkness
	Nom	Nom	2931	Automobile / automobiliste
	Verbe	Adjectif	1508	Kill / killable
Adjective cluster	Adjectif	Adjectif	1290	Strident / noisy

Tableau 6 : les relations lexicales entre lemmes dans wordnet [24]

Relation	Also Called	Definition	Example
Hypernym	Superordinate	From concepts to superordinates	*breakfast*^{1} → *meal*^{1}
Hyponym	Subordinate	From concepts to subtypes	*meal*^{1} → *lunch*^{1}
Instance Hypernym	Instance	From instances to their concepts	*Austen*^{1} → *author*^{1}
Instance Hyponym	Has-Instance	From concepts to their instances	*composer*^{1} → *Bach*^{1}
Part Meronym	Has-Part	From wholes to parts	*table*^{2} → *leg*^{3}
Part Holonym	Part-Of	From parts to wholes	*course*^{7} → *meal*^{1}
Antonym		Semantic opposition between lemmas	*leader*^{1} ⟺ *follower*^{1}
Derivation		Lemmas w/same morphological root	*destruction*^{1} ⟺ *destroy*^{1}

Tableau 7 : Relations de noms sur wordnet [24].

Relation	Definition	Example
Hypernym	From events to superordinate events	*fly*^{9} → *travel*^{5}
Troponym	From events to subordinate event	*walk*^{1} → *stroll*^{1}
Entails	From verbs (events) to the verbs (events) they entail	*snore*^{1} → *sleep*^{1}
Antonym	Semantic opposition between lemmas	*increase*^{1} ⟺ *decrease*^{1}

Tableau 8 : relation entre verbes dans wordnet [24].

4.1.6. Le root du wordnet :

La figure suivante représente le Top level des synsets dans wordnet où on peut distinguer 16 levels en tout dans l'hiérarchie du wordnet

Figure 10 : le root level du wordnet **[24]**

4.1.7. Les inconvinients de wordnet

- **Informations manquantes :** WordNet ne précise pas l'étymologie, la prononciation, les formes de verbes irréguliers et ne contient que des informations limitées sur l'usage des mots [25].
- **Profusion de sens :** Malheureusement WordNet est très précis dans la définition des sens où on a une granularité très fine des sens. Par exemple, le verbe « to give » n'a pas moins de 44 sens. Une telle profusion ne facilite pas la tâche de désambiguïsation lexicale [25].
- **Absence de relations pragmatiques :** WordNet ne matérialise pas d'une façon formelle tout le sens contenu dans les définitions des termes. Par exemple, l'information qu'un chat ne rugit pas figure dans la définition, mais ne se retrouve formalisée dans aucune relation. De même, des relations pragmatiques telles que savon / bain sont absentes dans WordNet.

4.1.8. La différence entre thesaurus, dictionnaire et wordnet :

Qu'est-ce WordNet ? Un dictionnaire ? Un thésaurus ? Les dictionnaires contiennent généralement des connaissances sur des lexiques. Quant aux thésaurus, leur structure est bâti autour des concepts et aident l'utilisateur à acquérir l'unité lexicale la plus appropriée lorsqu'il a un concept à rechercher. WordNet n'est ni un dictionnaire classique ni un thésaurus : il est en fait, un arrangement des traits de chacune de ces deux ressources lexicales.

Figure 11 : Ressources descendances de WordNet **[25]**

4.1.9. Applications du wordnet:

Pour les applications du TAL, le wordnet est utiliser dans : Word sens Désambiguation, Information retrieval, query expanded, Recherche multilingue, Corpus étiquetés par rapport à WordNet, indexation, régler le probléme de la polysémie (fréquence de synset), pour l'étiquetage sémantique de corpus ..ect [24].

4.1.10. EuroWordNet:

EuroWordNet est une base de données pour plusieurs langues européennes. La phase initiale du projet s'est achevée en 1999, avec la conception de la base de données, ainsi que la définition de types de relations, d'un haut d'ontologie (63 éléments partagé par toutes les langues) et d'un Index-Inter-Langues (basé sur la version 1.5 du WordNet de Princeton). EuroWordNet a produit des wordnets pour le néerlandais, l'italien, l'espagnol, l'allemand, le français, le tchèque et l'estonien. Les langues sont reliées ensemble par l'intermédiaire de l'Index-Inter-Langues. Il est ainsi possible de passer des mots dans une langue aux mêmes mots dans n'importe quelle autre langue. EuroWordNet permet donc une recherche d'information monolingue ou multilingue [25].

4.2. Thésaurus :

Un thésaurus est une Liste organisée de termes normalisés (descripteurs et non-descripteurs) servant à l'indexation des documents et des questions dans un système documentaire.

Les descripteurs sont reliés par des relations sémantiques (génériques, associatives et d'équivalence) exprimées par des signes conventionnels. Les synonymes (non-descripteurs) sont reliés aux descripteurs par la seule relation d'équivalence. On peut distinguer les thésaurus en fonction du :

- Mode de regroupement des termes
- La variété linguistique des termes (mono- ou multilingue).
- Des domaines de connaissances couverts (thésaurus spécialisé ou sectoriel, thésaurus encyclopédique).

une autre définition à partir du larousse en ligne : un thésaurus est un « Liste alphabétique de mots standards utilisés pour le classement de la documentation. » Il est donc un ensemble de termes, utilisés pour l'indexation (des descripteurs) ou non (non-descripteurs), qui sont relié entre eux avec des relations de synonymie, d'hiérarchie et

d'association. Il peut être spécialisé (souvent dans un domaine, mais cela peut être plus ou moins large) et qui se compose de termes hiérarchisés entre eux [27].

4.2.1. Les éléments du thesaurus:

L'organisation du thésaurus est hiérarchique. Constitué d'un ensemble structuré de concepts représentés par des termes. On distingue plusieurs types de termes dans un thésaurus, avec chacun un statut précis **[45]**. Les types de termes sont :

- Descripteurs (ou termes acceptés) : il s'agit de l'ensemble des mots autorisés pour indexer **[44].**
- Non-descripteurs (termes rejetés) : Il s'agit de synonymes, quasi-synonymes, abréviations ou variantes orthographiques du concept retenu comme descripteur. Ils sont utilisés à la recherche **[43].**
- Mots outils : des descripteurs qui ne peuvent être utilisés seuls, Descripteur + « au moins » descripteur **[44].**

4.2.2. les relations dans thesaurus :

Les relations sémantiques présentes dans thesaurus sont :

- Hiérarchie :
    - Termes génériques (TG) concepts principaux en référence aux autres termes et au domaine considéré **[41]**
    - Termes spécifiques (TS) concepts particuliers à l'intérieur du champ sémantique d'un terme générique **[41]**
- Association **:** Termes associés (TA)
- équivalence **:** Termes équivalents (EP / EM) variantes des termes spécifiques, et non descripteurs, termes non retenus pour représenter une notion, renvoie à un ou plusieurs descripteurs (synonymie)**[42].**
- Appartenance **:**microthésaurus (MT)

4.2.3. Application du thésaurus dans le TAL :

Il existe de nombreux cas où les données du thésaurus peuvent être une source principale d'application.

- Faciliter la description d'un domaine et à harmoniser la communication et le traitement de l'information **[46]**.
- En pratique, le thésaurus est un outil d'indexation de documents, Il donne la possibilité de représenter tout document par une sélection rigoureuse de mots précis, appelés mots-clés **[45]**.
- Les synonymes et dérivés peuvent être utilisés par le moteur de recherche pour étendre la requête de recherche d'origine, permettant de trouver des documents plus relatifs **[43]**.
- Synonymizer utilise le thésaurus comme principale source de synonymes pour les substitutions de mots **[42]**.
- Le moteur de traduction obtient la traduction, les dérivés et les hyperonymes lors de la traduction en texte intégral.

4.2.4. La différence entre thésaurus et ontologie

Un thésaurus est un outil utilisé dans le domaine de la représentation et de la recherche d'information qui représente un domaine de connaissances spécifiques à travers sa structure conceptuelle. Cette structure conceptuelle fournit une organisation sémantique en rendant explicites les relations conceptuelles et en restreignant le sens des termes qui les représentent. Le champ de la connaissance est structuré sur la base de relations conceptuelles hiérarchiques et associatives basées sur l'équivalence **[45]**.

Une ontologie est une représentation formelle et explicite de la structure conceptuelle d'un domaine de connaissance. L'ontologie est un support sémantique pour les mots qui sont décrits comme des objets linguistiques dans une base de données lexicale ou terminologique. Les relations conceptuelles représentées dans une ontologie sont extrêmement variées et dépendent du domaine de connaissance à structurer. Une ontologie est construite dans le but de partager et de réutiliser des informations stockées qui, une fois formalisées, peuvent être interprétées aussi bien par des personnes que par des programmes informatiques **[46]**.

4.3. Sentiwordnet :

Le SentiWordNet permet de fournir une extension pour WordNet, de telle sorte que tous les synsets puissent être associés à une valeur concernant la polarité négative, positive ou objectif. La version actuelle de SentiWordNet est le 3.0 qui représente la version améliorée de SentiWordNet 1.0 et elle est accessible gratuitement au public à des fins de recherche avec une interface Web. Cette extension étiquette chaque synset avec une valeur pour chaque catégorie entre 0,0 et 1,0. La somme des trois valeurs est toujours de 1, donc chaque synset peut avoir une valeur non nulle pour chaque sentiment, car certains synsets peuvent être positifs, négatifs ou objectif selon le contexte dans lequel ils sont utilisés **[43]**.

L'interface Web permet à l'utilisateur de rechercher n'importe quel synset appartenant à WordNet avec ses scores SentiWordNet associés. De plus, l'utilisateur peut voir une visualisation de ces scores. Chaque catégorie est liée à une couleur, qui est le rouge pour la négativité, le bleu pour l'objectif et le vert pour la positivité. Un exemple de la visualisation du synset good peut être vue sur la figure suivante. L'avantage d'utiliser des synsets au lieu de termes est d'offrir des scores de sentiment différents pour chaque sens d'un mot, car les polarités peuvent différer dans un mot en fonction du sens.

Figure 12 : polarité du mot good dans sentiwordnet **[43]**

4.3.1. Application du sentiworndet

Une utilisation typique de SentiWordNet consiste à enrichir la représentation du texte dans les applications d'exploration d'opinion (OM), en ajoutant des informations sur les propriétés liées aux sentiments des termes dans le texte. L'OM est une sous-discipline récente au carrefour de la recherche d'informations et de la linguistique informatique qui ne s'intéresse pas au sujet dont traite un document, mais à l'opinion qu'il exprime. OM dispose d'un riche ensemble d'applications, allant du suivi des opinions des utilisateurs sur les produits ou sur les candidats politiques telles qu'exprimées dans les forums en ligne, à la gestion de la relation client.

Afin de faciliter l'extraction des opinions du texte, des recherches récentes ont tenté de déterminer automatiquement la « polarité PN » des termes subjectifs, c'est-à-dire d'identifier si un terme qui marque un contenu opiniâtre a une connotation positive ou négative. Les recherches visant à déterminer si un terme est en effet un marqueur d'un contenu opiniâtre (un terme subjectif) ou non (un terme objectif) ont été, au contraire, beaucoup plus rares. SentiWordNet est la première ressource lexicale qui fournit un tel niveau de détail spécifique (le sens du mot représenté par un synset) et une couverture aussi large (tous les 115 000+ synsets WordNet) **[45]**.

4.3.2. La construciton du sentiwordnet

La méthode utilisée pour développer SentiWordNet est basée sur l'analyse quantitative des gloses associées aux synsets, et sur l'utilisation des représentations terminologiques vectorielles résultantes pour la classification semi-supervisée des synsets. Les trois scores sont dérivés en combinant les résultats produits par un comité de huit classificateurs ternaires, tous caractérisés par des niveaux de précision similaires mais un comportement de classification différent **[45]**.

4.4. FrameNet

Le projet FrameNet[7] construit une base de données lexicale de l'anglais basée sur des exemples d'annotation de la façon dont les mots sont utilisés dans des textes réels. L'idée de base est que pour comprendre la signification des mots dans une langue il faut d'abord avoir des connaissances sur leurs cadres sémantiques. Il a pour but le mapping du sens sur la forme via la théorie du cadre sémantique puisqu'elle est basée sur l'annotation manuel d'un corpus **[45][46]**. Les elements de base de framenet :

- **Frame :** structure conceptuel qui constitue une représentation et description d'un type d'événement (event), relation, ou entité (ITEM) et les participants à cette entité. Bien qu'il soit utile de faire des inférences sur les points communs sémantiques entre différentes phrases avec augmentation, ce serait encore plus utile si nous pouvions faire de telles inférences dans beaucoup plus de situations, à travers différents verbes et également entre les verbes et les noms **[38]**. Par exemple, nous aimerions extraire la similarité entre ces trois phrases :
- [$_{Arg1}$ The price of bananas] increased [$_{Arg2}$ 5%].
- [$_{Arg1}$ The price of bananas] rose [$_{Arg2}$ 5%].

[7] https://framenet.icsi.berkeley.edu/fndrupal/about

- There has been a [$_{Arg2}$ 5%] rise [$_{Arg1}$ in the price of bananas].

Notez que le deuxième exemple utilise le verbe « increased » différent (rose), et le troisième exemple utilise le nom « rise » plutôt que le verbe increased. Nous voudrions qu'un système reconnaisse que le prix des bananes est ce qui a augmenté, et que 5% est le montant qu'il a augmenté, peu importe si les 5% apparaissent comme l'objet du verbe augmenter ou comme un modificateur nominal d'augmentation du nom **[43]**. Considérez l'ensemble de mots suivant :

« reservation, flight, travel, buy, price, cost, fare, rates, meal, plane »

Il existe de nombreuses relations lexicales individuelles d'hyponymie, de synonymie, etc. entre de nombreux mots de cette liste. L'ensemble de relations qui en résulte ne constitue cependant pas un compte rendu complet de la manière dont ces mots sont liés. Ils sont clairement tous définis par rapport à un ensemble cohérent d'informations contextuelles de bon sens concernant le transport aérien **[33]**. Nous appelons la connaissance de base holistique qui unit ces mots un frame. L'idée que des groupes de mots sont définis par rapport à certaines informations de base est répandue dans l'intelligence artificielle et les sciences cognitives. Voici quelques exemples de phrases :

- [ITEM Oil] rose [ATTRIBUTE in price] [DIFFERENCE by 2%].
- [ITEM It] has increased [FINAL STATE to having them 1 day a month].
- [ITEM Microsoft shares] fell [FINAL VALUE to 7 5/8].
- [ITEM Colon cancer incidence] fell [DIFFERENCE by 50%] [GROUP among men].
- a steady increase [INITIAL VALUE from 9.5] [FINAL VALUE to 14.3] [ITEM in dividends]
- a [DIFFERENCE 5%] [ITEM dividend] increase...

À partir de ces exemples de phrases, notez que le frame comprend des mots cibles tels que monter, descendre et augmenter **[46]**. En fait, le frame complet se compose des mots suivants :

Frames	**Rôles principaux**
ATTRIBUTE	L'ATTRIBUT est une propriété scalaire que possède l'ITEM
DIFFERENCE	La distance par laquelle un ITEM change de position sur l'échelle
FINAL_STATE	Une description qui présente l'état de l'ITEM après le changement de la valeur de l'ATTRIBUT en tant que prédication indépendante
FINAL_VALUE	La position sur l'échelle où se termine l'ITEM
INITIAL_STATE	Une description qui présente l'état de l'ITEM avant le changement de la valeur de l'ATTRIBUT en tant que prédication indépendante
INITIAL_VALUE	La position initiale sur l'échelle à partir de laquelle l'ITEM s'éloigne.
ITEM	L'entité qui a une position sur l'échelle.

VALUE_RANGE	Une partie de l'échelle, généralement identifiée par ses extrémités, le long de laquelle les valeurs de l'ATTRIBUT fluctuent.
	Certains rôles non essentiels
DURATION	La durée pendant laquelle le changement a lieu.
SPEED	Le taux de variation de la VALUE.
GROUPE	Le GROUPE dans lequel un ITEM change la valeur d'un ATTRIBUT d'une manière spécifiée.

Tableau 9 : Les éléments du frame en position de changement sur un frame d'échelle **[45]**

FrameNet code également les relations entre les frames, permettant aux frames d'hériter les unes des autres, ou représentant les relations entre les frames comme la causalité (et les généralisations entre les éléments de frame dans différents frames peuvent également être représentées par l'héritage) **[43]**. Ainsi, il existe un changement de position Cause sur un référentiel qui est lié au Changement de position sur un référentiel par la relation de cause, mais qui ajoute un rôle AGENT et est utilisé pour des exemples causatifs tels que les suivants :

- [AGENT They] raised [ITEM the price of their soda] [DIFFERENCE by 2%].

Ensemble, ces deux cadres permettraient à un système de compréhension d'extraire la sémantique commune des événements de tous les usages verbaux et nominaux causatifs et non causatifs **[35]**.

4.4.1. Applications du framenet :

Du point de vue de l'étudiant, c'est un dictionnaire de plus de 13 000 sens de mots, la plupart avec des exemples annotés qui montrent le sens et l'usage. Pour le chercheur en traitement du langage naturel, plus les 200 000 phrases annotées manuellement liées à plus de 1 200 cadres sémantiques fournissent un ensemble de données d'entraînement unique pour l'étiquetage sémantique des rôles, utilisé dans des applications telles que l'extraction d'informations, la traduction automatique, la reconnaissance d'événements, l'analyse des sentiments, etc. Des bases de données de type FrameNet ont été créées pour un certain nombre de langues notamment l'espagnol, l'allemand, le japonais, le portugais, l'italien et le chinois et un nouveau projet travaille sur l'alignement des FrameNets entre les langues **[45]**.

4.5. ConceptNet

L'apprentissage automatique du langage peut être amélioré en lui fournissant des connaissances spécifiques et des sources d'informations externes. ConceptNet est un projet de représentation des connaissances, fournissant un grand graphe sémantique qui décrit les connaissances humaines générales et comment elles sont exprimées en langage naturel. Il comprend des mots et des expressions courantes dans n'importe quelle langue humaine écrite. Il fournit un large éventail de connaissances de base qu'une application informatique travaillant avec du texte en langage naturel devrait connaître. Nous présentons ici une nouvelle version de la ressource de données ouvertes liées « ConceptNet »

qui est particulièrement bien adaptée pour être utilisée avec les techniques TAL modernes **[46]**. Dans cette partie nous allons répondre aux questions suivantes :

1- Qu'est-ce que ConceptNet ?
2- Comment sa structure ?
3- Quelles sont les points forces et les faiblesses de ConceptNet?

4.5.1. Qu'est-ce que ConceptNet ?

ConceptNet est un réseau sémantique disponible gratuitement, conçu pour aider les ordinateurs à comprendre le sens des mots que les gens utilisent. Il est issu du projet de « crowdsourcing Open Mind Common Sense », lancé en 1999 à MIT Media Lab. Il s'est depuis développé pour inclure des connaissances provenant d'autres ressources participatives, des ressources créées par des experts et des jeux avec un objectif **[46]**.

4.5.2. Structure Et l'utilisation de ConceptNet :

La structure du conceptnet est similaire à celle de WordNet, mais il est plus riche en connexions. ConceptNet en est déjà à sa cinquième version (Speer et Havasi, 2013) Pour la construction des concepts, une plateforme en ligne a été créée et des phrases du type : « The effect of eating food is » et la question quelle est l'effet de manger de la nourriture?. Les réponses ont été collectées et 700 000 concepts ont été obtenus, réduits à 300 000 après l'application de filtres et des règles d 'extraction de patrons. ConceptNet est un graphe de connaissances qui relie des mots et des phrases du langage naturel avec des arêtes étiquetées. Ses connaissances sont collectées à partir de nombreuses sources, notamment des ressources créées par des experts, du crowdsourcing et des jeux avec un objectif. Il est conçu pour représenter les connaissances générales impliquées dans la compréhension du langage, en améliorant les applications en langage naturel en permettant à l'application de mieux comprendre le sens des mots que les gens utilisent **[46]**. Dans ConceptNet, les types de relations sont aussi très variés par exemple on trouve :

- **prerequisiteOf (condition préalable) :** « wake up in the morning » prerequisiteOf « eat breakfast » ;
- **generalisation (généralisation) :** « wake up in the morning » generalisation « wake up » ;
- **usedFor (utilisé pour) :** « ki tchen table » usedFor « eat breakfast »
- **location Of (localisation de) :** « in house » location Of « ki tchen table ».

Cette ressource lexicale contient plus de 21 millions d'arêtes et plus de 8 millions de nœuds. Son vocabulaire anglais contient environ 1 500 000 nœuds et il existe 83 langues dans lesquelles il contient au moins 10 000 nœud. ConceptNet vise à donner aux ordinateurs l'accès aux connaissances de bon sens, le genre d'informations que les gens ordinaires connaissent mais laissent généralement sous silence. Il représente un réseau sémantique qui représente des choses que les ordinateurs devraient savoir sur le monde, en particulier dans le but de comprendre des textes écrits par des personnes. Ses « concepts » sont représentés à l'aide de mots et d'expressions de plusieurs langues naturelles différentes. Contrairement à des projets similaires, il n'est pas limité à une seule langue telle que l'anglais. Il exprime plus de 13 millions de liens entre ces concepts et met l'ensemble des données à disposition sous une licence Creative Commons **[46]**.

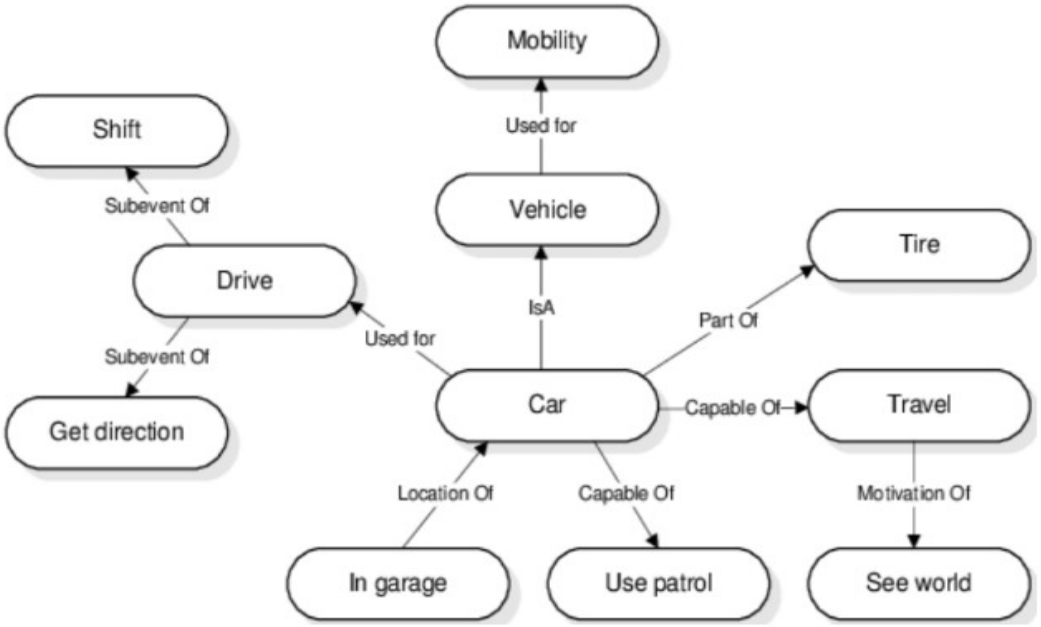


Figure 13 : Structure de Concepts liés à la voiture dans ConceptNet **[46]**

4.5.3. Les critiques De ConceptNet :

- Connaissances acquises automatiquement en intégrant d'autres bases.
- Très grande taille.
- Expressivité limité (réseau sémantique).
- Inférence limitée (analogie).
- Incomplet et hétérogène.

4.5.4. Exemples Sur ConceptNet :

À titre d 'exemples de concepts de sens commun, les auteurs citent : wake up in the morning (se lever le matin) , eat breakfast (prendre le petit déjeuner), full stomach (ventre plein), wake up (se réveiller) , in house (dans la maison) , kitchen table (tab le de cuisine), etc.

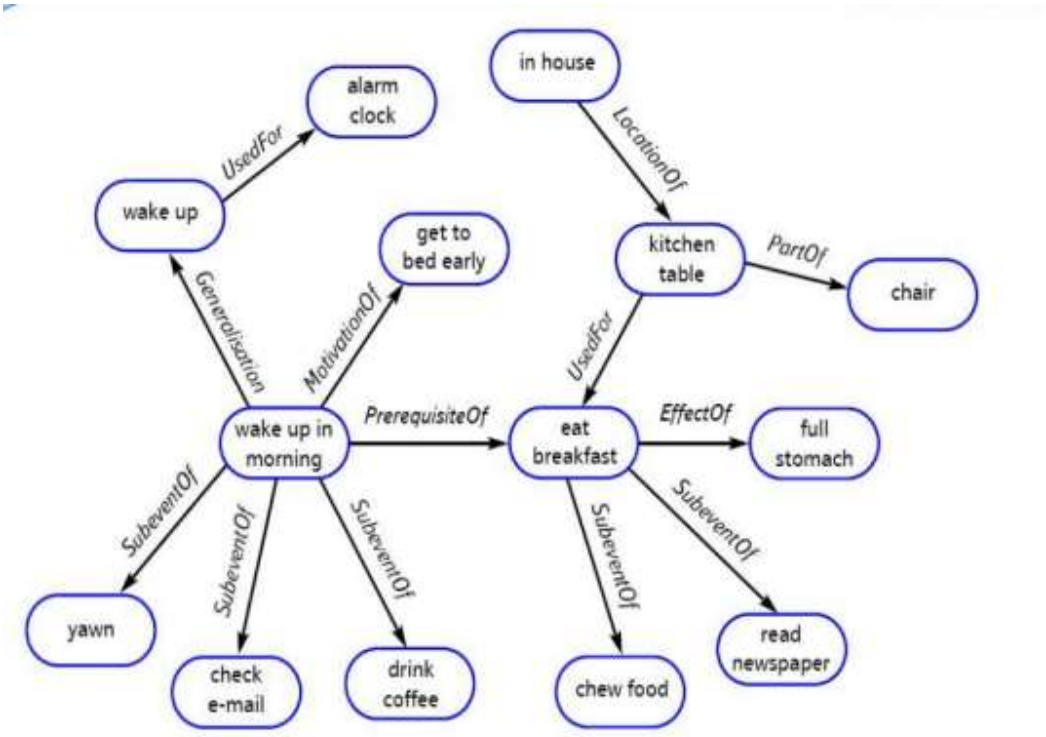


Figure 14 : un sous ensemble de conceptnet **[46]**

4.5.5. Développement ConceptNet :

ConceptNet a été développé dans le cadre de l'Open Mind projet Common Sense, un projet pour collecter les choses que les ordinateurs doivent savoir pour comprendre de quoi les gens parlent, qui s'est ensuite transformé en un projet international multi-hébergé appelé Common Sense Initiative informatique. Le développement de ConceptNet a lieu en tant que projet open source de Luminoso Technologies Le code qui construit et alimente ConceptNet est disponible sur GitHub[8] Le développement de ConceptNet 5 est dirigé par Robyn Speer, co-fondatrice De Luminoso. ConceptNet fournit une combinaison de fonctionnalités non disponibles dans d'autres projets de représentation des connaissances **[45]**.

- Ses concepts sont liés à des mots et des phrases en langage naturel qui peuvent également être trouvés en texte libre.
- Il comprend non seulement les définitions et les relations lexicales, mais aussi les associations de bon sens que les gens ordinaires font entre ces concepts.
- Les concepts ne sont pas limités à une seule langue ; ils peuvent provenir de n'importe quelle langue écrite.

[8] https://github.com/commonsense/conceptnet5

- Il intègre des connaissances provenant de sources avec des niveaux de granularité et des registres de formalité variables et les rend disponibles à travers une représentation commune. La figure suivante représente un résumé de conceptnet

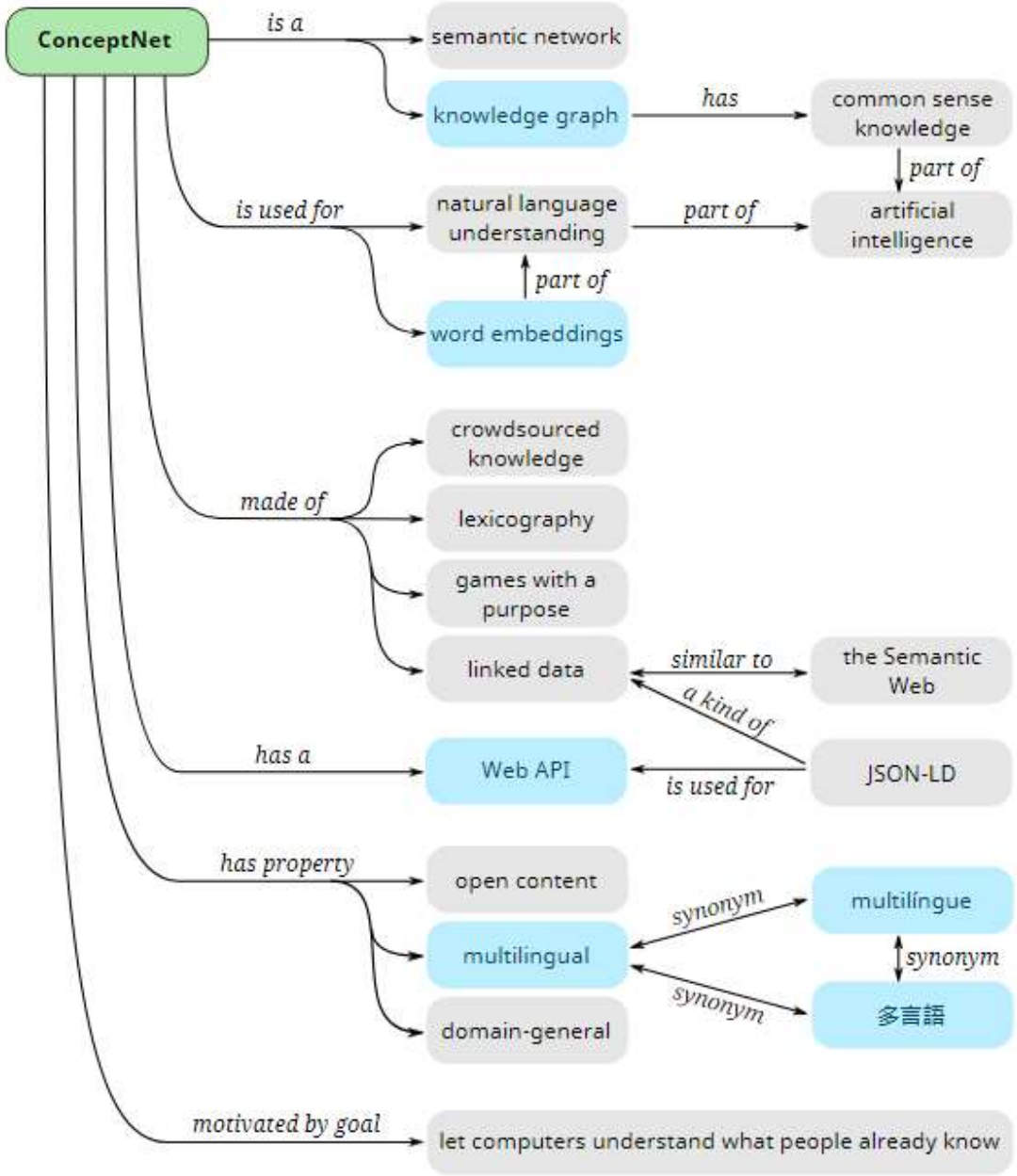


Figure 15 : résumé du contenu de ConceptNet **[46]**

4.5.6. Application du ConceptNet dans le TAL :

ConceptNet est utilisé pour créer des imbrications de mots - des représentations de la signification des mots sous forme de vecteurs, similaires à word2vec, GloVe ou fastText, mais en mieux. Ces incorporations de mots sont gratuites,

multilingues, alignées dans toutes les langues et conçues pour éviter de représenter des stéréotypes nuisibles. Leur performance sur la similitude des mots, dans et entre les langues, s'est avérée être à la pointe de la technologie à SemEval 2017 **[47]**.

4.6. Ontologie:

Une ontologie est une spécification explicite d'une conceptualisation. Le terme « conceptualisation » fait référence à un système de concepts. L'expression « spécification :explicite» signifie que la conceptualisation est représentée dans un langage (langue naturelle ou,langage :formel). Le terme «ontologie» est un emprunt à la philosophie. Il désigne (Petit ROBERT, 1979) : la partie de la métaphysique qui s'applique à l'être en tant qu'être, indépendamment de ses déterminations **[47]**

4.6.1. Concept :

Un concept est une entité structurée. Il peut se définir comme une entité composée de trois éléments distincts :

a- Le(s) terme(s) exprimant le concept en langue.

b- La signification du concept, appelée également « notion » ou « intension » du concept.

c- Le(s) objet(s) dénotés par le concept, appelé(s) également « réalisation » ou « extension » du concept.

4.6.2. Les types d'ontologies :

- **Les ontologies du domaine :** elles sont appelées de la sorte parce qu'elles expriment des conceptualisations spécifiques à un domaine. Elles rendent compte du vocabulaire d'un domaine spécifique au travers de concepts et de relations qui modélisent les principales activités, les théories et les principes de base du domaine en question. La plupart des ontologies existantes sont des ontologies du domaine, elles sont réutilisables pour plusieurs applications concernant le domaine pour lequel elles ont été créées car elles ont été conçues de façon aussi indépendante que possible du type de manipulations qui vont être opérées sur/ces :connaissances [47].
- **Les ontologies applicatives (ou ontologies d'application) :** Sont les ontologies les plus spécifiques, elles contiennent les connaissances requises pour une application particulière et ne sont pas réutilisables. Elles peuvent en outre inclure une ontologie de domaine.
- **Les ontologies génériques ou ontologies de haut niveau (upper ontology) :** Elles expriment des conceptualisations valables dans différents domaines de valeur relativement générale 8 comme les notions d'objets, de propriété, de valeur, d'état, ou encore des concepts de temps, d'espace d'événements, elles sont prévues pour être utilisées dans des situations diverses, et pour servir une large communauté d'utilisateurs [47].
- **Les ontologies de représentation :** Ce type d'ontologies regroupe les concepts utilisés pour formaliser les connaissances. Parmi les ontologies de représentation, on trouve des ontologies qui vont décrire les notions utilisées dans toutes les ontologies pour spécifier les connaissances, telles que les substances, les concepts, les relations etc. Par exemple, la « Frame-Ontology » est une ontologie de représentation. Elle définit de

manière formelle les concepts utilisés principalement dans les langages à base de frames : classes, sous-classes, attributs, valeurs, relations et axiomes. Les ontologies de représentation sont indépendantes des différents domaines de connaissances, puisqu'elles décrivent des primitives cognitives communes aux différents domaines [47].

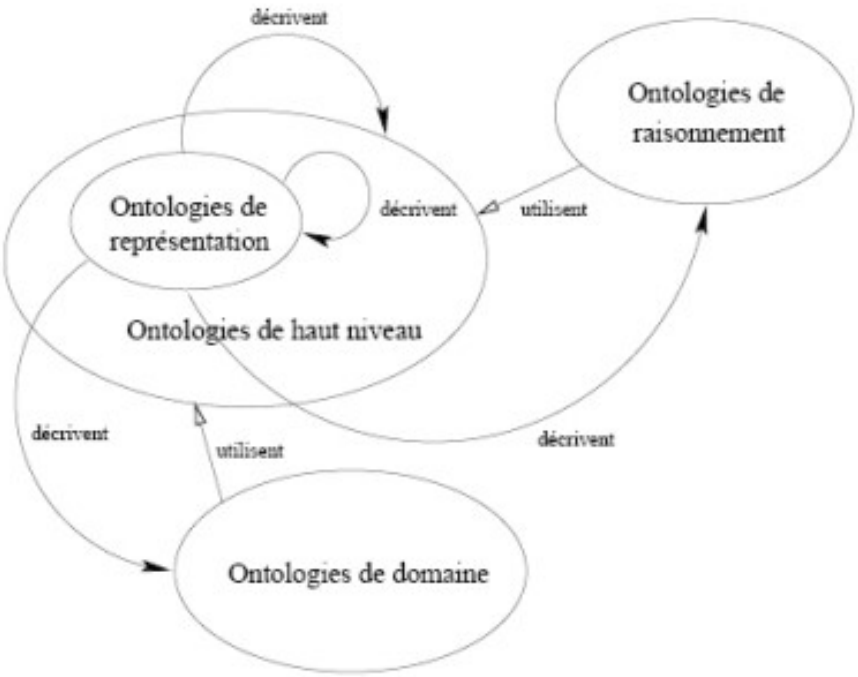


Figure 16 : Les diffèrent type d'ontologie [47]

Chapitre 5

L'encodage des ressources lexicales sémantiques

5. Introduction

Dans ce chapitre l'objectif est d'avoir une idée sur les languages necessaires (comme RDF/RDFs et OWL) pour le dévloppement des RLs sémantiques comme wordnet, thesaurus, onologie, framenet..ect.

5.1. RDF/RDFS

Comme vous pouvez le constaté que xml ne permet pas la representation sémantique et primitive [32]. Pour cela le rdf a fait son apparition qui a pour objectif de décrire les ressources (exemple wordnet) en exploitant les métadonnées. Il posséde une syntaxe XML ((mais ce n'est pas l'unique syntaxe) et il est représenté oar un triplet :

- **Ressources :** tous qui peut étre identifier par un URI (chaque concepte ou synset a un uri (ID)) **[32].**
- **Descreption :** attribut, caractéristiques et relations entre ressources (synonymie, antonymie …ect) **[32].**
- **Framework :** designe le language dans son ensemble. **(sujet predicat objet) :** RDF permet de de décomposé les descripteurs en triplets **[32]**.

Dans le cas de representation du wordnet il est appelé Modèle de graphe :

- Sommet de départ (sujet)
- Sommet d'arrivé (objet)

Comme le montre l'exemple dans la figure suivante synset {wagon, wagon} a une relation is-a avec synset 2 {wheeled, vehicule}.

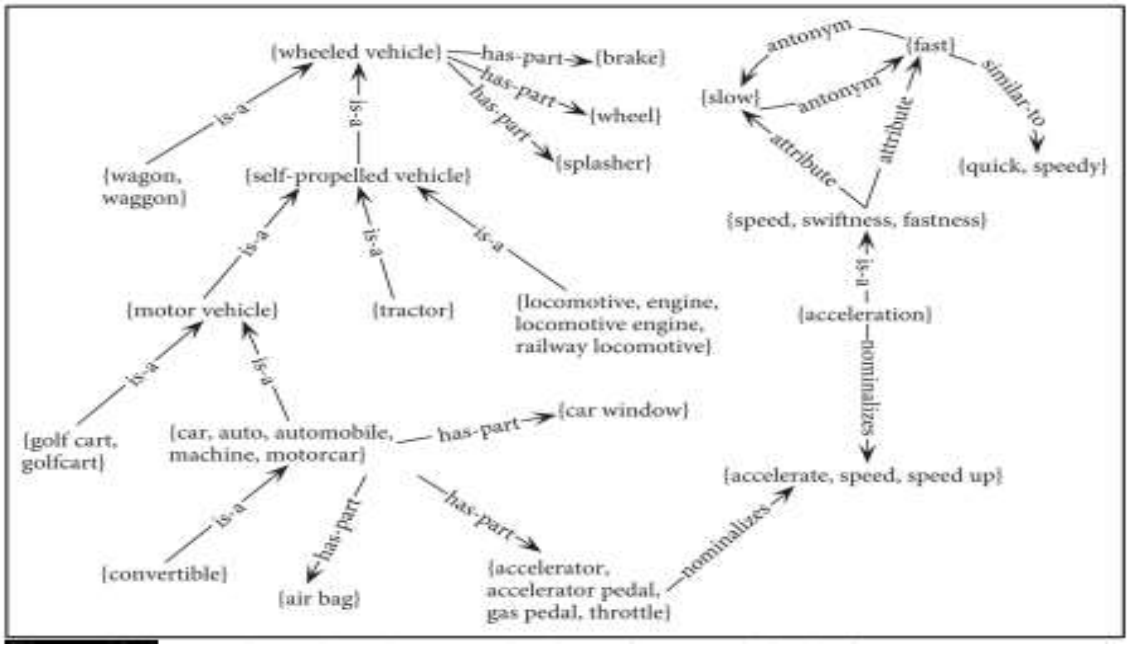


Figure 17 : exemple de synsets et leur relations dans wordnet **[27]**

- Le sujet ou l'objet représenté par un URI sera représenté par cercle et par un literau par un rectangle **[27]**.
- Sujet (sommet du graphe) sont toujours etiqueté par des uri mais object peut etre par letiraux mais relation obliger par des uris**[27].**
- Modèle ouvert permet à quiconque de faire des relations et descreptions

Pourquoi les URIs : Regroupe les concepts (synsets) qui partagent méme homonyms et holonyms et ainsi de construire un réseau sémantique et aussi pour son utilisation dans le web **[32]**.

5.1.1. Modélisation avec RDF :

On peut distinguer les Primitives clés avec : 7 pour les classes, 7 pour les propriétés et une pour les instances

- Primitives des Classes :

  - rdf:Statement : la classe des triplets contenant un sujet, une propriété et un objet.

  - rdf:Property : la classe des propriétés

  - rdf:Bag, rdf:Seq et rdf:Alt : les classes des collections.

  - rdf:List : la classe des listes RDF.

  - rdf:XMLLiteral : un type de donnée,qui permet de définir une classe pour les littéraux XML.

- Primitives du propriété :

- rdf:first et rdf:rest : représentent la relation entre une liste et son premier élément (le reste des éléments).

- rdf:predicate, rdf:subject et rdf:object : ils définissent les ressources propriété, le sujet et l'objet d'une déclaration (statement).

- rdf:type : pour définir la classe d'appartenance d'une ressource.

- rdf:value : pour définir la valeur d'une propriété lorsque celle-ci est une ressource structurée (un RDF statement).

- Instance

- rdf:nil : pour décrire une liste vide.

5.1.2. Syntaxe de sérialisation :

La syntaxe de serialisation utiliser pour le codage du RL comme wordnet est le XML/RDF comme le montre l'exemple suivant :

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-
  syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/">
  <rdf:Description rdf:about="http://www.univ-
  mlv.fr/~ocure">
    <dc:title>Page de olivier Cure</dc:title>
    <dc:author>Olivier Cure</dc:author>
  </rdf:Description>
</rdf:RDF>
```

5.1.3. Rdfs par rapport à rdf :

les points ajouter par RDF schema par rapport à RDF sont :

- Hierarchie de classe
- Notion de classes et subclasses
- Domaine range (precisant le sujet et le prédicat)

```
<rdf:RDF xml:base="http://inria.fr/2005/humans.rdfs"
 xmlns:rdf ="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
 <rdfs:Class rdf:ID="Man">
  <rdfs:subClassOf rdf:resource="#Person"/>
  <rdfs:subClassOf rdf:resource="#Male"/>
 </rdfs:Class>
</rdf:RDF>
```

Figure 18 : exemple de representation des propreités RDFs (class (man) is a subclass of person and male) [33]

```
<rdf:RDF xml:base="http://inria.fr/2005/humans.rdfs"
 xmlns:rdf ="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
 <rdf:Property rdf:ID="hasMother">
  <rdfs:subPropertyOf rdf:resource="#hasParent"/>
  <rdfs:domain rdf:resource="#Human"/>
  <rdfs:range  rdf:resource="#Woman"/>
 </rdf:Property>
</rdf:RDF>
```

Figure 19 : exemple de representation des properties RDFs (property (hasParent) + subproperty (hasMother) [33]

5.2. Ontology Web Language (OWL) :

OWL est un langage de définition d'ontologie qui permet de formaliser la sémantique des schémas. Comme vous le savez que RDFS est un langage pour définir des schémas et des vocabulaires pour RDF. RDS permet de définir des ontologies simples et OWL Fournit des primitives supplémentaires pour des ontologies plus complexes et des définitions plus riches pour les classes et propriétés. Il permet de tirer plus de conclusion et de faire plus d'inférences comparer au RDFS [47].

5.2.1. Les énoncés du OWL :

Permet de définir des classes et des propriétés grâce à des expressions logiques. Par Example une classe pourra être définit par l'union la disjonction ou l'intersection de d'autres classes. Une vue graphique des constructeurs logiques offert par le vocabulaire du OWL sont présenter dans la figure suivante :

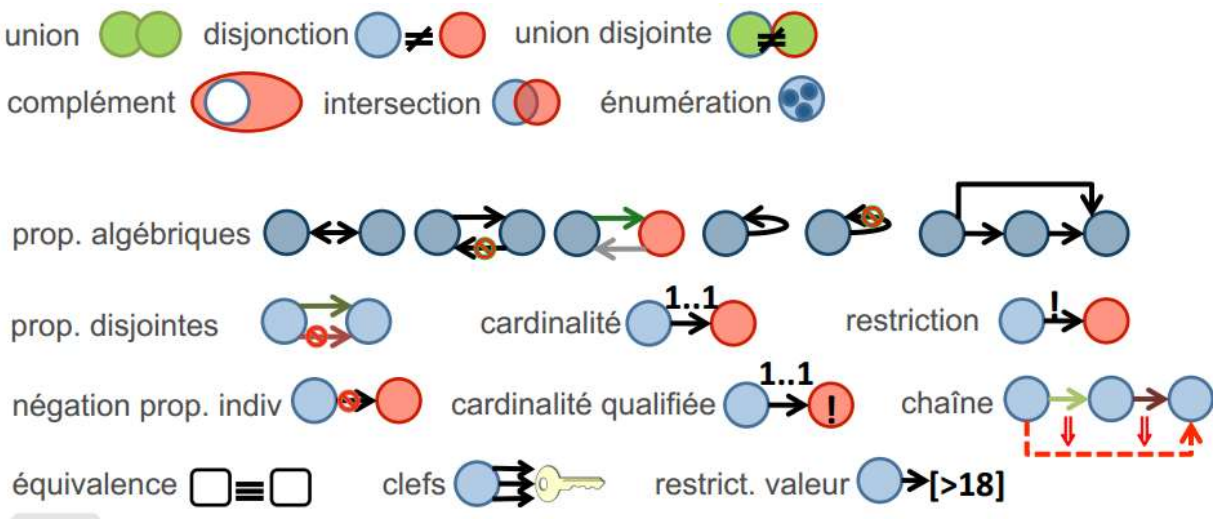


Figure 20 : les constructeurs logiques dans OWL [47]

Il est possible aussi de définir les propriétés algébriques des relations comme la transitivité et la réflexivité. Nous pouvons spécifier des restrictions sur les valeurs possibles (cardinalité des propriétés) il y'a aussi des négations de propriété, des équivalences.

5.2.2. Classe :

Une classe définit un groupe d'individus qui sont réunis parce qu'ils ont des caractéristiques similaires. Chacun de ces individus étant alors une « instance » de la classe. Une classe peut être défini de plusieurs manières [47]. Elle peut se faire directement par le nommage de cette classe. EX : une classe « Module » se déclare de la manière suivante :

<owl:Class rdf:ID="Module" />

5.2.3. Les relations entre les classes :

- **Classes énumérées :** Définir une classe en énumérant les instances de cette classe. Comme le montre l'exemple suivant la classe couleur yeux est définit par l'énumération des instances (bleu, vert, marron, noire) [47].

```
<owl:Class rdf : ID="couleurYeux">
    <owl:one Of rdf : parseType="Collection">
```

```
            <owl:Thing rdf:ID="Bleu"/>
            <owl:Thing rdf:ID="vert"/>
            <owl:Thing rdf:ID="marron"/>
            <owl:Thing rdf:ID="noire"/>
        </owl:oneOf>
    </owl:Class>
```

- **Classe union [47] :** L'ensemble des instances d'une classe est l'union des instances des classes considérer comme le montre l'exemple suivant :

```
<owl:Class rdf:ID="LegalAgent">
        <owl:equivalent Class>
            <owl:Class>
                <owl:unionOf rdf:parseType="Collection">
                    <owl:Class rdf:about="#Person"/>
                    <owl:Class rdf:about="#Group"/>
                </owl:unionOf>
            </owl:Class>
        </owl:equivalent Class>
</owl:Class>
```

- **Héritage [47] :** Il existe dans toute ontologie OWL une superclasse, nommée Thing, dont toutes les autres classes sont des sous-classes. Ceci nous amène directement au concept d'héritage, disponible à l'aide de la propriété subClassOf :

```
<owl:Class rdf:ID="Homme">
        <rdfs:subClassOf rdf:resource="#Humain" />
</owl:Class>
```

- **Classe définie par intersection :** Les classes définies par intersection permet de définir une classe comme étant l'intersection de d'autre classes c a dire une instance est de la classe man si elle est instance de la classe person et de la classe male [47]. Toute ressource commune aux classes est aussi dans la classe intersection.

```
<owl:Class rdf:ID="Man">
  <owl:equivalentClass>
    <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
```

```
            <owl:Class rdf:about="#Person"/>
            <owl:Class rdf:about="#Male"/>
          </owl:intersectionOf>
    </owl:Class>
  </owl:equivalentClass>
</owl:Class>
```

- **Classes définies par négation :** Comme dans l'exemple suivant on définit la classe edible comme étant le complément de la classe inedible. Les choses mangeables et non mangeables [47].

```
<owl:Classrdf:ID="Inedible">
    <owl:equivalentClass>
      <owl:Class>
        <owl:complementOf rdf:resource="#Edible"/>
      </owl:Class>
    </owl:equivalentClass>
</owl:Class>
```

- **La disjonction entre les classes :** Une ressource ne peut pas appartenir à deux classes en même temps. Dans l'exemple suivant on 'a la classe des carrés est disjointe avec la classe des cercles. Si nous exprimons qu'une ressource est instance de carré alors elle ne peut pas être une instance de cercle et réciproquement [47].

```
<owl:Class rdf:ID="Square">
    <owl:disjoint With rdf:resource="#Circle"/>
</owl:Class>
```

- **Disjonction de plusieurs classes :** Nous pouvons aussi indiquer qu'un ensemble de classes sont disjoint deux à deux. Une ressource ne peut, au plus, appartenir qu'à une seule de ces classes. Aucune ressource ne peut appartenir à deux classes en même temps [47].

```
<owl:All Disjoint Classes>
      <owl:members rdf:parseType="Collection">
              <owl:Class rdf:about="#Square"/>
              <owl:Class rdf:about="#Circle"/>
              <owl:Class rdf:about="#Triangle"/>
      </owl:members>
</owl:AllDisjointClasses>
```

- **Union avec disjonction :** Diviser une classe en une partition complète de sous classes [47].

```
<owl:Classrdf:about="#Passenger">
    <owl:disjointUnionOfrdf:parseType="Collection">
        <owl:Class rdf:about="#Adult"/>
        <owl:Class rdf:about="#Child"/>
        <owl:Class rdf:about="#Pet"/>
    </owl:disjointUnionOf>
</owl:Class>
```

5.2.4. Les instances de la classe (les individus):

La définition d'un individu consiste à énoncer un « fait », encore appelé « axiome d'individu », occurrence ou instance. Une occurrence s'exprime de la manière suivante :

```
<Humain rdf:ID="Pierre">
    <aPourPere rdf:resource="#Jacques" />
    <aPourFrere rdf:resource="#Paul" />
</Humain>
```

L'occurence écrite dans l'exemple précédent exprime l'existence d'un Humain nommé « Pierre » dont le père s'appelle « Jacques », et qu'il a un frère nommé « Paul ».

5.2.5. Propriété :

Une propriété a la capacité d'exprimer des faits au sujet de ces classes et de leurs instances. OWL fait la distinction entre trois types différents de propriétés :

- owl:ObjectProperty sont des relations entre des ressources. Pour définir les relations et les associations qui existent entre les classes en **reliant des instances à d'autres instances.**
- owl:DatatypeProperty : ont des valeurs littérales (typées). Permet de définir les types et les attributs de chaque classe ou relation en **reliant des individus à des valeurs de données.**
- owl:AnnotationProperty sont ignorées dans les inférences et utilisées pour documenter ou pour des extensions .

Une propriété d'objet est une instance de la classe owl:ObjectProperty, une propriété de type de donnée étant une instance de la classe owl:DatatypeProperty. Ces deux classes sont elles-mêmes sous-classes de la classe RDF rdf:Property.

5.2.6. La signature d'une ontologie :

- **owl:ObjectProperty :** Dans l'exemple ci-dessous, on apprend que la propriété enseigne a pour domaine la classe Enseignant (classe de départ) et pour image ou range la classe module (classe d'arriver). Elle relie des instances de la classe enseignant à des instances de la classe module.

```
<owl:ObjectProperty rdf:ID="enseigne">
    <rdfs:domain rdf:resource="#enseignant" />
    <rdfs:range rdf:resource="#module" />
</owl:ObjectProperty>
```

- **owl:DatatypeProperty :** Dans le cas d'une propriété de type de donnée (**DatatypeProperty)**, le range de la propriété est le type de donnée (attribut) et le domain pour le nom de l'attribut [47]. Par exemple, on peut définir la propriété de type de données anneeDeNaissance :

```
<owl:Class rdf:ID="dateDeNaissance" />
    <owl:DatatypeProperty rdf:ID="anneeDeNaissance">
    <rdfs:domain rdf:resource="#dateDeNaissance" />
    <rdfs:range rdf:resource="&xsd;positiveInteger"/>
</owl:DatatypeProperty>
```

Dans ce cas, anneDeNaissance fait correspondre aux instances de la classe dedateDeNaissance des entiers positifs [47].

5.2.7. La relation entre les propriétés : caractérisation des propriétés.

- **Propriétés symétriques :** Une propriété si elle existe entre deux ressources elle existe entre eux aussi dans l'autre sens. (ex. marié_avec) x R y ⇒ y R x [47].

```
< owl : symmetricProperty rdf :ID= "hasSpouse" />
```

- **Propriété asymétrique :** Si une propriété existe entre deux ressources elle ne peut pas exister en sens inverse ex : la relation hasChild (à pour enfant) est asymétrique. La relation entre les parents et les enfants est asymétrique par contre la relation entre les époux est symétrique. x R y ⇒ ¬ y R x [47].

```
< owl : AsymmetricProperty rdf :ID= "haschild" />
```

- **Propriété inverses :** Deux relations qui existent simultanément en sens inverse (ex. parent_de / enfant_de) x R1 y ⇔ y R2 x

```
< rdf : Property rdf :ID= "haschild" />
        < owl : inverseof rdf : resource = "#hasParent" />
</rdf:Property>
```

- **Propriété transitive :** les amis de mes amis sont mes amis. Une relation qui se propage de proche en proche (ex. Tom ami Jim ami Jules) x R y & y R z ⇒ x R z [47]

```
< owl : TransitiveProperty rdf :ID= “hasfriend” />
```

- **Propriétés disjointes :** Des relations qui ne peuvent pas exister en même temps sur le même sujet et le même objet [47].

```
<owl:ObjectProperty rdf:about="#hasSon">
    <owl:propertyDisjointWith rdf:resource="#hasDaughter"/>
</owl:ObjectProperty>
```

- **Propriétés réflexives :** Une relation qui relie tous les individus à eux-mêmes [47].

```
<owl :ReflexiveProperty rdf :about= « hasRelative”/>
```

- **Propriétés irréflexives :** Une relation qui ne relie aucun individu à lui-même. Ex : la propriété hasParent suivante est irréflexive parce que on ne peut pas être le parent de soi-même [47].

```
<owl : IrreflexiveProperty rdf :about= « #hasParent”/>
```

- **Propriétés chaînées :** Définir une propriété par une chaine de propriétés. Des relations qui mises bout à bout impliquent une autre relation (ex. parent + frère = oncle) x P y & y Q z ⇒ x R z [47].

```
<owl:ObjectProperty rdf:ID="uncle">
    <owl:propertyChainAxiom rdf:parseType="Collection">
        <owl:ObjectProperty rdf:about="#parent"/>
        <owl:ObjectProperty rdf:about="#brother"/>
    </owl:propertyChainAxiom>
</owl:ObjectProperty>
```

Cela implique que la propriété uncle peut être déduite à partir d’une occurrence de la propriété parent suivi d’une occurrence de la propriété brother. On va inférer la propriété uncle à chaque fois qu’on voit les propriétés Brother suivi de la propriété parent.la chaine des propriétés peut avoir la longueur quelconque [47].

- **Propriété fonctionnelle :** Une relation pour laquelle une ressource ne peut avoir qu’une valeur (ex. naissance) x R y & x R z ⇒y = z

```
<owl:FunctionalPropertyrdf:ID="birthDate" />
```

- **Propriété inverses fonctionnelles :** Une relation pour laquelle une même valeur implique la même ressource (ex. NSS). Deux personnes ont le même numéro de sécurité social alors on déduit qu'ils sont identiques. x R y & z R y ⇒x = z **[47].**

<owl:InverseFunctionalProperty rdf:ID="socialSecurityNumber" />

- **Equivalence de classes :** Il est possible de définir en OWL que deux classes sont équivalentes et ils rassemblent les mêmes ressources. Si une relation est de type human alors on peut déduire quel est de type personne et si une personne est de type person alors elle est de type human. Cela se fait grâce à la propriété prédéfinit owl :equivalentClass [47]
- **Equivalentpropreité :** ces deux types de propriétés expriment exactement la même relation.

ex:name owl:equivalentProperty my:label

- **Ressources identiques :** Deux URI qui identifient exactement la même chose. ex:Bill owl:sameAs ex:William
- **Ressources différentes :** deux URI dont on sait qu'ils identifient deux choses différentes [47].

ex:Good owl:differentFrom ex:Evil

- **Restriction des propriétés :** Une restriction de propriété permet de définir une classe par une contrainte qui porte sur les instances de classe et cette contrainte s'exprime par une restriction sur les types des valeurs possibles d'une propriété dans l'exemple suivant la classe herbivore est définie comme étant une sousclasse d'animal (les harbivores sont des animaux qui mangent des plantes). Comme vous voyer dans l'exemple suivant toutes les valeurs possibles de la propriété eats doivent être de type plante [47].

```
<owl:Class rdf:ID="Herbivore">
 <rdfs:subClassOf rdf:resource="#Animal"/>
 <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onPropertyrdf:resource="#eats" />
      <owl:allValuesFromrdf:resource="#Plant" />
    </owl:Restriction>
 </rdfs:subClassOf>
</owl:Class>
```

- **Restriction à une seule valeur pour une propriété :** La classe définie ne peut avoir qu'une seule valeur pour la propriété visée [47]

```
<owl:Class rdf:ID="Bicycle">
  <rdfs:subClassOf>
    <owl:Restriction>
    <owl:onPropertyrdf:resource="#nbWheels" />
    <owl:hasValue>2</owl:hasValue>
    </owl:Restriction>
 </rdfs:subClassOf>
</owl:Class>
```

- **Les contraintes sur les cardinalités :** contrainte sur le nombre de fois qu'une propriété peut être utilisée avec des valeurs différentes sur le même sujet : minimum, maximum, nombre exact [47].

```
<owl:Classrdf:ID="Person">
<rdfs:subClassOf>
<owl:Restriction>
<owl:onPropertyrdf:resource="#name" />
<owl:maxCardinality>1</owl:maxCardinality>
</owl:Restriction>
</rdfs:subClassOf>
</owl:Class>
```

Chapitre 6 :

Applications des RLs

6.1. Introduction

Les versions lisibles par machine des RLs ont été considérées comme une source probable d'informations à utiliser dans le traitement du langage naturel, car elles contiennent une énorme quantité de connaissances lexicales et sémantiques.

6.2. Utilisation des RLs dans la synthèse vocale :

La synthèse vocale est une technique informatique de synthèse sonore qui permet de créer de la parole artificielle à partir de n'importe quel texte. Pour obtenir ce résultat, elle s'appuie à la fois sur des techniques de traitement linguistique, notamment pour transformer le texte orthographique en une version phonétique prononçable sans ambiguïté, et sur des techniques de traitement du signal pour transformer cette version phonétique en son numérisé écoutable sur un haut parleur. Pour cela on utilise la RL « dictionnaire de prononciation » qui permet de faire le mapping entre le son et le texte. Néanmoins, l'analyse des dictionnaires en général peut être une opération très complexe et même l'extraction d'un champ, comme la prononciation, peut poser des problèmes. Plusieurs prononciations peuvent être données pour un mot-clé et le choix doit être fait. Si la prononciation varie pendant la flexion des noms et des adjectifs, le champ de prononciation reflète cette variation qui rend l'information difficile à extraire automatiquement [4].

6.3.Réduction des dimensionnalités et SÉLECTION Des FONCTIONNALITÉS

Le processus de sélection de caractéristiques utilisant une RL sémantique comme wordnet est utilisé pour découvrir des termes synonymes basés sur des références croisées. Comparant cette approche avec des méthodes statistiques telles que Chi2 et term frequency (TF). TF est une technique de sélection de caractéristiques qui utilise la présence et l'absence d'un terme dans un document pour sélectionner ses caractéristiques. Le Chi2 mesure le degré d'indépendance entre un terme et une catégorie. Il sélectionne des fonctionnalités fortement dépendantes d'une catégorie particulière.

L'approche basé sur wordnet explorera l'utilisation des synonymes et des sens des mots pour dériver un meilleur ensemble de caractéristiques pour la représentation des catégories afin d'obtenir une meilleure efficacité de catégorisation et réduire l'espace vectorielle de l'indexation **[1]**. Dans cette approche, la sélection des caractéristiques est basée sur des termes avec des sens de mots qui se chevauchent et qui coexistent dans une catégorie. La co-

occurrence de termes avec la même signature synset est utilisée comme indicateur de termes significatifs pour représenter une catégorie. Le référencement croisé se fait en vérifiant la liste des synsets nominaux et tous les sens pour la similitude dans les signatures.

6.4.La désambiguïsation du sens des mots (WSD)

Le WSD est un problème ouvert de traitement du langage naturel, qui régit le processus d'identification du sens d'un mot utilisé dans une phrase, lorsque le mot a plusieurs sens (polysémie). La solution à ce problème a un impact sur d'autres écrits liés à l'informatique, tels que l'analyse de discours, l'amélioration de la pertinence des moteurs de recherche, la résolution des anaphores, la cohérence, l'inférence, etc **[43]**.

6.5. Systèmes de dialogue

Dans l'histoire de l'IA, la principale mesure de l'intelligence a été linguistique, à savoir le test de Turing : un système de dialogue, répondant à la saisie de texte d'un utilisateur, peut-il fonctionner si naturellement que nous ne pouvons pas le distinguer d'une réponse générée par l'homme ? En revanche, les systèmes de dialogue commercial d'aujourd'hui sont très limités, mais remplissent toujours des fonctions utiles dans des domaines étroitement définis, comme nous le voyons ici :

S : Comment puis-je vous aider ?

U : Quand est-ce que Saving Private Ryan joue ?

S : Dans quel théâtre ?

U : Le théâtre Paramount.

S : Saving Private Ryan ne joue pas au théâtre Paramount, mais ça passe au Madison Theatre à 15h00, 17h30, 20h00 et 10h30.

Vous ne pouviez pas demander à ce système de fournir des instructions de conduite ou des détails sur les restaurants à proximité à moins que les informations requises n'aient déjà été stockées et que des paires de questions-réponses appropriées aient été intégrées dans le système de traitement de la langue. Observez que ce système semble comprendre les objectifs de l'utilisateur : l'utilisateur demande quand un film est diffusé et le système détermine correctement à partir de là que l'utilisateur veut voir le film. Cette inférence semble si évidente que vous n'avez probablement pas remarqué qu'elle a été faite, pourtant un système de langage naturel doit être doté de cette capacité afin d'interagir naturellement. Sans cela, lorsqu'on lui demande Savez-vous quand Saving Private Ryan joue ?, un système pourrait répondre de manière inutile par Oui. Cependant, les développeurs de systèmes de dialogue commercial utilisent des hypothèses contextuelles et une logique commerciale pour garantir que les différentes manières dont un utilisateur peut exprimer des demandes ou fournir des informations sont traitées d'une manière qui a du sens pour l'application particulière. Donc, si vous tapez Quand est ..., ou Je veux savoir quand ..., ou Pouvez-vous

me dire quand ..., des règles simples (RL)donneront toujours des temps de projection. Cela suffit pour que le système fournisse un service utile **[41]**.

6.6.Extraction des entités nommées

Les entités nommées sont des syntagmes nominaux définis qui font référence à des types spécifiques d'individus, tels que des organisations, des personnes, des dates, etc. le tableau suivant répertorie certains des types d'éléments les plus couramment utilisés.

Types entités nommés	Exemple
ORGANIZATION	Georgia-Pacific Corp., WHO
PERSON	Eddy Bonte, President Obama
LOCATION	Murray River, Mount Everest
DATE	June, 2008-06-29
TIME	two fifty a m, 1:30 p.m.
MONEY	175 million Canadian Dollars, GBP 10.40
PERCENT	twenty pct, 18.75 %
FACILITY	Washington Monument, Stonehenge
GPE	South East Asia, Midlothian

Tableau 10 : Types d'entités nommées couramment utilisés [42]

L'objectif d'un système de reconnaissance d'entités nommées est d'identifier toutes les mentions textuelles des entités nommées. Cela permet l'identification des relations dans l'extraction d'informations, elle peut également contribuer à d'autres tâches. Par exemple, dans Question Answering (QA), nous essayons d'améliorer la précision de la recherche d'informations en récupérant non pas des pages entières, mais uniquement les parties qui contiennent une réponse à la question de l'utilisateur. La plupart des systèmes d'assurance qualité prennent les documents renvoyés par la RI standard, puis tentent d'isoler l'extrait de texte minimal dans le document contenant la réponse. La question qui se pose Comment procédons-nous pour identifier les entités nommées ? Une option serait de rechercher chaque mot dans une liste de noms appropriée. Par exemple, dans le cas des localisations, nous pourrions utiliser un index géographique, ou un dictionnaire géographique, comme l'Alexandria Gazetteer ou le Getty Gazetteer comme indiqué dans 5.1.

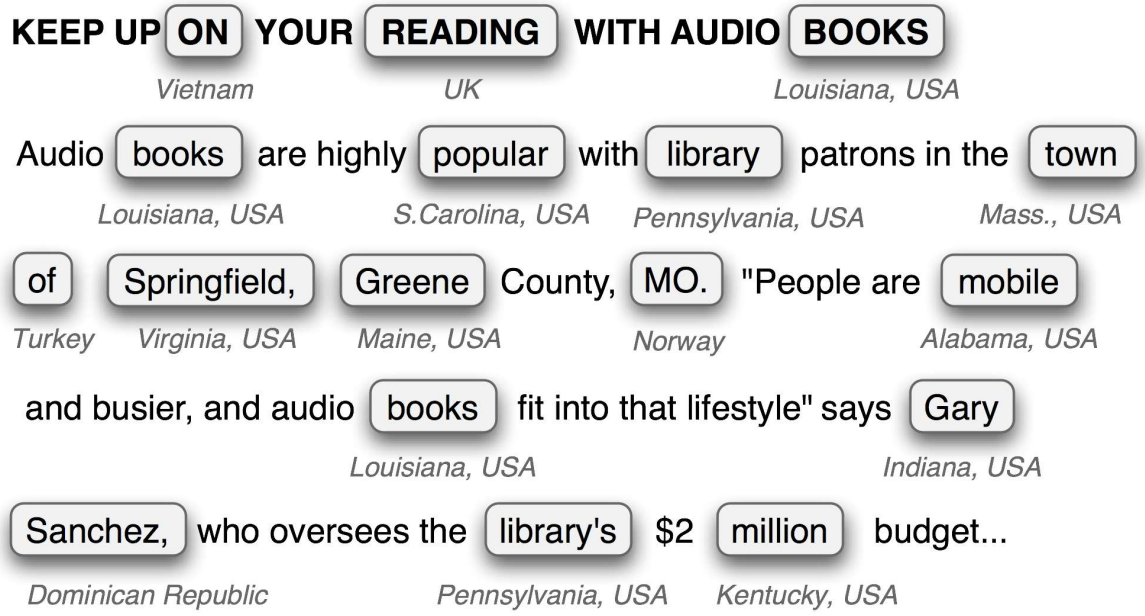


Figure 21 : Détection d'emplacement par recherche simple pour un article d'actualité : La recherche de chaque mot dans un dictionaire géographique [45].

6.7.Traduction automatique (TA) :

La TA est un domaine de recherche qui a plusieurs difficultés comme :

- Domaine du texte ex: artistique ou littéraire : Variation morphologique par example you peut étre tu ou vous .
- la taille des phrases : la longueur de la phrase à traduire joue un role important dans la qualité des résultats de la traduction.
- La typologie : Deux langues peuvent ne pas avoir les mêmes types de variations morphologiques et syntaxiques ce qui influence le résultat final.
- Il y'a certains concepts qui peuvent carrément ne pas avoir un mot associé dans la langue cible. Ex la traduction de la langue anglaise vers chinoise est difficile par rapport à la traduction de la langue anglaise vers française.
- L'ordre des mots : il y'a des langues où l'adjectif est toujour avant le nom et il y'a d'autre l'inverse Ex : français : maison bleu / anglais : blue house.
- Le lexique est varié surtout pour les mots sémantiquement ambigus
- une phrase peut avoir plusieurs traductions possibles

6.7.1. Architecture générale de la traduction automatique :

Chaque méthode de traduction suit les étapes illustrées dans la figure suivante :

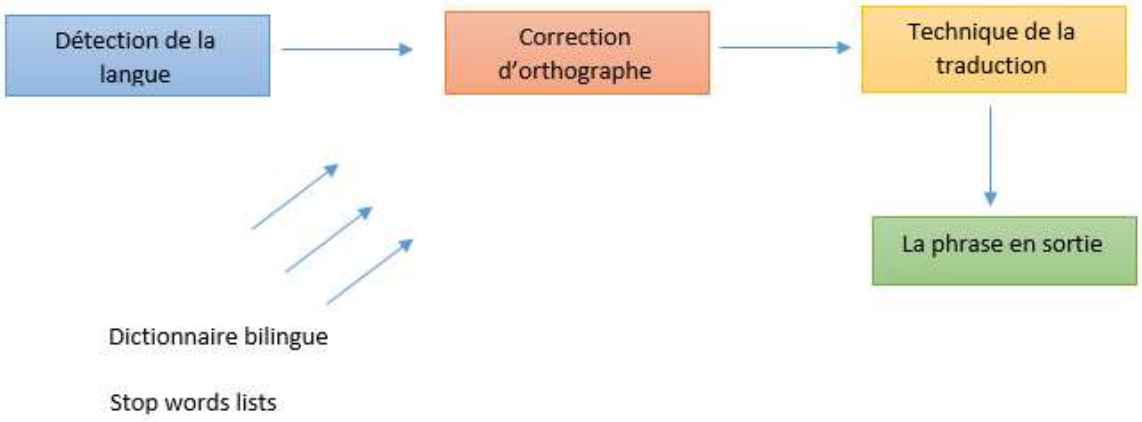


Figure 22 : architecture générale de la traduction automatique [49]

- **Détection de la langue** : pour détecter la langue d'une phrase plusieurs méthodes peuvent être utiliser comme :

1- Liste des mots vide : chaque liste des mots vide est spécifique à une langue et chaque texte doit impérativement contenir du mot vide et dans cette situation la méthode la plus simple est de comparer les mots de la phrase avec la liste des mots vide de chaque langue afin de détecter la langue.

2- Profil : chaque langue a un profile XML Utilisant. Pour cela on peut utiliser la méthode n-gram profil avec des méthode probabiliste (bayes) afin de Généré des profils à partir du wikepedia xml pour enfin détecter la langue.

3- langdetect : un API qui permet de détecter 49 languages.

6.7.2. Les techniques de traduction « langue-pivot »:

L'architecture générale des techniques de la traduction automatique languistique est illustré dans la figure suivante :

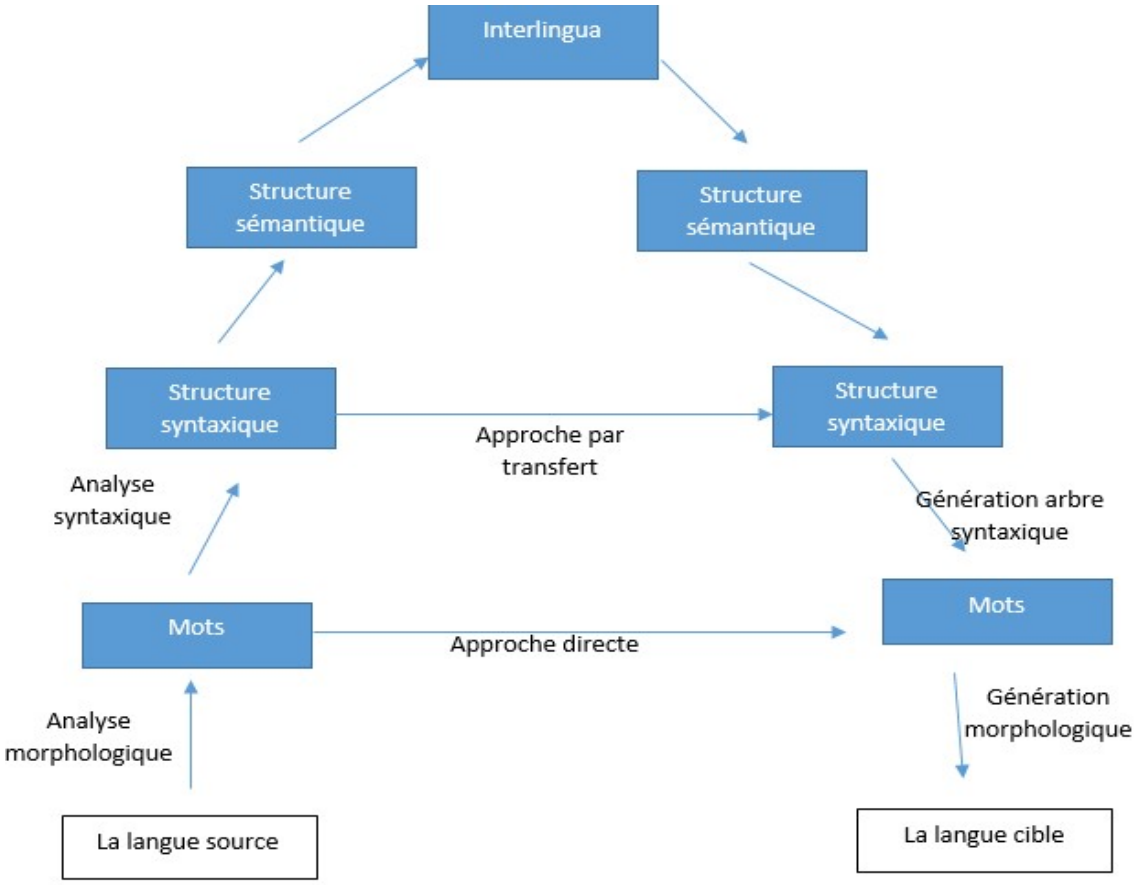


Figure 23 : architecture générale des techniques linguistiques de la traduction automatique « langue-pivot » [49]

6.7.3. Approche directe :

Dans cette approche on utilise seulement des dictionnaires bilingues pour faire la traduction directe en traitement la phrase source mot à mot. On cherche directement un mot dans la langue source et on le remplace par sa traduction dans la langue cible [49]. Les différentes étapes de cette approche son détaillé par la suite avec des exemples :

- **Analyse morphologique :** une lemmatisation est appliqué sur chaque mot afin de trouver le lemme pour cela on'a besoin d'un Etiquetage morphosyntaxique sans prendre en considération le lien entre la phrase source et cible [49].
- **Traduction morphologique directe :** associé à chaque mot source un mot cible (mapping directe) à partir du dictionnaire [49].

- **ré-ordonnancement:** Après la traduction des mots et des expressions figées, des règles simples de ré-ordonnancement peuvent s'appliquer, par exemple déplacer les adjectifs après les noms lors de la traduction de l'anglais vers le français. Ces régles sont specifique pour chaque pair de langues (cible/source) [49].
- **Génération morphologique :** qui permet de supprimer les informations inutiles générer lors de l'analyse morphologique [49].

Exemple :

Entrée: Mary didn't slap the green witch

- **Étape 1: Morphologie :**

Mary DO-PAST not slap the green witch

- **Étape 2: Transfert lexical:**

Maria PAST no dar una bofetada a la verde bruja

- **Étape 3: Ré-ordonnancement local:**

Maria no dar PAST una bofetada a la bruja verde

- **Étape 4: génération Morphologique:**

Maria no abofeteó a la bruja verde

6.7.4. Domaines d'application :

- traduction approximative est suffisante ex: traduction d'une recette
- le grand fabricant d'appareils médicaux Medtronics
- Systran et paho
- Systéme d'information et de réservation de vols
- le système METEO qui traduit des bulletins météorologiques
- Manuel logiciels

6.7.5. Approche par transfert

Dans l'approche par transfert, d'abord on analyse le texte d'entrée, puis on applique des règles pour transformer la structure syntaxique de la phrase source vers une structure syntaxique de la langue cible. Ensuite à partir de cette structure on génère la phrase en langue cible (on applique l'approche directe). Donc en plus de traduire les mots (transfert lexical) on aura besoin de regles pour traduire l'arbre syntaxique (transfert lexical).

à partir des mots une analyse syntaxique est effectuer afin d'extraire arbre syntaxique (forme qui expose certain structure syntaxique de la phrase source) afin de faire le transfert vers l'abre syntaxique de la langue cible, donc on'a deux types de transfert [49].

- transfert lexical pour la traduction des mots.
- transfert syntaxique pour traduire l'abre syntaxique

Un exemple de traduction d'abre syntaxique est illustrée dans la figure suivante de l'anglais vers japonais.

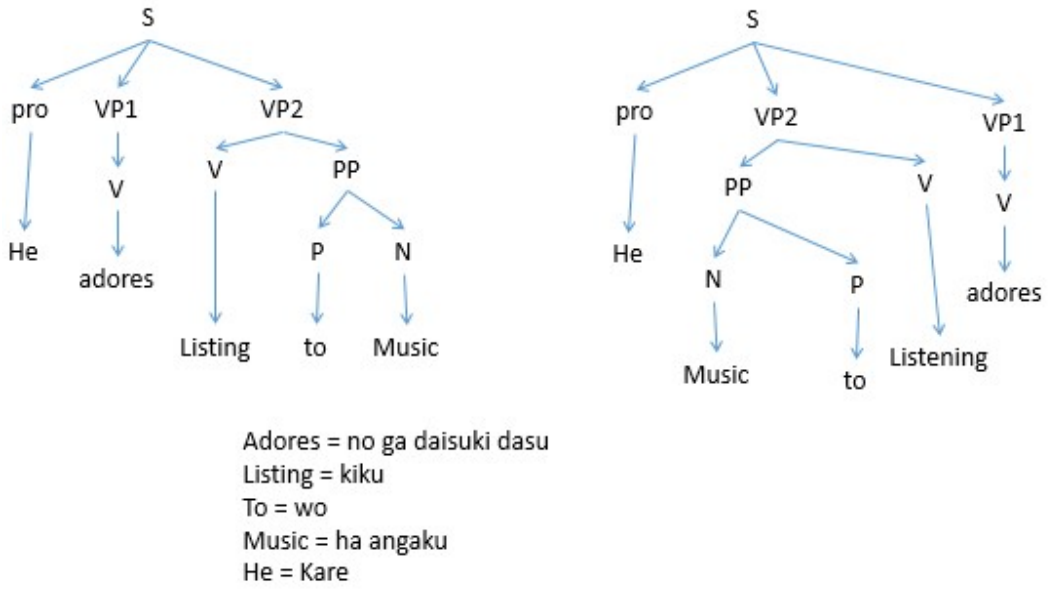


Figure 24 : traduction de l'arbre syntaxque de la langue anglaise vers la langue japonaise.

Pour faire la traduction syntaxique un ensemble de réglles doivent étre appliqué comme le montre l'Exemple suivante qui permet de traduire les régles de l'anglais vers le russe.

- Function DIRECT TRANSLATE MUCH/MANY (word) returns Russian translation
- if preceding word is how return skol'ko
- else if preceding word is as return stol'ko zhe
- else if word is much if preceding word is very return nil
- else if following word is a noun return mnogo else
- /* word is many */ if preceding word is a preposition and following word is a noun return mnogii else return mnogo

6.7.6. Approche interlingua :

L'idée derrière l'approche interlinga est de faire passer l'analyse vers une représentation indépendante de la langue. Dans les approches à langue-pivot, on analyse le texte en langue source en une représentation abstraite, appelée interlingua ou langue-pivot. A partir de cette représentation on génère ensuite un texte dans la langue cible. Le principe général de cette approche est résumé dans les points suivants :

- N'utilise plus de régles de transfert entre chaque langue

- En fait référence généralement aux concepts
- Cette méthode tend vers une représentation du sens de la phrase indépendante de la langue utilisée d'où le terme d'approche interlangue.
- L'idée est que l'approche à langue-pivot représente toutes les phrases qui veulent dire la même chose de la même manière, quelle que soit la langue d'origine.
- Utilise un étiqueteur de rôles sémantiques 3

6.7.7. **Technique statistique : phrase based model :**

Les techniques à pivot nécessitent beaucoup de connaissances linguistiques manuals et ne génèrent pas l'incertitude ce qui a fait l'apparition d'une approche statistique afin de permettre de :

- Gérer les ambiguités présentes dans le processus de traduction
- Utiliser des corpus au lieu de connaissances préétablie
- capter, d'une manière statistique, les régularités d'une langue en observant des phrases dans un corpus d'entraînement
- Se basé sur un modèle probabiliste entre la phrase source F et la phrase cible E

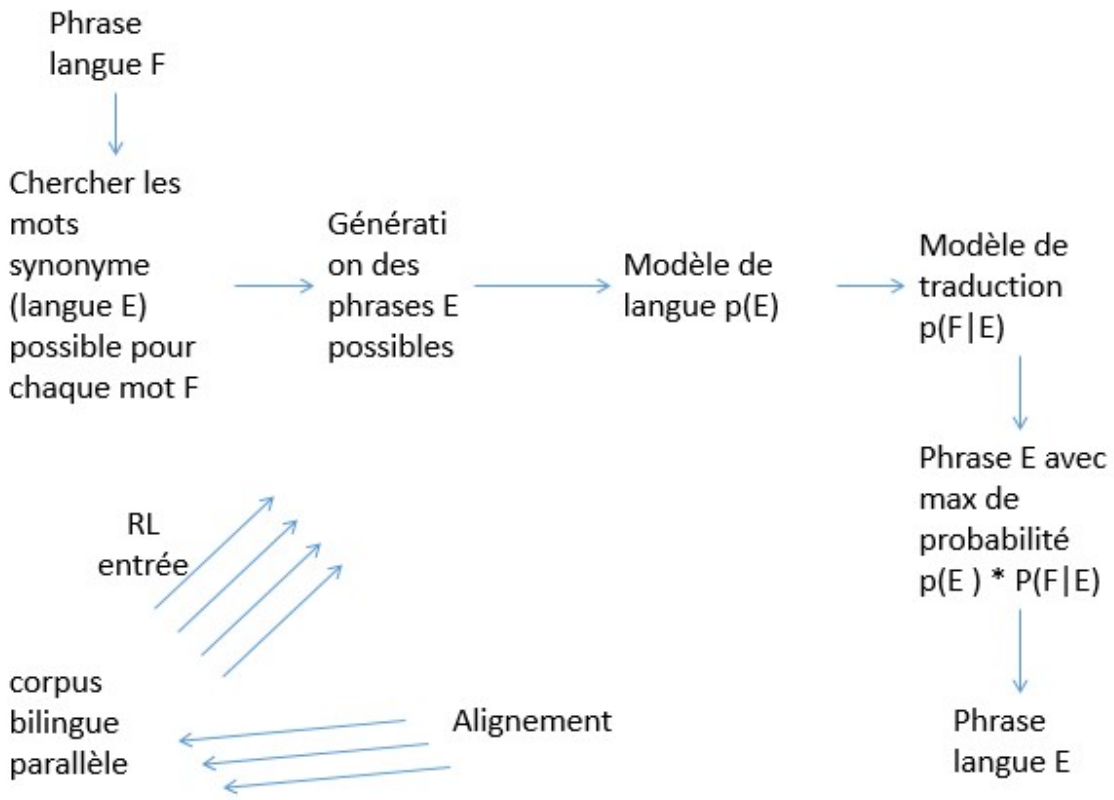


Figure 25 : architecture générale de l'approche « phrase based model » [49]

- Une approche statistique basée sur une base probabiliste entre un modèle de langue P(E) et un modèle de traduction P(F|E). à la fin un ensemble de phrases seront générées et la phrase avec maximum de probabilité sera choisis comme phrase cible.

$$\text{Argmax } P(F|E)*P(E)$$

- La traduction retournée doit étre bien construite en anglais (vérifier par le modèle de langue P(E)) et doit contenir l'information de la phrase source F(modèle traduction P(F|E)).

- **Corpus parallèle :** dans cette approche la RL corpus parallèle est utilisée. Cette RL contient paragraphe ou Les phrases sont placer à coter de ces traductions. Très précisément on parle de corpus bilingue est un corpus contenant un texte source et sa traduction dans une autre langue cible. Dans ce corpus une étape d'alignement est utilisée pour l'identification des phrases correspondante dans les deux moitiés du texte comme le montre la figure suivante **[49]**.

Exemple de corpus bilingue	
English	Japanese
How much is that **red umbrella**?	Ano **akai kasa** wa ikura desu ka.
How much is that **small camera**?	Ano **chiisai kamera** wa ikura desu ka.

Figure 26: exemple de corpus parallèle (anglais/japonais) [49]

Par la suie nous allons voir un Exemple détaillé de chaque étape de cette approche statistique :

Entrée : phrase F : The smart mouse plays violon

1- La langue cible : french

Etape 1 : rechercher dans un dictionnaire (anglais/français) les mots traduites pour chaque mot de la phrase **[49]**.

The	**Smart**	**Mouse**	**Plays**	**Violin**
La	connecté	Souris	Joue	Violon
Le	Futé	à souris	Lit	De violon
Les	Intelligent	De souris	Jeux	Lutherie
	Intelligente		Pièces	
	intelligents		Joue du violon	
		Souris s'amuse		

Etape 2 Génération des phrases : à partir du tableau précédent l'objectif est de construire les phrases possibles en français **[49]**. Ex

- **Phrase 1 :** les jeux à souris intelligents de violon
- **Phrase 2 :** le connecté souris s'amuse violon
- **Phrase 3 :** la souris intelligente joue du violon

Calculer probabilité à priori :

En appliquant le modèle de langue bigrammes : P(les|debut)*p(jeux|les)*p(à souris | jeux) * p (intelligents | à souris)* p(de violon | intelligents) **[49]**

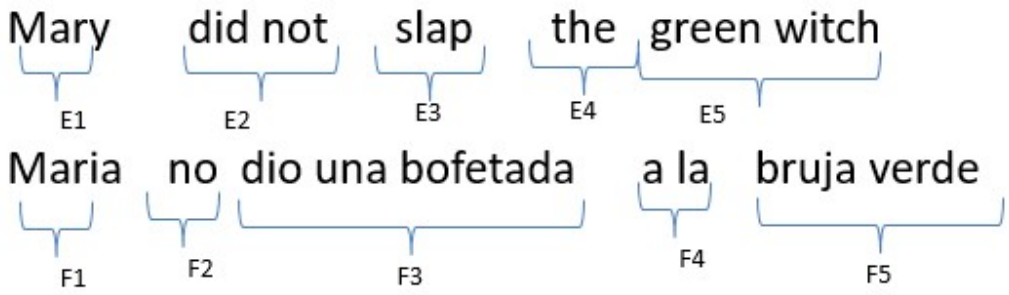
P(les | debut) = c(les|debut)/c(debut) = fréquence(les|debut)/fréquence(debut)

Exemple :

- Debut I am sam fin
- Debut sam i am fin
- Debut i do note like green eggs and ham fin
- P(i|sam)=0.5

- P(sam|debut)=0.33
- P(sam|am)=0.5
- P(am|i)=0.67
- P(do|i)=0.33
- P(fin|sam)=0.5

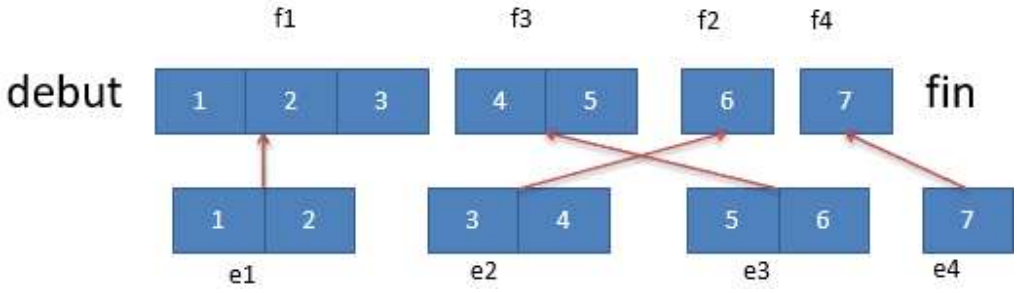
Modèle de traduction : Chaque syntagme ei permet de traduire fi avec probabilité de traduction p(fi | ei)d(ai-bi-1). Ex d'un corpus parallèle **[49].**



- P(F|E)=t(F|E)*d(ai bi-1)= t(maria | mary) *d(a1b0)*t(no|did not) * d(a2b1)*t(dio una bofetada|slap)* d(a3b2)*t(a la|the)*d(a4b3)*t(bruja verde|green witch)*d(a5b4)

Probabilité de distorsion (d) : L'ordre des syntagmes traduit fi est altéré selon la distribution du probabilité de distorsion.

- d(ai, bi-1)= $\propto^{|a_i - b_{i-1} - 1|}$
- D=1 si l'ordre ne change pas
- ∝:paramétre a fixer



- ai : la position du début du fi
- bi-1 : position du fin du mot fi-1
- alpha >1 : penaliser le reordoncement
- alpha < 1 : encourager le reordoncement
- alpha = 1 : équiprobable

Chapitre 7 :

Les exercices

7.1. Les exercices les types d'ambiguité :

Exercice 1:

Déterminez quel élément de la phrase cause l'ambiguïté

1- He gave her cat food
2- houari ne connaissait pas un invité à la fête.
3- salah aime les poèmes et les romans anglais.
4- Le juge a nié la demande du détenu parce qu'il était en colère.
5- Bonjour frère.
6- Le poulet est prêt à manger.
7- je veux que vous photographiiez une balle dans votre tête.

Exercice 2:

Construit l'arbre syntaxique pour chaque phrase :

1- la souris et le chat chassent le chien.
2- I hate annoying neighbors.
3- The mouse saw the cat on the mat with the hat.

Exercice 3 :

Mot	Root	Pourquoi	Stem	Pourquoi
Unpredictable				
Electives				
Bigger				
Uncountable				

Boy's				
Disagreement				

Exercice 4:

Décelez dans la liste ci-dessous des relations hyper/hyponimiques.

Métal, pantalon, parler, fatigué, chuchoter, vêtement, cuivre, éreinté, s'exprimer

Parmi les relations partie-tout exprimées ci-dessous, lesquelles s'appuient sur une méronymie détachable/nondétachable?

- La semelle de la chaussure
- Un morceau du mur
- Le tronc de l'arbre
- Les musiciens de l'orchestre
- Une région du globe
- Une touche du clavier
- Une part du gateau
- Les touristes du groupe

De quel type sont les antonymes suivants ? justifiez vos réponses

Gentil/méchant, élève/professeur, grave/aigu, pair/impair, prêter/emprunter

Exercice 5:

1- **Probléme d'abbréviation : lors du processus de recherche d'information un problème sera posé parce que RADAR sera pris différemment que Radio detection and ranging.**
2- **Problème :** dans les problèmes de recherche d'information on doit automatiquement calculer l'importance de chaque terme dans les textes en utilisant le terme frequency (TF). Malheureusement, les mots comme play et played sont les mêmes mais lors de l'indexation c'est deux termes vont être pris chacun indépendamment ce qui poser des problèmes et imprécision lors de calcule de similarité entre la requête et les documents. La question qui se pose comment résoudre ce problème ?
3- un grand probléme sera posé dans le processus de recherche d'information vu que am, is, was représentent le méme verbe be.

4- Comment résoudre le problème des mots composé dans le processus de recherche d'information ou d'apprentissage automatique
5- On veut savoir si nous pouvons calculer les sentiments des personnes à travers l'analyse de leur statut twitter.
6- Comment faire pour améliorer la qualité de traduction d'un traducteur automatique
7- Comment faire pour assurer une recherche d'information multilingue.
8- Comment nous pouvons savoir si deux phrases ont la même idée ou non
9- Comment faire pour assurer une recherche d'information utilisant darrija.
10- Proposer un système (avec des exemples) qui a pour objectif de remplacer un de vos proches décédé (façon de parler, de rire, ces habitudes…ect).
11- Proposer une solution non verbale dans le cadre du handicap qui permet d'interpréter les gestes des humains vers le son de la parole afin d'aider les personnes en situation de handicap à communiquer avec les gens.

7.2. Les exercices XML DTD :

Exercice 1:

Ecrire le document xml correspond à ce DTD

```
< ?xml version= 1.0 encoding= iso-8859-1 ?>
< !ELEMENT ville (mosque+, hopital*)>
< !ELEMENT mosque (nom, adresse, imame)>
< !ELEMENT nom (#PCDATA)>
< !ELEMENT adress (#PCDATA)>
< !ELEMENT imame (nom, prenom, date_naissance)>
< !ELEMENT prenom (#PCDATA)>
< !ELEMENT date_naissance (#PCDATA)>
< !ELEMENT hopital (nom, directeur*)>
< !ELEMENT directeur (#PCDATA)>
< !ATTLIST ville nomv CDATA #REQUIRED>
< !ATTLIST ville population (petite|moyenne|grande) #REQUIRED>
< !ATTLIST directeur nomdir CDATA #REQUIRED>
< !ATTLIST directeur prenomdir CDATA #IMPLIED>
< !ATTLIST prenom valeur CDATA #IMPLIED>
```

Exercice 2:

- **Corriger le document xml suivant pour qu'il soit bien formé**

- **Ecrire le document DTD correspond au document xml suivant**

- <laboratoire>
- <chef_labo number_labo= "1" > houari </chef_labo>
- <nom_labo> RI</nom_labo>
- <chef_labo number_labo= "2" > salah </chef_labo>
- <equipe>
- <membre number= "1"> ahmed <member>
- <2nom> bioinspirée </2nom>
- <membre number= "2">hamou <member>
- </laboratoire>
- <equipe>

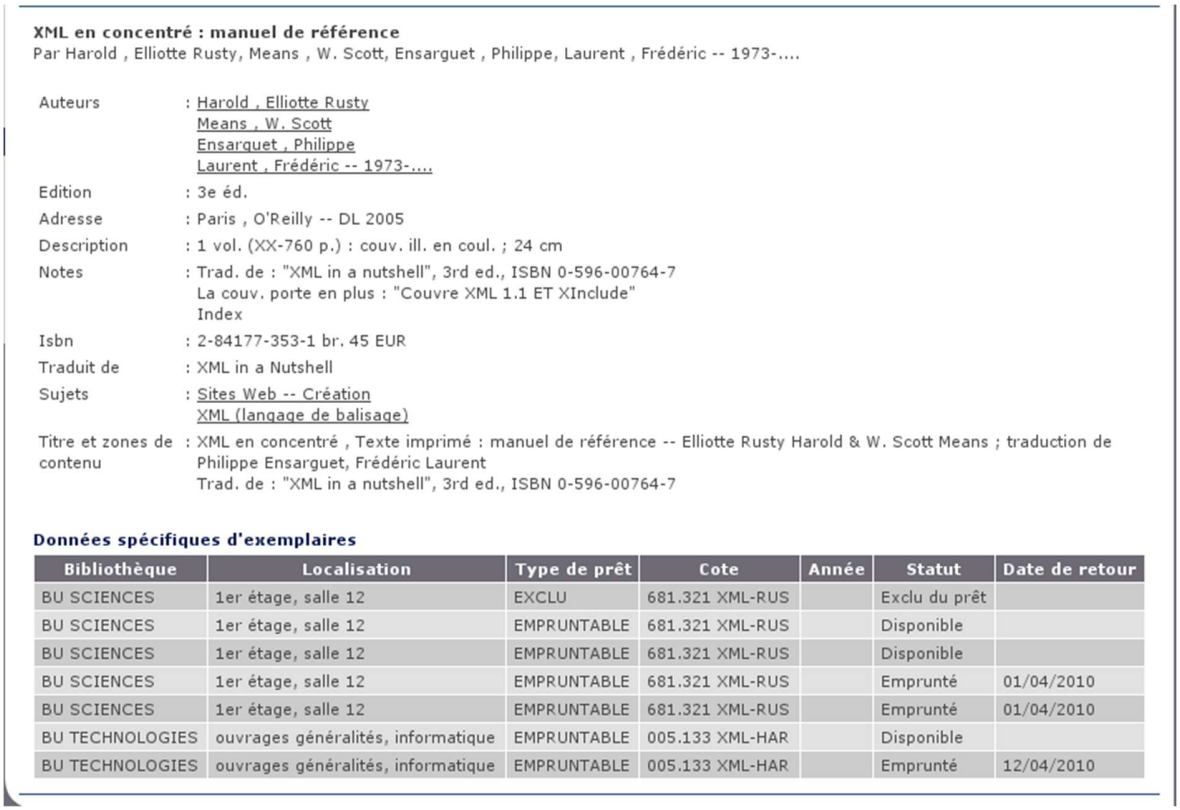
Exercice 3:

< !DOCTYPE textemath [
< !-- DTD pour décrire un texte contenant des formules mathématiques -->
< !ELEMENT textemath ((texte | formule)+)>
< !ELEMENT texte (#PCDATA)>
< !ELEMENT formule (|variable|valeur|somme|difference|produit|fraction|racine|puissance)
< !ELEMENT valeur (#PCDATA)>
< !ELEMENT somme (op1, op2)>
< !ELEMENT difference (op1, op2)>
< ! ELEMENT produit (op1, op2)>
< !ELEMENT fraction (op1, op2)>
< ! ELEMENT racine (op1)>
<!ELEMENT puissance (op1)>
< !ELEMENT op1 (Variable|valeur|formule)>
< ! ELEMENT op2 (|variable|valeur|formule)>
< !ELEMENT Variable (#PCDATA)>]>
< !ATTLIST racine ordre CDATA #IMPLIED>
< ! ATTLIST puissance exposant CDATA #REQUIRED>

Écrire un document XML valide (conforme à la DTD ci-dessus) destiné à transmettre l'énoncé suivant : Calculer la valeur de l'expression $\frac{x^4+\sqrt[3]{5}}{7\sqrt{3}}$ lorsque X prend la valeur 4

Exercice 4:

1) Concevoir un DTD servant de modèle pour des documents XML destinés à mémoriser le fonds documentaire d'une bibliothèque universitaire.



XML en concentré : manuel de référence
Par Harold , Elliotte Rusty, Means , W. Scott, Ensarguet , Philippe, Laurent , Frédéric -- 1973-....

Auteurs	: Harold , Elliotte Rusty Means , W. Scott Ensarguet , Philippe Laurent , Frédéric -- 1973-....
Edition	: 3e éd.
Adresse	: Paris , O'Reilly -- DL 2005
Description	: 1 vol. (XX-760 p.) : couv. ill. en coul. ; 24 cm
Notes	: Trad. de : "XML in a nutshell", 3rd ed., ISBN 0-596-00764-7 La couv. porte en plus : "Couvre XML 1.1 ET XInclude" Index
Isbn	: 2-84177-353-1 br. 45 EUR
Traduit de	: XML in a Nutshell
Sujets	: Sites Web -- Création XML (langage de balisage)
Titre et zones de contenu	: XML en concentré , Texte imprimé : manuel de référence -- Elliotte Rusty Harold & W. Scott Means ; traduction de Philippe Ensarguet, Frédéric Laurent Trad. de : "XML in a nutshell", 3rd ed., ISBN 0-596-00764-7

Données spécifiques d'exemplaires

Bibliothèque	Localisation	Type de prêt	Cote	Année	Statut	Date de retour
BU SCIENCES	1er étage, salle 12	EXCLU	681.321 XML-RUS		Exclu du prêt	
BU SCIENCES	1er étage, salle 12	EMPRUNTABLE	681.321 XML-RUS		Disponible	
BU SCIENCES	1er étage, salle 12	EMPRUNTABLE	681.321 XML-RUS		Disponible	
BU SCIENCES	1er étage, salle 12	EMPRUNTABLE	681.321 XML-RUS		Emprunté	01/04/2010
BU SCIENCES	1er étage, salle 12	EMPRUNTABLE	681.321 XML-RUS		Emprunté	01/04/2010
BU TECHNOLOGIES	ouvrages généralités, informatique	EMPRUNTABLE	005.133 XML-HAR		Disponible	
BU TECHNOLOGIES	ouvrages généralités, informatique	EMPRUNTABLE	005.133 XML-HAR		Emprunté	12/04/2010

7.3. Les exercices TEI, LMF :

Exercice 1 : en utilisant les standards du LMF

1- He boils a kettle of water and the kettle boils. Représenter l'extension syntaxique et morphologique du mot boil.

2- Visible, (a visible change of expression "obvious to the eye") seeable (capable of being seen; or open to easy view "a visible object"). Invisible (impossible or nearly impossible to see; imperceptible by the eye; "the invisible man"; "invisible mending"). Unseeable (impossible or nearly impossible to see; imperceptible by the eye;). Représenter l'extension morphologique et sémantique.

3- Happy (enjoying or showing or marked by joy or good fortune; "a happy smile"; "a happy marriage"). Glad (eagerly disposed to act or to be of service; "glad to help"). Unhappy

(experiencing or marked by or causing sadness or discontent; "unhappy over her departure"; "unhappy with her raise";). Représenter l'extension morphologique et sémantique.

Exercice 2: représenté les entrés lexicales suivantes en utilisant les directives du TEI.

1- Lexicography [lek-si-kog-ruh-fee] N. FS 1. (the editing or making of a dictionary) 2. (the principles and practices of dictionary making). Fr(Lexicographie) 1. Étude des mots d'une langue en vue de l'élaboration de dictionnaires. (lexicologie).

2- En 20-10-2016. Étudier [é-tu-di-é], étudie, étudies, étudions V. Infinitif 1. (Appliquer son esprit à l'étude des sciences, des lettres, etc.) 2. (Examiner attentivement.). En(study). syn survey (la recherche documentaire).

3- MCS abbre. For mouloudia club de saida (club de football en algérie).

4- **dresser** … (Theat) habilleur m, habilleuse f; (Comm: window dresser) étalagiste mf. she's a stylish ~ elle s'habille avec chic. (b) (tool) (for wood) raboteuse f; (for stone) rabotin m.

Exercice 3 : Extraire les entrées lexicales présentées par les documents suivants

(1)

```
<body>
<entry>
<form>
 <orth>brag</orth>
</form>
<gramGrp>
 <pos>vb</pos>
</gramGrp>
<form type="inflected">
 <orth>brags</orth>
<orth>bragging </orth>
 <orth>bragged</orth>
</form>
</entry></body>
```

(2)

```
<body>
<entry>
<form type=”simple”>
<orth>mining</orth>
<pro> ˈmīniNG <pro>
<syll> min·ing </syll>
</form>
<gramGrp>
<pos>n</pos>
</gramGrp>
<sens>
<def> the process or industry of obtaining coal or other minerals from a mine </def>
</body>
```

Exercice 4: Corriger le document TEI suivant et construit son DTD

<body>
<form>
<orth>étudions</orth>
<pron> é-tu-dions </pron>
<pron>[e.ty.djon]</pron>
</form>
<form >
<syll>étudies<syll>
<syll>étudie<syll>
<syll> étudiez <syll>
</form>
<per>5</per>
<gen>V</gen>
<ortho> present </ortho>

<moode>pluriel</moode>

<pose>indicatif<pose>
<number> singulier</number>
<tns> present </tns>
<form> indicatif </form>
</gramGrp>

<sens> chercher à apprendre, ou à acquérir une technique, un savoir-faire ou la connaissance de quelque chose

</sens>

<def> chercher à comprendre en examinant (étudier le comportement des concurrents, étudier la nature) **</def>**

<cit type="translation">

<def> study <def>

<quote> to communicate vocally <quote>

<quote> analyser avec attention, examiner (étudier un dossier délicat) **</quote>**

</cit>

<sens>

<lbl> the acte or process of studying </lbl>

</sens>

7.4. Les exercices stemming lemmatisation, corpus etiqueté

Exercice 1: ressource lexicale : les règles de racinisation de porter

Transformer les mots des textes suivants en leurs racines utilisant l'algorithme de porter

Texte 1:

Such as analysis can reveal features that are not easily visible from the variations in the individual genes and can lead to a picture of expression that is more biologically, transparent and accessible to interpretation.

Texte 2:

The human mind is not capable of grasping the Universe. We are like a little child entering a huge library. The walls are covered to the ceilings with books in many different tongues. The child knows that someone must have written these books. It does not know who or how. It does not understand the languages in which they are written. But the child notes a definite plan in the arrangement of the books - a mysterious order which it does not comprehend, but only dimly suspects.

- en utilisant l'algorithme de porter transformer les mots suivants en leur racines (stem) et détailler pour chaque mots les règles appliquées.

Liste des mots {Conformabli, graduation, studies, bing, warning, phishing, tanned, sensitiviti}

- Dans un système de recherche d'information (recherche adhoc) l'utilisation du stemming permet de diminuer le rappel et augmenter la précision (Vrais ou faux) avec explication.
- Quelle règle devrait être ajoutée pour correctement trouver la racine du mot pony.

Exercice 2: ressource lexical : dictionnaire pour la correction des erreurs d'orthographe et distance entre les unités lexicales.

Calculer la distance leveinshtein entre les pairs de mots suivants:

- Lexical / Lexic.
- Execution / intention.

Exercice 3 : ressource lexical : corpus étiqueté

- Calculer la probabilité de transition utilisant le bigram (MMC) modèle de markov caché et le corpus suivant et tracer le schéma de transition.
- Calculer la probabilité d'émission
- Étiqueter la phrase « la belle ferme la fenêtre » utilisant l'algorithme viterbi.

det N V det N

d0 La maman ferme la fenêtre fin

det N V Adj

d0 la ferme est jolie fin

det Adj V

d0 la belle pleure fin

le beau frére aime le foot

la belle ferme le robinet

- Calculer la probabilité de transition utilisant le trigramme MMC et le corpus étiqueté suivant.
- Calculer probabilité d'émission
- Etiqueté la phrase « the students pass test wait for teachers».

det N V Det N

d0 d1 the students pass the test fin

det N V P det N

d0 d1 The students wait for the pass fin

N V N

d0 d1 teachers test students fin

7.5. Les exercices traduction automatique

Exercice 1 : modèle de langue

Nous voulons faire la traduction de la phrase (f) « The beautiful closes the window » utilisant la technique probabiliste phrase based model avec :

1- Français comme langue cible (e).
2- Le tableau suivant regroupe liste des mots français pour chaque mot de la phrase obtenus à partir d'un corpus parallèle anglais français.

The	Beautiful	Closes	Window
Le	Beau	Ferme	Fenêtre
La	Belle	Termine	Guichet
Les	Jolie	Se ferme	Vitre
	Superbe	Clore	Crénant
	Magnifique		

3- Construction des phrases : à partir du tableau précédent plusieurs combinaisons sont possibles et plusieurs phrases peuvent être générées. Nous avons choisis cinq phrases suivant :

- Le beau ferme la fenétre
- La belle ferme la fenétre
- La belle ferme les fenétre
- La fenétre ferme la belle
- Le ferme la guichet beau

Q1 : Calculer la probabilité à priori P(e) avec un modèle de langue 2-grammes et la probabilité de distorsion (avec $\propto$=0.5) des phrases précédent utilisant le corpus suivant:

- d0 La maman ferme la fenêtre fin
- d0 la ferme est jolie fin
- d0 la belle pleure fin
- d0 le beau frère aime le foot fin
- d0 la belle ferme le robinet fin

4- le tableau suivant regroupe la probabilité de traduction de chaque mot de la phrase source (anglais) vers le français calculé à partir d'un corpus parallèle.

The	Beautiful	Closes	Window
Le =0.3	Beau =0.3	Ferme 0.4	Fenêtre=0.5
La =0.3	Belle =0.28	Termine =0.3	Guichet =0.2
Les =0.4	Jolie =0.12	Se ferme =0.2	Vitre =0.2
	Superbe =0.09	Clore =0.1	Crénant =0.1
	Magnifique =0.21		

Q2)- Calculer la probabilité à posteriori de chaque phrase en français générée dans l'étape 3 avec la phrase source.

Q3)- proposer une solution pour améliorer l'efficacité du modèle probabiliste à base de phrase.

Exercice 2 :

Nous voulons faire la traduction de la phrase (f) « les étudiants attendent les professeurs pour passer le test » utilisant la technique probabiliste phrase based model avec :

1- Anglais comme langue cible (e).
2- Le tableau suivant regroupe la liste des mots anglais pour chaque mot de la phrase obtenus à partir d'un corpus parallèle anglais français.

Les	Etudiants	Attendent	Professeur	Pour	Passer	Test	Le
Them	Students	Wait	Teachers	To	Pass	Quiz	It
The	Undergraduates	Wait for	Professors	For	Go	Test	Him
	Learners	Hold	Instructors	Toward	Play	Experiment	The
	Scholars			Regarding	Run		

3- Construction des phrases : à partir du tableau précédent plusieurs combinaisons sont possibles et plusieurs phrases peuvent être générées. Nous avons choisis cinq phrases suivant :

- The students wait for experiment to pass teachers
- The instructors to play quiz the students wait for
- The students wait for teachers to pass the test.

Q1)- Calculer la probabilité à priori P(e) avec un modèle de langue 3-grammes et 2-grammes. calculer la probabilité de distorsion (avec $\propto$=0.5) des phrases précédent utilisant le corpus d'entrainement suivant:

- **d0 d1 the students pass the test fin**
- **d0 d1 the students wait for the pass fin**
- **d0 d1 teachers test students fin**
- **d0 d1 the patients wait for doctor to pass the visit fin**
- **d0 d1 the students prepare a surprise for teachers fin**

Q2)- évaluer le corpus d'entrainement utilisant la mesure de perplexité afin de dire qu'elle est le meilleur modèle (2-grammes ou 3-grammes).

4- le tableau suivant regroupe la probabilité de traduction de chaque mot de la phrase source (français) vers l'anglais calculé à partir d'un corpus parallèle.

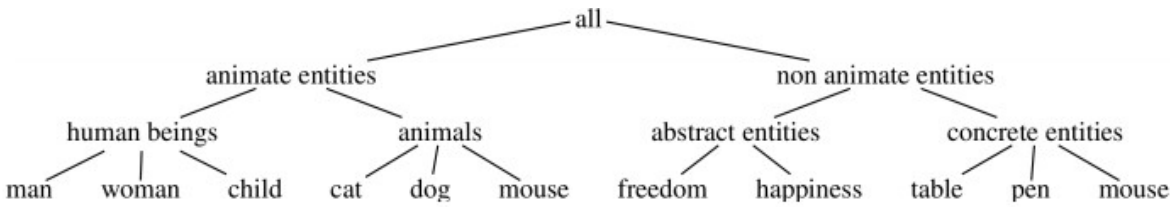
Les	Etudiants	Attendent	Professeur	Pour	Passer	Test	Le
Them =0.3	Students =0.5	Wait=0.47	Teachers =0.45	To =0.26	Pass =0.55	Quiz =0.3	It =0.3

The =0.7	Undergraduates=0.15	Wait for =0.33	Professors=0.35	For 0.34	Go=0.1	Test =0.5	Him =0.3
	Learners =0.2	Hold =0.2	Instructors =0.2	Toward =0.3	Play =0.15	Experiment=0.2	The =0.4
	Scholars =0.15			Regarding=0.1	Run 0.2		

Q3)- Calculer la probabilité à posteriori de chaque phrase en français générée dans l'étape 3 avec la phrase source.

7.6. Les exercices wordnet et similarité sémantique

Exercice 1 : Considérant la structure sémantique suivante d'un ensemble de noms:



Q1 : Quelle est la relation sémantique qui a été utilisée pour construire cet arbre?

Q2 : Citez une autre relation sémantique qui pourrait également être utile pour la construction de ressources sémantiques lexicales ?

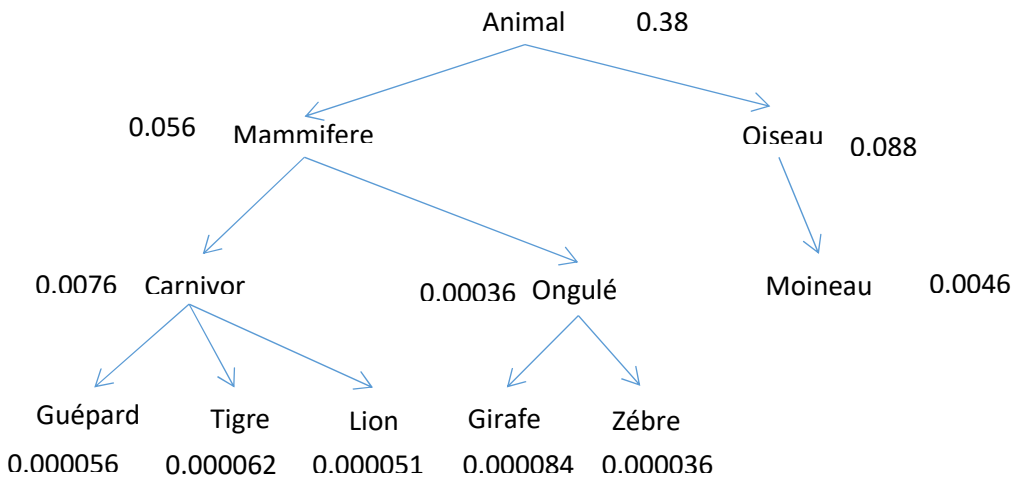
Q3 : Le mot "mouse" apparaît à deux endroits différents de l'arbre. Qu'est-ce que ça veut dire?

Considérez le court texte suivant: Cats are fighting dogs. There are plenty of pens on the table.

Q4: Quel traitement préliminaire faut-il effectuer sur ce texte pour le rendre adapté à l'utilisation de l'arbre sémantique précédente?

Q5 : Calcule la similarité sémantique à base de chemin entre toutes les paires de mots présents dans le texte ci-dessus et dans l'arbre (il en existe 6).

Exercice 2 : Voici l'arbre sémantique suivant avec la probabilité de chaque mot.



Q : calculer les similarités (**Resnik, lin, Jiangconrath**) entre les mots (girafe, moineau) ; (carnivor, zébre) ; (oiseau, animal)

Exercice 3 :

Le tableau suivant regroupe la fréquence de chaque mot dans chaque contexte

	c1	c2	c3	c4	c5
Work	2	1	4	0	1
Profession	0	1	0	2	3
Job	5	2	0	1	0
Occupation	0	0	2	0	6

Q : calculer la similarité distributionnelle entre chaque pair de mot utilisant la mesure cosinus et euclidienne.

7.7. Les exercices RDF RDFS, OWL, Ontology

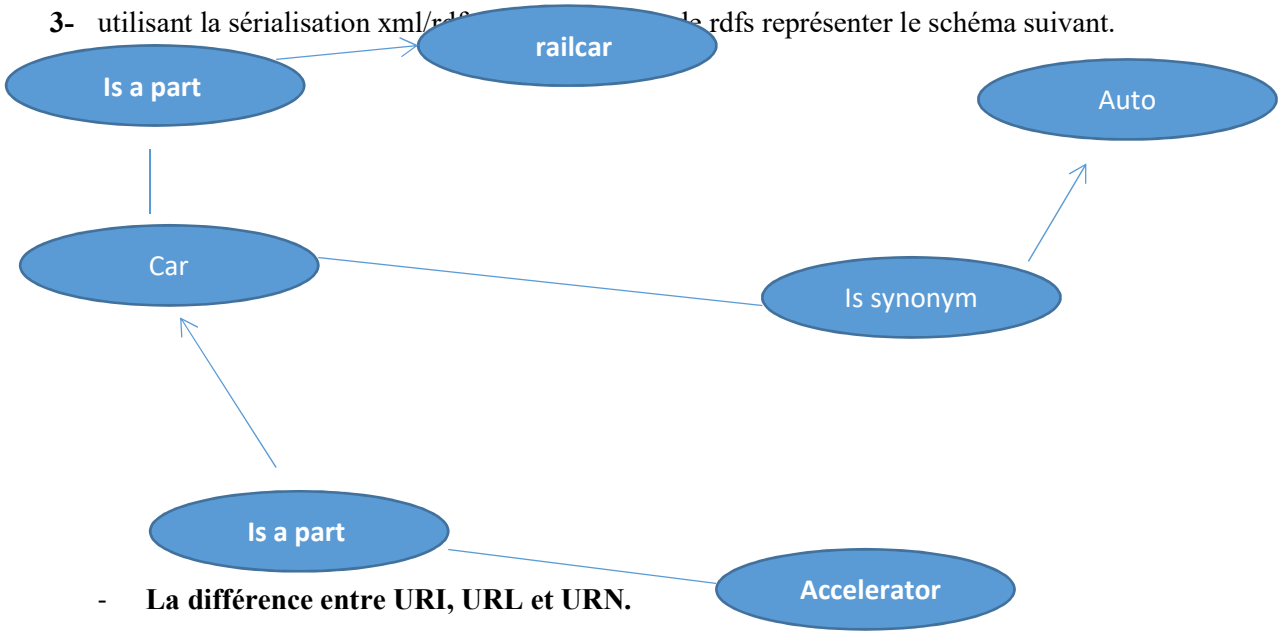
Exercice 1 :

1- **Représenté la phrase suivante utilisant la sérialisation rdf/xml. www.univ-saida.dz/11111111** a comme nom (www.w3.com/name) houari et il est le créateur (www.w3.com/creat) du site **www.univ-saida.dz.**

2- Que représente la portion du code xml/rdf suivant :

```
<rdf:RDF xml:base="http://inria.fr/2005/humans.rdfs"
 xmlns:rdf ="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
```

3- utilisant la sérialisation xml/rdfs représenter le schéma suivant.



- **La différence entre URI, URL et URN.**

Exercice 2 :

Q1 : Représentez les données suivant en triplet (Sujet, Prédicat, Objet).

Q2 : Ecrivez le code RDF/XML équivalent.

1- Le nom smart a smarting, smartness comme synonyms. Il est e kind of cognition, knowledge, moesis. L'adjectif smart a comme antonymy stupid et comme synonym shrewd. Le nom Smartness a brightness, cleverness comme synonyms. Adjective Stupid a dullard, dolt comme synonyms et kind of simpleton.
2- Le nom face a surface, human face, visual aspect comme synonyms. Il est une partie de head, human et man. Is a kind of countenance, visage. Phiz est a kind of face beard, mouth, eye sont des parties de face.

Exercice 3 :

La population que l'on souhaite décrire est composée d'humains, divisés en deux sous-classes Homme et Femme, et habitant dans une certaine ville. Un humain peut avoir un lien de fraternité avec un autre humain. Un homme peut être père d'un autre humain, et un homme et une femme peuvent être mariés.

Q1 : Établir le modèle entité/association décrivant les informations ci-dessus en indiquant les classes, les propriétés, les data propriétés, les restrictions, les cardinalités.

Q2 : Construit l'ontologie OWL sur le schéma obtenu de la question précédente.

Q3 : donner moi 3 instances en utilisant le langage OWL.

Exercice 4 :

La direction de recherche du département d'informatique entame l'automatisation de la gestion des projets de recherche. Cette gestion s'effectue actuellement par le traitement des informations sur :

- Les projets de recherche sont codés, recensés par année et décrits par un résumé du projet et son cout annuel.
- Les chercheurs participant à ces projets : sachant qu'un chercheur/chercheuse n'est rattachée qu'à un seul département.
- Un chercheur/une chercheuse peut avoir un lien de fraternité avec un autre chercheur/chercheuse.
- Un chercheur peut être marié à une chercheuse.
- Un chercheur peut être le père de d'autre chercheurs.
- L'ensemble des documents utilisés dans des projets sont décrit par leur date, leur titre, leur résumé ainsi que les mots clés qui leur sont associés (on peut avoir plusieurs mots clés par document).

- Plusieurs chercheurs peuvent participer à la rédaction d'un document. Ils deviennent alors ses auteurs.
- Les chercheurs sont des enseignants avec le grade Doctorat.

Q1 : Établir le modèle entité/association décrivant les informations ci-dessus en indiquant les classes, les propriétés, les data propriétés, les restrictions, les cardinalités.

Q2 : Construit l'ontologie OWL sur le schéma obtenu de la question précédente.

Q3 : donner moi 3 instances en utilisant le langage OWL.

Exercice 5 :

Une ville dispose de N cinémas. Chaque cinéma est équipé de plusieurs salles de projection. Un cinéma est caractérisé par un code qui l'identifier, une catégorie et une adresse. Une salle de projection est caractérisée par un code, une catégorie et par sa contenance (nombre de places). Un film peut être projeté dans plusieurs salles, à des moments différents, celui-ci est caractérisée par une catégorie, un titre et un réalisateur. Les spectateurs payant un billet d'entrée à un prix fixé en fonction de la catégorie du film, celles de la salle et du cinéma.

Q1 : Établir le modèle entité/association décrivant les informations ci-dessus en indiquant les classes, les propriétés, les data propriétés, les restrictions, les cardinalités.

Q2 : Construit l'ontologie OWL sur le schéma obtenu de la question précédente.

Q3 : la différence entre RDFs et OWL

Q4 : la différence entre ontologie, thésaurus, dictionnaire et wordnet.

Chapitre 8 :

Les Travaux pratiques

8.1. TP 1 : Opinion mining

Objectif du TP :

1- L'opinion **mining** (aussi appelé **sentiment analysis**) est l'analyse des sentiments à partir de sources textuelles dématérialisées sur de grandes quantités de données.
2- SentiWordNet est une ressource lexicale pour l'exploration d'opinion. SentiWordNet attribue à chaque synset de WordNet trois scores de sentiment: positivité, négativité, objectivité.
3- L'objectif du TP est de savoir comment utiliser le sentiwordnet pour analyser les sentiments des gens partagés à travers des **tweets** à travers le réseau social twitter.

Environnement de développement :

- Utiliser java comme langage de programmation
- Utiliser sentiwordnet 3.0 (http://swn.isti.cnr.it/)
- Vous pouvez trouver une classe d'exemple sur l'utilisation de sentiwordnet par java (http://sentiwordnet.isti.cnr.it/).

Les instructions à suivre :

- Extraction des tweets : Vous devez utiliser l'API java « Twitter4J » pour extraire les tweets en ligne et construire un dataset (ensemble de tweets).
- Prétraitement et vectorisation : Dans cette étape une partie de nettoyage et vectorisation utilisant les techniques du text mining pour transformer les tweets vers un ensemble de vecteurs.
- Calcule de scoring : Calculer le score de sentiment de chaque tweet en utilisant les résultats fournis par le sentiwordnet.

8.2. TP 2 : Traducteur automatique par un dictionnaire électronique

Objectif

Maitriser l'utilisation du dictionnaire électronique pour construire un système de traduction automatique à base d'un algorithme d'apprentissage par renforcement.

Environnement de développement :

- Télécharger Anaconda (disponible : https://www.anaconda.com/) python package pour l'installation de l'environnement python 2.7-3.6. Pour plus de détail concernant les étapes d'installation du python cliqué https://machinelearningmastery.com/setup-python-environment-machine-learning-deep-learning-anaconda/ . la plateforme open source anaconda est le moyen le plus rapide de faire de l'apprentissage automatique et l'intelligence artificiel utilisant python. Elle peut être utilisé sous différent systèmes d'exploitation (linux, windows….ect).
- Installer deep learning librery Keras 2.0 compatible avec TensorFlow (video montrant comment installer tenserflow utilisant anaconda https://www.youtube.com/watch?v=ZNWQN_g_ZsI), CNTK ou Theano (disponible https://pypi.org/project/Keras/) afin que vous puissiez utiliser directement les deux algorithmes convolutional networks network (CNN) et recurrent networks.

Les instructions à suivre :

1- Télécharger Dictionnaire anglais / français :

Pour le développement d'un traducteur automatique vous avez besoin d'un dictionnaire anglais / français (télécharger ici : www.manythings.org/bilingual/fra/) qui vas nous permettre d'avoir la signification en français de chaque mot anglais comme le montre les exemples suivant.
Go. Va ! MP3 Wow! Ça alors ! Jump. (Saute. Stop! Ça suffit ! Go on. Continuez. Poursuivez.

2- Dataset à télécharger :

le dataset europal est disponible en cliquant sur http://www.statmt.org/europarl/ et vous devriez choisir les langues français / anglais.

Néttoyage et préparation :

- Les textes doivent étre décomposer à un ensemble de phrases utilisant la représentation sac de phrases.
- Réaliser un nettoyage des phrases en éliminant tous les caractères non alphabétique en rendant le texte en minuscule.

3- Décomposer :

Une fois la phase de préparation de corpus est terminée alors le dataset sera décomposer en deux parties (test et apprentissage). 90% pour la partie d'apprentissage et 10% partie test.

4- Construction du modèle de prédiction :

Dans cette phase vous devez utiliser un algorithme du deep learning (convolutional neural network). Pour la phase d'apprentissage et de teste.

pour plus d'information concernant les instructions de construction suivre le lien ci-dessus **:**

https://machinelearningmastery.com/develop-neural-machine-translation-system-keras/

Pour plus de lecture sur le sujet de traduction automatique vous pouvez télécharger le livre « statistical machine translation ».

8.3. TP 3: Similarité sémantique

Objectif du TP :

L'idée générale de ce tp est de vous permettre d'exploiter et de passer à la machine vos connaissances sur le réseau sémantique wordnet en calculant la similarité sémantique entre deux textes.

Environnement de développement

Java :

- Télécharger et installer Wordnet 2.1 (lien de téléchargement : https://wordnet.princeton.edu/download/current-version)
- Synsmapping (https://wordnet.princeton.edu/download/current-version) .
- Télécharger API JWNL (vous trouviez une classe d'exemple pour l'utilisation de cette API) https://sourceforge.net/projects/jwordnet/

Python

- Utiliser NLTK et importer wordnet (from nltk.corpus import wordnet)
- Pour plus d'information https://pythonprogramming.net/wordnet-nltk-tutorial/

Les étapes à suivre :

Entrée : deux textes en anglais.

Etape 1 : représentation (tokenisation)

Chaque texte est divisé vers un ensemble de mots utilisant la représentation sac de mot en éliminant tous les caractères non alphabétiques.

Etape 2 : étiquetage morphosyntaxique

Dans cette étape l'objectif est de déterminer la catégorie grammaticale de chaque terme dans un texte en utilisant soit l'étiqueteur de brill soit l'étiqueteur probabiliste modèle de markov caché (déjà vu dans le module recherche et extraction d'information).

Etape 3 : lemmatisation

Cette étape est obligatoire si vous utiliser python NLTK alors cette étape sera réaliser automatiquement.

Etape 4 : choisir une méthode de désambiguïsation afin de choisir le synset le plus approprier à chaque mot parmi l'ensemble des sysnset valable dans wordnet.

Etape 5 : calcule de similarité en se basant sur la langueur du chemin entre les sysnsets.

8.4. TP 4 : Moteur de recherche sémantique

Objectif :

Réaliser un système de recherche textuel adhoc (moteur de recherche requête / documents pertinents) qui prend en considération l'aspect sémantique de la requête et les documents en utilisant wordnet.

Environnement de développement :

- Java : wordnet 3.0 et api jwnl.
- Python : NLTK

Les instructions :

- Réaliser une vectorisation des documents et la requête utilisant le wordnet.
- Après calculer la similarité sémantique entre la requête et les documents. Fixer un seuil de similarité et les documents ayant une similarité supérieure au seuil seront déclarer comme pertinent.

8.5. TP 5 : Reconnaissance vocale

Objectif : L'objectif de ce TP est de développer une application qui permet de transformer le texte parlé en texte écrit.

Environnement de développement :

- Télécharger anaconda et installer python
- Télécharger l'API Google Speech API v2 (disponible en : https://pypi.org/project/google-cloud-speech/) Cloud Speech-to-Text permet d'intégrer facilement les technologies de reconnaissance vocale de Google dans les applications de développement. Envoyer de l'audio et recevoir une transcription textuelle du service API Speech-to-Text.

Références :

[1] Aggarwal, C. C., & Zhai, C. (Eds.). (2012). Mining text data. Springer Science & Business Media

[2] Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. Emerging artificial intelligence applications in computer engineering, 160, 3-24.

[3] Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From data mining to knowledge discovery in databases. AI magazine, 17(3), 37-37.

[4] Tan, P. N., Steinbach, M., & Kumar, V. (2013). Data mining cluster analysis: basic concepts and algorithms. Introduction to data mining, 487-533.

[5] Aggarwal, C. C. (2015). Data mining: the textbook. Springer.

[6] Cios, K. J., Pedrycz, W., & Swiniarski, R. W. (1998). Data mining and knowledge discovery. In Data mining methods for knowledge discovery (pp. 1-26). Springer, Boston, MA.

[7] Han, J., Pei, J., & Kamber, M. (2011). Data mining: concepts and techniques. Elsevier.

[8] Romero, C., & Ventura, S. (Eds.). (2006). Data mining in e-learning (Vol. 4). WIT press.

[9] Romero, C., & Ventura, S. (2007). Educational data mining: A survey from 1995 to 2005. Expert systems with applications, 33(1), 135-146.

[10] Aggarwal, C. C., & Zhai, C. (Eds.). (2012). Mining text data. Springer Science & Business Media.

[11] Miner, G., Elder IV, J., Fast, A., Hill, T., Nisbet, R., & Delen, D. (2012). Practical text mining and statistical analysis for non-structured text data applications. Academic Press.

[12] Azuaje, F., Dubitzky, W., Black, N., & Adamson, K. (2000). Discovering relevance knowledge in data: a growing cell structures approach. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 30(3), 448-460.

[13] Yang, Q., & Wu, X. (2006). 10 challenging problems in data mining research. International Journal of Information Technology & Decision Making, 5(04), 597-604.

[14] Chen, F., Deng, P., Wan, J., Zhang, D., Vasilakos, A. V., & Rong, X. (2015). Data mining for the internet of things: literature review and challenges. International Journal of Distributed Sensor Networks, 11(8), 431047.

[15] Berkhin, P. (2006). A survey of clustering data mining techniques. In Grouping multidimensional data (pp. 25-71). Springer, Berlin, Heidelberg.

[16] Zaki, M. J., Meira Jr, W., & Meira, W. (2014). Data mining and analysis: fundamental concepts and algorithms. Cambridge University Press.

[17] Dua, S., & Du, X. (2016). Data mining and machine learning in cybersecurity. Auerbach Publications. [18] https://www.saedsayad.com/data_mining_map.htm (18/01/2018 à 15:00h)

[19] Bishop, C. M. (2006). Pattern recognition and machine learning. springer. [20] Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. Science, 349(6245), 255-260.

[21] Dietterich, T. G. (2000, June). Ensemble methods in machine learning. In International workshop on multiple classifier systems(pp. 1-15). Springer, Berlin, Heidelberg.

[22] Chen, M., Ebert, D., Hagen, H., Laramee, R. S., Van Liere, R., Ma, K. L., ... & Silver, D. (2008). Data, information, and knowledge in visualization. IEEE Computer Graphics and Applications, 29(1), 12-19.

[23] Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). The KDD process for extracting useful knowledge from volumes of data. Communications of the ACM, 39(11), 27-34.

[24] https://www.ranks.nl/stopwords (18/2/2018 à 18:00h)

[25] Mcnamee, P., & Mayfield, J. (2004). Character n-gram tokenization for European language text retrieval. Information retrieval, 7(1-2), 73-97. [26] Singhal, A. (2001). Modern information retrieval: A brief overview. IEEE Data Eng. Bull., 24(4), 35-43. [27] https://wordnet.princeton.edu/ (08/01/2017 à 18 :30 h)

[28] Aggarwal, C. C. (Ed.). (2014). Data classification: algorithms and applications. CRC press.

[29] Demuth, H. B., Beale, M. H., De Jess, O., & Hagan, M. T. (2014). Neural network design. Martin Hagan.

[30] Maulik, U., & Bandyopadhyay, S. (2000). Genetic algorithm-based clustering technique. Pattern recognition, 33(9), 1455-1465.

[31] J. B. MacQueen (1967): "Some Methods for classification and Analysis of Multivariate Observations, Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability", Berkeley, University of California Press, 1:281-297

[32] Bezdek, J. C. (1981): "Pattern Recognition with Fuzzy Objective Function Algoritms", Plenum Press, New York

[33] Krishnapuram, R., Nasraoui, O., & Frigui, H. (1992). The fuzzy c spherical shells algorithm: A new approach. IEEE Transactions on Neural Networks,3(5), 663-671.

[34] Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996, August). A density-based algorithm for discovering clusters in large spatial databases with noise. In Kdd(Vol. 96, No. 34, pp. 226-231).

[35] Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. Psychological review, 65(6), 386.

[36] Pitts, W., & McCulloch, W. S. (1947). How we know universals the perception of auditory and visual forms. The Bulletin of mathematical biophysics, 9(3), 127-147.

[37] Tu, J. V. (1996). Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes. Journal of clinical epidemiology, 49(11), 1225-1231.

[38] Safavian, S. R., & Landgrebe, D. (1991). A survey of decision tree classifier methodology. IEEE transactions on systems, man, and cybernetics, 21(3), 660-674.

[39] Van der Merwe, D. W., & Engelbrecht, A. P. (2003, December). Data clustering using particle swarm optimization. In The 2003 Congress on Evolutionary Computation, 2003. CEC'03. (Vol. 1, pp. 215-220). IEEE.

[40] Gowda, K. C., & Krishna, G. (1978). Agglomerative clustering using the concept of mutual nearest neighbourhood. Pattern recognition, 10(2), 105-112.

[41] Zander, S., Nguyen, T., & Armitage, G. (2005, November). Automated traffic classification and application identification using machine learning. In The IEEE Conference on Local Computer Networks 30th Anniversary (LCN'05) l (pp. 250-257). IEEE.

[42] HIROHATA, M., SHINNAKA, Y., IWANO, K., & FURUI, S. (2005, MARCH). SENTENCE EXTRACTION-BASED PRESENTATION SUMMARIZATION TECHNIQUES AND EVALUATION METRICS. IN PROCEEDINGS.(ICASSP'05). IEEE INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, 2005. (VOL. 1, PP. I1065). IEEE.

[43] Speer, R., Chin, J., & Havasi, C. (2017, February). Conceptnet 5.5: An open multilingual graph of general knowledge. In Thirty-first AAAI conference on artificial intelligence.

[44] Speer, R., & Lowry-Duda, J. (2017). Conceptnet at semeval-2017 task 2: Extending word embeddings with multilingual relational knowledge. arXiv preprint arXiv:1704.03560.

[45] Baker, C. F., Fillmore, C. J., & Lowe, J. B. (1998, August). The berkeley framenet project. In 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1 (pp. 86-90).

[46] Baker, C. F., Fillmore, C. J., & Cronin, B. (2003). The structure of the FrameNet database. International Journal of Lexicography, 16(3), 281-296.

[47] Guarino, N., Oberle, D., & Staab, S. (2009). What is an ontology?. In Handbook on ontologies (pp. 1-17). Springer, Berlin, Heidelberg.

[48] Francopoulo, G., George, M., Calzolari, N., Monachini, M., Bel, N., Pet, M., & Soria, C. (2006). Lexical markup framework (LMF). In International Conference on Language Resources and Evaluation-LREC 2006.

[49] Lehrberger, J. (2015). , Automatic Translation and the Concept of Sublanguage. In Sublanguage (pp. 81-106). De Gruyter.