REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
UNIVERSITE MOULOUD MAMMERI DE TIZI OUZOU
FACULTE DU GENIE ELECTRIQUE ET DE L'INFORMATIQUE
DEPARTEMENT D'INFORMATIQUE

En vue de l'obtention d'un Master Académique en informatique

Spécialité :conduite de projet informatique

Thème

Indexation temporelle d'une ontologie lexicale

Proposée et dirigée par :

Mme L. Bougchiche

réalisée par :

Melle BOUHACI Rym

Melle SLIMANI Taous

Année universitaire : 2016 /2017

Remerciement

Remerciement

D'abord nous remercions le Dieu de nous avoir donné santé, courage et foi pour réaliser ce travail.

Nous remercions chaleureusement notre promotrice, Melle Boughechiche, d'avoir proposé et dirigé ce travail avec ses orientations, ses précieux conseils, ses remarques constructives et surtout son sérieux et sa disponibilité.

Un grand merci aux membres du jury pour l'honneur qu'ils nous ont attribué pour évaluer et juger notre travail.

Enfin, un grand merci a nos chers parents à nos familles pour leurs soutiens et leurs encouragements.

Rym & Taous

Dédicaces

Dédicaces

C’est avec un immense plaisir que j’écris ces lignes, et que je tiens à exprimer ma profonde gratitude à mes très chers parents qui n’ont cessé de me soutenir, et de croire en moi pendant mes études, qui ont fait de moi ce que je suis aujourd’hui, je les remercie du fond du cœur et je vous déclare que vous êtes la source de mon inspiration et de mon courage. Je prie dieu le tout puissant de vs protéger du mal et vs récompenser de toutes les peines et sacrifices données aux quels je ne rendrai jamais assez.

Ainsi je dédie ce travail plus particulièrement pour la mémoire de mes grands-parents paternels et "Jiddah" qui m’avaient aidé pendant toute leurs vies.

Ainsi qu'à mes sœurs Thinhinane et Sarah et à mes frères Youva et Messipssa , qui sont ce que j’ai de plus chers . Qui ont toujours été là pour moi. Qui ont soulevé bien des fardeaux avec moi malgré leurs jeunes âges et qui je porte dans mon cœur.

A tous mes oncles et tantes, cousins et cousines et leurs enfants maternels et Paternels. Sans oublier mes grands-parents maternels.

À mon adorable enseignante « Boughchiche », aucun remerciement ne vaut ce que vous avez fait pour nous

À ma meilleure amie « LYZA», tu es vraiment la plus belle chose qui m'a arrivé à la fac,jt'adore

Je le dédie aussi à « Karima», tu as une place éternelle dans mon cœur ,

Sans oublier 'Kahina», malgré la distance mais j’aimerai pour toujours

à tous mes amis et les personnes qui ont participé soit de loin ou de proche dans cette réussite

Rym

Dédicaces

Je dédie ce modeste travail à mes parents, ma famille, mes proches, mes amies et enfin toute personne ayant contribué au bon accomplissement de ce modeste travail.

Taous

Sommaire

Sommaire

Sommaire

Sommaire

Sommaire

Sommaire

LISTE DES FIGURES

Introduction générale

INTRODUCTION GENERALE

La compréhension de la temporalité d'objets ou d'informations est une clé pour raisonner sur la manière dont le monde évolue. Par nature, le monde est en constant changement, le temps est donc une de ses caractéristiques les plus importantes. Les événements, les changements, les circonstances qui demeurent sur une certaine période sont tous liés par leur ancrage dans le temps. Le temps permet d'ordonner événements et états, d'indiquer leur durée, de préciser leur début et fin.

Dans les dernières années, on peut noter un intérêt croissant pour les applications de Traitement Automatique des Langues (TAL) et de Recherche d'Information (RI) qui peuvent analyser la masse de données numériques disponibles, avec une demande croissante pour une prise en considération de la dimension temporelle. Pour la recherche d'information, face à la quantité d'informations disponibles, proposer un accès aux documents ou aux textes via leur dimension temporelle est particulièrement pertinent.

Le Traitement automatique de la langue naturelle (TALN) ou des langues (TAL) est une discipline à la frontière de la linguistique, de l'informatique et de l'intelligence artificielle.

Elle concerne la conception de systèmes et techniques informatiques permettant de manipuler le langage humain dans tous ses aspects.

Ce travail se situe dans le domaine de traitement automatique des langues plus particulièrement dans le cadre de l'indexation temporelle d'une ontologie lexicale qui révèle de TAL.

Notre travail consiste à mettre en œuvre cette approche, pour cela on a utilisé la base de connaissances lexicale WordNet c'est-à-dire on a appliqué l'apprentissage automatique sur l'ensemble des concepts de cette ontologie et on a utilisé python pour l'implémenter .

Pour réaliser notre travail nous l'avons découpé en quatre parties :

La 1er porte sur des généralités sur le domaine de traitement automatique des langues(TAL), notamment les nivaux de traitement automatique des langues. Les domaines d'application ainsi que les avantages de traitement automatique des langues.

La 2eme porte sur l'indexation, nous définissons d'abord l'indexation ainsi que ses différents types, nous présentons ensuite la classification avec les méthodes de cette dernière.

La 3eme porte sur les ontologies ainsi que les différentes relations qui existe dans WordNet.

La 4eme porte sur l'indexation temporelle de wordNet, la conception et l'implémentation de l'approche proposée et l'expérimentation de notre approche en précisant les outils utilisés et le langage utilisés pour sa mise en œuvre.

Ce travail ce termine par une conclusion générale et quelques perspectives.

Chapitre I :

Traitement automatique de langue

I.1.Introduction :

Le traitement automatique des langues (T.A.L) est un domaine de recherches pluridisciplinaires, qui fait collaborer linguistes, informaticiens, logiciens, psychologues, documentalistes, lexicographes ou traducteurs, Il appartient au domaine de l'Intelligence Artificielle (I.A),Ceci fera l'objet de ce premier chapitre où on présentera l'historique de cette approche, et notamment ses différentes applications.

I.2.Le traitement automatique de langue :

On peut introduire le traitement automatique des langues avec plusieurs définitions :

I.2.1. Première définition [1] :

Le Traitement automatique des langues est l'ensemble des méthodes et des programmes qui permettent un traitement par l'ordinateur des données langagières.

Les acteurs de TAL sont séparés en deux catégories bien distinctes. D'un côté, les chercheurs qui réfléchissent aux méthodes, ils s'adressent à leur propre communauté de chercheurs, à leurs étudiants, ils s'adressent aussi d'une certaine façon aux industriels.
De l'autre côté, on trouve ces industriels eux même qui s'en occupent de la réalisation des produits, ils visent des publics de consommateurs, qui sont soit directement le consommateur individuel, soit d'autres entreprises qui vont se servir des technologies mises en œuvre par les industriels du TAL.

I.2.2.Deuxième définition [2] :

Le traitement automatique des langues naturelles (TALN) a pour but la création de programme informatique permettant de traiter automatiquement les langues naturelles.

Sachant que la langue naturelle désigne la langue écrite ou parlée par les êtres humains, par opposition aux langages artificiels, informatique, mathématique, ou logique, par exemple.

Le traitement porte sur les données linguistiques, les textes, codés dans une langue particulière. Sous cette dénomination générique, nous regroupons aussi les dialogues, écrits ou oraux, et des unités plus petites, comme les paragraphes ou les phrases.

D'une première approximation, le traitement est la transformation d'un objet d'entrée en un objet de sortie. Quand il porte sur la langue, le traitement peut être de deux types :

- Il peut agir sur des données linguistiques (des textes) pour les corriger, les condenser ou les traduire. Souvent, cette transformation comprend une étape intermédiaire qui vise à extraire des textes leur représentation, comme en (1) : elle est appelée analyse des langues naturelles.

(1)Texte1 ⟶ Représentation (Text1) ⟶ Texte 2
Analyse Génération

Le terme « représentation » désigne toute traduction du texte dans un système autre que la langue naturelle et qui rend explicites des informations implicites dans le texte : un ensemble de mot clés, un arbre syntaxique, une formule logique,...etc.
Dans ce premier type de traitement, l'entrée est donc un texte et la sortie un nouveau texte ou une représentation de texte.

- Le TAL peut aussi faire l'opération inverse : il prend alors en entrée la représentation de texte, pour produire un texte en langue naturelle. Cependant, on ne dispose pas généralement de la représentation de texte mais des données brutes, comme des tableaux ou des tables, qu'il faudra d'abord traduire en une représentation de texte, comme en (2), cette opération est appelée génération des langues naturelles.

(2) Données brutes ⟶ Représentation (texte) ⟶ texte
Analyse Génération

Ce traitement automatique nécessite évidement des outils divers que l'on peut grouper en trois catégories distinctes :

- **1ère catégorie :** linguistique, ils décrivent les diverses connaissances relatives aux langues ;
- **2ème catégorie :** Formels, ils expriment ces connaissances dans un formalisme qui convient à un traitement automatique ;
- **3ème catégorie :** Enfin, en informatique, ils utilisent cette description formelle de la connaissance dans une application informatique concrète. Il ne faut donc pas s'étonner de la diversité de TAL, qui fait intervenir des recherches dans différents domaines.

I.3. Les disciplines du TAL :

1. La linguiste informatique : qui développe des programmes de TAL, et définit, dans ce but, de véritable langages informatiques, spécialisés pour les applications de TAL.

2. La linguistique : qui fournit des théories explicites du savoir linguistique.

3 .L'informatique : qui permet d'optimiser les algorithmes et les programmes de traitement, mais aussi de développer des applications formelles (de résolution de problèmes par exemple).

4 .Les mathématiques : qui étudient les propriétés formelles des outils de traitement et de théories.

5. L'intelligence artificielle (IA) : qui s'occupe de la représentation des connaissances et de leur utilisation.

Remarque : Le traitement de données ainsi l'écriture sur fichiers, sauvegardes ou autre ne

font pas partie de TAL.

I.4. Histoire du traitement automatique de langue et la linguistique [2][3]

Les dates qui suivent sont des points de repère remarquables, présentant soit l'une des domaines informatiques ou linguistiques ou les deux à la fois.

I.4.1 Avant le XVIIIème siècle :

Les précurseurs de la linguistique sont les auteurs de grammaires descriptives d'une langue donnée, les précurseurs de l'informatique sont les mathématiciens qui décrivent des méthodes générales de calcul (comme résoudre des équations) et les inventeurs de "machines à calculer" mécaniques comme Pascal et Leibniz.

I.4.2 Année 1660 :

Publication de la "Grammaire générale et raisonnée" connue sous le titre "Grammaire de Port-Royal» d'Arnaud et Lancelot avec une ambition de décrire les règles du langage en termes de principes rationnels universels.

I.4.3 Aux XVIII et XIXème siècles :

Dans cette période, on compare les langues entre elles et on cherche à en conclure des lois d'évolution générales. Du rapprochement entre diverses langues comme latines, grecques, perses, germaniques, celtes, slaves, etc. , pour faire apparaitre l'hypothèse que toutes ont un "ancêtre commun" qui sera appelé plus tard "indo-européen" .Ce siècle est nommé « le règne de la linguistique comparative et historique ».

Le XIXème siècle connaît aussi de grands progrès en mathématiques, et voit naitre sont né la logique "booléenne" (ou "propositionnelle") par Boole puis la "logique des prédicats du 1er ordre" par Frege.
La conception des plans de machines l'ingénieur et mathématicien anglais Babbage, ayantles mêmes capacités de calcul que les ordinateurs actuels. Elles n'ont malheureusement pas pu être construites de son vivant.

I.4.4 Année 1916 :

Publication posthume (des notes de cours publiées par deux étudiants du "Cours de linguistique générale" du linguiste suisse Ferdinand de Saussure, Ce dernier introduit plusieurs différenciation et concepts importants :

- Il distingue les dimensions **diachronique** (évolution au cours du temps) et **synchronique** (rapports entre les signes à une époque donnée du langage). Les études historiques et comparatistes se sont focalisées sur la première de ces dimensions.

- Il distingue deux axes d'analyse d'un discours, en tant que suite de signes :
  - ✓ l'axe ***syntagmatique*** est celui de la succession linéaire des unités qui constituent le discours (un syntagme est une suite d'unités adjacentes) ;
  - ✓ l'axe ***associatif*** *ou* ***paradigmatique*** provient des liens que les signes présents dans le

discours entretiennent avec d'autres signes non présents dans le discours mais en rapport avec eux dans le système. Par exemple : suivant l'axe syntagmatique "un petit chat" est un syntagme dont le sens provient de la combinaison des signifiés de "petit" et "chat", mais ces sens eux-mêmes sont associés suivant l'axe paradigmatique avec d'autres ("petit" par opposition à "grand", etc.).

I.4.5 Années 30-35 :

Le "cercle de Prague" ou « l'école de Prague » (un groupe de critique littéraire et de linguistique) prolonge les analyses de Saussure et promeut une "linguistique structurale". Ses membres les plus connus sont Roman Jakobson et Nicolas Troubetzkoy. On leur doit notamment l'invention de **la phonologie** : étude des sons élémentaires (les phonèmes) qui jouent le rôle d'unités distinctives dans une langue donnée.

C'est aussi à Jakobson qu'on doit d'avoir identifié six *fonctions* permises par le langage dans un contexte de communication :

- la fonction expressive permet au locuteur d'exprimer ses sentiments ;
- la fonction conative permet d'agir sur le destinataire (donner un ordre...) ;
- la fonction référentielle permet d'informer sur le monde extérieur : il faut bien reconnaître que les modèles informatiques ont souvent tendance à limiter le langage à cette fonction ;
- la fonction phatique permet juste de s'assurer du bon fonctionnement de la "ligne" de communication ("allo"...) ;
- la fonction poétique met l'accent sur la forme du message plus que sur son contenu informationnel ;
- la fonction métalinguistique permet de parler du langage grâce au langage (comme le fait ce document !)

.

I.4.6 Année1936 :

Alan Turing (mathématicien anglais) propose un dispositif plus tard appelé "machine de Turing" qui donne une caractérisation mathématique précise à la notion d'algorithme. Cette proposition peut être considérée comme la date de naissance de l'informatique.

I.4.7 Année 1945 :

Von Neumann (mathématicien et physicien américain d'origine hongroise) définit dans un rapport le plan de construction des ordinateurs, tels qu'ils sont encore conçus de nos jours des prototypes plus rudimentaires ont été construits avant.

I.4.8 Année 1950 :

Alan Turing a publié un nouvel article dans lequel il décrit un test pour juger de la capacité des machines à penser : ce test, appelé depuis "test de Turing" est fondé sur un jeu de dialogue comme suit :

Un humain est placé dans une pièce et discute par clavier interposé avec une personne et un ordinateur.

Le test est considéré comme réussi si l'humain n'arrive pas à déterminer qui est l'autre humain et qui est l'ordinateur.

On évalue le comportement extérieur de la machine, ce quine garantit pas forcément la présence d'intelligence ou une compréhension de la langue.

Il prédit qu'en l'an 2000, des machines réussiront ce test.

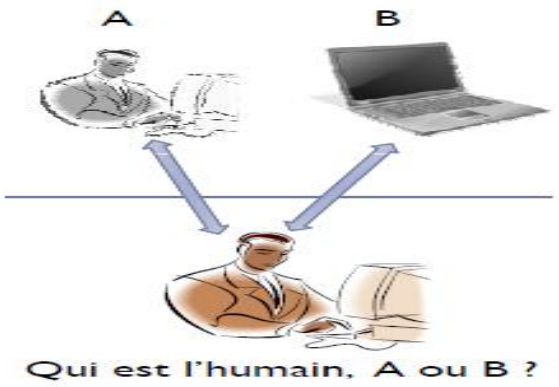

Figure I.1 Image le test de Turing

I.4.9 Année 1952 :

Première conférence sur la traduction automatique, organisée au MIT (Massachussets Institute of Technology) par Yehoshua Bar-Hillel. C'est l'époque de la guerre froide, la compétition russes/américains bat son plein. L'informatique, elle, n'en est qu'à ses balbutiements et les premiers programmes de traduction doivent se contenter de dictionnaires bilingues et de quelques règles de restructuration élémentaires.
Le terme "Intelligence artificielle" (IA), lui, est inventé lors d'une autre conférence.

I.4.10 Année 1954 :

La mise au point du premier traducteur automatique (très rudimentaire). Il contient quelques phrases russes, sélectionnées à l'avance, furent traduites automatiquement en anglais. Bien que le vocabulaire ne comptât que 250 mots et la grammaire 6 règles, cette expérience a déclenché de nombreux travaux dans ce domaine. C'est en effet l'époque où l'URSS remporte succès après succès dans la course à l'espace et où les militaires américains sont très désireux de suivre les publications techniques soviétiques, sans pour autant faire apprendre le russe à tous leurs ingénieurs.

I.4.11 Année 1966 :

En 1966, Les autorités qui ont investi des sommes considérables dans les recherches en TAL désirent des résultats concret et chargent le comité ALPC

(AutomaticLanguageAdvisoryCommittee) d'évaluer les progrès de cette discipline. Le rapport, influencé par les nombreux critiques et sceptiques qui turne en ridicule le domaine, est désastreux ; il conclut que les recherches dans leur état actuel ne sont pas rentables pour l'Etat Américain. Les subsides pour la TA sont donc coupés aux USA du jour au lendemain. Faute de moyens les chercheurs se tournent vers d'autres directions qui permettent de diversifier les recherches de TAL.

Deux programmes célèbres datent de cette époque : "Eliza" de Weizenbaum qui est simulation d'un dialogue avec un psychothérapeute et "Student" de Bobrow capable de faire une analyse syntaxique de textes simples, pour résoudre des problèmes algébriques du type « trouvez l'âge du capitaine ».

Les deux étaient fondés sur la recherche de "mots-clés" dans les données qu'on leur fournissait, mots-clés qui servaient à remplir les "trous" de formulaires définis a priori, sans prise en compte du contexte d'énonciation. Par exemple, dans un dialogue avec Eliza, dès que l'utilisateur mentionnait un lien de parenté ("father"/ "mother"/"brother"...), le programme enchaînait en demandant "Tell me about your [father/mother/brother...]".

la machine : bonjour, comment allez-vous ?
l'homme : je ne me sens pas très bien aujourd'hui.
la machine : et les autres jours ?
l'homme : ça dépend, mais sauriez-vous diagnostiquer mon mal ?
la machine : me croyez-vous capable de diagnostiquer votre mal ?

Figure I.2 : Extrait de dialogue avec le programme Eliza (exemple adapté au français par J-H Jayez).

L'informaticien Terry Winograd présente son programme intitulé SHRDLU (ce nom proviendrait de l'ordre décroissant de la fréquence des lettres en anglais : ETAOINSHRDLU...), permettant des interactions langagières avec un ordinateur sur un domaine restreint à un monde de blocs.

Ce monde est constitué d'un nombre fini d'objets de forme géométrique simple (cubes, boules, cylindres, pyramides, etc.), disposés dans un environnement limité (l'équivalent d'une table).
Les interactions se limitent à la possibilité de poser des questions sur l'état de ce monde simplifié ("Combien y a-t-il de cubes verts à droite de la boule rouge ?") et de donner des ordres permettant de le modifier ("Mettre le cylindre sur le cube bleu.").

L'originalité de ce programme est qu'il ne se contente plus de mots-clés ou de "traitements de surface" rudimentaires : le monde étant parfaitement circoncit, il pouvait être entièrement modélisé

I.3.12 Année 1975 et suivante :

A partir des années soixante-quinze, le TAL va en effet prendre de plus en plus d'importance sous l'effet de différents facteurs :

1. le regain d'intérêt pour le TAL, grâce au succès de programmes comme ELIZA, SHRDLU (qui font oublier l'échec de la traduction automatique).
2. Un effort académique et politique : action de certaines sociétés, comme l'association Of Computational Linguistics (ACL) ou les observations des industries de la langue, création de programme d'études dans plusieurs universités (par exemple les Survey of Computation a Courses, édités par l'ACL en 1986et 1994 ainsi que la revue T.A.L, vol 37, 1996 dédiée à l'enseignement du TAL), organisation de conférences périodiques et d'écoles d'été, etc.
3. Le développement des techniques informatiques (logiciel et matériel).
4. Le développement de l'informatique qui envahit tous les secteurs de la société.
5. La nécessité de traiter et de gérer une masse toujours croissante d'informations multilingues.

A cet égard, 1975 est une date importante pour le développement du TAL en Europe. La communauté Européen, qui doit faire face à l'accroissement alarmant du nombre de traductions, entrevoit la possibilité de recourir à la traduction automatique. L'année suivante, elle déclenche un plan d'action dont le but est de coordonner différents projets qui traitent du multilinguisme et, notamment, de TA quand elle annonce en 1976 l'installation d'un système de TA commercial, nommé Systran.

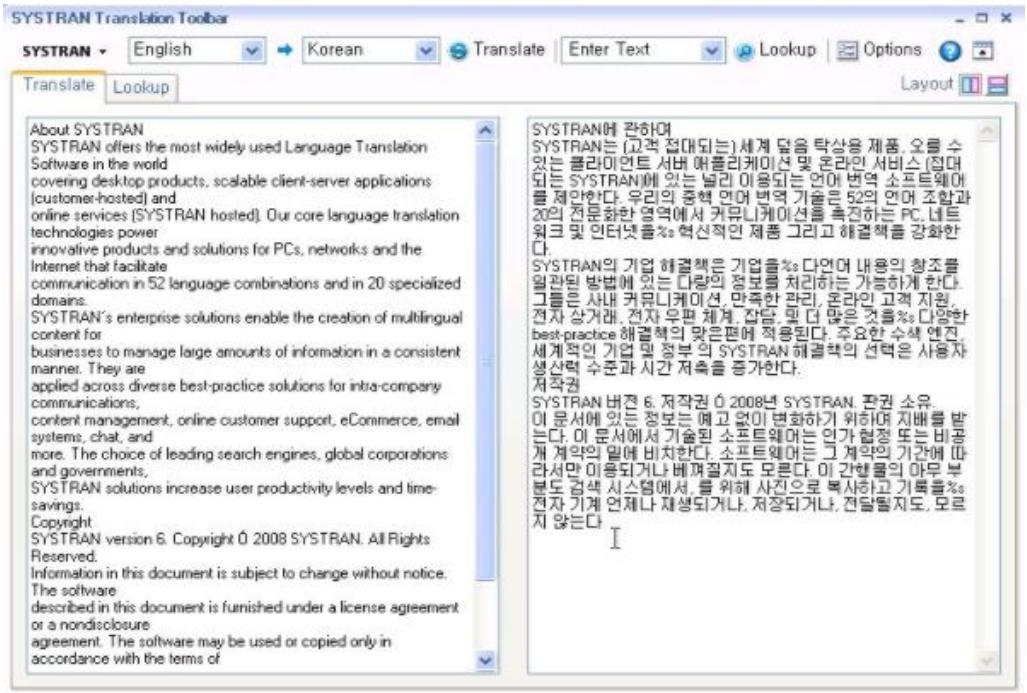

Figure I.3 : Traduction de l'anglais vers le coréen

La traduction automatique se fait connaitre du grand public et suscite à nouveau l'intérêt des firmes privées : elle est désormais une application rentable et les systèmes commerciaux se multiplient.

Depuis, différents programmes de TAL se succèdent, contribuant ainsi au développement des industries de la langue. Citons, à titre d'exemple, les programmes européens EUROTRA (1986-1992) de traduction automatique entre les neuf langues de la communauté européenne. Linguistic Research et Engineering LRE (1994/1998) pour le développement d'applications de TAL et de méthodes, outils et ressources de base et enfin, Langage Engineering LE(1994-998) qui continue les activités développées dans le cadre du programme LRE.

I.5.Les niveaux de traitement automatique des langues [4] :

La tradition distingue plusieurs niveaux de bonne formation et de représentation d'une phrase. On peut les répartir en trois ensembles : deux ensembles qui sont propres aux modes de réalisation oral ou écrit, et un ensemble commun à ces deux modes. Ainsi, pour l'oral, on distingue en particulier les niveaux prosodique, phonétique et phonologique, et pour l'écrit, les niveaux rendant comptent de l'orthographe et de la ponctuation. L'ensemble dit commun est constitué en particulier par les niveaux lexical, syntaxique, sémantique et pragmatique. C'est ce dernier ensemble qui nous intéresse ici.

- **la morphologie (lexique)** : qui concerne l'étude de la formation des mots et leurs variations de forme.
- **La syntaxe** : qui s'intéresse à l'agencement (la distribution) des mots et à leurs relations structurelles dans un énoncé (l'ordre des mots).
- **La sémantique** : Comprendre le sens des phrases.
- **La pragmatique** : c'est de prendre en compte le contexte d'énonciation.

Dans ce qui suit, nous présentons en détaille ces domaines sous l'angle de l'analyse automatique des textes en débutant par la morphologie, la syntaxe, la sémantique et en fin la pragmatique

I.5.1 La morphologie (lexique) :

Selon une vision information, un texte est défini comme une chaine de caractère. Son analyse commence par une première étape: la reconnaissance, d'unités linguistiques de base, les mots, et la mobilisation des informations associées.

Initialement le lexique, est la liste des mots de la langue, et qui associe à chaque mot les informations linguistique correspondante : catégorie syntaxique, traits morphosyntaxiques (genre, nombre, etc.), Plusieurs phénomènes amènent à préciser cette définition de lexique :

- Un mot peut exister sous plusieurs formes : en français, formes fléchies des noms,

adjectifs, etc., conjugaison des verbes. On peut alors considérer une forme canonique, ou lemme, pour chaque mot, qui sert d'entrée dans le lexique pour l'ensemble de ses formes fléchies (singulier pour le nom, masculin singulier pour l'adjectif, infinitif pour le verbe).

- Plusieurs mots peuvent se trouver partager une forme commune (*homographes*) :
  - ✓ « montre » est une forme du nom « montre » aussi bien Le document en sciences du traitement de l'information que du verbe « montrer »
  - ✓ « vu» est le participe passé du verbe « voire »
  - ✓ «vit » est le participe passé du verbe « vivre »

- Un mot peut être construit à partir d'un autre : par dérivation (« penser», «pensable »,« impensable ») ou par composition (« compter» + « gouttes » = « compte-gouttes »,« un » + « jambe » =« unijambiste », « sclérose » + « artère » = « artériosclérose »).

- Enfin, pour de multiples raisons, tous les mots possibles d'une langue ne sont ou ne peuvent être répertoriés a priori dans un lexique. D'une part, les noms propres constituent un inventaire ouvert. D'autre part, de nouveaux mots sont créés régulièrement (néologie) par dérivation et composition, mais aussi par siglaison, abréviation, emprunt,…etc.

Pour commencer l'analyse, la chaîne de caractères d'entrée doit utiliser un encodage déterminé (typiquement, pour le français, l'encodage ISO-latin-1), les caractères de contrôle (fin de ligne, etc.) état aux aussi normalisés. On élimine généralement les caractères non répertoriés.

A ce stade, plusieurs choix peuvent être appliqués, soit selon les séparateurs choisis : tous les caractères non alphabétiques (espaces, apostrophes, tirets...) ou les espaces seulement, ou bien selon que l'on prend en considération les « mots composés » (par exemple « pomme de terre » = une unité) ou pas.
En tout état de cause, on est généralement amené à distinguer la notion d'unité minimale (« token ») et celle de mot (associé à une information lexicale).

I.5.2 La syntaxe :

Afin de pouvoir reconnaitre quels sont les mots qui fonctionnent ensemble dans une phrase, le premier niveau de modélisation consiste à constituer des classes de mots (catégories syntaxiques, parties du discours) possédant un fonctionnement similaire: Nom (N), Verbe (V), Adjectif (A), etc.

Certaines unités, par accident (homographes :*« la »*, *« est »*) ou de façon plus systématique(*« normale »* : A => N, *« coronarographie »* N =>*« coronarographie »* V =>*« coronarographie »* V), peuvent être *ambiguës*entre plusieurs catégories (ambiguïté catégorielle ou lexicale).

Les relations syntaxiques entre les mots d'une phrase peuvent se représenter de plusieurs façons. Le modèle en constituants considère des groupes de mots, ou syntagmes, généralement centrés sur un mot de tête (N, V, etc.), et les modélise par des catégories spécifiques (syntagme nominal ou SN, syntagme verbal ou SV, syntagme adjectival ou SA, etc.). Ces syntagmes peuvent eux-mêmes être éléments d'autres syntagmes, et la structure d'une phrase est alors un arbre de constituants.

Le modèle en dépendance considère directement les mots de tête (recteurs, ou régissant), et leur attache les mots qui en dépendent (régis).

La structure d'une phrase est alors un arbre de dépendance. Des équivalences existent entre les deux modèles.

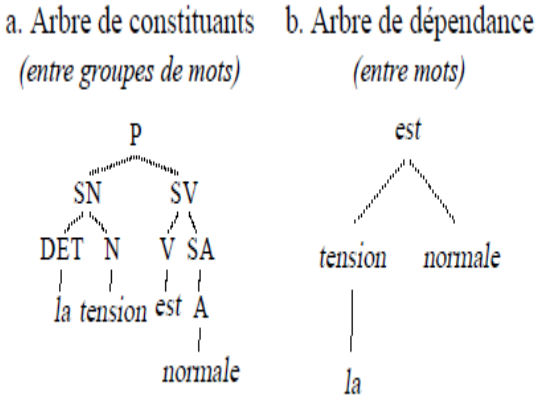

Figure I.4 : Représentation syntaxique d'une phrase.

Même sans ambiguïté lexicale, une phrase peut donner lieu à plusieurs structures syntaxiques (ambiguïté structurelle).

Exemple : La phrase *« je vois un homme avec un télescope »* dans laquelle*« avec un télescope»* peut désigner la manière dont je vois l'homme (attachement au verbe *« vois »*, complément circonstanciel de manière) ou au contraire une caractéristique de l'homme (avec un attachement au nom *« homme »*, complément de nom). Des informations sémantiques, voire pragmatiques (comme ce serait le cas ici), sont nécessaires pour déterminer l'interprétation la plus appropriée de ce genre de phrase.

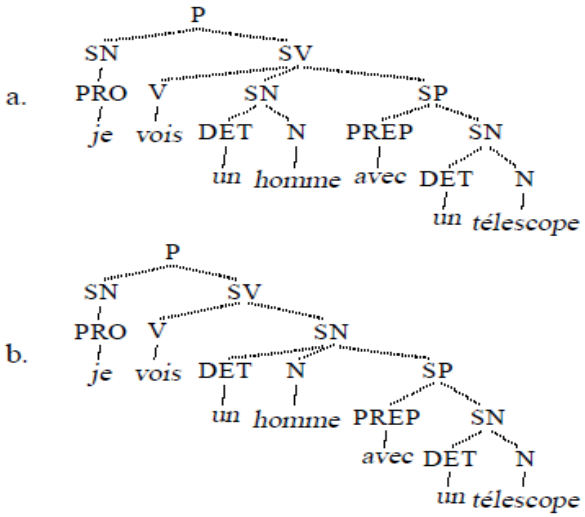

Figure I.5 : Ambiguïté structurelle

Les propriétés intrinsèques des mots restreignent le type de relations syntaxiques qu'ils peuvent avoir. C'est en particulier le cas des verbes, qui régissent ou *sous-catégorisent* de zéro à trois ou quatre « arguments » :

« Il pleut. » *Pleuvoir ()*
« Jean marche » *marché (X)*
« Jean mange une pomme. » *Manger(X, Y)*
« Jean offre un cadeau Marie. » *Offrir(X, Y, Z)*
« Jean interdit à Médor de sortir. » *Interdire(X, Y, Z)*

Des relations plus précises entre mots ou syntagmes sont utiles à l'interprétation des phrases. Les relations grammaticales classiques (sujet-verbe,verbe-objet, verbe-objet-indirect,etc.) permettent de représenter la fonction des groupes de mots les uns par rapport aux autres. Les relations entre pronom et antécédent, et plus généralement entre anaphore (pronom, mais aussi nom) et antécédent, mobilisant encore davantage sémantique et pragmatique, assurent des mises en relation qui peuvent se situer à distance plus grande et qui sont très utiles en recherche d'information.

I.5.3. Sémantique :

Le premier niveau de modélisation pour la sémantique consiste à constituer des classes de mots (catégories sémantiques). Ces classes regroupent des mots dont le sens est proche, ou au minimum (pour des classes générales) des mots qui possèdent certaines propriétés sémantiques communes.

En sémantique aucune classification universelle n'existe (la constitution d'une classification universelle risque même d'être théoriquement impossible). Les classifications que l'on pourra utiliser (par exemple, les catégories générales de WordNet) reflètent nécessairement un point

de vue, une prise de position culturelle ou ontologique spécifique.

I.5.3.1 Mot à plusieurs sens

Un mot pourra posséder plusieurs sens. Par exemple, on pourra distinguer l'*«artère »*– vaisseau sanguin de l'*« artère »*–avenue, même si le second est un sens figuré du premier. Le contexte permet en général de déterminer quel sens est à l'œuvre dans un énoncé.
Les mots d'une langue entretiennent un réseau riche de relations sémantiques paradigmatiques: hyperonymie / hyponymie (*« vaisseau »*/*« artère »*), méronymie (partie d'un tout : *« vaisseau »* / *« système cardiovasculaire »*),antonymie (*« malin »* / *« bénin »*) et autres contraires, etc.

I.5.3.2 Relation pragmatique

Dans un énoncé, les relations grammaticales sont le support de relations sémantiques syntagmatiques. Par exemple, les différents actants d'un événement jouent différents rôles thématiques: agent, thème, source, destination, …etc. Ainsi, dans *« Jean donne un livre à Marie. »*, les rôles par rapport à l'événement *« donne »* pourront être : *« Jean/agent, source donne un livre/thème à Marie/destination. »*

I.5.3.3 La représentation sémantique

La représentation sémantique finale que l'on vise à associer à un énoncé dans un système de TAL dépend de l'objectif de ce système. Cet objectif peut être l'extraction d'informations spécifiques, comme c'est le cas dans les tâches définies dans les campagnes MUC d'évaluation de systèmes d'analyse de textes. Par exemple, l'évolution des postes d'une personne dans une ou plusieurs entreprises était l'une des tâches de la campagne MUC6.
Un éventail d'informations plus large peut aussi être recherché. La représentation doit alors être plus complète, comme dans le système MENELAS.

I.5.4 Pragmatique :

L'interprétation d'un énoncé dépend de son contexte. Dès que l'on veut traiter plus d'une phrase (et même pour une seule phrase), cette dimension intervient.

Le co-texte désigne le texte qui précède (et suit) la phrase courante. Deux facteurs concourent à faire qu'une phrase s'insère bien dans un texte :

– **La *cohésion :*** régit la continuité du texte. Elle est assurée par l'emploi d'anaphores, l'homogénéité du thème, un emploi judicieux d'ellipses, etc.

– **La *cohérence :*** détermine l'intelligibilité du texte. Elle s'appuie sur des structures de discours ainsi que sur les relations causales, temporelles,…etc., entre les événements décrits.

Au-delà du texte lui-même, les conditions d'énonciation et les connaissances partagées complètent le contexte d'un énoncé. L'interprétation devra donc faire appel à des connaissances sur le monde (scénarios, plans, etc.). L'identification de structures de discours

(structure de dialogue, structure argumentative, etc.) est également nécessaire selon le type de texte. De façon générale, une représentation de la situation décrite par un énoncé demande d'effectuer des inférences, ces dernières consistent en l'interprétation, pour un contenu propositionnel donné, une signification supérieur à la somme de ce qui a été simplement énoncé, en faisant intervenir des éléments de contexte intra et extratextuels issus à la fois de l'entourage linguistique et de l'univers de référence des interlocuteurs.

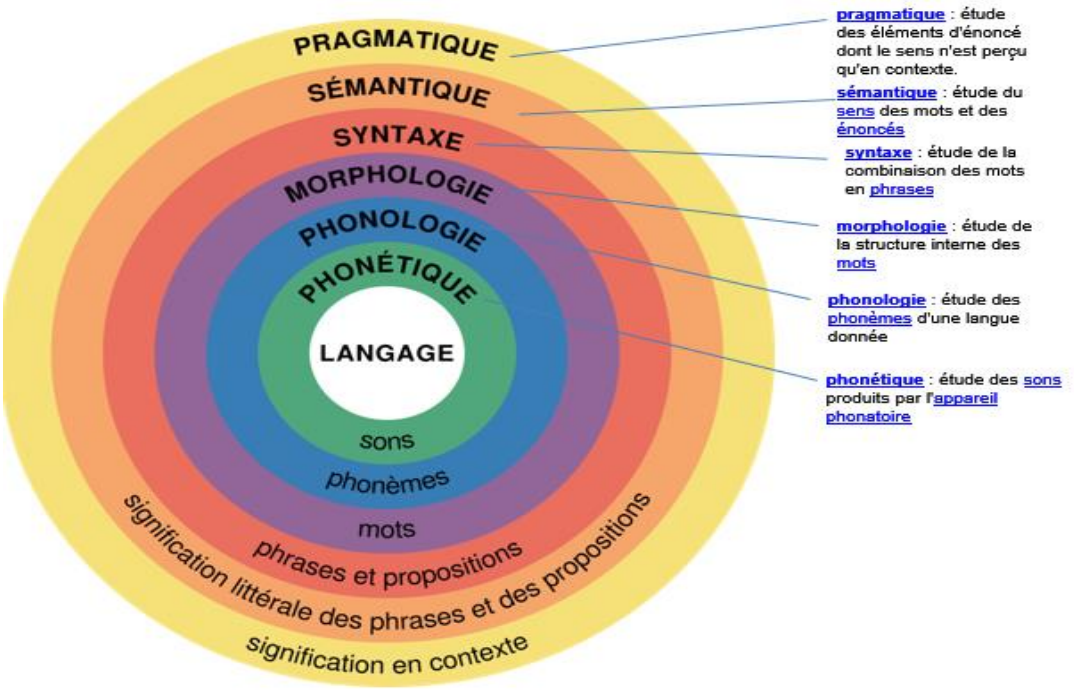

Figure I.6 : Récapitulatif des niveaux d'analyse de TAL

I.6 Les applications du TALN [5]

Concernant les applications, la demande de TALN provient, de deux tendances « lourdes » :

1-D'une part la nécessité de concevoir des interfaces de plus en plus ergonomiques.

2- D'autre part la nécessité de pouvoir traiter (produire, lire, rechercher, classer, analyser, traduire) de manière de plus en plus « intelligente » les informations disponibles sous forme textuelle, de manière à pouvoir résister à leur prolifération exponentielle.

Les applications des techniques de TAL sont donc nombreuses et variées. Nous avons regroupé ces applications en trois grandes familles qui sont :

I.6.1 Le traitement documentaire :

Les applications les plus immédiates TALN sont celles qui visent à faciliter le

traitement par l'humain des immenses ressources disponibles en langage naturel, comme par exemple :

I.6.1.1 La traduction automatique :

Cette application, qui a historiquement suscités les premiers efforts de recherche en TALN, reste un enjeu économique et politique de première importance. Si de tels traducteurs existaient, il serait sans doute beaucoup moins crucial de recourir, pour assurer une large diffusion à des documents.

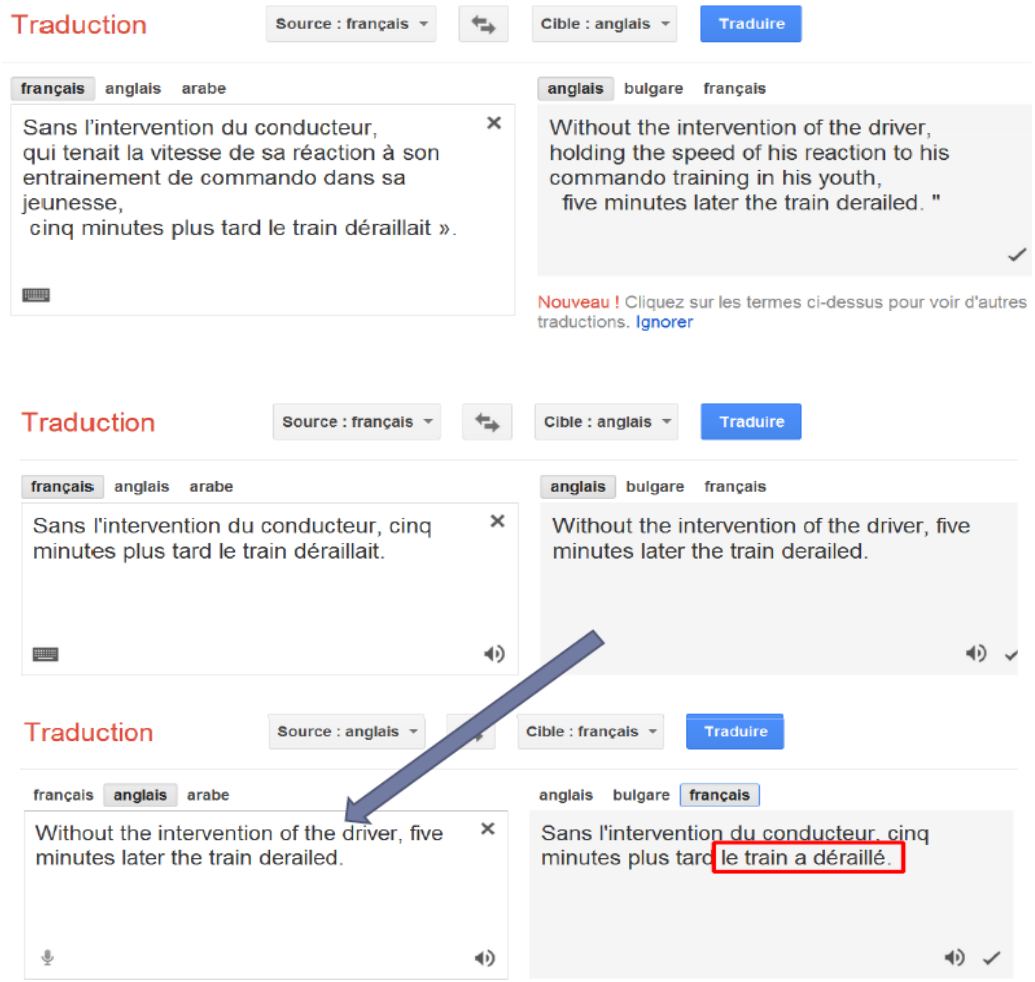

Figure I.7 : Traducteur automatique en ligne

I.6.1.2 La recherche de documents :

Intéressants dans des bases documentaires. La prolifération des outils de recherche documentaire sur la toile, qui traitent quotidiennement des millions de requêtes, montrent bien l'importance de la demande en la matière. Les performances de ces moteurs témoignent du chemin qu'il reste à parcourir dans ce domaine. Si Google semble aujourd'hui sortir du lot, d'autres moteurs de recherche valent certainement la peine d'être connus.

Figure I.8 : Exemple de recherche document

I.6.1.3 Le routage :

Classement ou l'indexation automatique de documents électroniques sont des variantes applicatives du paradigme de la recherche documentaire.

Plus complexe est la tache de trouver (ou de produire à la demande) des réponses précises aux questions de l'utilisateur (tache de « question-réponse »).

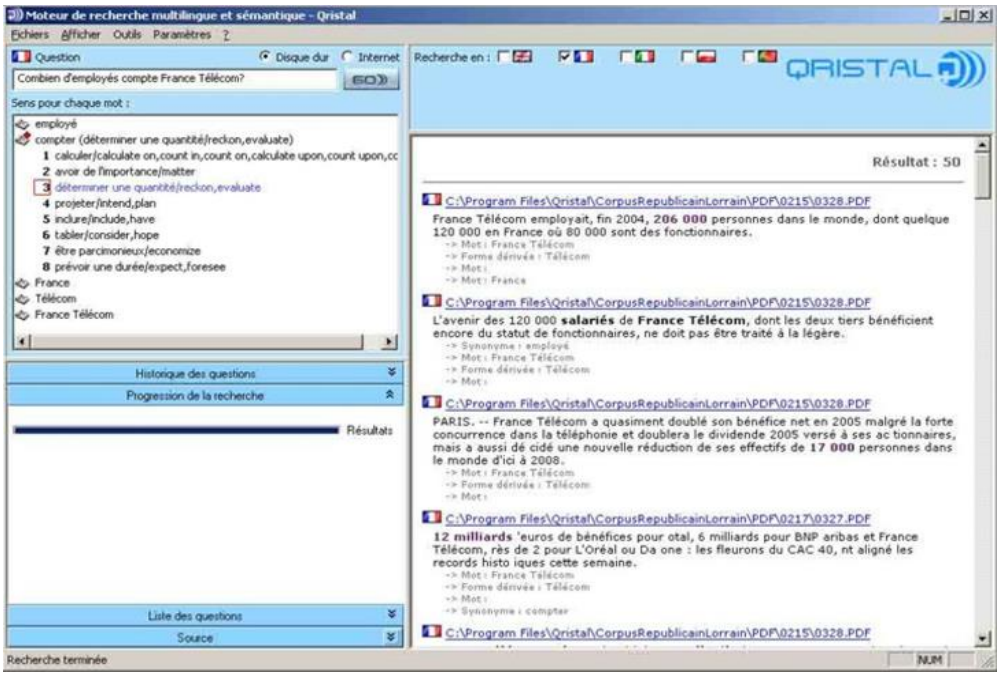

Figure I.9 : Exemple d'un system « question-réponse »

I.6.1.4 L'analyse d'un corpus :

Une application typique de ce domaine consiste à fournir des outils de visualisation et d'exploration dynamique de champs disciplinaires (scientifiques).

Exemple : Le logiciel *Chrono liner développé par* ***Delphine Battistelli et Charles Teissèdre***. À partir du corpus initial, une étape de filtrage à l'aide mots-clés et d'intervalles de dates (étape 1) permet de constituer un corpus thématique sur une période spécifiée par l'utilisateur (par exemple, Egypte+Moubarak de 2011 à 2012). Deux étapes principales sont ensuite à distinguer. L'étape 2 permet de constituer automatiquement une frise chronologique des événements saillants - autrement appelée CE - associée à la thématique. Un événement saillant est présenté sur une CE sous la forme d'une phrase issue d'une dépêche de l'AFP .Le nombre d'événements figurant sur une CE est déterminé par l'utilisateur. Il peut choisir de visualiser les 10 ou 20 événements les plus saillants par exemple.

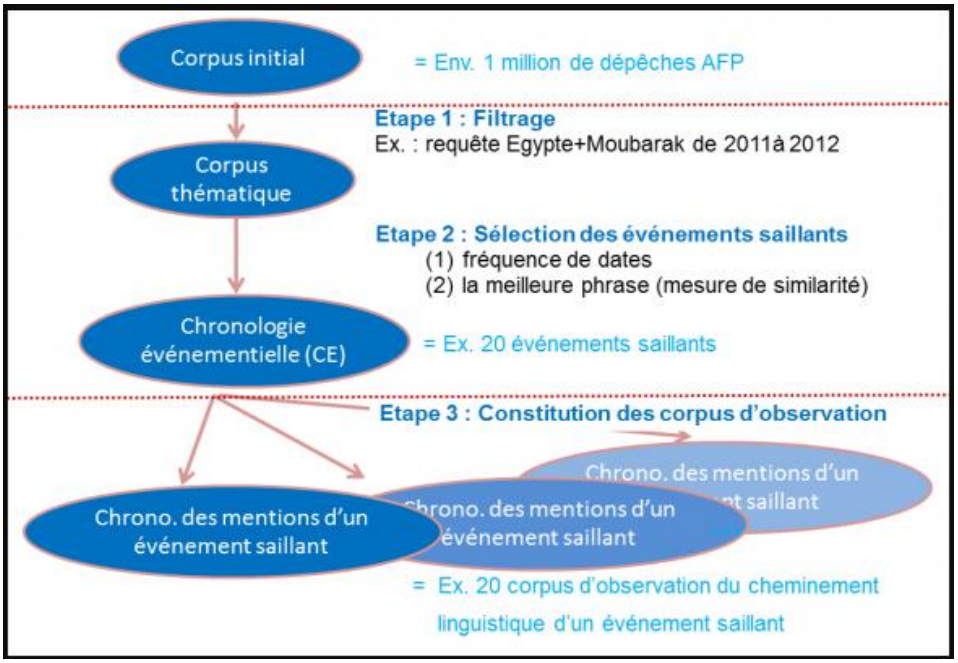

Figure I.10 : Exemple l'architecture générale de Chronoliner

I.6.1.5 La reconnaissance des mots composés : La mise en œuvre des grammaires régulières locales, qui vont détecter toutes les occurrences des patrons typiques de production des mots-composé. Ainsi en Français, ces grammaires détecteront les séquences N Adj (. langage naturel), N PREP N (réseau de neurones, machine à écrire).

I.6.2 la production de document :

Dans le domaine de l'aide à la production de texte (la génération de textes), les applications du TALN sont également nombreuses.

I.6.2.1 Auto-correcteurs : (par exemple pour les handicapés)
Exemple une **plage Braille** affiche le contenu textuel de l'écran en caractères Braille et au même moment, une voix lit le texte. Le logiciel qui permet de faire cela est un lecteur d'écran. Le lecteur d'écran permet également à une personne aveugle d'utiliser l'environnement graphique.

Figure I.11 exemple de système Braille

I.6.2.2 la reconnaissance optique de caractères : De nombreux systèmes commerciaux sont aujourd'hui disponibles, avec des performances très satisfaisantes : Recognita, Omnipage, ScanWorX... etc.

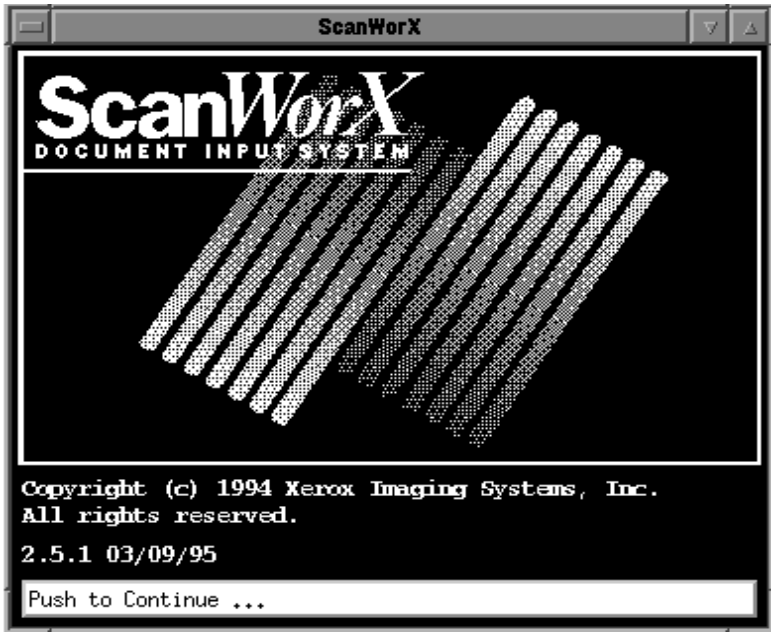

Figure I.12 : Interface de ScanWorX

I.6.2.3 les correcteurs d'orthographe ou de syntaxe : De tels correcteurs sont aujourd'hui disponibles dans la majorité des systèmes de traitement de texte commerciaux, avec des performances variables suivant les mécanismes de correction mis en œuvre, qui vont de la recherche lexicale tolérante à l'analyse syntaxique partielle ou complète de la phrase.

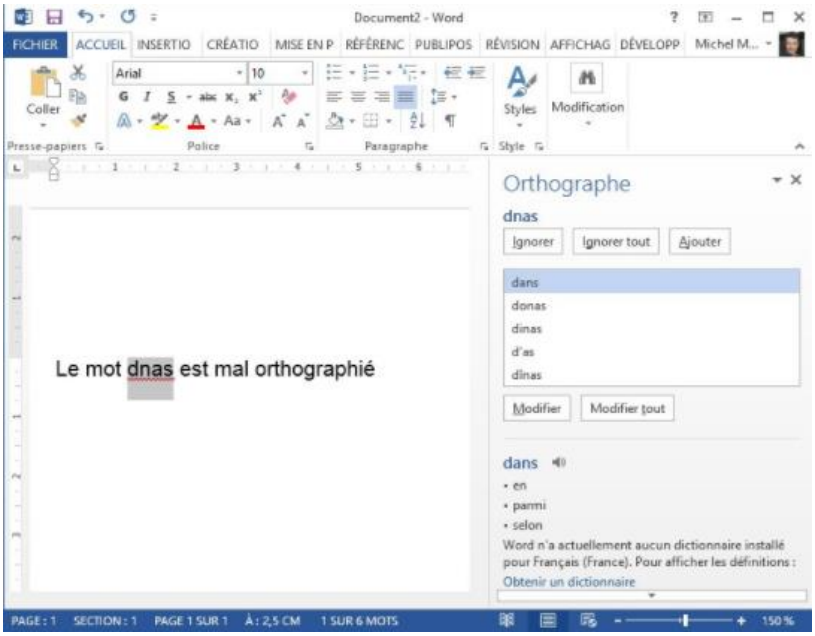

Figure I.13 : Correcteur intégré dans l'office

I.6.2.4 la génération automatique de documents : à partir de spécifications formelles. En fait, de nombreux secteurs d'activité impliquent la production massive de textes très stéréotypés à partir de spécifications plus ou moins formelles (textes juridiques, compte-rendu d'exploration d'une base de données, rapports d'analyses statistiques, documentations techniques, etc). Pour cette classe de documents, il est parfaitement possible de générer automatiquement, sinon des textes complètement définitifs, du moins des versions préliminaires qui seront ensuite finalisés par des rédacteurs humains

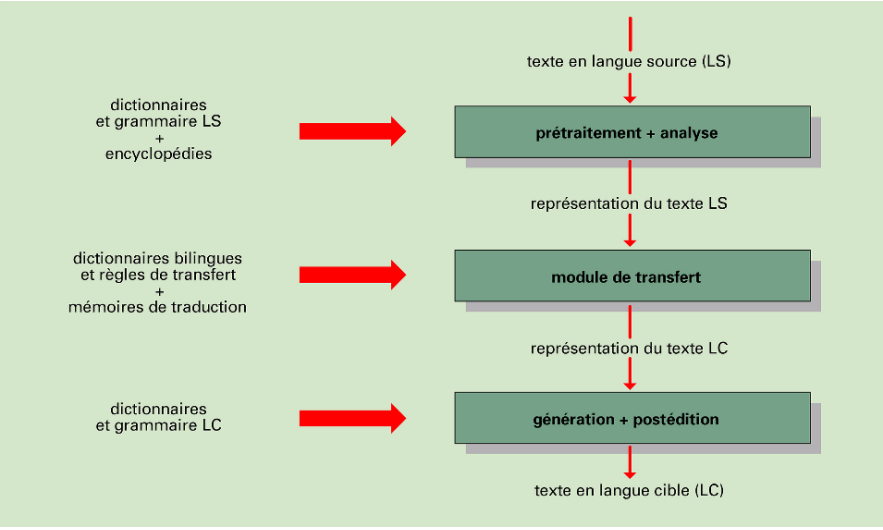

Figure I.14 : Schéma d'un générateur d'un texte

I.6.2.5 les correcteurs « stylistiques », ou les aides intelligentes à la rédaction intégrant des thésaurus, des connaissances sur les « bonnes » pratiques rédactionnelles, etc.

Voici un exemple de ce qu'une correctrice orthographique nouvelle génération **Pro Lexis** en l'occurrence est capable de faire. Le texte suivant truffé de fautes de types différents a été soumis au logiciel *:*

Jusqu'a l'anniversaire de mes vingts-deux ans, les mois et les années sont passé sans que je m'en aperçois. Je n'aie pas connu de tristesse profonde, ni de vraie querelles familiales, et je n'ai pas subis les tiraillements pénibles de parents a problèmes. J'ai pas eu faim ni froid ni peur. Je n'ai vécu la solitude que lorsque je l'ai recherché et j'ai toujours trouvé, quant j'en ai eu besoin, quelqu'un à qui parler. Mon père, aîné de huit enfants, avaient des yeux bleus-clairs, des yeux bleus magnifique. J'aurais bien aimé avoir quelques demis-frères ou surs, en tous cas, je n'en aurais pas été contrariée, j'en suis certain. Toi aussi, mon enfant, tu m'as reprochée ne pas m'être mariée, de ne pas t'avoir donnés de frère ou sur, de ne pas faire ce que j'aurais du Ce n'est pas un hasard, cette maman qui t'a mis au monde, élevé depuis dix sept ans n'est pas ta mère Ce pratique est courante. Dans ces années quatre-vingts, ma grand-mère, précisément âgée de quatre-vingt ans, me ne le reprochait pas. A l'époque, on multiplier.

Le résultat final montre le fort potentiel qui caractérise aujourd'hui les aides logicielles à la correction de textes.

« Jusqu'à l'anniversaire de mes **vingt** -deux ans, les mois et les années sont **passés** sans que je m'en **aperçoive.** Je n'**ai** pas connu de tristesse profonde, ni de **vraies** querelles familiales, et je n'ai pas **subi** les tiraillements pénibles de parents **à** problèmes. **Je n'ai pas eu** faim ni froid ni peur. Je n'ai vécu la solitude que lorsque je l'ai **recherchée** et j'ai toujours trouvé, **quand** j'en ai eu besoin, quelqu'un à qui parler. Mon père, aîné de huit enfants, **avait** des yeux **bleu clair**, des yeux bleus **magnifiques**. J'aurais bien aimé avoir quelques **demi-frères** ou surs, en tout cas, je n'en aurais pas été contrariée, j'en suis certain. Toi aussi, mon enfant, tu m'as **reproché de ne pas** m'être mariée, de ne pas t'avoir **donné** de frère ou sur, de ne pas faire ce que j'aurais **dû** Ce n'est pas un hasard, cette maman qui t'a mis au monde, élevé depuis **dix-sept ans** n'est pas ta mère **Cette** pratique est courante. Dans ces années **quatre-vingt**, ma grand-mère, précisément âgée de **quatre-vingts** ans, **ne me le reprochait pas. A l'époque, on multipliait ces tentatives.**

*L'anniversaire de mes **vingt** -deux ans, les mois et les années sont **passés** sans que je m'en **aperçoive.** Je n'**ai** pas connu de tristesse profonde, ni de **vraies** querelles familiales, et je n'ai pas **subi** les tiraillements pénibles de parents **à** problèmes. **Je n'ai pas eu** faim ni froid ni peur. Je n'ai vécu la solitude que lorsque je l'ai **recherchée** et j'ai toujours trouvé, **quand** j'en ai eu besoin, quelqu'un à qui parler. Mon père, aîné de huit enfants, **avait** des yeux **bleu clair**, des yeux bleus **magnifiques**. J'aurais bien aimé avoir quelques **demi-frères** ou surs, en tout cas, je n'en aurais pas été contrariée, j'en suis certain. Toi aussi, mon enfant, tu m'as **reproché de ne pas** m'être mariée, de ne pas t'avoir **donné** de*

frère ou s ur, de ne pas faire ce que j'aurais ***dû*** *Ce n'est pas un hasard, cette maman qui t'a mis au monde, élevé depuis* ***dix-sept ans*** *n'est pas ta mère* ***Cette*** *pratique est courante. Dans ces années* ***quatre-vingt****, ma grand-mère, précisément âgée de* ***quatre-vingts*** *ans,* ***ne me le reprochait pas. A l'époque, on multipliait ces tentatives »***

Au vu de ce test on constate que si la correction n'est pas parfaite à 100%, le résultat est amplement satisfaisant. On remarque en outre que la vérification n'est pas simplement lexicale et grammaticale mais également syntaxique. Le module de traitement syntaxique d'une phrase fonctionne par comparaison, construisant le schéma syntaxique de la phrase qui débute par une majuscule initiale et se termine par un point final, d'exclamation ou d'interrogation en vérifiant la compatibilité des catégories syntaxiques des mots dans l'ordre dans lequel ils sont placés tout en s'assurant que cet ordre correspond à l'un des schémas syntaxiques connus. Ceci explique que *je ne le reprochait pas* soit aussitôt rectifié par *ne me le reprochait pas*. Il est toutefois bon de préciser qu'une mauvaise ponctuation perturbe grandement le fonctionnement du traitement syntaxique puisque le logiciel divise la phrase en plusieurs segments se basant pour ce faire sur la ponctuation, les prépositions et les subordonnants. Pour en revenir brièvement au traitement grammatical, le fait que la plupart des bases de données des logiciels susnommés sont constituées sur la base de la grammaire Le Bon Usage de Goosse-Grevisse, dont la réputation n'est plus à faire, est un gage de qualité.

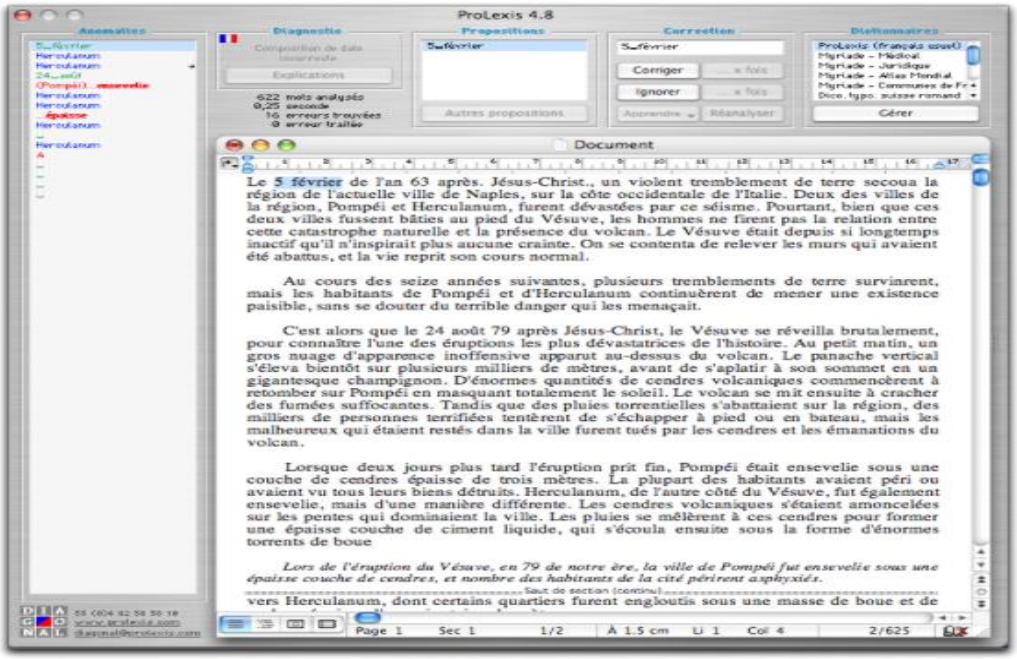

Figure I.15 : Traducteur ProLexis 4.8

I.6.2.6 l'apprentissage :

Il est Assisté par ordinateur des langues naturelles comme titre d'exemple on cite le jeu Fun English: Apprenez l'anglais – qui un jeux d'apprentissage de langue pour les enfants de 3 à 10 ans, pour apprendre à lire, parler et épeler, il combine un cours structuré de langue anglaise à des jeux amusants et engageants.

Le cours de langue anglaise est divisé en leçons. Chaque leçon d'anglais enseigne un vocabulaire de base et présente les mots dans plusieurs contextes pour aider à l'apprentissage et à la mémorisation.

Fun English utilise des voix masculines et féminines à la fois avec des accents anglais et américain. Les voix utilisent des tons et expressions différents pour que les apprenants puissent percevoir les subtilités de la prononciation.

Figure I.16 : Jeu Fun English

I.6.3 Les interfaces naturelles :

Dernier domaine d'application, qui est sans doute celui dans lequel la demande de traitements linguistiques est la plus forte, le domaine des interfaces

I.6.3.1 l'interrogation en langage naturel de bases de données :

La mise en place de multiples applications de ce type commencent à se mettre en place sur la toi le. Comme traduction langage naturel vers SQL via ses interfaces ou de moteurs de recherche sur la toile.

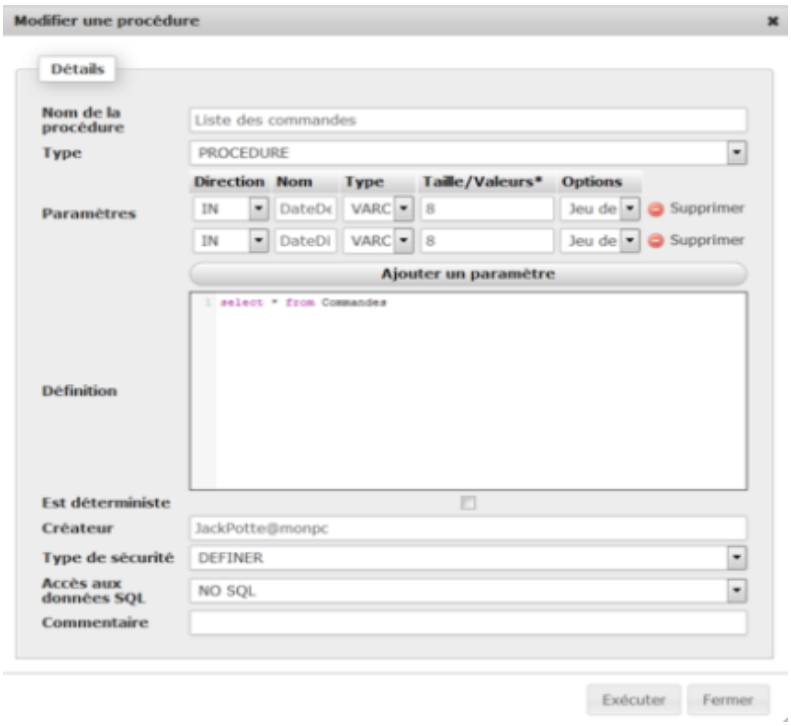

Figure I.17 : Interface d'interrogation d'une base de données

I.6.3.2 les interfaces vocales :

Qui mettent en œuvre de manière variable suivant les applications des modules de reconnaissance de parole, synthèse de parole, génération et gestion de dialogue, accès aux bases de connaissance,..., chacun de ces modules demandant des traitements spécifiques (désambiguïsation morphosyntaxique et identification de syntagmes pour la synthèse, grammaires stochastiques pour la reconnaissance de la parole...).

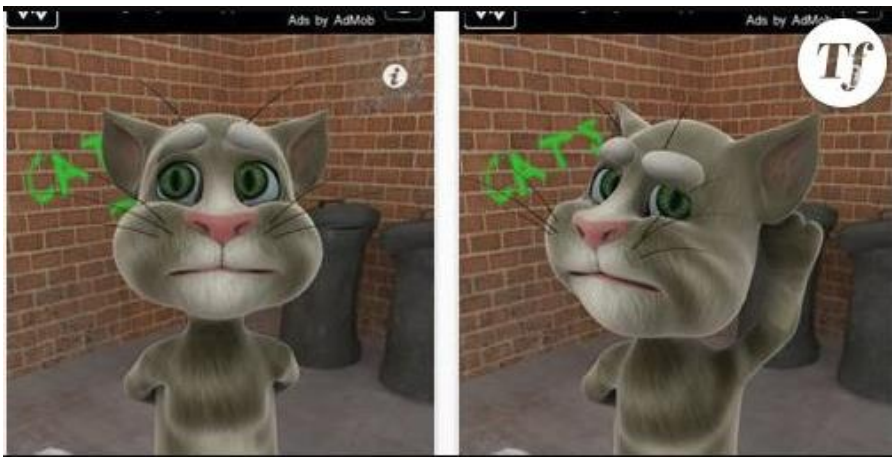

Figure I.18 : L'application «Tom le chat qui parle »

I.7 Les difficultés du TALN [5]:

Les difficultés que l'on rencontre en TALN sont principalement de deux ordres, et ressortent soit de l'ambiguïté du langage, soit de la quantité d'implicite contenue dans les communications naturelles

I.7.1 Ambiguïté

Le langage naturel est ambigu, et ce à quelque niveau qu'on l'appréhende. Cette ambiguïté, loin d'être marginale, est un de ses traits caractéristiques. On peut d'ailleurs voir là le résultat d'un compromis inévitable entre d'un côté une capacité d'expression quasi illimitée, et de l'autre des contraintes liées à la limitation des ressources physiologiques mises en œuvre (taille de la mémoire à long et court-terme, densité de l'espace phonétique, contraintes articulatoires, etc).

Cette ambiguïté se manifeste par la multitude d'interprétations possibles pour chacune des entités linguistiques pertinentes pour un niveau de traitement, comme en témoignent les exemples suivants :

– ambiguïté des lettres dans le processus d'encodage orthographique : comparez la prononciation du i dans lit, poire, maison ;

– ambiguïté des terminaisons dans les processus de conjugaison et d'inflexion : ainsi un /S/ final marque à la fois le pluriel des noms, des adjectifs, et la deuxième (parfois également la première) personne du singulier des formes verbales ;

– ambiguïté dans les propriétés grammaticales et sémantiques (i.e. associées à son sens) d'une forme graphique donnée : ainsi manges est ambigu à la fois morpho-syntaxiquement, puisqu'il correspond aux formes indicative et subjonctive du verbe manger), mais aussi sémantiquement. En effet, cette forme peut aussi bien référer (dans un style familier) à un ensemble d'actions conventionnelles (comme de s'assoir à une table, mettre une serviette, utiliser divers ustensiles, ceci éventuellement en maintenant une interaction avec un autre humain) avec pour visée finale d'ingérer de la nourriture (auquel cas il ne requière pas de complément d'objet direct) ; et à l'action consistant à effectivement ingérer un type particulier de nourriture (auquel cas il requiert un complément d'objet direct), etc.

Comparez en effet : (a) Demain, Paul mange avec ma sœur.

(b) Paul mange son pain au chocolat. Ainsi que les déductions que l'on peut faire à partir de ces deux énoncés : de (a), on peut raisonnablement conclure que Paul sera assis à une table, disposera de couverts,....tout ceci n'est pas nécessairement vrai dans le cas de l'énoncé (b).

– **ambiguïté de la fonction grammaticale des groupes de mots**, illustrée par la phrase :

Il poursuit la jeune fille à vélo. Dans cet exemple «à vélo » est soit un complément de manière de poursuivre (et c'est-il qui pédale), soit un complément de nom de fille (et c'est elle qui mouline) ;

– **ambiguïté de la portée des quantificateurs**, des conjonctions, des prépositions. Ainsi, dans « Tous mes amis ont pris un verre» on peut supposer que chacun avait un verre

différent, mais dans «Tous les témoins ont entendu un cri » il est probable que c'était le même cri pour tous les témoins. De même, lorsque l'on évoque «les chiens et les chats de Paul », l'interprétation la plus naturelle consiste à comprendre de Paul comme le complément de nom du groupe les chats et les chiens ; cette lecture est beaucoup moins naturelle dans les chiens de race et les chats de Paul ;

– ambiguïté sur l'interprétation à donner en contexte à un énoncé : En comparant la « signification » de non, dans les deux échanges suivants : Si je vais en cours demain ? Non (négation) avec «Tu vas en cours demain ! Non ! » (J'y crois pas)

I.7.2 Implicite :

L'activité langagière s'inscrit toujours dans un contexte d'interaction entre deux humains, sensément dotés d'une connaissance du monde et de son fonctionnement telle que l'immense majorité des éléments de contexte nécessaires à la désambiguïsation mais aussi à la compréhension d'un énoncé naturel peuvent rester implicites.

La situation change du tout au tout dès qu'une machine tente de s'insérer dans un processus de communication naturel avec un humain : la machine ne dispose pas de cette connaissance d'arrière-plan, ce qui rend la compréhension complète de la majorité des énoncés difficile, voire impossible, si l'on ne dispose pas de bases de connaissance additionnelles, donnant accès à la fois à un savoir sur le monde (ou le domaine) en général (connaissance statique) et sur le contexte de l'énonciation (connaissance dynamique).

Un exemple typique est la désambiguïsation du référent du pronom personnel, il dans les trois énoncés suivants : Le professeur envoya l'élève chez le censeur parce qu'a :
a. ... il lançait des boulettes (il réfère probablement à l'élève)
b. ... il voulait avoir la paix (il réfère probablement au professeur)
c. ... il voulait le voir (il réfère probablement au censeur)

En l'absence de telles connaissances, bien d'autres problèmes de compréhension deviennent pratiquement insurmontables : pensez par exemple aux ellipses, aux métaphores, et, plus généralement, aux figures de style.

Fort heureusement, il existe de nombreuses applications pour lesquelles ces difficultés peuvent être, dans une large mesure, circonscrites en restreignant, le cadre des textes analysés à un sous-domaine particulier (textes juridiques, textes scientifiques, serveur d'information spécialisé dans les informations sportives...), il devient possible d'une part d'ignorer un grand nombre d'ambiguïtés, en particulier sémantiques (par exemple dans le contexte de textes juridiques, on pourra probablement négliger la possibilité qu'un avocat marron désigne un fruit un peu trop mûr) ; et d'autre part de représenter formellement un grand nombre des connaissances nécessaires à la compréhension des énoncés du domaine considéré.

En fait, certains domaines d'activité ou contextes d'interactions spécifiques semblent

restreindre de manière drastique l'ensemble des énoncés possibles (ou acceptables), simplifiant de manière considérable le traitement de ces véritables sous-langages par une machine.

I.8 Les outils de TAL [4] :

On distingue deux grandes familles des outils de traitement automatiques des langues :

I.8.1 La famille des étiqueteurs :

Constitue les programmes et dispositifs ayant pour fonction d'associer des informations (étiquetées) distinctive à des mots, phrase, et passage d'un texte ou document.

- **Les étiqueteurs morphosyntaxiques** : permettent d'ajouter ou d'associer des informations morphologique et syntaxique aux mots d'un corpus, d'un texte ou d'un document écrit ou orale.
- **Les étiqueteurs prosodiques** : L'ensemble d'outils permettant d'ajouter ou d'associer des informations prosodique aux mots d'un corpus, d'un texte ou d'un document. Désormais aux corpus oraux.
- **Les étiqueteurs sémantiques :** qui permettent d'ajouter ou d'associer des informations sémantiques aux mots d'un corpus, d'un texte ou d'un document écrit ou orale.

On peut illustrer par beaucoup des étiqueteurs existants comme LeTreeTagger qui est un outil pour annoter du texte avec des informations sur la partie de la parole et le lemme. Il a été développé par Helmut Schmid dans le projet de TC à l'Institut de linguistique computationnelle de l'Université de Stuttgart en 1994. L'TreeTagger a été utilisé avec succès dans la langue allemande, anglais, française, italienne, néerlandaise, espagnole, bulgare, russe, portugaise, galicienne, chinoise, swahili, slovaque, slovène, latine, estonienne, polonaise, roumaine, tchèques.

```
ABR Abreviation
ADJ Adjectif
ADV Adverbe
DET:ART Article
DET:POS Pronom Possessif (ma, ta, ...)
INT Interjection
KON Conjunction
NAM Nom Propre
NOM Nom
NUM Numéral
PRO Pronom
PRO:DEM Pronom Démonstratif
PRO:IND Pronom Indefini
PRO:PER Pronom Personnel
PRO:POS Pronom Possessif (mien, tien, ...)
PRO:REL Pronom Relatif
PRP Préposition
PRP:det Préposition + Article (au,du,aux,des)
PUN Ponctuation
PUN:cit Ponctuation de citation
SENT Balise de phrase
SYM Symbole
VER:cond Verbe au conditionnel
VER:futu Verbe au futur
VER:impe Verbe à l'impératif
VER:impf Verbe à l'imparfait
VER:infi Verbe à infinitif
VER:pper Verbe au participe passé
VER:ppre Verbe au participe présent
VER:pres Verbe au présent
VER:simp Verbe au passé simple
VER:subi Verbe à l'imparfait du subjunctif
VER:subp Verbe au présent du subjunctif
```

Figure I.19 : Jeu d'étiquettes utilisé par tree tagger pour le français

I .8.2 Famille des correcteurs :

Représentent tous les outils permettant d'assurer les différentes fonctions de corrections.

- **Les correcteurs grammaticaux** : permettent de corriger la structure grammaticale d'un texte.

- **Les correcteurs orthographique/lexicaux** : détectent les erreurs orthographiques d'un texte et proposent la correction (le correcteur intégré dans l'office)

- **Les correcteurs stylistiques** : l'ensemble des programmes qui corrigent le style d'un texte. Ils éliminent les répétions, constatent les même ensemble de mots souvent présents dans le texte, signalent les phrases trop longues.

On peut citer comme exemple d'outils le correcteur orthographique et grammatical « Cordial », conçu par la société toulousaine Synapsys Developpement, qui est le fruit de

l'effort d'une équipe d'une équipe d'une dizaine de développeurs et linguistes. Le mot « Cordial » est un acronyme pour « CORrecteur D'Imprécisions et Analyseur Lexico-syntaxique ».

I.9. Les avantages de traitement automatique des langues

Les bénéfices du traitement du langage naturel sont innombrables, dans ce qui suit nous allons citer quelques exemples

1. Le traitement des langues naturelles peut être exploité par les entreprises pour améliorer l'efficacité des processus de documentation, améliorer la précision de la documentation et identifier les informations les plus pertinentes provenant des grandes bases de données. Par exemple, un hôpital peut utiliser le traitement du langage naturel pour tirer un diagnostic spécifique des notes non structurées d'un médecin et attribuer un code de facturation

2. Des meilleurs résultats pour n'importe quelle recherche de mots-clés ou de recherche par texte, la recherche sémantique fournit des résultats fidèles à la forme : exactement ce que les clients recherchent. Plus de bizarreries dans les résultats ou des efforts gaspillés dans le perfectionnement des termes de recherche : TAL fournit les résultats applicables dès que le client frappe "recherche".

3. Les clients sont humains, ce qui signifie qu'ils sont faillibles. Ils font des erreurs d'orthographe, confondent les marques avec les produits et oublient les détails lorsque ces erreurs se produisent. Le TAL connecte les points pour continuer la recherche sans interruption, même en cas de fautes de frappe ou de mauvaises informations qui pourraient autrement diverger les résultats.

4. Plus de données extraites signifie plus de données pour la croissance La mesure de ce que les clients recherchent est essentielle pour améliorer les entreprises. Grâce à l'énorme profondeur de données présentée par la TAL, nous sommes en mesure de cultiver ces données à un degré énorme, en découvrant les habitudes et les tendances des clients dans l'ensemble de la base de consommateurs. Ces données peuvent être appliquées à travers de nombreuses facettes de votre entreprise, du marketing au référencement, des campagnes de marketing aux ventes et aux promotions et au-delà.

I.10 Conclusion

Généralement, les modèles linguistiques existants se limitent à la description des aspects lexicaux et syntaxiques du langage, mais ce niveau de représentation est insuffisant dans la reconnaissance et la compréhension du langage, même quand il s'agit de phrases très simples. Il est devenu plus intéressant, du point de vue de la représentation des connaissances, de s'orienter vers les modèles sémantiques et pragmatiques qui font actuellement l'objet de recherches très prometteuses. L'approche par attributs sémantiques,

pour lever les ambiguïtés non résolues aux niveaux morpho-lexical et syntaxiques s'avère trop limitée. Des systèmes s'appuyant sur une formalisation lexicale poussée, telle l'approche de Melcuk, semblent plus prometteurs.

Si certains systèmes informatiques en matière de TAL semblent actuellement satisfaisants pour des langues européennes, la représentation, la saisie, l'édition et le traitement des langues extra-européennes restent largement inexplorés. En outre, la plupart des modèles élaborés pour la formalisation de sous-ensembles du langage naturel, au niveau morpho-syntaxique et sémantique, ne sont pas paramétrables ; souvent dédiés à des langues indo-européennes, ils sont difficilement adaptables aux autres familles de langues (ex. langues sémitiques). Par ailleurs, le contexte économique international actuel conduit à un besoin de plus en plus croissant en outils de traitement multilingue, d'entrées/sorties et de traduction de documentations techniques et scientifiques.

Chapitre II :Indexation

I. Introduction :

L'indexation est l'opération qui consiste à décrire et à caractériser un document à l'aide de la représentation des concepts contenus dans ce document, c'est-à-dire à transcrire en langage documentaire les concepts après les avoir extraits du document par une analyse. La transcription en langage documentaire se fait grâce à des outils d'indexation tels que les thesaurus, et les classifications.
Dans ce chapitre nous allons traiter la notion d'indexation et de classification.

II. présentation de l'indexation :

L'indexation consiste à représenter le contenu d'un document par des expressions linguistiques. « L'indexation est une activité intellectuelle qui a un but pratique, retrouvé rapidement de l'information. » (Weinberg 2009, 2287, notre traduction) , «Indexer, c'est l'acte de pointer vers. » (Jacob & Shaw 1998, 156, notre traduction) Par l'association de localisateurs aux expressions linguistiques, l'index peut permettre de retrouver l'évocation d'un concept dans un ensemble documentaire, que ce soit un document dans une collection ou une section particulière dans un document.**[07]**

Le processus d'indexation comporte deux étapes principales L'analyse des documents afin d'identifier les concepts qu'il comprend, et la représentation de ces concepts en termes d'indexation. Les concepts sont issus de l'analyse pratiquée sur les documents à indexer (identification et sélection des concepts) et les termes d'indexation sont choisis pour représenter ces concepts au mieux par la personne qui indexe. Ce dernier choix s'effectue en fonction de plusieurs paramètres tels que l'unité de traitement, l'institution et ses habitudes de fréquentation, les sujets fréquemment recherchés ou le vocabulaire employé par les usagers dans leurs requêtes.**[10]**

La (Figure II .1) ci-dessous résume les différentes étapes:

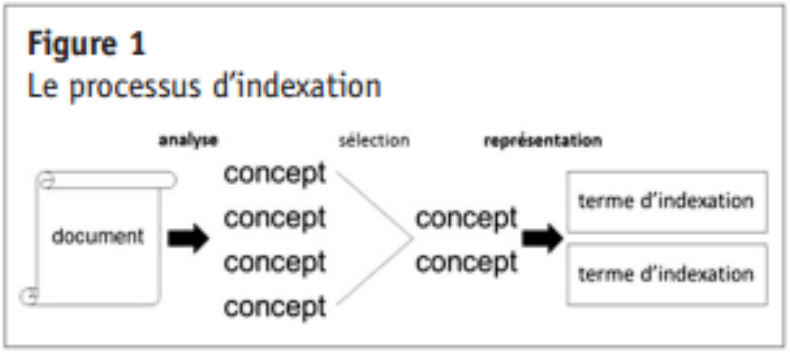

Figure II.1 : présentation du processus d'indexation

II.2.Type d'indexation :

Techniquement le processus d'indexation peut être manuelle (expert en indexation), automatique (ordinateur) ou semi-automatique (combinaison des deux).

- **Indexation manuelle :** chaque document de la collection est examiné par un spécialiste expert du domaine ou un documentaliste afin d'identifier les descripteurs.
- **Indexation automatique :** est un processus entièrement informatisé (automatisé).elle met en œuvre des méthodes et des techniques issues des traitements automatique des langues(TAL).
- **Indexation semi-automatique :** est une hybridation des deux types d'indexations précédentes, ainsi les résultats obtenus par une indexation automatique sont exposé a un documentaliste pour les valider ou pour les enrichir.

II.3. forme d'indexation : on peut distinguer deux forme d'indexation

- **Indexation conceptuelle :**

L'indexation conceptuelle a pour objectif d'indexer un document en utilisant les descripteurs issus d'une base de connaissances comme wordNet. Plus spécifique que l'indexation sémantique, l'indexation conceptuelle permet de pondérer les concepts correspondant à des termes (appartenant au document ou non) afin de sélectionner les meilleurs.

- **Indexation temporelle :**

L'indexation temporel a pour objectif de déterminer le temps par exemple la classification (présent, past, future) de tous les mots appartenant a un document.

L'indexation fait appel a une classification. Elle consiste à décrire finement le contenu d'un document dans le but de le rendre accessible par le catalogue par diverses entrées.**[09]**

Ainsi, il est possible d'utiliser une classification pour indexer un document, c'est-à-dire en traduire le contenu de façon normalisé même si les classifications ne sont pas les seuls langage utilisé pour décrire le contenu d'un document.**[08]**

Les classifications proposent une organisation des connaissances en un système totalement ordonné de classes et de sous classes. Leur structure est dite hiérarchique ou encore arborescente car elles parcourent le savoir des notions générales aux objets particuliers.**[08]**

III. Définition de la classification:

La classification est la plus populaire du Data Mining, c'est l'opération qui consiste à regrouper des objets (individus ou variables) en un nombre limité de groupes, classes (ou segments, ou clusters), les classes de la classification regroupent les objets ayant des caractéristiques similaires et séparent les objets ayant des caractéristiques différentes.

Les méthodes de classification sont de plus en plus utilisées en Data Mining: de janvier 2006 au début d'avril 2007, il y a eu 222 267 brevets industriels déposés aux Etats Unis, dont 500 se servant de la classification **[10]**.

III .1.Terminologies :

Avant d'aller plus loin dans ce chapitre, on préfère introduire certaines terminologies, synonymes de classification, qui peuvent amener à confusion :

Classification : terme généralement employé par les auteurs français. Les anglo-saxons l'emploient dans un autre sens, ils disent « classification » pour designer la technique que les français appellent « classement ».

Segmentation : terme employé en marketing.

Clustering : terme anglo-saxon le plus courant.

Taxinomie et taxonomie : utilisé en biologie et zoologie.

Nosologie : utilisé en médecine.

Notion de classe [12] :

Une classe est composée d'un certain nombre d'objets similaires que l'on peut regrouper ensemble.

- Une classe est un ensemble d'entités qui sont semblables, alors que les entités provenant de classes différentes ne sont pas semblables.
- Une classe est un agrégat de points dans l'espace de représentation des données tels que la distance entre le centre et un point de cette classe est moins importante que celle entre le centre de cette classe et n'importe quelle point d'une autre classe.
- Les classes peuvent être décrites comme des régions connexes de l'espace contenant relativement une grande densité de points.

Notion de similarité [12]:

Il s'agit de définir des groupes d'objets tels que la similarité entre objets d'un même groupe soit maximale et que la similarité entre objets de groupes différents soit minimale.

En Pratique, la similarité entre objets est estimée par une fonction calculant la distance entre ces objets.

Ainsi deux objets proches selon cette distance seront considérés comme similaires, et au contraire, deux objets séparés par une large distance seront considérés comme différents.

Le choix de cette mesure de distance entre objets est alors très important.

Très souvent il s'agit d'un choix arbitraire qui traite tous les attributs de la même manière

IV. Approches de la classification : Il existe deux types d'approches :

- Classification supervisée.
- Classification non supervisée.

Ces deux approches se différencient par leurs buts et leurs méthodes.

IV .1.Classification supervisée [12] :

La classification supervisée (catégorisation**)** vise à classer des objets selon des catégories bien définies au préalable.
Dans ce type de classification, les classes sont prédéfinies avec une description des données. Lorsqu'une nouvelle donnée arrive, on la compare avec la description de chaque classe et on la met dans celle qui lui ressemble le plus.

Dans cette approche on connaît les classes possibles et on dispose d'un ensemble d'objets déjà classés, servant d'ensemble d'apprentissage. Le problème est alors d'être capable d'associer a tout nouvel objet sa classe la plus appropriée, en se servant des exemples déjà étiquetés (déterminer, par exemple, a partir d'une base de données exprimant les problèmes cardio-vasculaires d'un ensemble de patients, quels risque cardio-vasculaire peut avoir un nouveau patient) [**07]**.

IV .1.1 Les méthodes de classification supervisée[12] :

Il existe plusieurs méthodes de classification supervisée :

- K plus proches voisins
- Arbres de décisions
- Naïve Bayes (simple Bayes)
- Réseaux de neurones
- Programmation génétique
- Machines à support de vecteurs(SVM).

❖ K plus proches voisins ou K-NN (« k-nearest neighbors »):

La méthode de k-NN se distingue des autres méthodes a cause du fait qu'elle n'a pas d'étape d'apprentissage : construction d'un modèle a partir d'un échantillon d'apprentissage (ex : réseau de neurones, arbres de décision….). Un modèle est : un échantillon d'apprentissage, une fonction de distance et une fonction de choix de la classe en fonction des classes des voisins les plus proches.

La classification de chaque individu d'opère en regardant, parmi les individus déjà classées, la classe des K individus qui sont les plus proches voisins(ou en calculant la moyenne dans le voisinage de la variable à prédire).

- La valeur de K sera choisie de sorte d'obtenir la meilleure classification possible.
- Ce choix est la principale difficulté de cet algorithme.

Algorithme K-NN :

Objectif : affecter a une classe une nouvelle instance.
Donnée : un échantillon de m enregistrements classées (x, c(x)).
Entrée : un enregistrement y.

1. déterminer les k plus proche enregistrement de y.
2. combiner les classes de ces k exemples en une classe c.

Sortie : La classe de y est c(y)=c.

- **Les avantages :**

Mise en œuvre simple.
Pas de modèle à construire.
Domaine de compétence bien délimité.
Efficace pour des classes réparties de manière irrégulière.
Utilisable pour des données hétérogènes(ou incomplète).

- **Les inconvénients:**

Pas de modèle construit
Nécessite capacité de stockage et puissance de calcul
Besoin de nombreuses données de référence
Choisir K (alternative: choisir la taille du voisinage)

❖ l'arbre de décision

L'arbre de décision est une technique permettant de classer des documents par division hiérarchique en sous classes voir illustration sur la figure II.1 suivante :

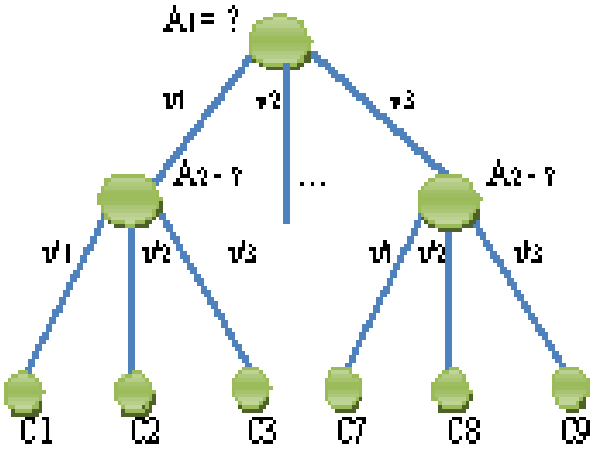

Figure II.2 : Principe de l'arbre de décision

Formellement un arbre de décision est composé :
- D'un ensemble de nœuds internes (y compris la racine de l'arbre) appelés nœuds de décision.
- D'un ensemble de feuilles qui constituent les classes cibles.

De manière plus détaillée avec un arbre de décision on a :
- Un nœud représente une classe de plus en plus fine depuis la racine.
-Un attribut sert d'étiquette de classe (attribut cible à prédire), les autres permettant le partitionnement.

Un chemin de la racine à une feuille peut être exprimé par le résonnement suivant :

(A1=V1) & (A2=v'1) ⟶ C1
(A1=V1) & (A2=v'2) ⟶ C2
(A1=V1) & (A2=v'3) ⟶ C3

(A1=v3) & (A2=v'1) ⟶ C7
(A1=v3) & (A2=v'2) ⟶ C8
(A1=v3) & (A2=v'3) ⟶ C9

Génération de l'arbre de décision [14] : La génération de l'arbre passe par deux phases :
1. Construction de l'arbre : L'arbre peut atteindre une taille élevée.
2. Elaguer l'arbre (Pruning) : Identifier et supprimer les branches qui représentent du "bruit" (Améliorer le taux d'erreur).

Algorithme [14] :

Construction de l'arbre :

-Au départ, toutes les instances d'apprentissage sont à la racine de l'arbre.

-Sélectionner un attribut et choisir un test de séparation sur l'attribut, qui sépare le "mieux" les instances. La sélection des attributs est basée sur une heuristique ou une mesure statistique.

-Partitionner les instances entre les nœuds fils suivant la satisfaction des tests logiques.

-Traiter chaque nœud fils de façon récursive.

-Répéter jusqu'à ce que tous les nœuds soient des terminaux. Un nœud courant est terminal si:

- Il n'ya plus d'attributs disponibles.
- Le nœud est "pur" : toutes les instances appartiennent à une seule classe.
- Le nœud est "presque pur" : la majorité des instances appartiennent à une seule classe (Ex : 95%).
- Nombre minimum d'instances par branche.
- Etiqueter le nœud terminal par la classe majoritaire.

Elaguer l'arbre obtenu (*Pruning*) :

-Supprimer les sous-arbres qui n'améliorent pas l'erreur de la classification → arbre ayant un meilleur pouvoir de généralisation, même si on augmente l'erreur sur l'ensemble d'apprentissage.

-Eviter le problème de sur-spécialisation (over-fitting). En d'autres termes, on a appris "par coeur" l'ensemble d'apprentissage, mais on n'est pas capable de généraliser.

Pour éviter le problème de sur-spécialisation, soit les deux approches suivantes :

- Pré-élagage : Arrêter de façon prématurée la construction de l'arbre.
- Post élagage : Supprimer des branches de l'arbre complet ("fully grown")**.**

- **Les avantages [14] :**
  - Compréhensible pour tout utilisateur (lisibilité du résultat –règles -arbre)
  - Justification de la classification d'une instance (racine → feuille)
  - Tout type de données
  - Robuste au bruit et aux valeurs manquantes

  Attributs apparaissent dans l'ordre de pertinence → tâche de pré-traitement (sélection d'attributs)
  - Classification rapide (parcours d'un chemin dans un arbre)
  - Outils disponibles dans la plupart des environnements de Data Mining

- **Les inconvénients [14] :**
  - Sensibles au nombre de classes: performances se dégradent
  - Evolutivité dans le temps: si les données évoluent dans le temps, il est nécessaire de relancer la phase d'apprentissage.
  - Définition des noeuds de niveau n+1 dépend de celle du niveau n.

❖ Classification Bayésienne naïve[12]

Le Simple Bayes est un classifieur probabiliste. Une fois les probabilités *a posterioi {P*,ci} estimées par le classifieur, il parait naturel d'affecter une nouvelle observation à la classe ayant la probabilité la plus élevée.
La théorie de la Décision Bayésienne établit que la stratégie qui consiste à affecter une nouvelle observation à la classe ayant la plus grande probabilité *a posteriori* est optimale, c'est-à-dire génère un plus petit nombre d'erreurs que toute autre stratégie, ce résultat peut être généralisé à une situation d'un grand intérêt pratique.
Dans le chapitre IV nous allons reprendre cette technique de classification de manière détaillée.

❖ Réseaux de neurones [11]:

Un réseau de neurones est composé de plusieurs neurones interconnectés. Un poids est associé à chaque arc. A chaque neurone on associe une valeur.

Les réseaux de neurones sont utilisés pour leur capacité à apprendre à partir d'exemples bruités comme les caméras ou les micros (reconnaissance de forme ou de son). Mais ils sont aussi utilisables pour des problèmes où les méthodes symboliques (arbres de décisions) sont souvent utilisées. Leur performance est alors équivalente.
Les réseaux de neurones sont appropriés si la compréhension de la fonction apprise par le réseau n'est pas essentielle. Avec un arbre de décision, l'opérateur humain peut toujours visualiser l'arbre et « comprendre » comment la machine décide. Avec un réseau de neurone, des techniques de visualisation existent, mais elles demandent généralement plus d'expertise que l'analyse d'un arbre de décision (qui peut être visualisé sous forme de règles).

La version la plus simple d'un réseau de neurone est le perceptron. Les perceptrons sont capables d'apprendre des fonctions linéairement séparables comme AND, OR, NAND, NOR. Le perceptron ne peut par contre pas apprendre le XOR. C'est d'ailleurs une critique adressée en 1969 par Minsky et Papert et les perceptrons ont été oubliés pendant quelques années...

❖ Programmation génétique [11]:

C'est une méthode générale qui peut être utilisée après n'importe qu'elle méthode précédente, par exemple avec les arbres de décisions. En entrée, un algorithme génétique reçoit une population de classifieurs non optimaux. Le but du programme génétique est de produire un classifieur plus optimal que chacun de ceux de la population d'origine.
D'une façon simple, cela consiste à extraire les meilleures parties de chaque classifieur d'origine et de les mettre ensemble pour produire un nouveau classifieur. Cela suppose de pouvoir comparer l'efficacité d'un classifieur. Un résultat important de la méthode est qu'après chaque itération on obtient un classifieur meilleur qu'avant. On peut donc arrêter les itérations à tout moment, même si le résultat n'est pas l'optimum.

❖ **Machines à support de vecteurs (ou SVM) [11]:**

Cette technique - initiée par Vapnik - tente de séparer linéairement les exemples positifs des exemples négatifs dans l'ensemble des exemples. Chaque exemple doit être représenté par un vecteur de dimension n. La méthode cherche alors l'hyperplan qui sépare les exemples positifs des exemples négatifs, en garantissant que la marge entre le plus proche des positifs et des négatifs soit maximale. Intuitivement, cela garantit un bon niveau de généralisation car de nouveaux exemples pourront ne pas être trop similaires à ceux utilisés pour trouver l'hyperplan mais être tout de même situés franchement d'un côté ou l'autre de la frontière. L'efficacité des SVM est supérieure à celle de toutes les autres méthodes sur la classification de textes. Son efficacité est aussi très bonne pour la reconnaissance de formes. Un autre intérêt est la sélection de Vecteurs Supports qui représentent les vecteurs discriminant grâce auxquels est déterminé l'hyperplan. Les exemples utilisés lors de la recherche de l'hyperplan ne sont alors plus utiles et seuls ces vecteurs supports sont utilisés pour classer un nouveau cas. Cela en fait une méthode très rapide.

IV.2.La classification non supervisée (ou clustring) [06]

La classification non supervisée est un des champs les plus importants dans l'exploitation de données, dans cette approche de classification, les classes possibles ne sont pas connues à l'avance, et les exemples disponibles sont non étiquetés, le but est donc de regrouper dans un même cluster (ou groupe) les objets considèrent comme similaires, pour constituer les classes.

Dans ce cas, le problème est alors de définir cette similarité entre objets. Typiquement, la similarité entre objets est estimée par une fonction calculant la distance entre ces objets. Une fois cette fonction distance défini, la tâche de clustering consiste à réduire au maximum la distance entre membre d'un même cluster, tout en augmentant au maximum la distance entre clusters [**07**].

On rencontre une grande quantité de méthode de clustering, mais, étant donné leur nombre considérable, nous nous limitons à en citer que deux parmi les plus utiliser en pratique

- La classification hiérarchique.
- La classification non hiérarchique à partitionnement.

IV.2.1 .But du clustering [9] :

Le but du clustering est de regrouper dans un même cluster les données considérées comme similaire pour constituer des classes.

Cette similarité est estimée par la fonction calculant la distance entre les données. La tache du clustering consiste à réduire au maximum la distance entre membre d'un même cluster, toute en augmentant au maximum la distance entre clusters.

IV.2.2. Différentes approches du clustering :

On peut regrouper les approches du clustering dans deux grandes familles : classification hiérarchique et classification non hiérarchique (par partitionnement).

IV.2.2 .1.Classification hiérarchique (CH) [6]:

Etant donné un ensemble d'observations, une hiérarchie sur cette ensemble est une collection de groupes d'observations (clusters) tels que :

L'ensemble complet des données est un cluster.

Chacune des observations est un cluster (singleton).

Etant donnée deux clusters d la hiérarchie ou bien ils n'ont aucune observation en commun, ou bien l'un est inclus dans l'autre (pas de chevauchement).

Une telle structure peut se représenter par un « dendrogramme » ou un « arbre ».

La figure ci-dessous représente un exemple d'une hiérarchie :

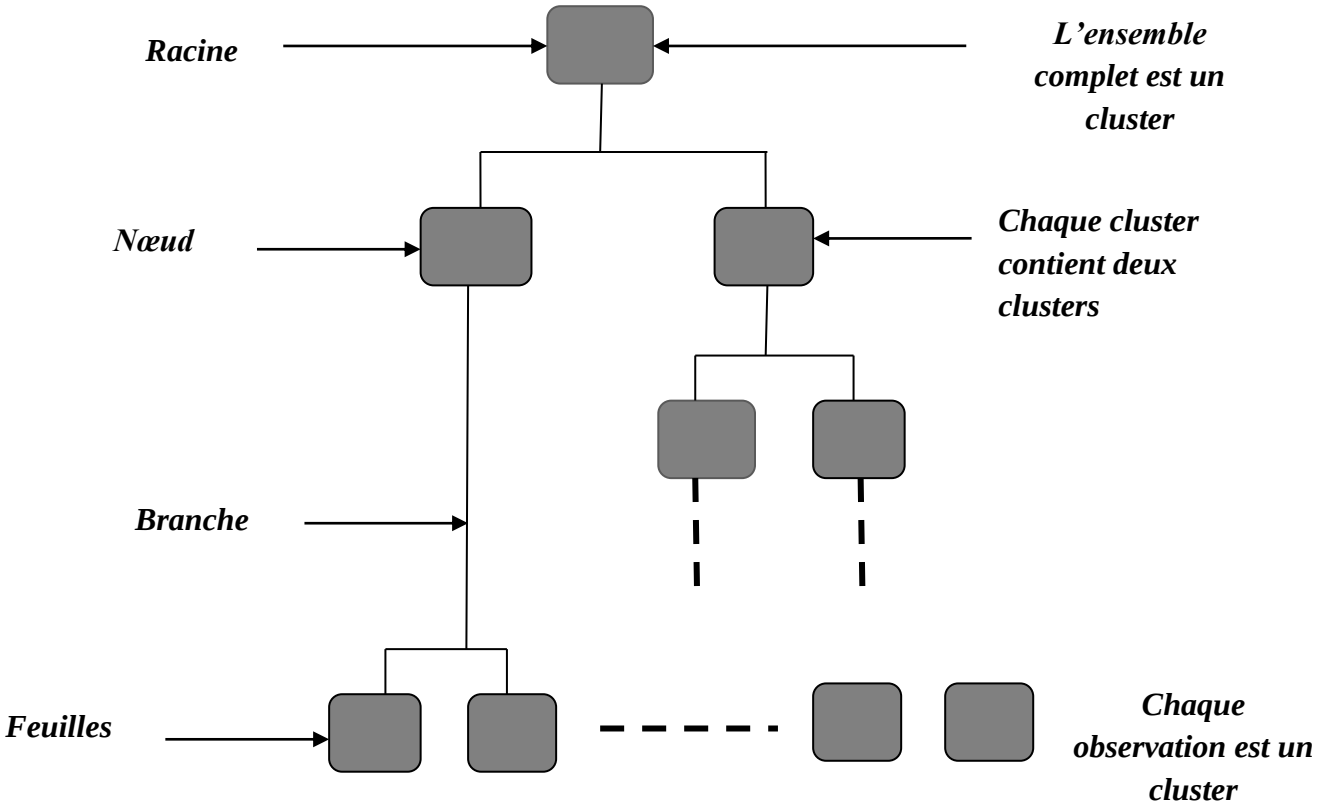

Figure II. 3 : Une hiérarchique

Le cluster contenant toute les observations s'appelle la « **racine** » de l'arbre.

Au bas de l'arbre, les singletons (clusters ne contenant qu'une seule observation) s'appellent des « **feuilles ».**

Chaque cluster dans l'arbre s'appelle un « **Nœud** ».

Chaque cluster (a l'exception des singletons) a habituellement deux enfants « enfant ».

Une ligne joignant un nœud a l'un de ses enfants s'appelle une **« branche ».**

Tous les chemins menant d'un nœud au feuilles qui en dépendant n'ont pas nécessairement le même nombre de branche.

Hiérarchie et typologie :

Une hiérarchie permet de construire beaucoup de topologies : toute section supérieure de l'arbre définit est un ensemble de clusters partitionnant l'ensemble des données, donc une typologie consiste en une partition de l'ensemble des données en clusters qui sont :

- ✓ Compacts.
- ✓ Bien séparés les uns des autres et facilement interprétables.
- ✓ Facilement interprétables.

La construction de typologies à partir de hiérarchie doit résoudre deux problèmes :

Construire la hiérarchie :

Etant donnée un cluster, quelle est la meilleure manière de le scinder en deux **« enfant »,** ou bien a l'inverse comment choisir deux clusters dans le but de les fusionner en un unique cluster parent, ces deux questions donnent naissance respectivement aux hiérarchies descendantes et ascendantes.

Dans la classification hiérarchique les clusters crées sont emboités de manière hiérarchique les uns dans les autres. On distingue alors :

- ➢ La CH ascendante (ou agglomerative) qui part des documents seuls (singleton) que l'on regroupe en sous-ensemble, qui sont a leur tour regroupé, et ainsi de suite.
- ➢ La CH descendante (ou division) qui part de l'ensemble de tous les documents et les fractionne en un certain nombre de sous-ensemble, chaque sous-ensemble et ainsi de suite.

Pour déterminer les classes à fusionner, on utilise le critère d'agrégation. Le même critère peut être utilisé au sens inverse comme critère de division pour fractionner les classes.

Critère d'agrégation :

Le critère d'agrégation permet de comparer les classes deux a deux pour sélectionner les classes les plus similaires suivant un certain critère. Les critères classiques sont le plus proche voisin, le diamètre maximum, la distance moyenne et la distance entre les centres de gravités.

- ✓ **Plus proches voisin :** la distance entre la classe C_p et la classe C_q est le plus petite distance entre un élément de cp et un élément de cq :

$$D(Cp,Cq)=\min\{distance(i,j) ; i \in Cp \text{ et } j \in Cq\}.$$

- ✓ **Diamètre maximum :** la distance entre la classe Cp et la classe Cq est la plus grande distance entre un élément de Cp et un élément de Cq :

$$D(Cp, Cq)=\max\{distance(i, j) ; i \in p \text{ et } j \in q\}$$

- ✓ **Distance moyenne :**

La distance entre la classe Cp et la classe Cq est la moyenne des distances entre les éléments de cp et les éléments de cq :

$$D(C^p, C^q) = \frac{\sum_{i,j} Dist\{i, j\}}{Cardinal(Cp)*Cardinal(Cq)}, \quad i \in Cp \text{ et } j \in Cq$$

- ✓ **Distance entre centres de gravité :** Si *Gp* est le centre de gravité de la classe *Cp* et si *Gq* est le centre de gravité de la classe *Cq* alors la distance entre la classe *Cp* et la classe *Cq* est la distance entre leurs centres de gravités.

D (*Cp, Cq)*=D (*Gp, Gq*)

Ce critère n'a de sens que si le calcul du centre de gravité possède lui-même un sens sur les données de l'étude.

La classification hiérarchique est la plus connue et la plus ancienne des méthodes de classification. Cette problématique est souvent rencontrée dans divers domaines, notamment les plus anciens tel dans, la classification des espèces, puis en anthropologie , en chimie, en microbiologie et en génétique, en gestion, et plus récemment en RI sur web. Pour rappeler, les méthodes hiérarchiques sont caractérisées, soit par le critère d'agrégation de clusters similaires, la stratégie est alors dite ascendante ou agglomération, soit par le critère de division de clusters dissimilaires, c'est la stratégie descendant.

IV.2.2.2 .La classification hiérarchique ascendante [15]:

La classification hiérarchique ascendante (ou agglomerative) procède par fusions successives des clusters déjà existants, on utilise le critère d'agrégation. A chaque étape, les deux clusters qui vont fusionner sont ceux dont la « distance » est la plus faible.

La classification ascendante hiérarchique (CAH) considère initialement toute les observations comme étant des clusters ne contenant qu'une seule observation (singleton), et leur distance est généralement définie par leur distance euclidienne.

Dans cette classification, on commence d'abord par réunir dans un cluster à deux observations les deux observations les plus proches, le processus continu en fusionnant à chaque étape les deux clusters les plus proches au sens de la distance choisie.

Le processus s'arrête quand les deux clusters restant fusionnent dans l'unique cluster contenant toutes les observations.

Les méthodes ascendantes partent des singletons (classe à un seul élément) et procèdent par agrégations successives, elles consistent à effectuer une suite de regroupements en classe de moins en moins fines en agrégeant à chaque étape les objets ou les groupes d'objets les plus proches selon le principe suivant :

***Principe de la méthode**

- initialement chaque individu est supposé représenter un groupe.
- Trouver les deux groupes les plus proches.
- Grouper ces deux groupes en un nouveau groupe.
- Itérer jusqu'au nombre de groupes désiré.

Le schéma suivant illustre cette démarche :

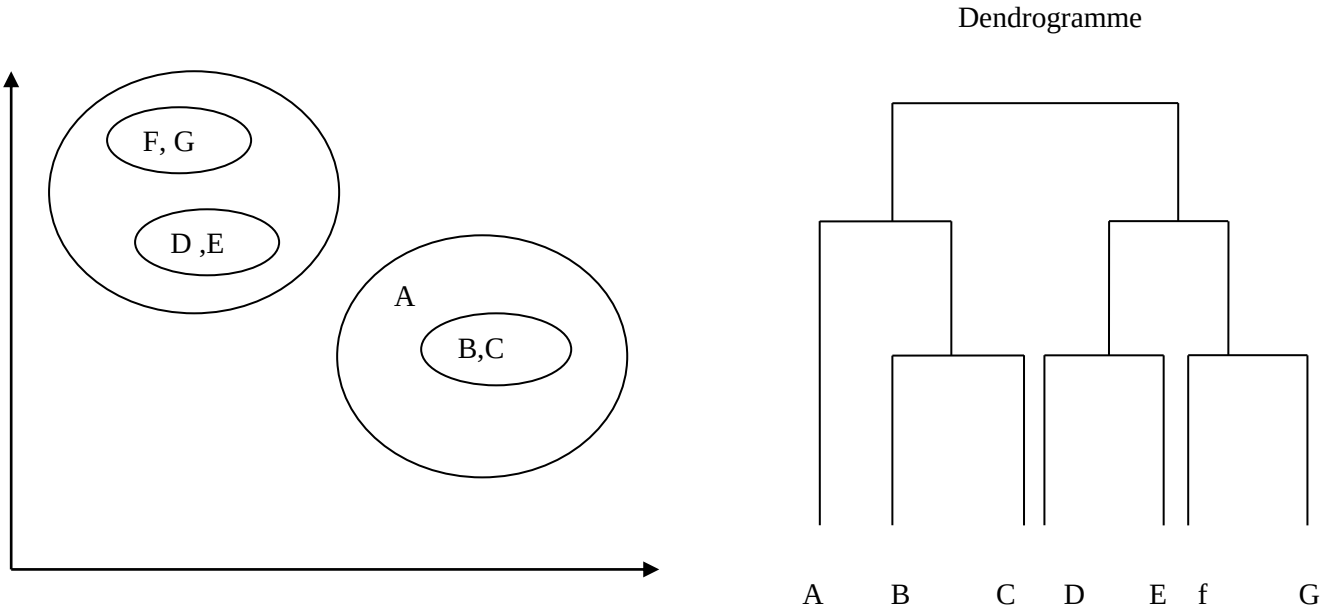

Figure II.4 : clustering hiérarchique

***Variante de la méthode**

- calcule les distances de tous les points deux par deux.
- Associer tous les points dont la distance ne dépasse pas un certain seuil.
- Calculer le centre de chaque cluster.
- Répéter le processus avec les centres et un nouveau seuil jusqu'à l'obtention du nombre de clusters souhaité.

IV .2.2.3.La Classification hiérarchique descendante [15]:

La classification hiérarchique descendante (division) procède de façon inverse.

Elle considère l'ensemble des données comme un gros cluster unique et le scinde en deux clusters « descendants ». La scission s'opère de façon a ce que la distance entre les deux descendants soit la plus grande possible, de façon a créer deux clusters bien séparés. Cette procédure est ensuite appliquée a chacun des descendants (procédure récursive) jusqu'à ce qu'il ne reste que des clusters ne contenant qu'une seule observation (singleton).

❖ **Comparaison entre l'approche ascendante et descendante [15] :**

A première vue, ces deux approches semblent être comme des images symétriques l'une de l'autre.

Le dendrogramme [3] :

Puisque les méthodes hiérarchiques fusionnent les groupes a des degrés décroissants de ressemblance, il est naturel de représenter les résultats de la classification au moyen d'une structure arborescente que l'on appelle dendrogramme.

L'algorithme de construction d'un dendrogramme :

i. Placer les observations selon un ordre quelconque de gauche a droite.

ii.S'il ne reste qu'un seul groupe, on termine.

iii .Prendre les groupes compris entre les groupes qui fusionnent et les déplacer rapidement a la droite de la dernière observation du groupe fusionnée situé le plus a droite.

iv. Retourner a ii.

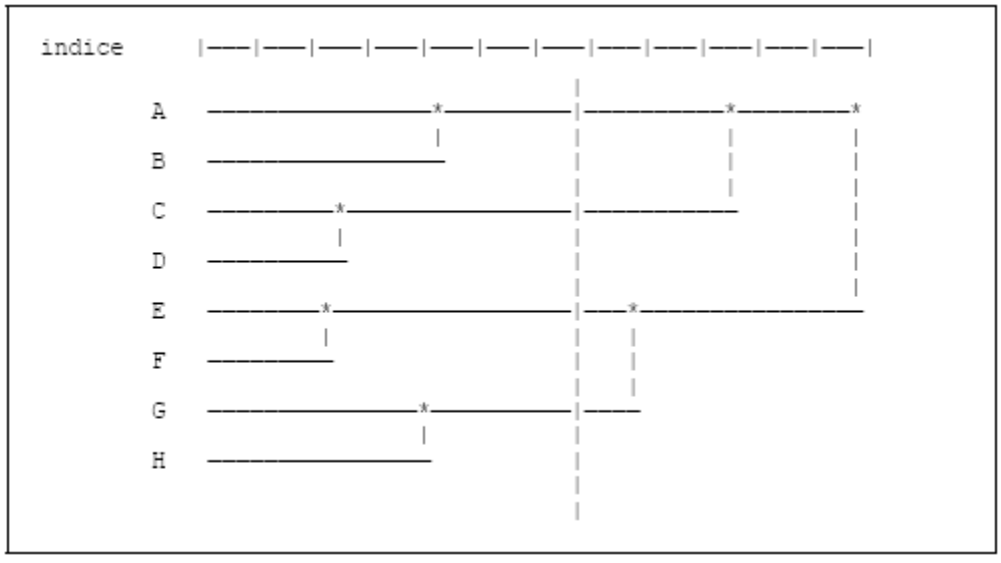

Figure II.5: Dendrogramme représentant une classification a huit éléments

IV.3. Classification non hiérarchique (par partitionnement) [15]:on distingue deux types, partitionnement dur et partitionnement doux :

- **Partitionnement dur :**

Il consiste à découper l'espace des observations en un certain nombre de régions disjointes, définies par des frontières, et de détecter que toutes les observations situées dans une même région de l'espace appartiennent a une même classe.

Chaque classe est représenter par un prototype, observation virtuelle sensée être la plus représentative de la population de la classe. Le prototype d'une classe sera le plus souvent le barycentre des observations de la classe.

Ces prototypes sont positionnés de façons itérative dans les zones a forte densité, et les observations sont affecter aux classes sur la base d'un critère de proximité aux différents prototypes.

Il existe de nombreuses technique de partitionnement, la plus connu étant k-means(k-moyennes) et ses variantes.

- **Partitionnement doux :**

L'idée selon laquelle chacune des classes réelles, sous-jacentes, occupe une région limitée de l'espace peut paraitre irréaliste.

Mélanges de modèles :

Cette approche reprend l'ide selon laquelle les classes ont des distributions multi normales.

Pour un nombre supposé de classes K donnée, on considère alors l'ensemble des données est en fait un échantillon tiré de K distribution multi normales, chacune affecté d'une probabilité a priori. On cherche alors les centres et matrices de covariances de ces distributions, ainsi que les probabilités a priori, de façon a maximiser les vraisemblances des données, ce qui se fait par l'algorithme classique dit « EM » (Expectation Maximization).

Classification floue :

Dans les mélanges de modèles, les classes se chevauchent mais chaque observation appartient à une classe et une seule (bien que la détermination de cette classe ne puise être faite que de façon probabiliste). En classification floue, a l'inverse, on reconnait qu'une observation peut appartenir simultanément à plusieurs classes a des degrés divers dont la somme est égal a un.
La classification floue affecte alors a chaque observation des degrés d'appartenance aux diverses classes d'une façon cohérente avec la répartition

géométrique des données. Pour chaque observation, la somme des degrés d'appartenance a chacune des classes est égale a un.

Distance et densité [15] :

Les classes rencontrées dans les applications ont souvent des distributions uni-modales : à partir d'un noyau central, la densité des observations décroit de façon monotone dans toutes les directions de l'espace. Beaucoup de techniques de classification non supervisée s'appuient sur cette image et portent une attention particulière aux ensembles d'observations ayant entre elles de faibles distances (régions de forte densité). Elles le font de diverses façons :

- En utilisant les distances entre observations pour construire les classes (ex : méthodes hiérarchique).
- En reconnaissant que les zones peuplées mais de faible inertie autour de leurs barycentres sont des zones de forte densité (k-means).
- En faisant de l'estimation de densité de façon paramétriques (modèles de mélanges) ou non paramétriques (méthodes basées sur l'estimation de densité par k-premiers voisins).

Qualité d'une partition [15]: un algorithme de classification non supervisée produira toujours des classes. Mais ces classes reflètent-elles la structure de données ? Chaque classe abrite-t-elle un cluster dense et ses envions immédiats, comme le souhaite l'analyste ?

Il ya deux raison pour lesquelles une topologie peut être non satisfaisante :

1. les données ne sont que faiblement structurées. Elles ne contiennent pas de cluster bien identifiable, et sont plus ou moins uniformément réparties.

2. il existe des clusters bien identifiables, mais l'analyse n'a pas réussi a les mettre en évidence.

Il est donc indispensable de pouvoir quantifier la quantité d'une partition, c'est-à-dire d'apprécier le fait que la partition correspond à des clusters présents dans les données.

Il existe de nombreux critères de qualité d'une partition, ce qui signifie qu'aucun n'est universellement satisfaisant. Parmi les plus utilisé, citons :

- Divers indices comparant la variance intra-classe à la variance totale (R-square, semi-partial R-square, Root-Mean-square standard déviation…).
- Des indices « pseudo-statistiques », comme le F de Fisher.
- Des indices dérivés des indices classiques des modèles linéaires (principalement AIC et BIC).

Et de nombreux autres….

Pratique de la classification non supervisée [15] :

Le contenu théorique des techniques de classification non supervisée est souvent simple, ce qui laisse à penser, a tort, que la classification non supervisée est une activité dénuée de difficultés.

- **Nature des variables :**

La grande majorité des technique de classification, reposant sur des notions métriques (distance, densité), exigeant que les variables soient quantitatives. Cependant, il est rare que les données ne soient décrites de façon native que par des variables quantitative. Il faut donc procéder a une phase initiale de codage des variables nominales sous forme nominale ceci se fait :

Soit en codant les modalités des variables nominales sous forme disjonctive complète.

Soit en soumettant les variables quantitatives a une analyse des correspondances multiples (ACM, une méthode statistique d'analyse des données) et en remplaçant dans la table des données les variables nominales pour les facteurs issus de l'ACM.

Ya t-il des clusters dans les données

Le plus souvent, il n existe pas de raison forte de croire en l'existence de tels clusters naturels et pourtant, même sur des données uniformément distribuées, et donc sans structure, les algorithmes produiront des classes. Mais le résultat d'une classification non supervisée (« typologie ») n'a de sens que si les classes identifiées correspondent effectivement à des clusters présents dans les données.

Affirmer qu'un ensemble de données contient des clusters bien définis est difficile. Deux idées directrices d'aborder cette question :

- La stabilité des classes.
- L'influence du nombre de classes.

Plus précisément :

- **Stabilité :**

Plusieurs typologies sont construites en laissant à chaque fois de coté un sous-ensemble des données choisi aléatoirement. Si les clusters sont bien définis, cette faible réduction du nombre d'observations utilisées pour construire les typologies aura peu d'influence sur le résultat final, et les diverses typologies obtenues seront très similaires.

Par contre, si les données sont plus ou moins uniformément distribuées, les ensembles de classe finals seront très différents (penser a des points uniformément distribuées, les ensembles de classes finales seront très différents).

- **Nombre de classe :**

Si l'analyse est conduite avec le bon nombre de classes, il est possible d'obtenir une typologie particulièrement convaincante. Par contre, si les données ne sont que faiblement structurées, toutes les topologies obtenues seront également « mauvaises », quel que soit le nombre de classes.

Toute les méthodes non supervisée permettent a l'analyste de choisir le nombre de classes de partition finale (niveau de la coupure de l'arbre en classification hiérarchique, nombre de prototype en K-means ...etc).

Cette possibilité est une arme à double tranchant. Si la population contient effectivement des clusters bien identifiables, mais le choix du nombre de classes finales ne correspond pas au nombre de clusters, alors la partition :

Soit regroupera à tort des clusters séparés (pas assez de classes).

Soit découpera en plusieurs classes des clusters homogènes (trop de classes).

Dans le premier cas, la typologie sera dite « trop grossière », ou ayant « une granularité trop forte ». Dans le deuxième cas, la typologie sera dite « trop fine », ou ayant une « granularité trop faible ».

La recherche du nombre approprié de classes est toujours une phrase indispensable dans la construction d'une classification de donnée, mais elle est longue est souvent ambigüe.

Classification sur variables [5]:

Il est possible non supervisé non pas sur des observations mais sur des variables. La dissimilarité entre deux variables (similaire a la distance entre observations) est en général définie a partir de leur coefficient de corrélation, des variables fortement corrélées sont considérées comme proches.

Une telle classification peut être utile lorsque les variables sont nombreuses et présentent un fort risque de colinéarité. Apres classification, toutes les variables d'une classe seront alors remplacées par une unique variable synthétique représentant au mieux l'ensemble des variables de la classe.

V .Technique de classification :

Les nombreuses techniques de classification sont nées le plus souvent de la nécessité de résoudre des problèmes pratiques issus de domaines très variés.

- **Méthodes statistiques:**

Ces méthodes présentent le grand avantage qui permettent de regrouper les observations totalement inconnues c'est à dire sur lesquelles on ne dispose d'aucune information que celle qui peut être extraite des observations à classer elle mêmes.

Ces méthodes considère que les données forme un échantillon aléatoire (x_1,x_2,x_3,…x_n) issu d'une population , et de s'appuyer sur l'analyse de la distribution de probabilité de cette population pour définir une classification elle repose sur l'étude de la fonction de densité de probabilité sous jacente a la distribution de l'ensemble des observations soumise a l'analyse.

Les méthodes statistiques sont divisées en deux catégories :

- Les méthodes non paramétriques.
- Les méthodes paramétriques.

- **Méthodes non paramétriques :**

Dans ce type de méthodes on se base beaucoup plus sur la fonction de probabilité si cette dernière n'est pas connue a priori, alors on peut l'estimer et cela en faisant appel a l'estimateur du noyau de k plus proche voisin. Le travail consiste a la recherche des modes de la FDP(fonction de densité de probabilité) estimé. Les région de l'espace caractérisées par une grande densité d'observations correspondent aux noyaux des classes et peuvent être localisées en détectant les modes de FDP.

Il existe plusieurs méthodes pour détecter les modes :

a)Recherche des maxima locaux :

Les modes peuvent être détectés en remontant les pentes de la FDP selon la direction de son gradient. L'assignation de l'observation se fait par rapport au mode le plus proche.

b) analyse de convexité :

Dans ce cas les modes sont assimilés à des régions de l'espace ou cette fonction est concave.

c)extraction des contours des modes :

Les modes sont considérés ici comme des régions délimitées par les contours.

Apres réalisation d'un filtrage de la FDP, des opérations différentielles.

- **Méthodes paramétriques :**

Dans ces méthodes la fonction de densité de probabilité est connue a priori.

Le problème de la classification est de déterminer les caractéristiques d'un mélange de FDP représentant les distributions des observations provenant de chacune des classes en présence dans l'échantillon a analysé.

Méthodes métrique :

On distingue deux principale approches de type métrique permettant de classifier les objet, l'une consiste a établir la hiérarchie des classes tandis que l'autre effectue le partitionnement de l'espace des observation.

- **Classifications hiérarchiques :**
  La plupart des méthodes représentent les résultats de classification sous forme d'une structure plane.
  Il est cependant naturel de représenter la connaissance ou l'objet sous forme d'une hiérarchie de classe ou de concepts. Dans une classification hiérarchique une classe peut être divisée en plusieurs sous classes, l ensemble des classes formant alors une hiérarchie (représenté par un arbre).
  Les algorithmes de classification hiérarchiques permettent de construire ce type de résultat, on distingue deux types d'approche :
  - Les méthodes agglomératives parlent d'un grand nombre de classes (éventuellement une classe par objet) et fusionnent les classes similaires entre elles.
  - Les méthodes divisées partent de l'ensemble de données et le divise en classe qui sont alors divisées récursivement.

- **La classification non hiérarchique :**

Les méthodes non hiérarchiques cherchent à diviser la population initiale en groupes disjoints tels que selon un critère choisi a priori, deux individus d'un même groupe ont entre eux un minimum d'affinité. Il existe dans la littérature statistique une profusion de méthodes et de critères de classification no hiérarchique. Mais ces méthodes s'apparentent à deux familles.

- ✓ La famille des méthodes de type « centre mobile », telle que la méthode de nuée dynamique.
- ✓ La famille des méthodes de type « relationnelles » telle que la méthode des partitions centrale, la méthode de l'analyse relationnelle…etc.

- **Techniques de partitionnement :**
  L'idée générale est de découper l'espace des observations en un certain nombre de régions disjointes, définies par les frontières, et de décréter que toutes les observations situées dans une même région de l'espace appartiennent à une même classe. Chaque classe est représentée par un prototype.

Ces prototypes sont positionnées de façon itérative dans les zones a forte densité, et les observations sont affectées aux classes sur la base d'un critère de proximité aux différents prototypes.
Il existe de nombreuses techniques de classification par partitionnement, la plus connue étant K-means (ou « K-moyennes ») et ses variantes.

- **Algorithme des centres mobiles :**

L'algorithme des centres mobiles considère initialement un centre de gravité pour chaque classe. Il consiste à diviser l'ensemble des observations(ou des individus) en un nombre de classe prédéterminé par l'utilisateur. Pour chaque classe une observation est sélectionnée aléatoirement parmi l'ensemble des observations dans le but de construire son centre de gravité initial. Chaque observation ensuite est affectée au centre le plus proche. Chaque observation ensuite est affectée au centre le plus proche.

On obtient ainsi une première partition de l'ensemble des observations. Une fois que toutes les observations ont été affectées a leurs classes le centre de gravité de chaque classe est réactualisé et le processus est réitéré jusqu'à la convergence de l'algorithme obtenue en minimisant a chaque itération une fonction cout ou fonction de ressemblance.

- **Algorithme des k means [09]:**

L'algorithme k-means, créé par McQueen, 1967 est l'algorithme de clustering le plus connu et le plus utilisé car il s'avère être très simple à mettre en oeuvre et efficace. Il suit une procédure simple de classification d'un ensemble d'objets en un certain nombre k de clusters, k fixé à priori. Dans cet algorithme, chaque clusters est caractérisé par son centre qui se trouve être la moyenne des éléments composant le cluster.
L'algorithme k-means s'exécute en 4 étapes :

- **Etape 1 :** choisir aléatoirement k objets qui forme ainsi les k clusters initiaux.
- **Etape 2 :** affecter les objets a un cluster. Pour chaque objet x, le prototype qui lui est assigné est celui qui est le plus proche de l'objet, selon une mesure de distance, (habituellement la distance euclidienne est utilisée).
- **Etape 3 :** une fois tous les objets placés, recalculer les centres des k clusters.
- **Etape 4 :** répéter les étapes 2 et 3 jusqu' à ce qu'il n'y ait plus de changement dans les clusters.

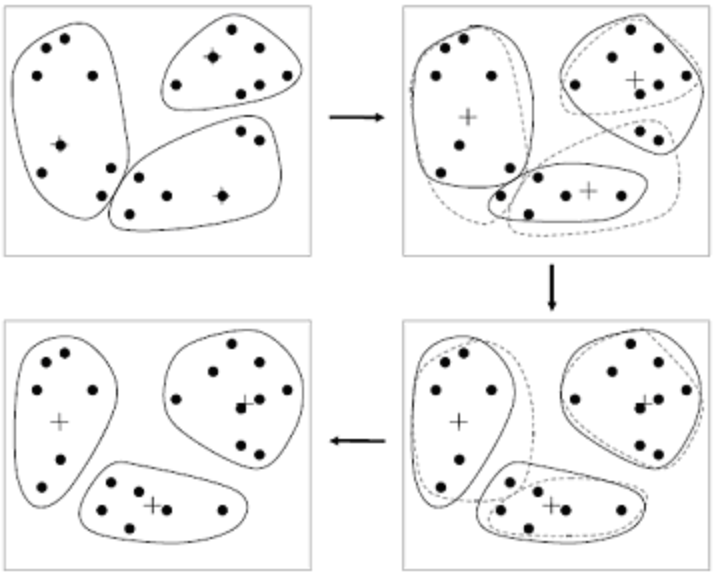

Figure II.6 : Différentes étapes de l'algorithme *McQueen* .

VI. Conclusion

Dans ce chapitre, nous avons présenté une vue générale de l'indexation et l'unité de la classification de données dans le processus d'indexation. Nous avons introduits des concepts appliqués. Dans notre travail, nous utiliserons l'algorithme de classification probabiliste naive bayes afin de l'appliquer sur la classification des éléments d'une base de connaissances ontologiques. La présentation des ontologies ainsi que les concepts de bases associés fera l'objet du prochain chapitre.

Chapitre III :WordNet

I .Introduction :

La masse de plus en plus croissante d'information dans tous les domaines a généré un besoin capital d'organisation et de structuration des contenus de documents, disponibles généralement sur le web. Les ontologies sont un moyen prometteur qui ne cesse de donner ses preuves. Leurs applications sont multiples : indexation, recherche d'informations, web sémantique, e-learning…

Dans ce chapitre nous allons présenter les notions de base d'une ontologie nécessaires à la compréhension de la structure de l'ontologie linguistique WordNet .

II. Définition d'une ontologie : [26]

En philosophie, l'ontologie (de *onto-* tiré du grec « étant », participe présent du verbe « être ») est l'étude de l'être en tant qu'être, c'est-à-dire l'étude des propriétés générales de ce qui existe.

Par analogie, le terme est repris en informatique et en science de l'information, où une ontologie est l'ensemble structuré des termes et concepts représentant le sens d'un champ d'informations, que ce soit par les métadonnées d'un espace de noms, ou les éléments d'un domaine de connaissances. L'ontologie constitue en soi un modèle de données représentatif d'un ensemble de concepts dans un domaine, ainsi que des relations entre ces concepts. Elle est employée pour raisonner à propos des objets du domaine concerné. Plus simplement, on peut aussi dire que l' « ontologie est aux données ce que la grammaire est au langage ».

Les concepts sont organisés dans un graphe dont les relations peuvent être :

- des relations sémantiques .
- des relations de subsomption.

L'objectif premier d'une ontologie est de modéliser un ensemble de connaissances dans un domaine donné, qui peut être réel ou imaginaire.

Les ontologies sont employées dans l'intelligence artificielle, le Web sémantique, le génie logiciel, l'informatique biomédicale ou encore l'architecture de l'information comme une forme de représentation de la connaissance au sujet d'un monde ou d'une certaine partie de ce monde. Les ontologies décrivent généralement :

- individus : les objets de base.
- classes : ensembles, collections, ou types d'objets.
- attributs : propriétés, fonctionnalités, caractéristiques ou paramètres que les objets peuvent posséder et partager.
- relations : les liens que les objets peuvent avoir entre eux.
- événements : changements subis par des attributs ou des relations.
- métaclasse (web sémantique) : des collections de classes qui partagent certaines caractéristiques.

Selon Gruber, « l'ontologie est une spécification explicite d'une conceptualisation », c'est-à-dire qui permet de spécifier dans un langage formel les concepts d'un domaine et leurs relations.

II .1. Type d'ontologie [21]:

Les ontologies peuvent être classées suivant le degré de formalisme, selon les objets modélisés et selon le degré de granularité.

II.1.1.Selon le degré de formalisme :

[Uschold & Gruninger] a identifié selon le degré de formalisme quatre types d'ontologies : les ontologies informelles, les ontologies semi-informelles, les ontologies semi-informelles, les ontologies semi-formelles, et les ontologies rigoureusement formelles.

- **Les ontologies informelles :** elles sont exprimées en langage naturel.
- **Les ontologies semi-informelles** : elles sont exprimées sous une forme limitée restreinte et structuré du langage naturel(en utilisant des modèles). Pour ce faire des patrons on été mise en œuvre.
- **Les ontologies semi-formelles :** elles sont exprimées dans un langage défini artificiellement.
- **Les ontologies formelles :** Elles sont exprimées dans un langage contenant une sémantique formelle, des théorèmes et des preuves de propriétés telles que la robustesse, l'exhaustivité, la complétude et la constance.

II.1.2.Selon les objets modélisés [21] :

Les ontologies ont été regroupées dans [Gomez-pérez & al 2004] en se basant sur les objets modélisés afin de répondre à un but précis. Ces ontologies sont classées selon : les ontologies supérieures, les ontologies du domaine, les ontologies de tâche, les ontologies d'application, les ontologies de représentation, et les ontologies de raisonnement.

- **Les ontologies supérieures :**

Dites aussi de haut niveau « top level ontologies ou upper level ontologies » [Guarino 1998], ou alors « ontologies génériques ou noyaux d'ontologies » ou « méta- ontologies » ou « Ontologies de sens commun/général». Ces ontologies sont universelles, réutilisables, et référença bles à partir des concepts des autres niveaux d'ontologies.

Les ontologies supérieures (upper level) représentent des concepts généraux comme l'espace, le temps ou la matière. Elles sont universelles. Les concepts des trois autres types d'ontologie peuvent y faire référence.

Elles comportent des concepts abstraits (généraux) subsumant les concepts existant dans les différents domaines. Une ontologie de haut niveau est généralement conçue afin de réduire

les incohérences des termes définis plus bas dans la hiérarchie.

Les ontologies génériques définissent des concepts considérés comme génériques à plusieurs domaines. WordNet [Miller 1988] par exemple est une ontologie dont le but est de représenter la langue naturelle anglaise.

WordNet est un système de références lexicales dont la conception a été inspirée par les théories de la mémoire linguistique humaine. Elle est composée d'ensembles de synonymes appelés synsets, où chaque terme est regroupé en classes d'équivalence sémantique. Chaque ensemble de synonymes représente un concept particulier. Chaque terme appartient de plus à une catégorie lexicale donnée (nom, verbe, adverbe, adjectif). Un terme peut appartenir à plusieurs synsets et à plusieurs catégories lexicales. Les ensembles de synonymes sont associés par des relations sémantiques : généricité/spécificité, antonymie (relation entre ensembles de mots qui, par leur sens, s'opposent).

- **Les ontologies du domaine :**

Ce type d'ontologie décrit un vocabulaire appartenant à un domaine générique donné tel que la médecine. Elles ne sont pas propre à une tâche précise et présentent une bonne précision et se rapportent à un certain type d'artifacts. Une ontologie informatique est une représentation de propriétés générales de ce qui existe dans un formalisme supportant un traitement rationnel [Gan 2006][Ontologie]. C'est le résultat d'une formulation exhaustive et rigoureuse de la conceptualisation d'un domaine. Cette conceptualisation est souvent qualifiée de partielle car, en l'état de l'art, il est illusoire de croire pouvoir capturer dans un formalisme toute la complexité d'un domaine. Notons aussi que le degré de formalisation d'une ontologie varie avec l'usage qui en est envisagé.

Exemple :

- **L'ontologie médicale :** est l'étude de qui «est» en médecine et du processus de leur formation. Elle s'intéresse à la généalogie des entités médicales: les maladies, les signes cliniques, les syndromes cliniques, les symptômes, les lésions, les syndromes lésionnels, les anomalies biologiques et les anomalies radiologiques.**[30]**

- **Ontologies linguistiques :** Il s'agit d'ontologies servant à décrire le vocabulaire d'une langue. Elles sont plus particulièrement destinées à être utilisées dans une perspective de Traitement Automatique de la Langue (TAL).WordNet de l'Université Princeton (Felbaum, 1997) en est sans doute la plus connue des ontologies linguistiques. Il s'agit d'une base de données lexicales de l'anglais (appelée ontologie à cause de la structure hiérarchique à laquelle elle répond), qui comprend différents types d'entités lexicales (mots-composés, collocations, locutions), mais dont l'unité lexicale constitue l'unité linguistique principale de description. Sa structure n'est cependant pas organisée autour des unités lexicales individuelles. En effet, la base de connaissance est organisée principalement à partir de la relation d'hyper-/hyponymie connectant non seulement les unités lexicales, mais des regroupements d'unités lexicales synonymes, appelés synsets.

Chaque synset fonctionne comme unité de structuration de l'ontologie, et représente plus qu'un mot lui-même ; dans une perspective cognitive, le synset s'approche d'une unité psycholinguistique de raisonnement. En ce sens, WordNet peut bel et bien être considéré comme une ontologie, dans la mesure où des concepts (et pas seulement des « mots ») y sont explicitement représentés. **[22]**

- **Les ontologies de tâche :**

Ces ontologies sont spécifiques à une tâche générique, telle que la vente, et indépendamment des autres du domaine d'application.

- **Les ontologies d'application :**

Ces ontologies correspondent à l'exécution d'une tâche particulière et leur domaine d'application est restreint. Elles sont souvent des spécialisations des ontologies du domaine et des ontologies de tâche.

- **Les ontologies de représentation :**

Ce type d'ontologies est un cas particulier d'ontologie supérieure qui regroupe des concepts déjà utilisé pour formaliser les connaissances. Indépendamment des domaines [Guarino & al 1994] puisqu'elles décrivent des primitives cognitives communes. Parmi les ontologies de représentation, on trouve « frame ontology ».qui définit, de manière formelle. Les concepts utilisés particulièrement dans des langages à base de frame : classes, sous- classes, attributs, valeurs, relations et axiomes [Gruber 1993].

- **Les ontologies de raisonnement :**

Ces ontologies regroupent les processus de raisonnement appliqués aux connaissances qui forment eux-mêmes un domaine de connaissances. On parle particulièrement d'ontologies développées pour représenter des connaissances génériques mises en œuvre lors de la résolution automatique de problème.

II.1.3.Selon la granularité [21] :

La classification suivante est en fonction du degré de granularité, c'est-à-dire quel niveau de détail des objets de la conceptualisation est préconisé. En fonction de l'objectif opérationnel, une connaissance plus ou moins fine du domaine est nécessaire et des propriétés considérées comme accessoires dans un contexte peuvent se révéler indispensables dans un autre. Ce type d'ontologies est classé suivant : granularité fine et granularité large.

- **Granularité fine :**

Correspondent à des ontologies très détaillées, possédant aussi un vocabulaire plus riche capable d'assurer une description détaillée des concepts pertinents d'un domaine ou d'une tâche [Furst 2004].

- **Granularité large :**

Correspondent à un vocabulaire moins détaillé. Les ontologies de haut niveau ont une granularité large, du fait que les notions sur lesquelles elles portent peuvent être raffinées par des notions plus spécifiques [Furst 2004] .

II.2.Les éléments d'une ontologie [21] :

La connaissance dans les ontologies est principalement formalisée en utilisant cinq types de composants qui sont [Gruber 1993] : classes ou concepts, relations entre concepts, fonctions, axiomes, instances et rôles.

- **Classe ou concept :**

Les connaissances portent sur des objets auxquels on fait référence à travers des concepts qui sont habituellement organisés sous forme hiérarchisée dans l'ontologie.

Un concept peut représenter un objet matériel, une notion, une idée [Uschold &King 1995] et peut être divisé en trois parties :

Un terme ou un label qui est l'expression linguistique utilisée couramment pour y faire référence.

Une notion qui désigne, au sens de la représentation des connaissances, l'intention du concept. Elle contient sa sémantique qui est définie à l'aide de propriétés, d'attributs, de règles et de contraintes.

Un ensemble d'objets auxquels le concept fait référence forment l'extension du concept, autrement dit les instances.

- **Les relations :**

Les relations représentent un type d'interaction entre les concepts du domaine. Elles lient les concepts primitifs (ou simples) entre eux pour construire des représentations conceptuelles complexes. Elles sont formellement définies comme n'importe quel sous ensemble d'un produit de n ensembles : R dans C1XC2XC3 XC4….X Cn.

Selon [Guarino &Carrara 1995] et [Kassel 2002], les principales relations jugées utiles à la modélisation d"une ontologie sont ; « instance de», « sorte de », « appartenance à », « dépendance » et «subsomption (is a) ». Cette dernière est implicite et a un statut particulier car elle définit un lien de généralisation qui structure la hiérarchie ontologique.

On dit qu'un concept C1 (concept père) subsume un concept C2 (concept fils) si toute propriété sémantique de C1 est également propriété sémantique de C2 et si C2 est plus spécifique que C1 l'extension du fils est plus spécifique que C1. L'extension du fils est donc forcément plus réduite que celle de son père mais son intension est plus riche. Par exemple, le concept pathologie subsume le concept « pneumonie ».

- **Les fonctions :**

Les fonctions sont un cas particulier de relations dans lesquelles le nième élément de la relation est unique pour les (n-1) éléments précédents. Formellement, des fonctions sont définies comme :

F : C1XC2XC3XC4…Cn-1 ⟶ Cn ou les Ci sont des concepts.

Exemple de fonction binaire : la fonction « mère de » ou carrée de.

Exemple de fonction ternaire : le prix d'une voiture usagée sur lequel on peut se baser pour calculer le prix d'une voiture d'occasion en fonction de son modèle, de sa date de construction et de son kilométrage.

- **Les axiomes (ou les règles)**

Les axiomes sont des expressions qui sont toujours vraies. Ils ont pour but de définir dans un langage logique la description des concepts et des relations permettant de représenter leur sémantique. Ils représentent les intentions des concepts et des relations du domaine et, de manière générale, les connaissances n'ayant pas un caractère strictement terminologique [Staab& Maedche 2000]. Leur inclusion dans une ontologie peut avoir plusieurs objectifs :

- Définir la signification des composants.
- Définir des restrictions sur la valeur des attributs.
- Définir les arguments d'une relation.
- Vérifier la validité des informations spécifiées ou en déduire de nouvelles.

- **Les instances**

Les instances constituent des valeurs concrètes et des occurrences pour les concepts et les relations. Par exemple lapin est une instance du concept « animal ».

- **Les roles :**

Une entité peut être caractérisée par un rôle [Sowa 2000] et ceci en définissant quelques rôles qu'elle peut jouer dans sa relation avec une autre entité. Exemple : le type « humain » est un type qui dépend de la forme interne de l'entité, mais la même entité peut être caractérisée par des rôles du type : mère, employé.

II. 3.Caractéristiques d'une ontologie : [23]

- **Les ontologies sont structurées:**

Ceci signifie qu'elles sont exprimées dans une langue qui a une syntaxe et basée sur les mathématiques pour leur signification. Comme les concepts sont exprimés formellement, ils peuvent être traités par des programmes informatiques. Les concepts qui existent dans des techniques de modélisation traditionnelles (schéma relationnel, UML, …) sont seulement semi formels. Ces dernières ne peuvent donc pas être manipulées automatiquement par de logiciels sans un effort considérable de programmation de manière à faire ressortir leurs significations.

- **Les ontologies sont lisibles par les humains :**

Ceci signifie qu'elles peuvent être développées, partagées, et compris non seulement par des programmes informatiques, mais aussi par les communautés d'experts de domaine ainsi que des utilisateurs potentiel.

- **Les ontologies sont vastes :**

Les ontologies sont conçues dans le but d'inclure la signification appropriée des concepts liés à un domaine, pas simplement celles requises pour une application particulière. Cela veut dire que si toute la signification des concepts est capturée par une ontologie, elle peut être comprise, modifiée, et contrôlée par n'importe quel expert de domaine.

- **Les ontologies sont partageables:**

Elles sont construites sur la base de bibliothèques communes de concepts fondamentaux et elles sont utilisables à travers de multiples domaines d'application. Ceci facilite la combinaison des ontologies développées séparément pour permettre la communication entre les systèmes d'information qui doivent partager des informations basées sur des concepts communs.

II. 4.Langage de développement d'une ontologie [21]:

Une des principales décisions à prendre dans le procédé de développement d'ontologies consiste à choisir le langage (ou l'ensemble des langages) dans lequel l'ontologie sera exprimée et utilisée. Parmi les langages développés pour les ontologies et les plus fréquemment utilisés, certains sont basés sur la syntaxe de XML, tel que XOL (Ontology Exchange Language), OML (Ontology Markup Language), RDF (Ressource Description Framwork) et RDF schéma.

Trois autres langages sont établis sur RDF (S) pour améliorer ses caractéristiques : OIL (Ontology Inference Layer) DAML+ OIL et OWL (Web Ontology Language) qui est une révision de DAML + OIL qui utilise la conception et l'application de DAML + OIL et qui tend à s'imposer.

III.Présentation de WordNet :

WordNet (Miller, 1995) est une base de données lexicale développée depuis 1985 par des linguistes du laboratoire des sciences cognitives de l'université de Princeton. C'est un réseau sémantique de la langue anglaise, qui se fonde sur une théorie psychologique du langage. La première version diffusée remonte à juin 1991. Son but est de répertorier, classifier et mettre en relation de diverses manières le contenu sémantique et lexical de la langue anglaise. Le système se présente sous la forme d'une base de données électronique qu'on peut télécharger sur un système local. Des interfaces de programmation sont disponibles pour de nombreux langages. S'il n'est pas exempt de critiques (granularité très fine, absence de relations paradigmatiques…), WordNet n'en reste pas moins l'une des ressources de TAL les plus populaires. **[24][20]**

WordNet ressemble superficiellement à un thésaurus, en ce qu'il regroupe les mots en fonction de leurs significations. Cependant, il existe des distinctions importantes. Tout d'abord, WordNet interconnecte non seulement des formes de mots - chaînes de lettres - mais des sens spécifiques des mots. En conséquence, les mots qui se trouvent à proximité les uns des autres dans le réseau sont sémantiquement désambiguïsés. Deuxièmement, WordNet étiquette les relations sémantiques entre les mots, alors que les groupements de mots dans un thésaurus ne suivent aucun motif explicite autre que la similarité. **[24]**

III.1. Conception et Structure de WordNet[27] :

On peut considérer WordNet comme un graphe ou un réseau sémantique, souvent qu'on qualifie d'ontologie légère (Light Ontology), ou chaque nœud représente un concept du monde réel. La conception de WordNet est basée sur les théories de la représentation des connaissances mentales : mémorisation des mots et concepts d'une manière hiérarchique, en utilisant la relation d'inclusion (qui lie, par exemple, des triplets comme «animal », « oiseau», et « Chardonnet »).

Exemple : Un concept peut être un objet tel que « Car » une entité tel que « Teacher » ou un concept abstrait tel que « art ». Chaque nœud est constitué d'un ensemble de mots, où chacun représente le concept associé à ce nœud. Un nœud peut être vu comme un ensemble de mots dont chacun représente le même concept. Exemple : Le concept « car » est représenté par l'ensemble de mots {car, auto, automobile, motocar}.

Dans la terminologie de WordNet cet ensemble est nommé « Synset ».
WordNet offre des descriptions détaillées et précises des mots. Leur structuration sur un axe ontologique a un fondement psychologique. Il résulte de cette approche qu'il arrive parfois que l'on rencontre plus de 20 sens pour un verbe, par exemple le verbe « ***give*** » à 27 sens.

III.2synset

La composante atomique sur laquelle repose le système entier est le synset (*synonym set*), un groupe de mots interchangeables.

Le synset (ensemble de synonymes) est la composante atomique sur laquelle repose WordNet. Un synset correspond à un groupe de mots interchangeables, dénotant un sens ou un usage particulier. Un synset est défini d'une façon différentielle par les relations qu'il entretient avec les sens voisins.**[20]**

WordNet manipule les unités lexicales non pas par des mots mais par un ensemble de synonymes ou « Synset », groupes de mots ou de phrases qui expriment le même concept.
Des différences de sens entre les membres d'un «Synset» se montrent dans différentes restrictions de sélection.

Exemple : les nœuds suivant correspondent aux différents sens de "mouse" dans WordNet

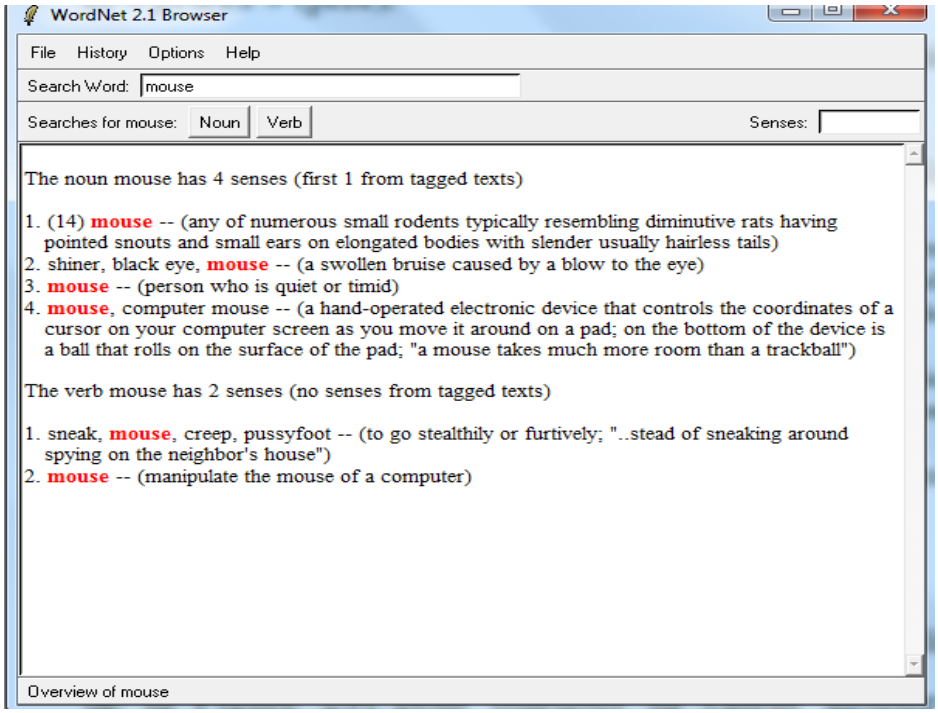

Figure III.1: noeuds correspondent aux différents sens de "mouse" dans WordNet

III.3.Organisation de wordNet [22]

WordNet décompose le lexique en cinq catégories : noms, verbes, adverbes et mots fonctionnels. Chacune de ces catégories a sa propre structure interne. « Ce sont des expériences sur les associations de mots qui ont mis en évidence, à l'origine, que l'organisation varie d'une catégorie syntaxique à l'autre. »

La relation principale entre les mots dans WordNet est la synonymie, entre les mots fermé et fermé ou la voiture et l'automobile. Les synonymes - mots qui désignent le même concept et sont interchangeables dans de nombreux contextes - sont regroupés en ensembles non ordonnés (syntaxe). Chacun des 117 000 syntaxes de WordNet est lié à d'autres syntaxes au moyen d'un petit nombre de "relations conceptuelles". De plus, un synset contient une brève définition ("gloss") et, dans la plupart des cas, une ou plusieurs phrases courtes illustrant l'utilisation Des membres synset. Les formes de mots avec plusieurs significations distinctes sont représentées en autant de syntaxes distinctes. Ainsi, chaque paire form-meaning dans WordNet est unique.

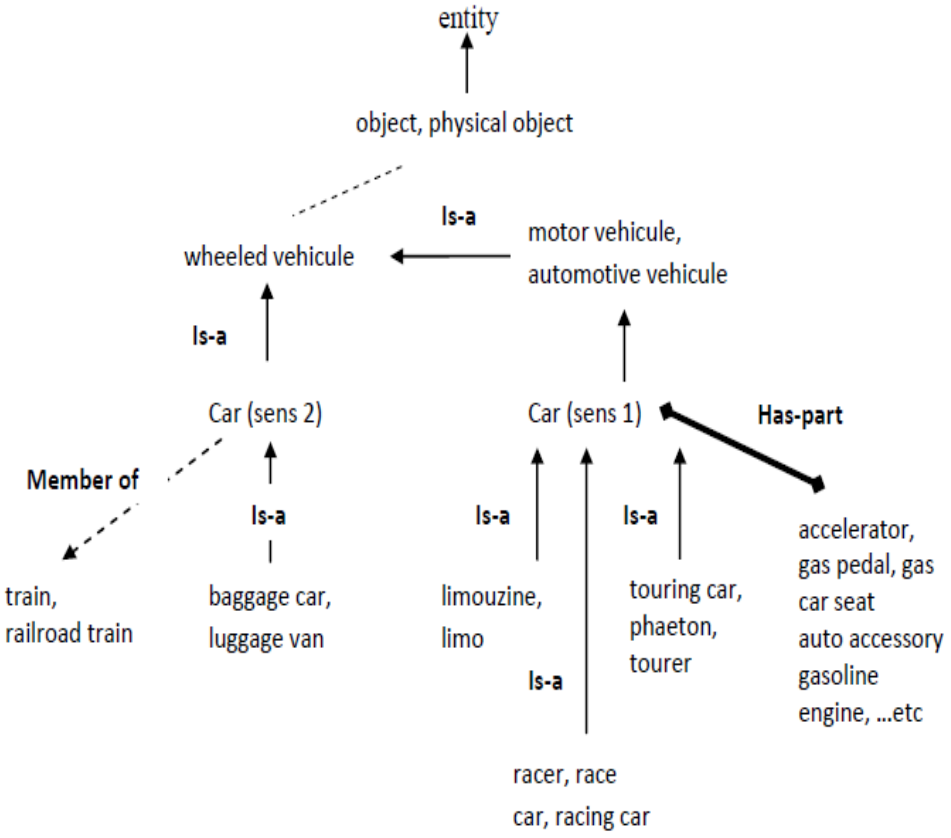

Figure III.2 : Exemple de sous hiérarchie dans WordNet correspondent au concept "car".

Mesures de similarité : Une utilisation possible de l'ontologie fournie par WordNet est la définition de métriques heuristiques de « distance sémantique » entre les synsets. Cette métrique est basée sur la distance à parcourir dans le graphe, combinée ou non avec le Contenu Informationnel. Elle permet de quantifier la similarité de deux concepts. Elle peut également servir dans un cadre de désambiguïsation lexicale.**[20]**

III .4. Les relations dans WordNet

Deux relations fondamentales interviennent dans WordNet, notamment celle entre les « *word form* » appelés *relations lexicales* (par exemple : la synonymie), et celle qui associent les « *word meaning* » appelés relation sémantiques (par exemple : l'hyperonymie).
La figure ci-dessous nous résume les différentes relations qui existent dans wordnet :

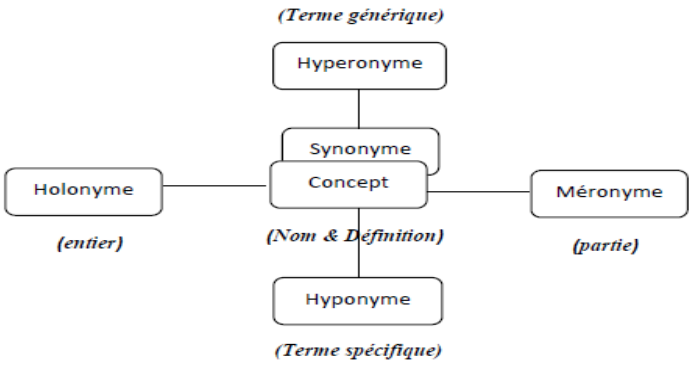

Figure III.3 Principales relations sémantiques dans WordNet

Voilà, une définition des relations les plus importantes dans WordNet :
Synonymie: relation reliant deux concepts équivalents ou proches (fragiles / fragiles).
C'est une relation symétrique.

Antonyme : relation reliant deux concepts opposés (petit / grand). Cette relation est symétrique.

Hyperonymie: relation reliant un concept-1 à un concept plus général-2 (Tulipe / fleur).

Hyponymie: relation reliant un concept-1 à un concept plus spécifique-2. C'est la réciproque de l'hyperonymie.

Meronymie: relation reliant un concept-1 à un concept-2 qui est l'un de ses Parties (fleur / pétale), l'un de ses membres (forêt / arbre) ou une substance constituée de (Vitre / verre).

Holonymie: Relation reliant un concept-1 à un concept-2 dont il est l'un des les parties. C'est le contraire de la relation meronymie.

III .4 .Application de wordnet: WordNet est utilisé en recherche d'informations :

- pour représenter les documents.
- pour étendre la requête de l'utilisateur (ajout de synonymes, par exemple, pour augmenter le rappel, c'est-à-dire la proportion de documents pertinents rapportés) .
- Acquisition de relations sémantiques.
- la désambiguïsation sémantique (l'étiquetage sémantique de corpus, la structuration et catégorisation des documents).

III.5 . Limites de WordNet [20]

- **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.
- **Profusion de sens pour un mot donné** : La contrepartie de son importante couverture est que WordNet est très précis dans le sens des définitions. On a une granularité très fine des sens. Par exemple, le verbe to give (« donner ») n'a pas moins de 44 sens. Une telle profusion ne facilite pas une tâche de désambiguïsation lexicale.
- **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.

III.6.WordNets pour d'autres langues que l'anglais [20]

- **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. (À notre connaissance, les ressources pour le français ont été fournies par la société MemoData sur la base de son Dictionnaire Intégral.) 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.
- **BalkaNet :** BalkaNet prolonge la base de données d'EuroWordNet avec d'autres langues européennes : tchèque, roumain, grec, turc, bulgare, et serbe.

III.7.Statistique de wordNet [28]:

WordNet est un système d'une étonnante ampleur : la version la plus récente (2.1) répertorie plus de 200 000 mots de classes ouvertes, pour lesquelles l'ajout d'éléments lexicaux est possible, ainsi que plus de 115 000 synsets. Pourtant, son statut de projet « en développement » implique toutefois que certaines de ses composantes sont incomplètes. À chaque nouvelle version, le lexique s'enrichit de nouveaux mots, et des relations sémantiques sont ajoutées, modifiées, ou encore rendues désuètes.
WordNet étant un logiciel libre, celui-ci comprend, outre les définitions des mots, l'ensemble des sources utiles pour l'accès aux données du dictionnaire [27]. La version 3.0 de wordnet comporte les statistiques suivante[28] :

Réseau	formes	synsets	Paires mot-sens
Noms	117798	82115	146312
Verbes	11529	13767	25047
Adjectifs	21479	18156	30002
Adverbes	4481	3621	5580
Total	155287	117659	206941

Figure III.4. Statistique sur wordNet

IV.Conclusion :

Dans ce chapitre, nous avons présenté brièvement les notions les plus couramment utilisées dans le domaine d'ingénierie ontologique qui est vu comme un sous-domaine de l'ingénierie des connaissances. Nous avons aussi présenté la base de connaissance lexicale wordNet qui est sans doute le précurseur et la référence en matière de base lexicales sémantiques et informatiques devenue peu à peu à caractère ontologique.

Chapitre IV : Idexation temporel de WordNet

I. Introduction :

Connaitre le temps d'un document (un discours par exemple), soit implicitement ou explicitement, est en effet important pour pouvoir l'exploiter en recherche(RI) d'information ou en traitement automatique des langues(TAL) ou dans un autre domaine. La plupart des travaux qui existent dans la littérature basés sur l'aspect temporel des documents se focalisent sur l'exploitation de la date du document ou de séries temporelles[1].
Dans ce chapitre, nous allons présenter en détail la construction d'une ontologie temporelle, ce qui peut contribuer au succès des applications liées au temps. .Nous allons construire la TempoWordNet, une base de connaissances lexicale où chaque synset de WordNet est augmenté avec sa valeur temporelle intrinsèque.

II.Motivation et objectifs :

Le temps joue un rôle crucial dans toute recherche d'information, Il est exploité dans plusieurs de ses tâches telles que l'extraction de l'information, la détection et le suivi de sujet, detection de temps et le clustring. Depuis sa création, toutes les informations temporelles incorporées dans des documents n'ont pas été pleinementutilisé par les applications RI pour une meilleure satisfaction des utilisateurs et des fonctionnalités de recherche supplémentaires.

Dans les textes en langage naturel, différents types d'informations temporelles peuvent être associées. La plupart de ses types (information temporelle) associésà un document sont métadonnées (son temps de création ou de le temps la modification).

Ce type d'informations peut facilement être identifié, accessible et utilisé pour plusieurs tâches,par exemple la recherche de temps conscient, la classification, le classement temporel, etc. Il est important dementionner que les méta-informations ne sont utiles que dans certains contextes spécifiques.
Supposons qu'un rapport de document de presse à propose d'un futur événement compte tenu uniquement de la méta-base de données c'est-à-dire du document. Là le temps de création entrainera des informations parasites sur l'heure de l'évènement. Mais pour faire utiliser ces informations temporelles latentes, habituellement. Des taggers sont appliqués pour extraire et normaliser les expressions temporelles contenues dans les documents.

les expressions temporelles peuvent être regroupées en deux principales catégories : expressions temporelles absolues divisées en expression temporelles explicites(par exemple 2 mars 2015) ou implicites (exemple le la fête des travailleurs) et des expression temporelle relatives (par exemple : l'année dernière).Cependant , des milliers de mot existant tels que « passé », « présent », « avenir »etc. qui possède clairement une dimension temporelle par exemple une phrase comme « votre iPone 4ou5 pourrait ressentir comme nouveau avec le prochain Ios9 » est clairement transportant une indexation future. Le mot «prochain » a une annotation claire du futur bien ignorée par la plupart des indicateurs de temps (taggers) temporels existants.

1 une série temporelle est une suite d'observation d'une variable ya des dates différentes.

La plupart des tâches TAL comme on nous avons déjà vu dans le premier chapitre,se fondent sur un vocabulaire sensible au facteur temps. Au contraire, les systèmes T-IR qui n utilisent pas généralement des informations reliées au temps dans un langage bien qu'ils puissent bénéficier de quand faire face au problème récurrent de manque des indicateurs explicites. Pour remédier à ce problème nous avons l'intention de construire une ontologie temporelle

III. WordNet temporel :

III.1.La description de TempoWordnet :

Les aspects temporels sont fondamentaux à l'interprétation du langage naturel. On peut dire qu'il n'y a pas de phrase ou expression dont l'interprétation ne comporte pas les aspects temporels. Par conséquent, notre approche vise une annotation temporelle du langage naturel basée sur l'ontologie qui doit tenir compte de ces aspects temporels. A cet effet, nous avons besoin d'une ontologie ou un vocabulaire logique qui nous permet de représenter l'indexation temporelle du sens d'une phrase, énoncé ou discours. Au lieu de construire une nouvelle ontologie, nous construisons une ontologie temporelle nommée WordNet temporelle basée sur WordNet[Fellbaum, 1998a, Miller, 1995]. Qui était présentée dans le chapitre précédent.

Notre travail (WordNet temporelle) consiste d'enrichir tout synsets de WordNet avec des dimensions temporelles .Autrement tous les synsets de WordNet seront automatiquement étiquetés avec des dimensions temporelle (passé, présent,future).

La figure suivante montre le processus général de notre approche qui reçoit en entrée les synset de WordNet et qui produit en sortie une ontologie WordNet temporelle.

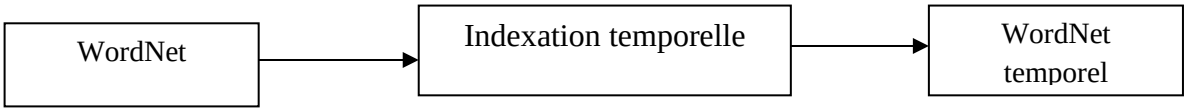

Figure IV.1 : schéma descriptif de notre approche

La méthode suivie pour indexer l'ontologie WordNet passe par deux étapes essentielles qui sont : la construction du corpus et l'étiquetage temporel.

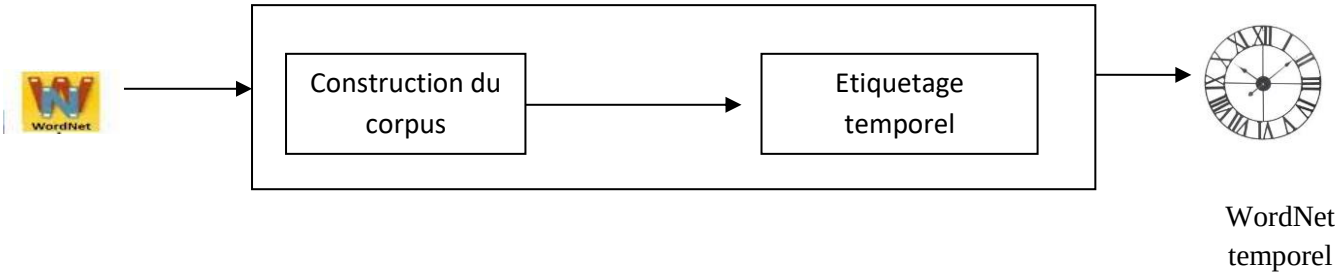

Figure IV.2: Processus d'indexation temporelle

III.2 .La construction de classifieur :

Pour construire un classifieur, le système doit être formé en utilisant l'ensemble de formation, il doit être testé à l'aide d'un jeu de tests. Donc, l'ensemble de données doit être partitionné dans un ensemble de formation et des ensembles de tests. La validation croisée en K-fois est utilisée dans cette recherche, où K est fixé à 10 en fonction de la précédente établie dans des recherches antérieures (Dai et al., 2007;Genkin et al., 2007; Mullen et Collier, 2004).

L'avantage de la validation croisée K-fold est que tout le jeu de données des échantillons sont utilisés et proposés pour la formation et test. Cela garantit que le système produit des solutions fiables.

Trois mesures quantitatives sont utilisées: précision, Rappel et F-mesure (Forman, 2003; Lodhi et al.,2002). Étant donné que la sortie du classificateur naive bayes est une matrice de confusion qui montre le nombre de documents attribués à chaque classe. Certains documents sont attribués correctement pendant que d'autres sont mal classés

vrai classe → ↓ classé	positif	négatif
positif	VP	FP
négatif	FN	VN
total	P	N

Figure IV.3 :matrice de confision

III.3.La validation croisée (cross validation) :

La validation croisée est une méthode statistique d'évaluation et de comparaison des algorithmes d'apprentissage en divisant les données en deux segments: l'un utilisé pour l'apprentissage ou la formation d'un modèle et l'autre utilisé pour valider le modèle. En cas de validation croisée typique, les ensembles de formation et de validation doivent se transmettre en rondes successives, de sorte que chaque point de données ait une chance d'être validé. La forme de base de la validation croisée est la validation croisée du k-fold. D'autres formes de validation croisée sont des cas spéciaux de k-fold cross-validation ou impliquent des cycles répétés de k-fold cross-validation.

III. 4.La construction de corpus :

Notre corpus est constitué de tous les synsets de WordNet dont nous avons besoin d'établir une liste de concepts étiquetés qui vont servir à prédire la classe du reste des concepts de WordNet.

Nous avons sélectionné les synsets utilisés comme de bons paradigmes pour past,present,future. Par exemple, des mots comme "yesterday", "previously", "remember" sont de bons mots paradigmatiques pour la classe past, "current", "existing", "presently" pour le present et "prophecy", "predict", "tomorrow" pour future.

Le choix d'ensemble initial de synsets est une étape cruciale dans la procédure, avec leurs propriétés qui doivent être préservées au long du processus d'expansion. Par conséquent, le choix de la liste imparfaite aura d'énormes conséquences.

Pour garder les synsets les plus pertinents pour chaque classe de temps, une première sélection a été faite par plusieurs personnes grâce à des discussions libres de groupe intensif e et surtout avec des gens qui maitrisent la langues anglaise. Chaque participant a été encouragé à réfléchir à haute voix et de proposer autant de mots que possible.

Après avoir obtenu autant de mots, nous avons consulté la base de donnée WordNet pour assurerla presence de chaque catégorie grammaticale existant dans WordNet(ie Noun, Adjectif, adverbe et Verbe) serait présent dans les ensembles des seeds (graines) pour les classes past,present,future

Chaque synset dans WordNet contient un ou plusieurs mots, Nous avons sélectionné les synsets exprimant les annotations temporelles énumérées par les individus.

Enfin, nous avons procédé à un processus d'accord sur la liste des seeds (past,present,future).

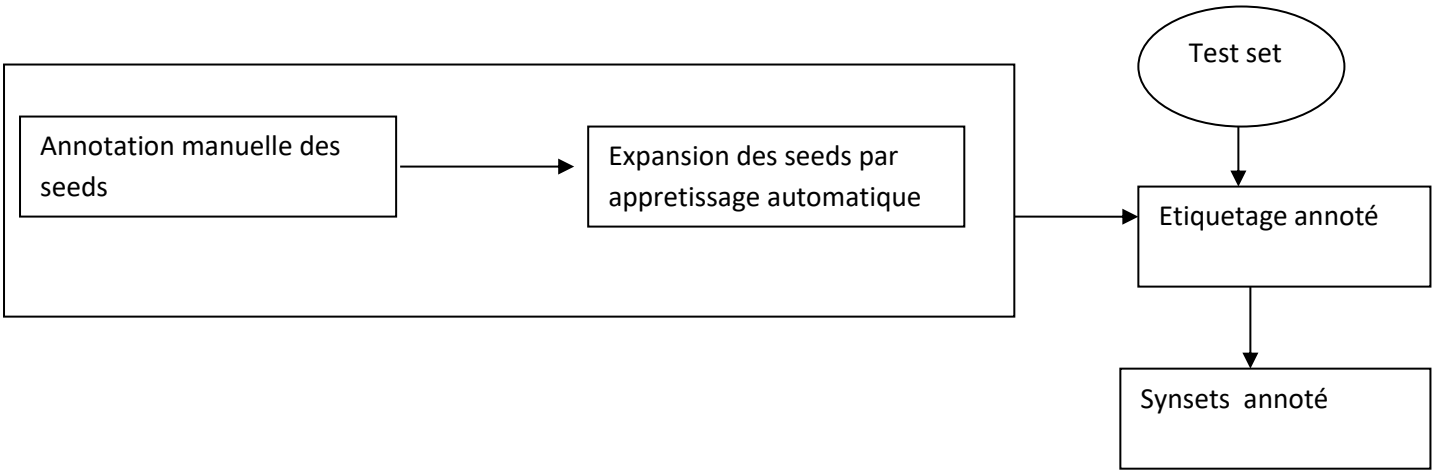

Figure IV.4. Construction du corpus

```
seeds = ['past.n.01',
         'past.a.01',
         'yesterday.n.01',
         'yesterday.n.02',
         'yesterday.r.01',
         'yesterday.r.02',
         'commemorate.v.02'
        ,'previously.r.01',
         'present.n.01',
         'present.a.01',
         'present.a.02',
         'now.n.1',
         'now.r.03',
         'nowadays.r.01',
         'today.n.01',
         'ongoing.a.01',
         'existing.a.01',
         'current.a.01',
         'future.n.01',
         'future.a.01',
         'future.a.02',
         'tomorrow.n.01',
         'tomorrow.n.02',
         'tomorrow.r.01',
         'predict.v.01',
         'expected.a.01',
         'prophesy.v.01',
         'aforethought.a.01']
```

IV.5.Liste des seeds extraite de notre programme

Après avoir sélectionné les seeds . nous avons récupéré leurs glossaires extrait de wordNet puis nous les avons stocker dans un fichier : « seeds-glossaire.txt » comme le montre la figure suivante :

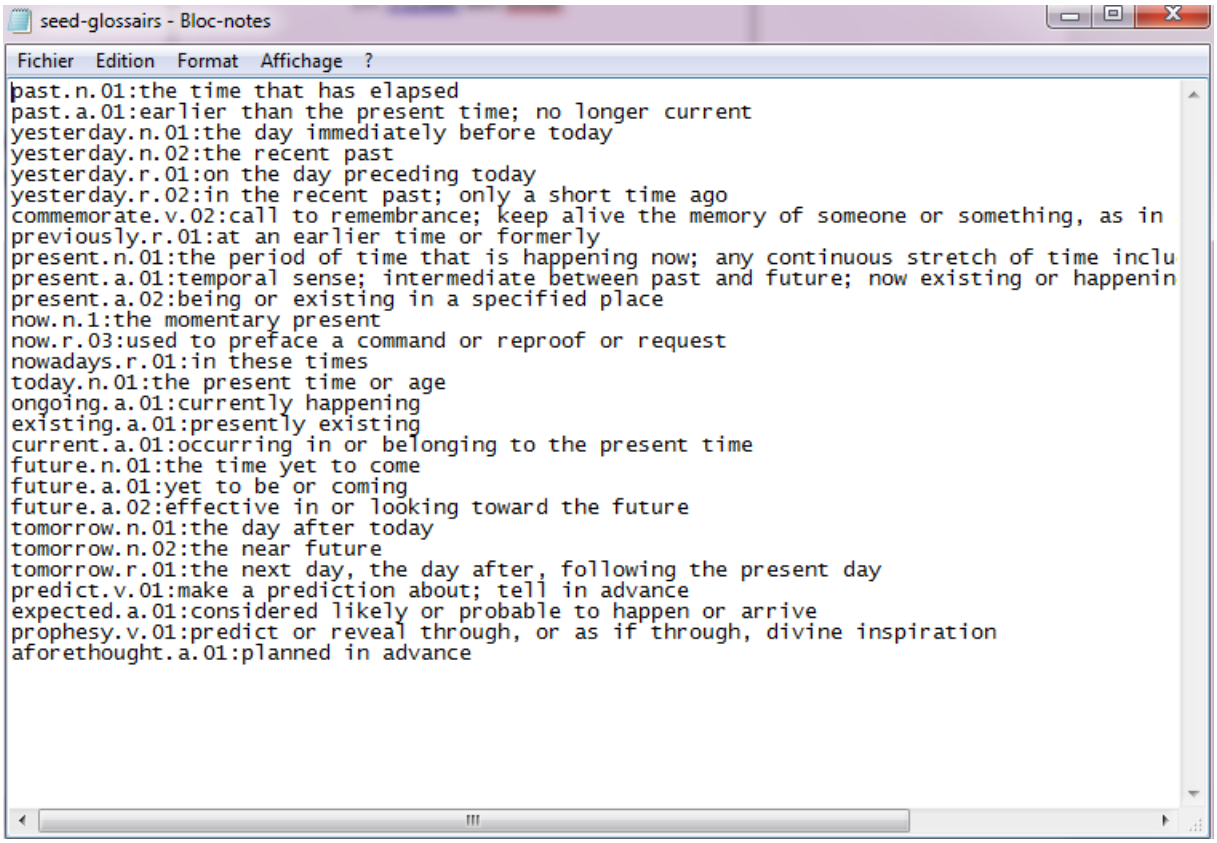

Figure IV.6:le fichier « seed-glossairs.txt »

En suite, nous avons filtré l'ensemble des glossaires en enlevant les stopsword (mots d'arret : je,moi….) et l'articulation pour avoir en résultat la liste suivante qui contient 74 éléments :

['formerly', 'predict', 'specified', 'period', 'alive', 'existing', 'past', 'likely', 'used', 'including', 'something', 'sense', 'remembrance', 'happen', 'happening', 'yet', 'momentary', 'ceremony', 'looking', 'divine','probable', 'inspiration', 'stretch', 'continuous', 'currently', 'next', 'current', 'age', 'intermediate','call', 'memory', 'preface', 'speech', 'consideration', 'occurring', 'tell', 'today', 'belonging', 'someone', 'presently', 'reveal', 'preceding', 'earlier', 'elapsed', 'prediction', 'reproof', 'moment', 'coming','immediately', 'come', 'day', 'present', 'recent', 'ago', 'advance', 'short', 'longer', 'effective','considered', 'command', 'times', 'request', 'make', 'keep', 'near', 'future', 'place', 'planned', 'arrive', 'time', 'following', 'toward', 'temporal']

Nous avons calculé les occurrences de chaque attributs dans le glossaire de chaque seeds dont l'objectif est de construire notre model d'apprentissage ,la figure IV.7 montre une partie des résultats de cette opération

Out[6]:

	formerly	predict	specified	period	alive	existing	past	likely	used	including	...	keep	near	future	place	planned	arriv
past.n.01	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
past.a.01	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
yesterday.n.01	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
yesterday.n.02	0	0	0	0	0	0	1	0	0	0	...	0	0	0	0	0	0
yesterday.r.01	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
yesterday.r.02	0	0	0	0	0	0	1	0	0	0	...	0	0	0	0	0	0
commemorate.v.02	0	0	0	0	1	0	0	0	0	0	...	1	0	0	0	0	0
previously.r.01	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
present.n.01	0	0	0	1	0	0	0	0	0	1	...	0	0	0	0	0	0
present.a.01	0	0	0	0	0	1	1	0	0	0	...	0	0	1	0	0	0

Figure IV.7 l'occurrence des attributs dans les seeds

Nous avons fait une sélection selon la classe des seeds ,les 3 figures suivants montrent les tableau résultats :

	formerly	predict	specified	period	alive	existing	past	likely	used	including	...	keep	near	future	place	planned	arrive
past.n.01	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
past.a.01	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
yesterday.n.01	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
yesterday.n.02	0	0	0	0	0	0	1	0	0	0	...	0	0	0	0	0	0
yesterday.r.01	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
yesterday.r.02	0	0	0	0	0	0	1	0	0	0	...	0	0	0	0	0	0
commemorate.v.02	0	0	0	0	1	0	0	0	0	0	...	1	0	0	0	0	0
previously.r.01	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0

Figure IV.8:table des synsets de la classe past

	formerly	predict	specified	period	alive	existing	past	likely	used	including	...	keep	near	future	place	planned	arrive	time
present.n.01	0	0	0	1	0	0	0	0	0	1	...	0	0	0	0	0	0	2
present.a.01	0	0	0	0	0	1	1	0	0	0	...	0	0	1	0	0	0	0
present.a.02	0	0	1	0	0	1	0	0	0	0	...	0	0	0	1	0	0	0
now.n.1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
now.r.03	0	0	0	0	0	0	0	0	1	0	...	0	0	0	0	0	0	0
nowadays.r.01	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	1
today.n.01	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	1
ongoing.a.01	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
existing.a.01	0	0	0	0	0	1	0	0	0	0	...	0	0	0	0	0	0	0
current.a.01	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	1

Figure IV.9 :table des synset de la classe present

	formerly	predict	specified	period	alive	existing	past	likely	used	including	...	keep	near	future	place	planned	arrive	ti
future.n.01	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	1
future.a.01	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
future.a.02	0	0	0	0	0	0	0	0	0	0	...	0	0	1	0	0	0	0
tomorrow.n.01	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
tomorrow.n.02	0	0	0	0	0	0	0	0	0	0	...	0	1	1	0	0	0	0
tomorrow.r.01	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
predict.v.01	0	1	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
expected.a.01	0	0	0	0	0	0	0	1	0	0	...	0	0	0	0	0	1	0
prophesy.v.01	0	1	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
aforethought.a.01	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	1	0	0

Figure IV.10 :Table des synset de la classe Future

Pour réaliser notre approche on a utilisé un classifieur bayésiens :

Le principe d'un classificateur naïf bayésien consiste à maximiser la probabilité Pr(y|d), soit la probabilité d'occurrences de la classe de prédiction y connaissant la représentation de la nouvelle donnée x (on suppose donc ici d = d(x) = (d1, d2, ..., dn)), et ce pour toutes les classes y ∈ Y et toutes les composantes qui interviennent dans la définition de l'espace de représentation D. Pour cela, on fait appel à la règle de Bayes.

***Règle de Bayes*.** Soient A et B deux évènements. La règle de Bayes dit alors que la probabilité de l'évènement A sachant l'évènement B (Pr(A|B)) peut se calculer à l'aide des probabilités des évènements A et B (Pr(A) et Pr(B)) et connaissant la probabilité de
l''évènement B sachant l'évènement A (Pr(B|A)) par la formule suivante :
Pr(A|B) = Pr(B|A) Pr(A) / Pr(B)

Application à la classification. En appliquant la règle de Bayes à la problématique de la Classification, on obtient l'équation suivante :
Pr(y|d) = Pr(d|y) Pr(y) / Pr(d)
Les probabilités de l'expression de droite doivent être estimées, à l'aide du corpus d'apprentissage S(l'ensemble des seeds), afin de calculer la quantité qui nous intéresse, soit P(y|d) :
– Pr(y) est la probabilité d'observer la classe y.
– Pr(d) est la probabilité d'observer la représentation d.
– Pr(d|y), la vraisemblance de l'évènement « observer la représentation d » si s ∈ S est de classe y. Ce terme est plus difficile à estimer que le précédent.

Le classifieur Naïve Bayes est utilisé avecles caractéristiques concurrentes pour choisir la meilleure conception afin d'améliorer la précision de la classification.

Dons notre cas : l'application de naive bayes consiste à calculer les probabilités de chaque classe et la probabilités de chaque mot sachant les trois classes (past,présent.future)

Nous avons obtenu les résultats suivants :
P (past)= `0.285714285714 et la p(present)= 0.357142857143`

```
p(future)= 0.357142857143
```

Pour calculer la probabilités des mots dans chaque classes,on utilise la formule suivane :
P(attribut/classe)=(nk+1)/n+|vocabulaire|

sachant :**nk**: nombre d'occurrence de la caractéristique (features qui seront utilisés dans le rest e de notre etude),n : nombre des mots dans la classe.
|vocabulaire| : la taille du corpus de features.

```
[[ 0.01351351  0.01333333  0.01315789  0.01298701  0.01282051  0.01265823
   0.0125      0.01234568  0.01219512  0.01204819  0.01190476  0.01176471
   0.01162791  0.01149425  0.01136364  0.01123596  0.01111111  0.01098901
   0.01086957  0.01075269  0.0106383   0.01052632  0.01041667  0.01030928
   0.01020408  0.01010101  0.01        0.00990099  0.00980392  0.00970874
   0.00961538  0.00952381  0.00943396  0.00934579  0.00925926  0.00917431
   0.00909091  0.00900901  0.00892857  0.00884956  0.00877193  0.00869565
   0.00862069  0.01709402  0.00847458  0.00840336  0.00833333  0.00826446
   0.00819672  0.00813008  0.00806452  0.008       0.00793651  0.00787402
   0.0078125   0.00775194  0.00769231  0.00763359  0.00757576  0.0075188
   0.00746269  0.00740741  0.00735294  0.00729927  0.00724638  0.00719424
   0.00714286  0.0070922   0.00704225  0.01398601  0.00694444  0.00689655
   0.00684932]
 [ 0.01351351  0.01333333  0.01315789  0.01298701  0.01282051  0.01265823
   0.0125      0.01234568  0.01219512  0.01204819  0.01190476  0.01176471
   0.01162791  0.01149425  0.01136364  0.01123596  0.01111111  0.01098901
   0.01086957  0.01075269  0.0106383   0.01052632  0.01041667  0.01030928
   0.01020408  0.01010101  0.02        0.00990099  0.00980392  0.00970874
   0.00961538  0.00952381  0.00943396  0.00934579  0.00925926  0.00917431
```

Figure IV.11 : probabilité conditionnel des features

III.5. Expansion des seeds :

Après avoir fini avec l'ensemble initial des seeds, nous avons appliqués des relations syntaxiques (synonyme ,hypernymie) afin d'expansé l'ensemble précedent :

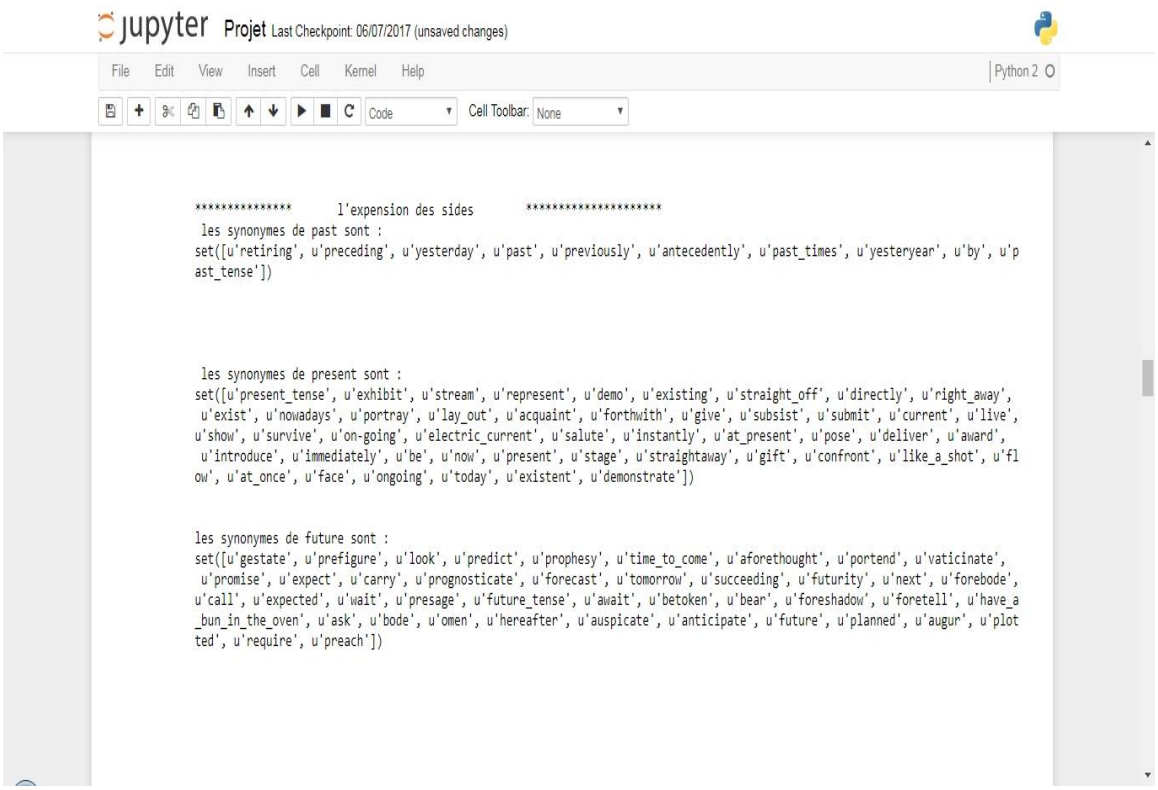

Figure IV.12 . résultats d'expansion en appliquant la synonymie

Dans la figure suivante ,Nous allons vous présenter notre modèle des variables :

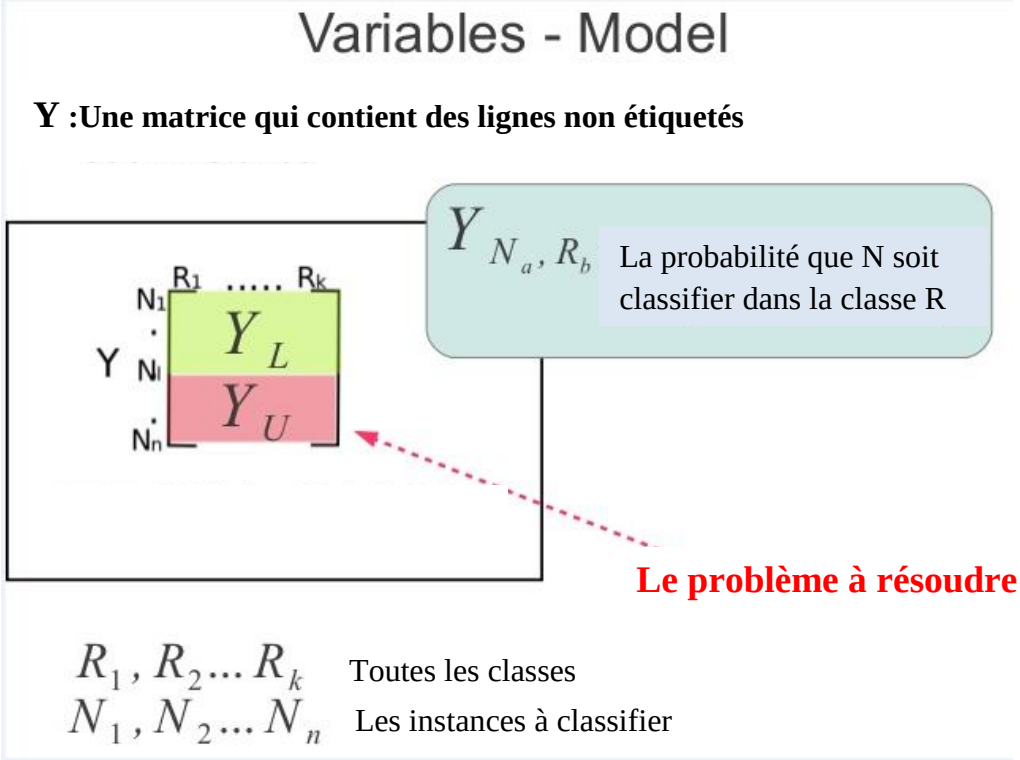

Figure IV.13 : Une matrice Y avec des lignes non classés

Afin de résoudre le problème et avoir Y totalement étiquetée. Nous avons appliqué l'apprentissage semi-supervisé (apprendre avec un peu de données labellisées et beaucoup de données non labellisées) en implémentant « label propagation » .les étapes de l'algorithme sont montrées dans la figure suivante :

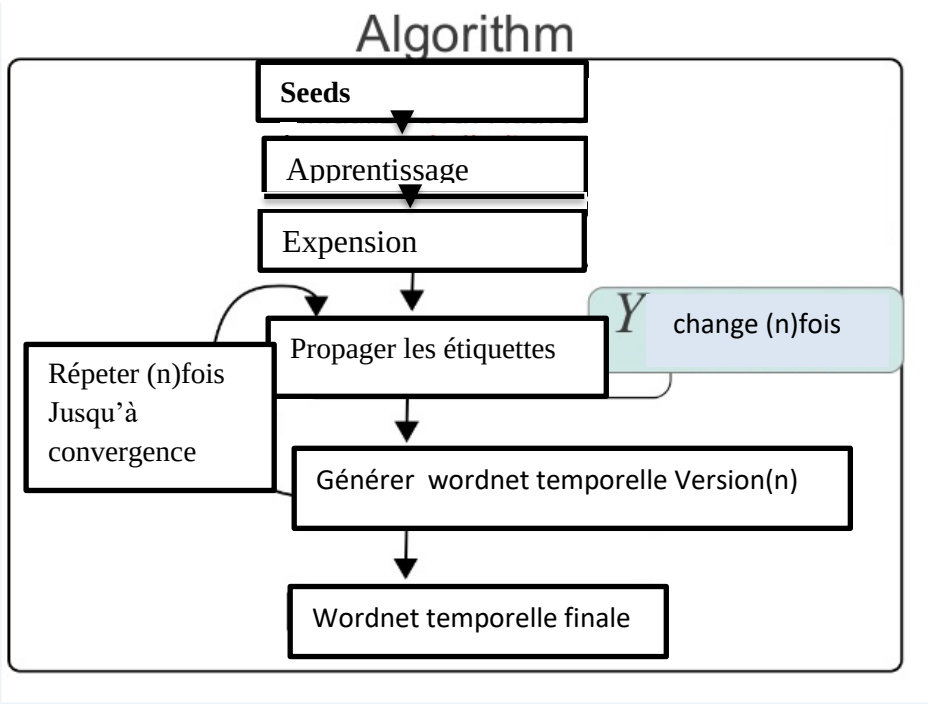

Figure IV.14 : Notre approche en inclurant l'algorithme « label propagation »

IV. Implementation de notre approche : Nous allons décrire maintenant l'environnement de programmation de notre approche.

IV.1.Environnement et outils d'implémentation :

Notre travail a été réalisé sous windows7. Anaconda Notebook version 2.3 comme environnement de développement, et le choix de python comme langage de programmation.

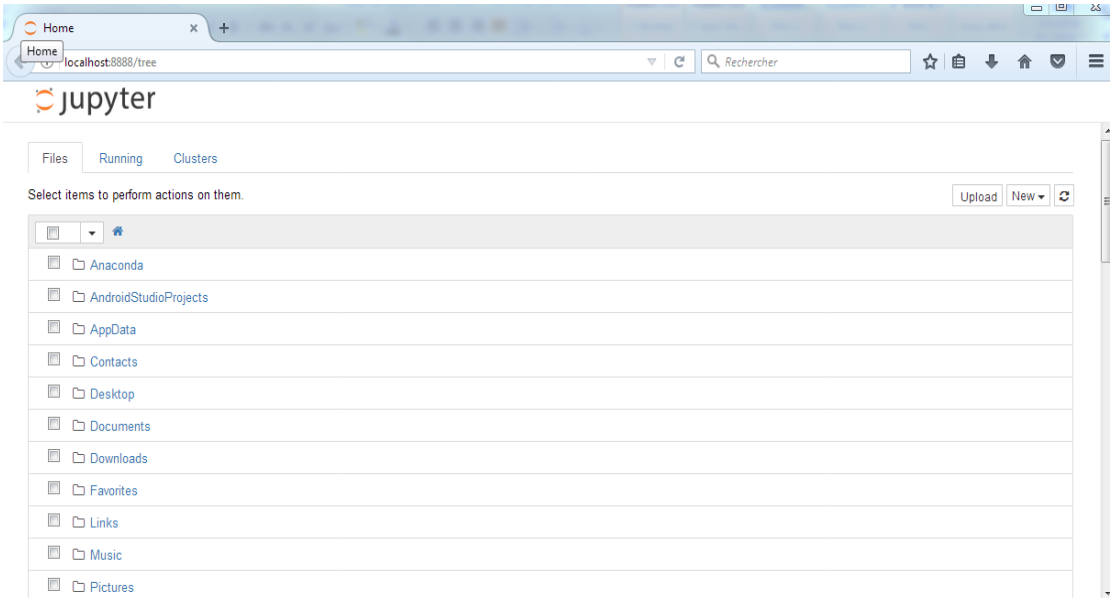

Figure IV.15.Présentation de l'interface de l'environnement de travail

IV.2.présentation du langage de programmation python : [25]

Python est un langage portable, dynamique, extensible, gratuit, qui permet (sans l'imposer) une approche modulaire et orientée objet de la programmation. Python est développé depuis 1989 par Guido van Rossum et de nombreux contributeurs bénévoles.

- ***Caractéristiques du langage***

Python estportable, non seulement sur les différentes variantes d'*Unix*, mais aussi sur les OSpropriétaires: *MacOS*, *BeOS*, *NeXTStep*, *MS-DOS* et les différentes variantes de *Windows*. Unnouveau compilateur, baptisé JPython, est écrit en Java et génère du *bytecode* Java.

- Python est gratuit, mais on peut l'utiliser sans restriction dans des projets commerciaux.
- Python convient aussi bien à des scripts d'une dizaine de lignes qu'à des projets complexes deplusieurs dizaines de milliers de lignes.

La syntaxe de Python est très simple et, combinée à des types de données évolués (listes,dictionnaires,...), conduit à des programmes à la fois très compacts et très lisibles. Afonctionnalités égales, un programme Python (abondamment commenté et présenté selon lescanons standards) est souvent de 3 à 5 fois plus court qu'un programme C ou C++ (ou même Java) équivalent, ce qui représente en général un temps de développement de 5 à 10 fois pluscourt et une facilité de maintenance largement accrue.

Python gère ses ressources (mémoire, descripteurs de fichiers...) sans intervention duprogrammeur, par un mécanisme de comptage de références (proche, mais différent, d'un*garbage collector*).

- Il n'y a pas de pointeursexplicites en Python.
- Python est (optionnellement) multi-threadé.
- Python est orienté-objet. Il supporte l'héritage multipleet la surcharge des opérateurs. Dansson modèle objets, et en reprenant la terminologie de C++, toutes les méthodes sont virtuelles.Python intègre, comme Java ou les versions récentes de C++, un système d'exceptions, quipermettent de simplifier considérablement la gestion des erreurs.

Python est dynamique(l'interpréteur peut évaluer des chaînes de caractères représentant desexpressions ou des instructions Python), orthogonal (un petit nombre de concepts suffit àengendrer des constructions très riches), réflectif(il supporte la *méta programmation*, parexemple la capacité pour un objet de se rajouter ou de s'enlever des attributs ou des méthodes, oumême de changer de classe en cours d'exécution) et introspectif(un grand nombre d'outils dedéveloppement, comme le *debugger* ou le *profiler*, sont implantés en Python lui-même).
Comme *Scheme* ou *SmallTalk*, Python est dynamiquement typé. Tout objet manipulable par leprogrammeur possède un type bien défini à l'exécution, qui n'a pas besoin d'être déclaré àl'avance.

- Python possède actuellement deux implémentations. L'une, interprétée, dans laquelle lesprogrammes Python sont compilés en instructions portables, puis exécutés par une machine

virtuelle (comme pour Java, avec une différence importante: Java étant statiquement typé, il estbeaucoup plus facile d'accélérer l'exécution d'un programme Java que d'un programme Python).
L'autre génère directement du *bytecode* Java.

- Python est extensible: comme *Tcl* ou *Guile*, on peut facilement l'interfacer avec desbibliothèques C existantes. On peut aussi s'en servir comme d'un langage d'extension pour dessystèmes logiciels complexes.
- La bibliothèque standardde Python, et les paquetages contribués, donnent accès à une grandevariété de services : chaînes de caractères et expressions régulières, services UNIX standard(fichiers, *pipes*, signaux, sockets, threads...), protocoles Internet (Web, News, FTP, CGI,HTML...), persistance et bases de données, interfaces graphiques.

- Python est un langage qui continue à évoluer, soutenu par une communauté d'utilisateursenthousiastes et responsables, dont la plupart sont des supporters du logiciel libre. Parallèlementà l'interpréteur principal, écrit en C et maintenu par le créateur du langage, un deuxièmeinterpréteur, écrit en Java, est en cours de développement. Enfin, Python est un langage de choix pour traiter le XML.

    - ✓ **Natural Language Toolkit(NLTK)** est une bibliothèque logicielle en Python permettant un traitement automatique des langues. En plus de la bibliothèque, NLTK fournit des démonstrations graphiques, des données-échantillon, des tutoriels, ainsi que la documentation de l'interface de programmation (API).

V.Conclusion :

Dans ce chapitre nous avons vu la phase de conception de notre approche, qui donne un aperçu sur l'environnement d'expérimentation et une explication de la démarche suivie qui nous a permis de réaliser notre approche, ainsi l'environnement de développement de notre travail en spécifiant les outils et les langages utilisés.

Conclusion générale

CONCLUSION GENARAL

Notre projet se situe dans le domaine de traitement automatique des langues, il porte sur l'implémentation et l'évaluation d'une indexation temporelle d'une ontologie lexicale.

Pour mener a terme notre travail, on a donné un aperçu général sur le traitement automatique des langues, ensuite on a présenté l'indexation et quelques méthodes de classification ce qui nous a permis d'enrichir nos connaissance pour le bon déroulement de notre travail. De même nous avons défini la notion d'ontologie, la base de connaissance lexicale wordNet ainsi que ces composants. Enfin nous avons défini et suivi, l'indexation temporelle de wordNet, le langage de programmation 'python' et l'environnement 'Notebook' afin de bien concevoir notre système.

Ce travail a permis d'aborder le domaine de traitement automatique des langues et plus précisément :

- Découvrir le domaine de traitement automatique des langues.
- Approfondir nos connaissances sur le domaine de traitement automatique des langues.
- Découvrir la notion d'ontologie ainsi que la base de connaissance wordNet. Découvrir le langage de programmation python.

D'après nos résultats, nous pensons que l'utilisation de ces modèles peut encore évoluer vers plus performance.

Bibliographie

Référence bibliographique

[1] :http://www.technolangue.net/imprimer.php3?id_article=274(date d'accès 01/03/2017)

[2] :BOUILLON,Pierette. « Traitement automatique des langues naturelles ».ChampsLinguistiques.Champs Linguistiques : Recueils, ISSN 1377-7971 *Universités francophones*, ISSN 0993-3948.Edition Duculot AUPELF UREF. Bruxelles. De Boeck Supérieur, 1998, 2801111813,9782801111819. 245pages. (date d'accès 01/03/2017)

[3] :http://www.Traitement Automatique du Langage Naturel.html(date d'accès 01/03/2017)

[4] : JACQUEMIN.Christian,ZWEIGenbaum.Pierre. « Traitement automatique des langues pour l'accès au contenu des documents » .[Document électronique]. Paris.Anass.2007. (date d'accès 05/03/2017)

[5] :YVON ,François. « une petite introduction au Traitement Automatique des langues Naturelles » .[document éectronique].france,2007 . (date d'accès 06/03/2017)

[6] : http://ldelafosse.pagesperso-orange.fr/Glossaire/Tal.html(date d'accès 06/03/2017)

[07] : Laure Amélie Guitard. Indexation par sujet en archivistique et en bibliothéconomie,2017 .

[08] : Dictionnaire encyclopédique de l'information et de la documentation / dir. Serge Cacaly. 3e édition, Paris : A. Colin, 2008. (date d'accès 20/03/2017)

[09] :HAL.Etude de faisabilité de mise en place d'une indexation semi-automatique avec un thésaurus spécialisé en archéologie .

[10] : Patrice Bertrand « Méthodes de classification : k-means », Université Paris-Dauphine, 2010 .

[11] : Alleb Kamel, Doufene Abdelkrim, « Classification automatique de document XML », ingénieur d'état en informatique, institut national de formation en informatique (INI), 2008.

[12] : Bernard Fertil, « Reconnaissance des formes : classement d'ensembles d'objets », directeur de recherche CNRS, 2008.

[13] : Beloucif Safia & Fodil Hamida,«Implémentation d'une méthode de classification par les algorithmes génétiques», mémoire d'ingénieur d'état en informatique, université Mouloud Mammeri de Tizi-Ouzou, 2009.

[14]:http://docs.happycoders.org/orgadoc/artificalintelligence/classificationdocument/classification.pdf.(date d'accès 20/03/2017) .
[15] : E.G Talbi « Fouille de données (Data Mining), un tour d'horizon », laboratoire informatique fondamentale de Lille .

[16] : Bertrand Liaudet, « Cours de Data Mining », 2008

[17] : Bertrand Liaudet, «Cours de data mining: modélisation non supervisée : classifications automatiques », 2008 .

[20] : François-Régis Chaumartin. WordNet et son écosystème : un ensemble de ressources linguistiques de large couverture.

[21] : Sini Ghenima.Méthode et outils pour la gestion des workflow-modélisation des processus de gestion pour l analyse,2013 .

[22] : Alain Polguère , Ophélie Tremblay Une ontologie linguistique au service de la didactique du lexique,2014 .

[23] :zaidi soraya. Une plateforme pour la construction d'ontologie en arabe : Extraction des termes et des relations à partir de textes (Application sur le Saint Coran),2012.

[24]:https:///wordnet.princeton.edu/wordnet/(date d'accès 22/04 /2017).

[25]: Gerard swinnen.Apprendre a programmé avec python.

[26] : https://fr.wikipedia.org/wiki/Ontologie_(informatique) (date d'accès 18/04/2017).

[27] : http://dictionnaire.sensagent.leparisien.fr/WordNet/fr-fr/(date d'accès 05/05/2017).

[28] : http://wordnet.princeton.edu/wordnet/man/wnstats.7WN.html(date d'accès 20/05/2017).