REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

Université **MOULOUD MAMMERI** TIZI-OUZOU
Faculté de Génie Electrique et d'Informatique
Département d'Informatique

Mémoire de fin d'études

En vue de l'obtention du diplôme de Master 2
en informatique

Option : Systèmes Informatiques

Thème :

Temporal Annotation of English Text

Dirigé par

Mme BOUGCHICHE L.

Réalisé par

Mlle BOUABACHA FAZIA

Promotion 2016*/2017*

Remerecement

A l'occasion de ce fameux mémoire de fin d'étude, dans l'objectif d'obtention du master 2. Nous tenons à remercier le bon dieu d'abord, qui nous a donné la force de travailler, et la capacité d'étudier.

Nous remercions aussi madame BOUGHCHICHE, qui nous a consacré son temps précieux pour nous guider et diriger vers un excellent travaille.

De même nous remercions l'ensemble des jurys, qui nous ont jugés avec droiture.

Toutes personnes ayant contribuées de près ou de loin à l'aboutissement de ce projet.

Dedicace

Je dédie ce merveilleux travaille à mes chères pères et mères, qui sont la cause de mon existence, et qui m'ont rempli d'amour et d'affection. Ainsi à mon petit frère MOHAMED ISLAM, qui est au collège, et que j'aime beaucoup. J'espère qu'un jour pourra réaliser tout ses rêves. A mon oncle dieu le bénisse et sa femme. A toute la famille et tout mes amis et tous ceux qui m'aime.

B. FAZIA

SOMMAIRE

Chapitre4 : Conception et réalisation.

Conclusion génarale

Liste des figures

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'annotation 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 connaissance lexicale WordNet c'est-à-dire on a applique l'apprentissage automatique sur l'ensemble des concepts de cette base de connaissance.

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 la classification, nous présentons la classification avec les méthodes de cette dernière.

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

La 4eme porte sur l'annotation 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.

CHAPITRE 1

Traitement automatique des langages (TAL)

Chapitre 1 : traitement automatique des langages (TAL).

Introduction :

Le Traitements Automatique des Langues (TAL) est une discipline qui associe étroitement linguistes et informaticiens. Il repose sur la **linguistique**, les **formalismes** (représentation de l'information et des connaissances dans des formats interprétables par des machines) et l'**informatique**.

Le TAL a pour objectif de développer des logiciels ou des programmes informatiques capables de traiter de façon automatique des données linguistiques.

Pour traiter automatiquement ces données, il faut d'abord expliciter les règles de la langue puis les représenter dans des formalismes opératoires et calculables et enfin les implémenter à l'aide de programmes informatiques.

Les principaux domaines du TAL sont:
• le traitement de la parole;
• la traduction automatique ;
• la compréhension automatique des textes;
• la génération automatique de textes ;
• la gestion électronique de l'information et des documents existants (GEIDE).

I. Définition :

Le TAL est l'ensemble des méthodes et des programmes qui permettent un traitement par l'ordinateur des données langagières, mais quand ce traitement tient compte des spécificités du langage humain. Il y a des traitements de données langagières (écritures sur fichiers, sauvegardes ou autres) qui ne font pas partie du traitement automatique des langues. [François Yvon]

Le TAL est destiné à deux sorte de publics différents aussi : Les chercheurs s'adressent à leur propre communauté de chercheurs, à leurs étudiants et ils s'adressent aussi d'une certaine façon aux industriels. Quant aux industriels, 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.
Ainsi le TAL est destiné à deux sorte d'individus. Les linguistiques et les informaticiens. Dans ce qui suit, on verra le rapport qu'il y ait entre les deux types d'individus.

II. Dialogue entre les linguistes et les informaticiens :

Dans le domaine de la recherche, on peut séparer deux catégories bien nettes entre linguistes et informaticiens, parce qu'il n'y a pas dans ce domaine des purs linguistes et des purs informaticiens. Il y a des gens qui ont une formation initiale plutôt en informatique et qui se sont formés à la linguistique. Par ailleurs il y a d'autres personnes, dont la formation initiale est plutôt en linguistique ou en langues, qui se sont formés à l'informatique.

III. Les techniques mises en œuvre :

Pour faire le traitement automatique des langages on a besoin de deux pôles ; les linguistiques et les informaticiens. Ce qui fait, il faut avoir des techniques linguistiques et des techniques pragmatiques.

1- D'un côté les techniques linguistiques ou à base de linguistique. Elles sont plutôt le fait des chercheurs et elles consistent à avoir une représentation, une modélisation des langues et des données langagières. Les techniques linguistiques permettent de développer d'ailleurs une recherche linguistique pure de modélisation des données langagières. C'est une recherche en linguistique et une recherche en informatique, puisqu'il s'agit de définir des modèles et des algorithmes sur ces modèles de données langagières.

2- D'un autre côté, il y a des techniques plus pragmatiques, parmi lesquelles les techniques dominantes sont des techniques à base de statistiques et de probabilités avec des apprentissages sur des corpus de données. À partir de ces apprentissages sur lesquels on fait des calculs de fréquence, on en déduit des probabilités qui permettent de donner des résultats avec un certain degré de fiabilité.

IV. Les freins technologiques :

La linguistique est une science humaine. Il s'agit de modéliser le comportement humain. Il y a des freins épistémologiques plutôt que technologiques à la modélisation du comportement humain, parce que les êtres humains ont une liberté de se comporter. D'une certaine façon, on peut dire que le langage n'est pas réductible à une machine et qu'une machine n'est pas susceptible de résoudre complètement des problèmes de modélisation du comportement humain et en particulier du comportement langagier.

Exemple : l'ambigüité lexicale du langage. Comme :

- Souris : forme verbales de sourire, nom féminin singulier et pluriel.
- Petit : adjectif ou nom masculin singulier
- La : déterminant ou pronom personnel féminin singulier, nom masculin
- Mousse : formes verbales de mousser, nom masculin, nom féminin.

V. Domaines d'application :

Le traitement automatique des langages est utilisé dans diverses applications, qu'on peut résumer ainsi.

- Recherche d'informations : ça fait partie du TALN il ya des algorithmes qui sont capable de comprendre la sémantique du texte; exemple test de Turing **(voir le point VIII de ce chapitre)**.
- Reconnaissance de la parole : dans ce cas, on doit manipuler un signal acoustic qui est un signal sonore et qui correspond au son de la voie d'une personne. A partir de ce

signal, on doit déterminer la phrase qui a été formulée. En passant par des phonèmes qui représentent des syllabes ensuite des words qui représentent des mots.

- Système de réponse automatique (WATSON) : c'est un système de « questing and answering » comme le système « WATSON IBM », qui a participé à une émission et qui a réussi à battre les deux meilleurs joueurs « Brad et Ken ».
- la traduction automatique (historiquement la première application, dès les années 1950) : on donne un texte dans un langage (langue source) par exemple en anglais et qui sera traduit dans un autre langage (langue cible) par exemple Français ;

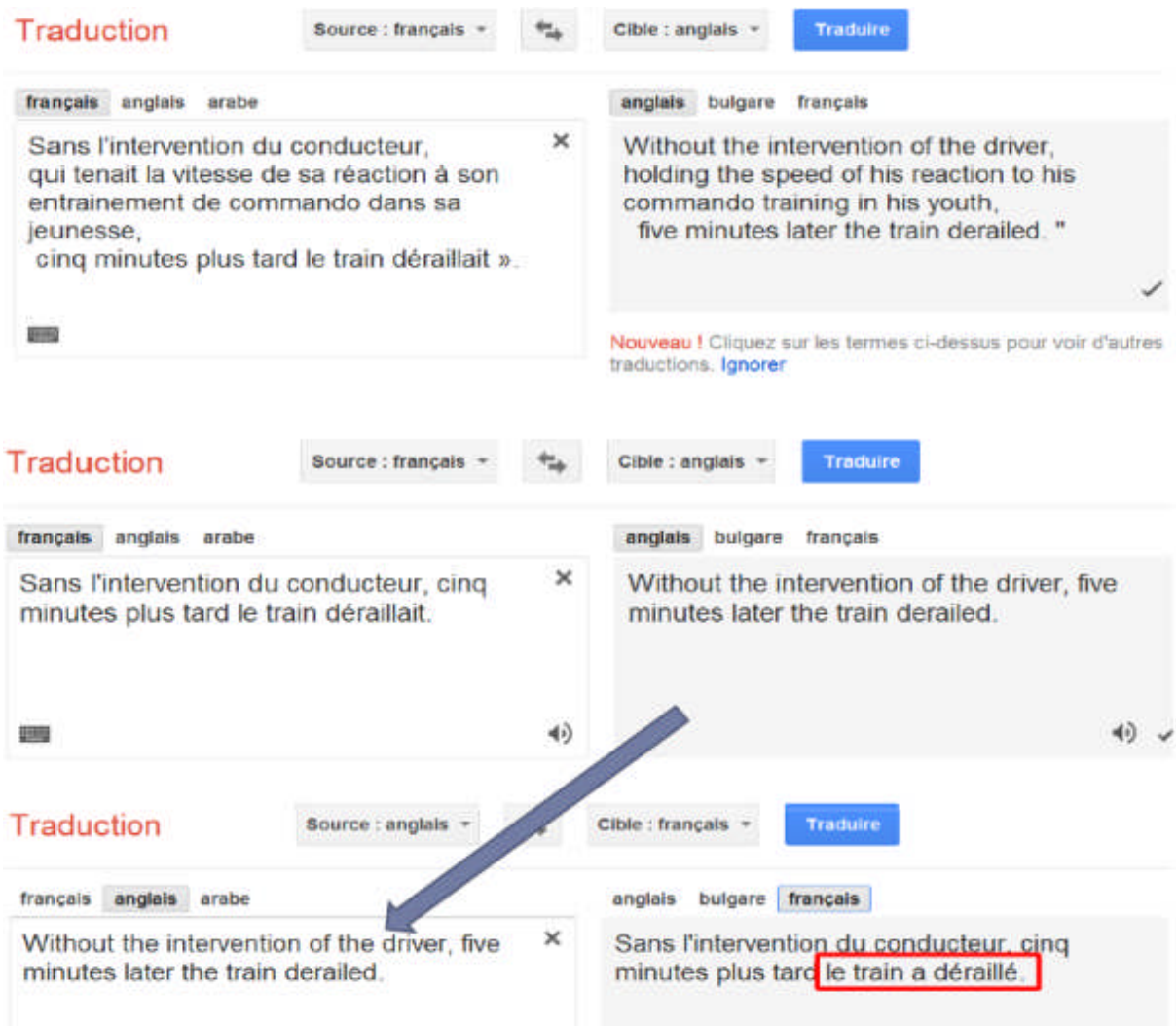

Figure 1 : Traduction automatique d'un texte.

- la génération automatique de textes ; exemple générer un récit, document technique à partir de données structurée.

- la correction orthographique ; exemple correcteur associé à des traitements de textes.

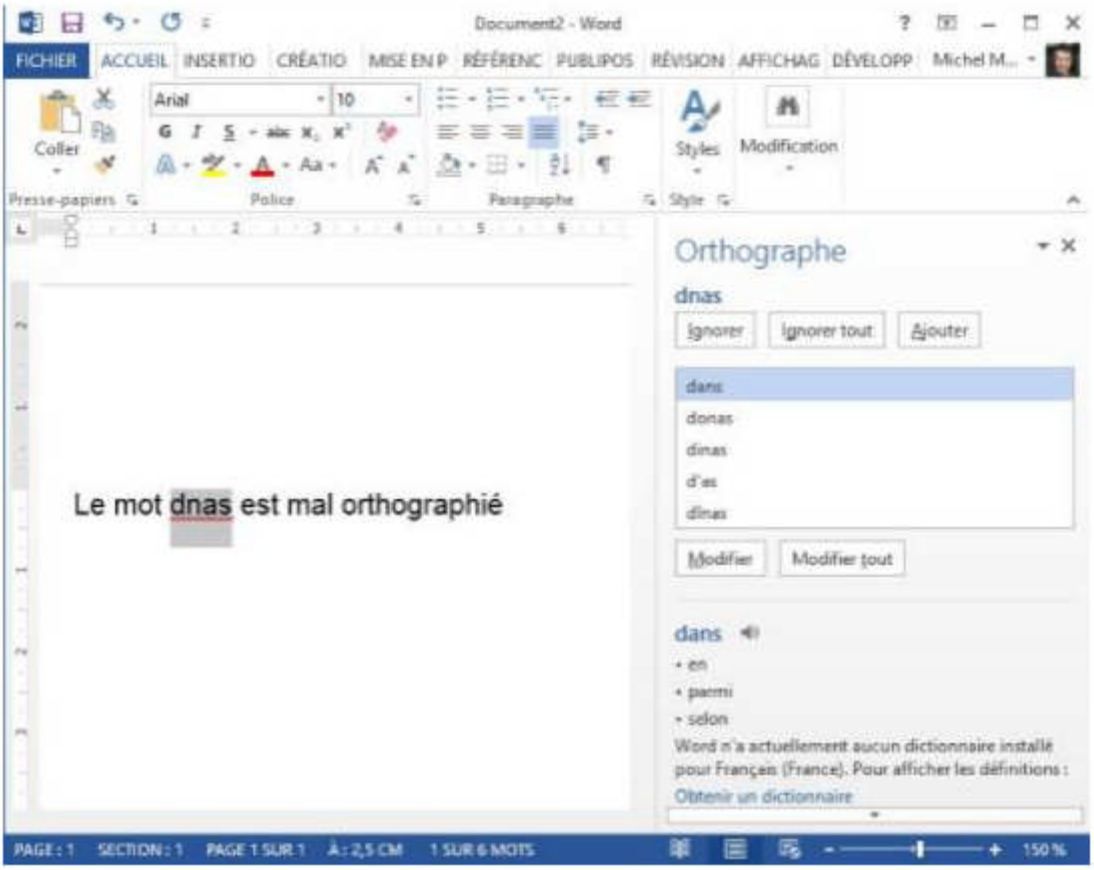

Figure 2 : Correction orthographique.

- le résumé automatique de texte ; exemple résumé automatique.
- Assister une personne dans l'apprentissage d'une langue ; exemple didactique des langues
- Assister une personne handicapée dans la formulation d'énoncé ; exemple communication assistée.

Les applications en relation avec le traitement du signal :

- la reconnaissance automatique de la parole ;
- la synthèse de la parole ;
- le traitement de la parole.

Les applications en relation avec l'extraction d'information :

- la recherche d'information et la fouille de textes ;

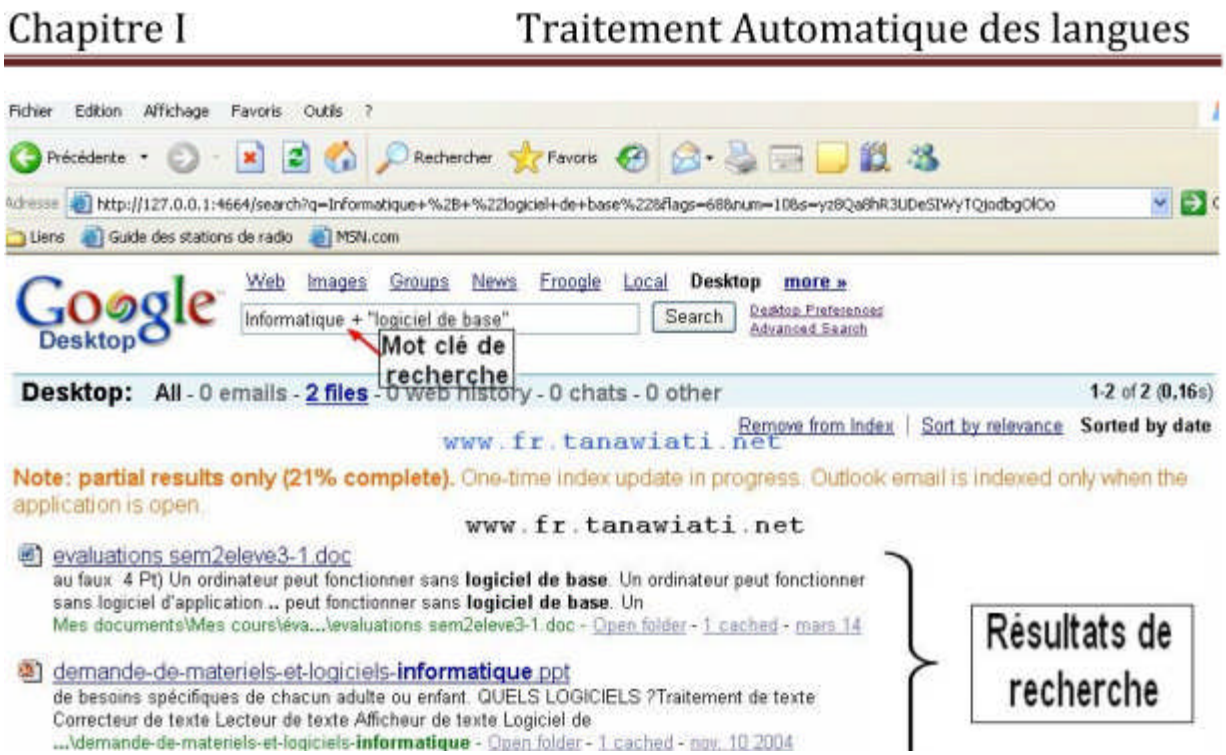

Figure 3 : Recherche d'informations.

- la reconnaissance d'entités nommées ; étant donné un texte, déterminer les noms propres, tels que des personnes ou des endroits;
- l'annotation sémantique ;
- la classification et la catégorisation de documents ;
- la détection de coréférences.
- Systèmes de tutorat intelligents notamment pour l'enseignement des langues ;
- L'analyse des sentiments.
- La recommandation automatique de documents ;
- La détection des langues et des dialectes tant à partir des textes ou des énoncés parlés.

VI. La bibliométrie :

La bibliométrie est l'utilisation du traitement automatique des langues sur des publications scientifiques.

La première étude d'envergure a été réalisée en 2013 à l'occasion de l'anniversaire de l'*Association for Computational Linguistics* (ACL) avec un atelier intitulé « *Rediscovering 50 Years of Discoveries in Natural Language Processing* ».

La même année a eu lieu l'action NLP4NLP, opération de bibliométrie d'application des outils de TAL aux archives du TAL depuis les années soixante jusqu'à nos jours par Joseph Mariani, Gil Francopoulo et Patrick Paroubek. Il s'agit par exemple de déterminer automatiquement quels sont les inventeurs des termes techniques que nous utilisons actuellement. Un autre champ d'étude est de déterminer quels sont les copier-coller éventuels que les chercheurs du TAL effectuent quand ils écrivent un article scientifique.

Un exemple de bibliométrie, l'accès généralisé aux données de citation via Google Scholar, et la mise à disposition d'outils bibliométriques. Comme « Publish ».

Google Scholar a été lancé en version beta à la fin de l'année 2004 et référence les articles scientifiques. A partir de Google Scholar, l'outil bibliométrique Publish permet de calculer quelques indicateurs bibliométriques par auteur, revue ou article.

VII. TAL statistique :

Les utilisations statistiques du traitement du langage naturel reposent sur des méthodes stochastiques, probabilistes ou simplement statistiques pour résoudre certaines difficultés, particulièrement celles qui surviennent du fait que les phrases très longues sont fortement ambiguës une fois traitées avec des grammaires réalistes, autorisant des milliers ou des millions d'analyses possibles. Les méthodes de désambiguïsation comportent souvent l'utilisation de corpus qui veut dire collection de textes/documents et d'outils de formalisation. Le TAL statistique comporte toutes les approches quantitatives du traitement linguistique automatisé, y compris la modélisation, la théorie de l'information, et l'algèbre linéaire. La technologie pour TAL statistique vient principalement de l'apprentissage automatique et le data mining, tous deux en tant qu'ils impliquent l'apprentissage à partir des données venant de l'intelligence artificielle.

Exemple : statistique des mots du roman Tom Sawyer

Word frequency	Frequency of frequency
1	3993
2	1458
3	4781
4	5874
5	6597
6	112
7	74
8	58
9	758
10	22
11-50	12
51-100	12
>100	12

Word frequency : désigne le nombre de répétition du mot.
Frequency of frequency : désigne combien de fois le mot a été répété.

On remarque que les mots qui n'apparaissent pas beaucoup sont les plus répété.
Ces mots sont appelés fréquents et appartiennent à des classes fermées comme les déterminants (un, son,….)
La fréquence f d'un mot est inversement proportionnelle à son rang r tel que :
F * r = k
Et k : est une constante.
r : l'ordonnancement décroissant des mots.
On appelle cette loi : loi de Zipf

On a parlé dans ce sous titre de data-mining qui veut dire « différents types d'application » : classification qu'on verra plus tard, régression dont la variable d'intérêt est continue comme la fabrication industrielle, et segmentation dont les variables d'entrée servent à créer des groupes comme la segmentation (clustering) des clients dans le cadre d'une compagne marketing.

VIII. La chaine de traitements « standard » :

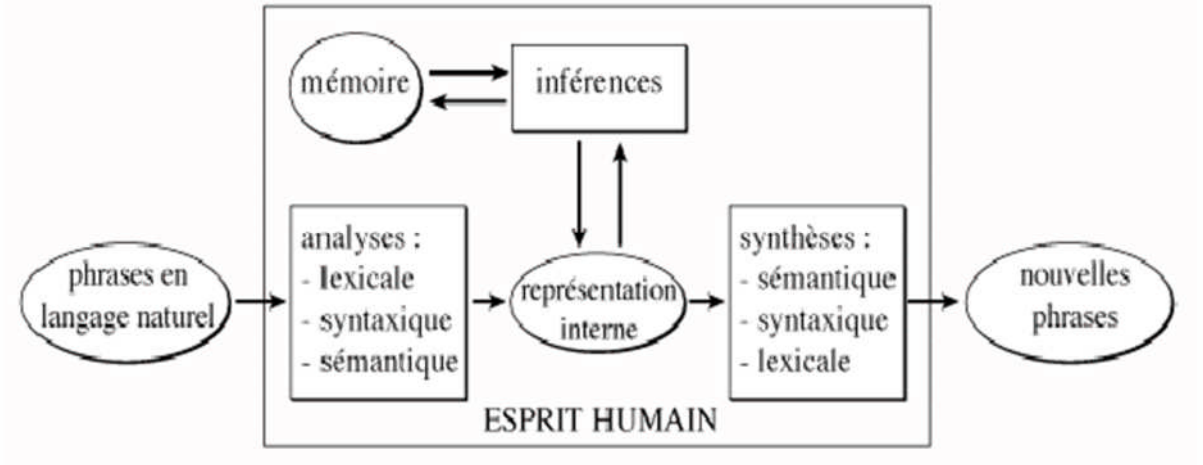

Figure 4 : chaine de traitements classique de la compréhension du langage.

Dans ce schéma, ici encore très simplificateur (il n'intègre pas, par exemple, la composante orale du langage), les données figurent dans des ovales, tandis que les traitements sont représentés dans des rectangles. Les deux principaux rectangles, intitulés respectivement "analyses" et "synthèses", correspondent aux deux tâches principales accomplies par les locuteurs d'une langue. Le fait de bien les séparer provient de l'observation de patients

souffrant de lésions cérébrales qui affectent en particulier une de ces compétences et pas l'autre. Il n'est pas nécessaire pourtant de faire l'hypothèse que chacun des traitements évoqués dans ce schéma soit réalisé par une aire cérébrale spécifique. Il suffit de considérer qu'il met en évidence un enchaînement des *fonctions*, indépendamment de leur "implantation" dans un substrat biologique concret.

Suivant ce schéma, *comprendre un énoncé* revient donc à le transformer, via une "analyse", en une représentation interne, tandis que pour en *produire* ou en *générer* un, il faut traduire linguistiquement une telle représentation via une "synthèse". Chacune de ces tâches nécessite de prendre en compte l'ensemble des niveaux d'analyse identifiés précédemment, mais dans un ordre différent et en faisant chaque fois des hypothèses différentes sur ce qui est connu et ce qui doit être accompli.

Maintenant, pour construire un système artificiel complet avec lequel des humains pourraient interagir via une langue naturelle, une première approche possible consiste à tenter de reproduire une architecture de ce genre, en traduisant les "fonctions" en programmes. C'est en quelque sorte le projet du "traitement automatique du langage naturel" dans sa forme originelle. Il a donné lieu à de très nombreux travaux ces 50 dernières années, beaucoup des outils formels imaginés pour modéliser certains des traitements sont en fait exploitables à la fois en analyse et en synthèse (c'est le cas, par exemple, des automates et grammaires formelles). D'autres sont plus spécifiques.

Pourtant, le schéma de la **figure 1** n'est instancié dans sa totalité dans aucun système informatique existant. C'est un cadre idéal théorique, très influencé par une conception *cognitiviste* de l'esprit humain, où les "représentations internes" sont en général de nature symbolique. Pour une application particulière, on se limite la plupart du temps à implémenter une toute petite portion de cette architecture.

Mais il est aussi possible d'envisager une toute autre approche, qui ne se soucie pas de crédibilité psychologique, et se concentre sur l'efficacité pragmatique de ses programmes. Cette mutation est représentative de l'évolution qu'a connue l'"intelligence artificielle" ces dernières années. Après avoir longtemps tenté d'imiter le fonctionnement de l'esprit humain, les chercheurs du domaine essaient plutôt désormais d'exploiter au mieux les capacités de mémoire et de calculs de leurs machines. Des *modèles symboliques formels*, on est passé aux *modèles statistiques fondés sur l'analyse des données*.

- **Machine de « Turing » :**

Le "test de Turing" est un jeu au cours duquel un juge humain, en situation de dialogue médiatisé par une machine avec une entité distante, doit décider au bout d'un temps fixé à l'avance si son interlocuteur est un humain ou un programme. Son origine remonte à un article philosophique que Turing, l'inventeur de l'informatique, a fait publier en 1950. Selon lui, un programme qui "passerait ce test" -autrement dit qui ne serait pas identifié comme tel par le juge- devrait se voir reconnu les mêmes capacités que celles attribuées "naturellement" aux humains -en particulier celle de penser... Ce test fait donc des capacités linguistiques en situation de dialogue le critère principal de l'"intelligence".

Il existe à l'heure actuelle une compétition qui propose d'instancier le "test de Turing" dans un cadre contrôlé : elle s'appelle le "Loebner Price", du nom du généreux donateur qui offre un prix et une médaille au vainqueur.

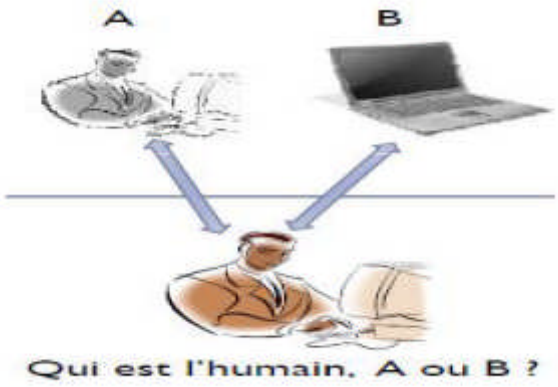

Figure 5 : Machine de Turing.

IX. Niveaux de formation et de représentation :

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 d'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.

L'étude des aspects lexicaux et syntaxiques permet de définir le caractère syntaxiquement bien formé des expressions linguistiques, et d'en donner une représentation syntaxique. Dans l'étude des aspects sémantiques, on distingue deux étapes. La première, purement formelle, consiste à étudier hors contexte le sens des expressions linguistiquement bien formées pour leur associer une représentation sémantique correspondant à leur sens littéral. C'est lors de

cette étape que l'on rend compte des phénomènes conceptuels pour décider du caractère conceptuellement bien formé des représentations sémantiques, ou en d'autres termes, pour vérifier les présuppositions lexicales.

X. 1. Lexique et syntaxe :

On veut définir un sous ensemble de phrases du Français au moyen d'un lexique et d'une grammaire en vue d'analyser et de synthétiser automatiquement des phrases de ce sous ensemble.

Il n'est pas question pour nous de couvrir l'ensemble du vocabulaire et de la syntaxe du Français. Deux contraintes définissent la couverture de notre grammaire :

- Cette grammaire doit être suffisamment générale pour constituer un noyau utilisable dans différentes applications. Sa définition formelle doit faciliter de futures extensions.
- Une représentation sémantique de type logique doit pouvoir être associée à chaque phrase définie par la grammaire.

Un lexique définit l'ensemble des mots (et des expressions) au moyen desquels des phrases peuvent être construite. A chaque mot est associé un ensemble d'informations sur la base desquelles les autres niveaux de représentation et de bonne formation pourront être établis.

Une grammaire définit au moyen de règles de réécriture les phrases d'un langage en les structurant en chaines de constituants et de sous constituants aussi appelées **catégories**. Parmi les catégories, on distingue les catégories lexicales et les catégories syntaxiques. Les catégories lexicales (comme exemple, article, préposition, nom commun, verbe) constituant les catégories de base. Les catégories syntaxiques (comme par exemple, phrase, groupe nominal, groupe verbal) constituent les catégories supérieurs. Elles structurent des suites de catégories de base et/ou de catégories supérieures.

XI. 2. Sémantique :

D'une façon générale, on s'intéresse au traitement automatique des expressions linguistiques de type phrase. Une assertion ou une question sont des types de phrases. Une assertion présuppose des connaissances sur un modèle et en exprime d'autres. Une question présuppose des connaissances sur un monde

Et exprime une demande de connaissances sur ce monde. Qu'ils s'agissent d'une assertion ou d'une question, une phrase formule donc des connaissances sur un monde, c'est-à-dire exprime une situation particulière mettant en jeu des individus et des relations. La représentation sémantique d'une phrase donnée doit permettre d'exprimer sans ambigüité et de façon précise cette situation particulière. Cette situation particulière est-elle ou non en contradiction avec celle précédemment décrite ? La valeur sémantique de cette représentation doit nous permettre d'en décider, et cela en lui attribuant une valeur de vérité : vrai ou faux. On ne peut parler de valeur sémantique d'une phrase que par référence à un ensemble de situations. Pour nous, la représentation et la valeur sémantique d'une phrase sont indissociables. On parlera alors du sens d'une phrase, et le traitement du langage naturel qui nous intéresse ici est celui qui consiste à obtenir automatiquement ce sens.

A nos yeux, deux aspects restent fondamentaux dans la nature de la représentation

sémantique recherchée. On s'intéresse d'une part à savoir quelle est la valeur d'une représentation sémantique particulière par rapport à un ensemble de connaissances décrivant un univers de référence. On peut appeler cet aspect **évaluation.** D'autre part, on s'intéresse aux façons dont les expressions de ce langage de représentation sont reliées entre elles, et puis particulièrement aux inférences réalisables au sein de l'ensemble des expressions. On peut appeler ici cet aspect **déduction.**

Les deux aspects essentiels que sont l'évaluation et la déduction font de la **logique** le seul langage pertinent pour la représentation sémantique des phrases.

En effet, c'est le seul à posséder une syntaxe et une sémantique rigoureusement définies, et à offrir une théorie déductive rigoureuse. La syntaxe du langage de la logique définit l'expression correcte du langage : les **termes** et les **formules.** Parmi ces dernières, on distingue les formules atomiques qui sont les plus élémentaires. La sémantique du langage de la logique définit les règles de calcul permettant d'associer à toute formule une valeur de vérité (vrai ou faux) à partir des valeurs de vérité des formules atomiques. Les valeurs de vérité des formules atomiques constituent ce qu'on appelle une **interprétation (**ensemble de situations de référence**).** La théorie déductive permet de mettre en relation (implication logique) les représentations logiques. Pour un ensemble donné de connaissances représentées par un ensemble de formules, elle permet de déduire d'autres connaissances non explicites.

Comment associer une représentation sémantique aux phrases d'un langage ?

On fait l'hypothèse que la représentation sémantique (de type logique ou pas) d'une phrase se présente sous la forme d'une structure non ambiguë. Si une phrase est ambiguë, plusieurs représentations sémantiques non ambiguës lui seront donc associées. On fait aussi l'hypothèse que la représentation sémantique associée à une phrase est dépendante des éléments qui constituent cette phrase. Elle peut être alors composée sur la base des représentations sémantiques que l'on peut associer aux constituants de la phrase. On parle alors de **sémantique compositionnelle.** Le principe d'une sémantique compositionnelle consiste à :

- Associer à chaque catégorie (lexicale et syntaxique) un type de structure sémantique pour la construction de la représentation sémantique.
- Associer à chaque règle de grammaire une règle de composition combinant les différentes structures sémantiques associées aux différentes catégories de la règle.

Une telle composition peut être exprimée au moyen du formalisme du **lambda-calcul.** Dès qu'une phrase est lexicalement et syntaxiquement bien formée. Une (ou plusieurs, en cas d'ambigüité) représentations sémantiques peut lui être ainsi associée. C'est sur la base de cette représentation qu'il sera décidé si la phrase et ou non conceptuellement et contextuellement bien formée.

XII. 3. Modèle conceptuel :

Une phrase est conceptuellement bien formée si la représentation sémantique associée décrit une situation conceptuelle possible, c'est-à-dire si les relations et les individus ou objets qu'elle met en jeu sont compatibles. L'expression d'une telle compatibilité ne relève pas de contraintes syntaxiques ou grammaticales mais d'une modélisation extralinguistique de l'ensemble des situations de références potentielles, modélisation spécifiée au moyen de ce qu'on appellera le modèle conceptuel.

Une phrase comme :

« ***Une pomme dort »***

Est syntaxiquement et grammaticalement bien formée, et on peut naturellement lui associer une représentation sémantique. Cependant si le modèle conceptuel exprime que seuls les individus animés peuvent dormir, et qu'une pomme n'est pas un individu animé, alors on dira que la phrase ci-dessus est conceptuellement mal formée.

XIII. 4. Modèle contextuel:

Il y a des phénomènes qui ne peuvent être résolus qu'en tenant compte du contexte dans lequel une phrase est formulée. Ce contexte est déterminé par le **monde de référence** (par exemple le contenu de la base de données interfacée) et par le discours, c'est-à-dire par l'ensemble des phrases précédemment énoncées. Les représentations du monde de référence et du discours forment le **modèle contextuel.** C'est dans le cadre de ce modèle que sont traités en particulier les présuppositions existentielles et le pro formes. Par exemple, une expression comme :

« ***La date de naissance du fils de Max »***

Est conceptuellement bien formée mais présuppose l'existence (présupposition existentielle) de Max, du fils de Max (Max a au moins un fils). Cette existence ne peut être vérifiée qu'en consultant le monde de référence.

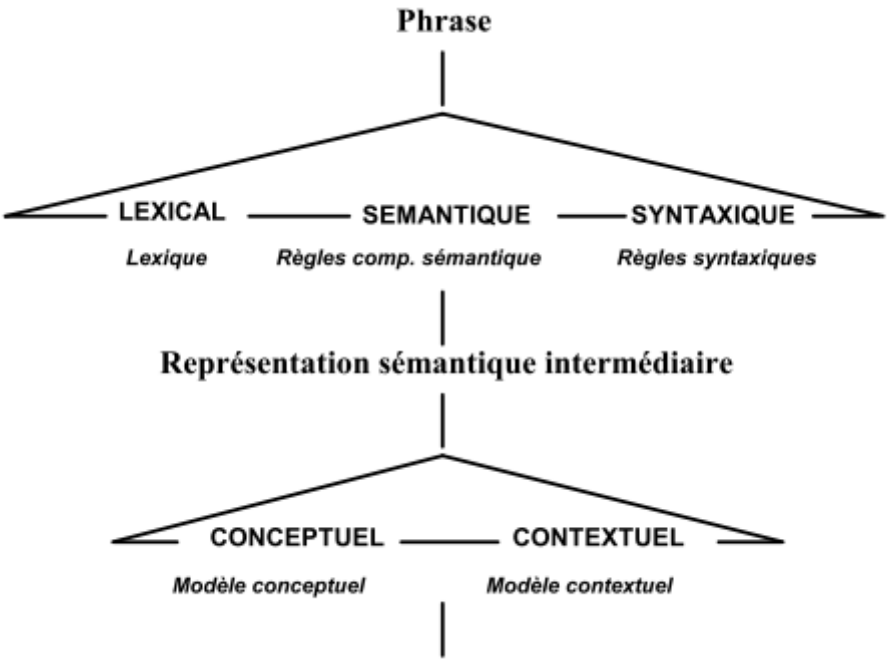

Figure 6 : Représentation sémantique finale.

XIV. Analyse et synthèse de phrases :

Le processus d'analyse peut être réalisé de plusieurs façons. Une façon de faire est
(1) de fondre dans un même formalisme les différentes connaissances (lexicales, syntaxiques, sémantiques, conceptuelles, contextuelles) relevant des différents niveaux de représentation et de bonne formation, et
(2) de réaliser l'analyse en une seule étape. Cette approche a l'inconvénient de conduire à la réalisation de systèmes difficilement extensibles et portables vers d'autres applications.
Une autre façon de faire consiste
(1) à séparer les différents types de connaissances dans des modules distincts, et
(2) à réaliser séquentiellement, les différentes analyses propres aux différents niveaux de représentation et de bonne formation. Ce qui donne, dans l'ordre : l'analyse lexicale, l'analyse syntaxique, l'analyse conceptuelle, l'analyse contextuelle.

Une autre approche, et celle qui nous intéresse ici, consiste à séparer les différents types de connaissances dans des modules distincts et à conduire en même temps les différentes analyses correspondant aux niveaux de bonne formation et de représentation. Si la phrase contient par exemple une incohérence conceptuelle, le principe est d'interrompre au plus tôt l'analyse syntaxique du reste de la phrase. On s'intéresse aussi à synthétiser des phrases. Là aussi, on veut produire des phrases en utilisant en même temps, comme en analyse, les connaissances relevant des différents niveaux de représentation et de bonne formation. Mieux encore, on souhaite que le même système de traitement automatique puisse fonctionner en analyse et en synthèse. Ainsi, par exemple, lorsqu'une incohérence lexicale, syntaxique ou conceptuelle est détectée en analyse en un point particulier d'une phrase, le système doit être capable de produire en synthèse l'ensemble des mots attendus à ce point précis de la phrase et qui conduiront d'une phrase bien formée à tous les niveaux.

Conclusion

On a vu pour l'instant le TAL qui est un domaine d'application de l'informatique qui concerne l'interaction langagière entre l'ordinateur et l'être humain. Dans le chapitre suivant, on parlera de la classification. Avec ses deux types, et on verra aussi les méthodes utilisées dans chaque type.

.

CHAPITRE 2

Les méthodes de classification (supervisées et non supervisées)

Introduction :

Le but de la classification est d'obtenir une représentation schématique simple d'un tableau rectangulaire de données, dont les colonnes sont des descripteurs de l'ensemble des observateurs, placées en lignes.

L'objectif le plus simple d'une classification est de répartir les données en groupes d'observations homogènes, chaque groupe étant bien différencié des autres. On veut en général, obtenir des sections à l'intérieur des groupes principaux, puis des subdivisions plus petites de ces sections, et ainsi de suite. En bref, on désire avoir une hiérarchie, c'est-à-dire une suite de partitions « emboitées », de plus en plus fines, sur l'ensemble d'observations initial.

Une telle hiérarchie peut avantageusement être résumée par un arbre hiérarchique ***(Figure 7)*** dont les nœuds (m, n, p, q) symbolisent les diverses subdivisions de l'échantillon ; les éléments de ces subdivisions étant les objets (a, b, c, d, e), placés à l'extrémité inférieure des branches qui leur sont reliées.

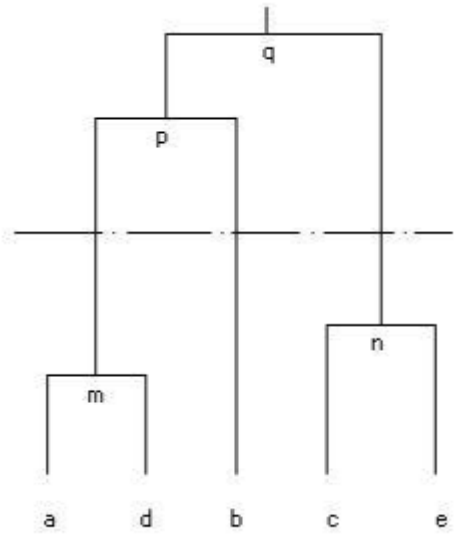

Figure 7 : exemple d'arbre hiérarchique.

a, b, c, d, e : les objets (les individus).
m, n, p, q : les nœuds de l'arbre.
Le trait horizontal mixte indique un niveau de troncature définissant une portion en trois classes.
On remarque que, si on coupe cet arbre à un niveau intermédiaire entre n et p, on obtient une partition en trois classes de l'ensemble étudié, à savoir les parties {a, d}, {b}, {c, e}.

La différence entre la classification et le classement. Dans un classement on affecte les objets à des groupes préétablis : c'est le but de l'analyse discriminante que de fixer des règles pour déterminer la classe des objets. La classification est donc, en quelque sorte, le travail préliminaire au classement, savoir la recherche des classes « naturelles » dans le domaine étudié.
Alors quelles sont les méthodes de classification les plus connues.

Chapitre 2 : Les méthodes de classification (supervisées et non supervisées)

I. Définition de la classification :

Classification : action de constituer ou construire des classes.
Classe : ensemble d'individus (ou d'objets) possédants des traits de caractères communs.
Exemples

- De classification : règne animal, disque dur d'un ordinateur, division géographique de la France.
- De classe : classe sociale, classe politique.

II. Problèmes et méthodes de la classification automatique :

Un algorithme est une description minutieuse de toutes les opérations à effectuer pour obtenir la solution concrète d'un problème. Ainsi on peut parler de l'algorithme permettant de trouver la racine carrée d'un nombre, ou bien pour obtenir le plus grand commun diviseur de deux nombres entiers, etc. …Il ne faut pas confondre algorithme et programme informatique : il peut y avoir plusieurs façons de programmer un même algorithme.

Supposons que l'on veuille, par exemple, construire une hiérarchie, l'une des manières de « bien poser » le problème pourrait être de choisir un critère évoluant la fidélité de la représentation hiérarchique au tableau initial des données, et de trouver ensuite un algorithme construisant la meilleure hiérarchie au sens de ce critère. Malheureusement on ne sait pas faire cela sauf pour des échantillons très petits, ou pour des critères sans intérêt. La solution qui consiste à examiner l'ensemble de toutes les hiérarchies possibles. Le nombre de hiérarchies croit si vite avec le nombre d'objets que, même avec de puissants ordinateurs, il n'est pas réaliste de vouloir les envisager toutes. C'est pourquoi l'on a recours à des heuristiques, c'est-à-dire des algorithmes dont on considère qu'ils sont suffisamment raisonnables vous donner des résultats satisfaisants.

Grossièrement on peut distinguer trois grands types parmi ces heuristiques. Il y a d'abord les algorithmes construisant une hiérarchie par agrégation successive d'objets, puis de groupes, en fonction des distances entre objets ou groupes. On les appelle « constructions ascendantes de hiérarchies », en abrégé CAH. A l'inverse « les constructions descendantes de hiérarchies », en abrégé CDH, procèdent par dichotomies successives. Dans celles-ci l'ensemble tout entier est d'abord scindé en deux, puis chacune de ces parties est, à son tour subdivisée, et ainsi de suite. Dans le troisième groupe de méthodes on peut rassembler toutes celles qui se limitent à l'élaboration d'une partition. Par des algorithmes très diverses, ces méthodes ont pour objectif de détecter les zones à forte densité dans l'espace des observations.

III. Domaines d'application et points de vocabulaire :

La classification a un rôle à jouer dans toutes les sciences et techniques qui font appel à la statistique multidimensionnelle. Citons tout d'abord les sciences biologiques : botanique, zoologie, écologie, … ces sciences utilisent également le

terme de « taxinomie » pour désigner l'art de la classification. De même les sciences de la terre et des eaux : géologie, pédologie, géographie, étude des pollutions, font grand usage de classifications.

La classification est fort utile également dans les sciences de l'homme : psychologie, sociologie, linguistique, archéologie, histoire, etc. … et dans les techniques dérivées comme les enquêtes d'opinion, le marketing, etc. … Ces dernières emploient parfois des mots de « typologie » et « segmentation » pour désigner la classification, ou l'une de ses innombrables variantes. Citons encore la médecine, l'économie, l'agronomie, et nous en oublions certainement !

Dans toutes ces disciplines, la classification peut avoir deux types. Classification supervisée et classification non supervisée.

Dans ce qui suit, on essayera de voir la différence entre une classification supervisée et une classification non supervisée. On expliquera aussi les différentes méthodes pour les deux types de classification.

1) Classification supervisée :

L'objectif de la classification supervisée est principalement de définir des **règles permettant de classer** des objets dans des classes à partir de variables quantitatives ou qualitatives caractérisant ces objets. Les méthodes s'étendent souvent à des variables Y quantitatives (régression).

Figure 8 : schémas de classification supervisée.

On dispose au départ d'un **échantillon d'apprentissage** dont le classement est connu. Cet échantillon est utilisé pour l'apprentissage des règles de classement.

Il est nécessaire d'étudier la **fiabilité** de ces règles pour les comparer et les appliquer, évaluer les cas de sous apprentissage ou de sur apprentissage (complexité du modèle). On utilise souvent un deuxième échantillon indépendant, dit de variation ou de test.

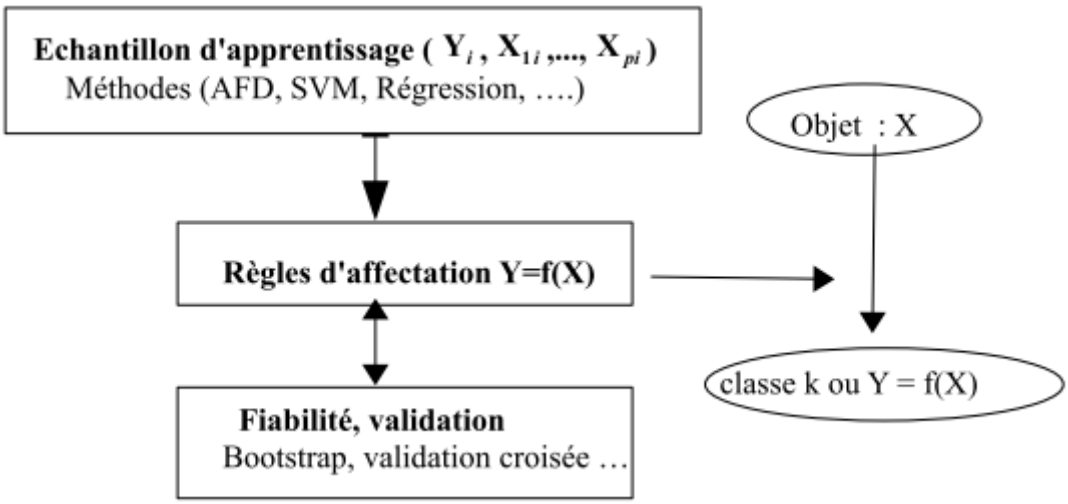

Figure 9 : Processus de classification.

A partir de là on commence avec les méthodes de la classification supervisée.

1. 1. Arbre de décision :

Un arbre de décision est un classifieur représenté sous forme d'arbre tel que :

- Les nœuds de l'arbre testent les attributs
- Il y a une branche pour chaque valeur possible de l'attribut testé
- Les feuilles spécifient les catégories (deux ou plus)
- Ils fonctionnent facilement sur des données qualitatives

- **Arbres de classification** : la variable expliquée est de type nominal (facteur). A chaque étape du partitionnement, on cherche à réduire l'impureté totale des deux nœuds fils par rapport au nœud père.
- **Arbre de régression** : la variable expliquée est de type numérique et il s'agit de prédire une valeur la plus proche possible de la vraie valeur.
  Construire un tel arbre consiste à définir une suite de nœuds permettant de faire une partition des objets en deux groupes sur la base d'une des variables explicatives. Il convient donc :
- De définir un critère permettant de **sélectionner le meilleur nœud** possible à une étape donnée.
- De définir quand s'arrête le découpage, en définissant un **nœud terminal** (feuille).
- D'attribuer au nœud terminal la classe ou la valeur la plus probable.

- D'**élaguer l'arbre** quand le nombre de nœuds devient trop important ou sélectionner un sous arbre optimal à partir de l'arbre maximal.
- Valider l'arbre à partir d'une validation croisée ou d'autres techniques.
- Critère de sélection d'un nœud :

La construction d'un nœud doit réduire de façon optimale le désordre des objets. Pour mesurer ce désordre, on définit l'entropie d'une variable qualitative Y à q par :

$$H(Y) = -\sum_{k}^{q} P(Y=k)\times\log(P(Y=k)) \text{ avec la convention } 0\log(0)=0$$

H(Y) est un critère d'incertitude de la variable Y.
On peut ensuite définir l'entropie de Y conditionnée par une variable X ayant q' modalités.

$$H(Y|X) = -\sum_{k}^{\kappa} P(Y=k, X=k')\times\log(P(Y=k|X=k')) = \sum_{k'} P(X=k')H(Y|X=k')$$

Plus X nous renseigne sur Y plus l'entropie conditionnelle diminue. A l'extrême, H(Y|Y)=0. On choisira le nœud de façon à réduire au maximum le désordre.

1. 2. Réseaux de neurone :

Cette méthode repose sur la notion de neurone formel. Un neurone formel est un modèle caractérisé par des signaux d'entrée (les variables explicatives par exemple), une fonction d'activation f,

$f\left(\alpha_0+\sum_i \alpha_i\times x_i\right)$, f peut être linéaire, à seuil, stochastique et le plus souvent sigmoïde. Le calcul des paramètres se fait par l'apprentissage.

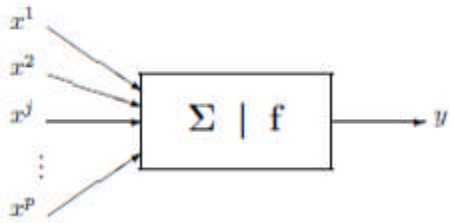

Figure 10 : schéma général du réseau de neurones.

Les neurones sont ensuite associés en couche. Une couche d'entrée lit les signaux entrant, un neurone par entrée xj, une couche en sortie fournit la réponse du système. Une ou plusieurs couches cachées participent au transfert. Un neurone d'une couche cachée est connecté en entrée à chacun des neurones de la couche précédente et en sortie à chaque neurone de la couche suivante. De façon usuelle et en régression (Y quantitative), la dernière couche est constituée d'un seul neurone

muni de la fonction d'activation identité tandis que les autres neurones (couche cachée) sont munis de la fonction sigmoïde.
En classification binaire, le neurone de sortie est muni également de la fonction sigmoïde tandis que dans le cas d'une discrimination à *m* classes (Y quantitative), ce sont *m* neurones avec fonction sigmoïde, un par classe, qui sont considérés en sortie.

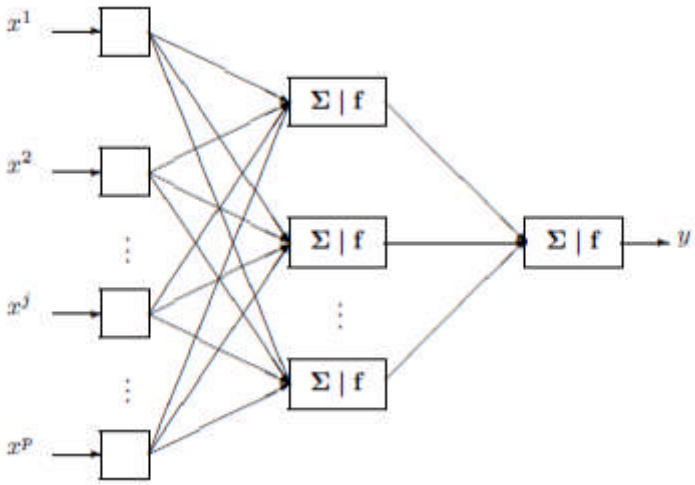

Figure 11 : Réseau de neurone.

- <u>Les modèles utilisés en réseaux de neurone :</u>

1- Structure classique

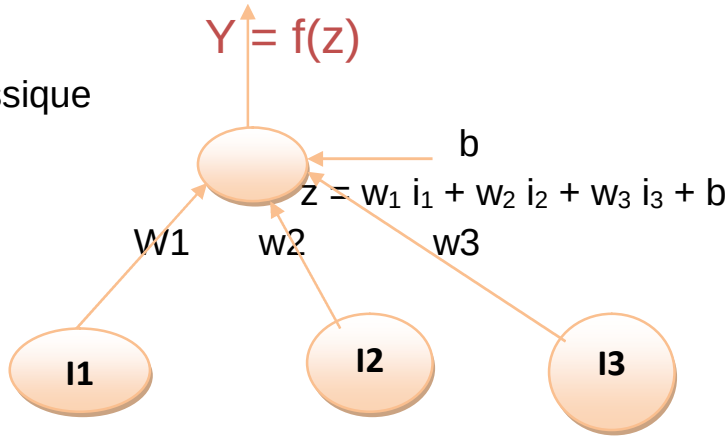

Figure 12 : la structure classique.

On distingue plusieurs entrées (w_i) dans un neurone. Et ces entrées seront multipliés par un point selon leur importance et seront sommés et on ajoute une constante b. Après on aura en sortie la fonction de transfert y = f(z).

2- (Multi Layer Perceptron)

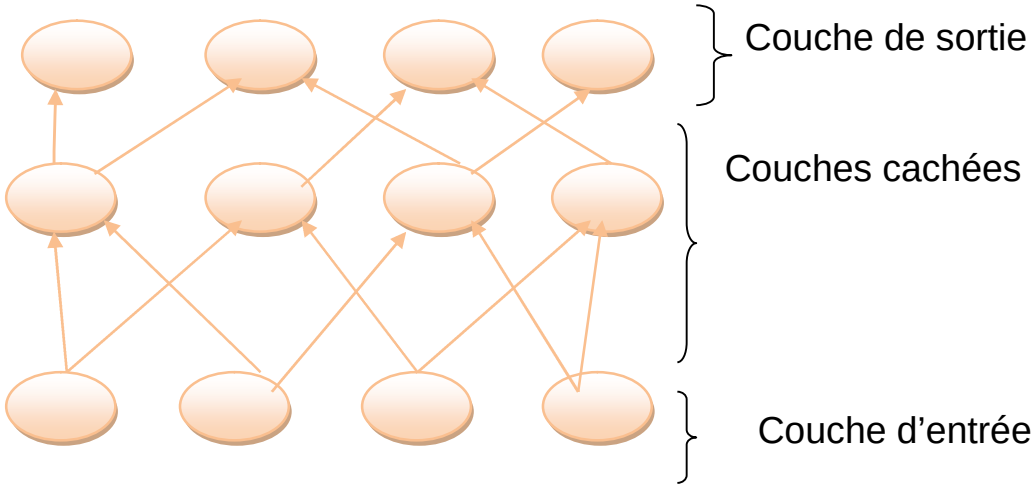

Figure 13 : MLP.

Paramètres de complexité :

Le modèle dépend de plusieurs paramètres

- **L'architecture du réseau :** nombre de couches cachées (une ou deux en général) et le nombre de neurones par couche.
- **Le nombre d'itération :** l'erreur maximale tolérée est un terme de régularisation (decay).

Les paramètres de réglage sont difficiles à définir correctement. On peut utiliser library (e1071) par exemple pour rechercher des valeurs optimales.

1. 3. Support Vector Machine :

Les entrées X sont transformées en un vecteur dans un espace de Hilbert F. Dans le cas d'un classement en 2 classes, on détermine un hyperplan dans cet espace F. La solution optimale repose sur la propriété que les objets sont les plus éloignés possibles de l'hyperplan, on maximise ainsi les marges.

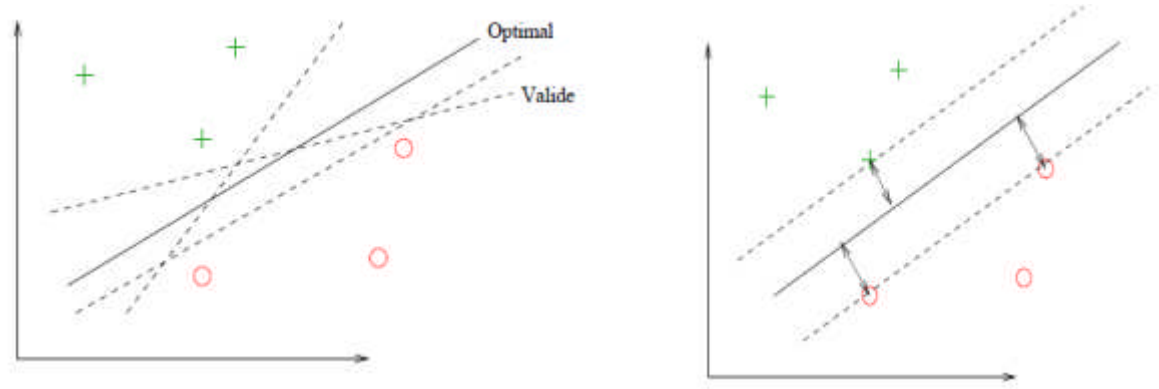

Figure 14 : Support Vector Machine.

Soit *x* le vecteur associé. On définit $f(x)=\omega x+\beta$ et l'hyperplan $\omega x+\beta=0$. La distance d'un point au plan est donnée par

$$d(x) = |\ \omega x + \beta\ | / \| \omega \|$$

Le classement est correcte si $yf(x)>0$, ou à un coefficient près $yf(x)\geq 1$.

- Ajustement :

Maximiser la marge revient à minimiser ||w|| ou $||w||^2/2$ sous les contraintes $y_i f(x_i) \geq 1$.
On utilise la méthode des multiplicateurs de Lagrange en ne conservant que les vecteurs x_i les plus proches de l'hyperplan (vecteurs de supports).
Lorsque tous les car ne sont pas séparables, on introduit un terme d'erreur :

$$y_i\ \ f(x_i)\ \ \geqslant\ 1-\ \xi_i$$

La transformation en vecteur ne fait intervenir que l'expression du produit scalaire dans F. on recherche en fait directement l'expression du produit scalaire à partir des coordonnées initiales à l'aide d'une fonction *k* appelée noyau. On distingue les noyaux linéaires, polynomiaux, gaussien …
S'adaptant aux différentes problématiques rencontrées.

1. 4. Régression logistique :

La régression logistique est l'une des méthodes d'analyse multi variables, qui est un ensemble de méthodes qui prennent en compte plusieurs variables explicatives. Un exemple de régression logistique est la présence de maladie cardiovasculaire en fonction de l'âge ou de catégorie chez l'homme

Nous étudions la présence chez l'homme de maladie cardiovasculaire en fonction de l'âge ou de catégorie d'âge AGRP.

On note Y (1 ou 0) la réponse et X la variable explicative. On pose $\pi_x = P(Y=1|X=x)$. π_x ne peut pas être décrit par une fonction linéaire de *x*, nous allons pour cela utiliser **une fonction de lien**, ici la **fonction logit** log (*y/1-y*) qui elle s'exprime linéairement en fonction de *x.* on a alors

$$\log \frac{\pi_x}{1-\pi_x} = \beta_0 + \beta_1 \ x$$

Et inversement avec sa fonction réciproque (sigmoïde).

$$\pi_x = \exp(\beta_0 + \beta_1 \ x)/[1+\beta_0 + \beta_1 \ x]$$

$\frac{\pi_x}{1-\pi_x}$ Définit un rapport de probabilité (odds). Si X est une variable binaire (Homme, Femme), ce rapport devient le rapport des probabilités d'avoir la maladie des hommes et femmes, soit le coefficient multiplicateur de la maladie si on est homme par rapport à une femme. Si odds est 2, on a deux fois plus de chance d'être malade en étant un homme.

En présence de plusieurs variables explicatives, on complexifie le modèle linéaire. On parle ici de **modèle linéaire généralisé GLMM.**

- Test

Il existe trois types de tests :

- **Test du maximum de vraisemblance ;**
  On définit la déviance du modèle étudié par :

Déviance(modèle) = -2 log (vraisemblance du modèle/vraisemblance du modèle saturé)

= Null déviance – Residual Deviance

Le modèle saturé est le modèle ou les fréquences y_i / n_i observées sont utilisées.

- **Test de Wald :**

  Le test de wald est un test paramétrique économétrique dont l'appellation vient du mathématicien hongrois Abraham Wald avec une grande variété d'utilisations. Chaque fois que nous avons une relation au sein des éléments de données qui peuvent être exprimées comme un modèle statistique avec des paramètres à estimer, et tout cela à partir d'un échantillon, le test de wald peut être utilisé pour « tester la vraie valeur du paramètre » basé sur l'estimation de l'échantillon.

- **Test du score :**

Le test de score est un test statistique d'une hypothèse nulle simple selon laquelle un paramètre θ est égal à une valeur particulière θ – {0}. C'est le test le plus puissant puisque la valeur réelle de θ est proche de 0.

1.5. Réponse binaire – courbe ROC :

On se place dans le cas d'une réponse binaire + / -. On obtient un résultat de prédiction de la forme

O\P	-	+	Total
-	TN	FP	N
+	FN	TP	P

Avec

- TP (true positives) : les prédits positifs qui le sont vraiment.
- FP (false positives) : les prédits positifs qui sont en fait négatifs.
- TN (true négatives) : les prédits négatifs qui le sont vraiment.
- FN (false négatives) : les prédits négatifs qui sont en fait positifs.
- P (positives) : tous les positifs quelque soit l'état de leur prédiction. P = TP + FN.
- N (negatives) : tous les négatifs quelque soit l'état de leur prédiction. N = TN + FP.

On définit alors la spécificité et la sensibilité :

- La **sensibilité** est : TP / (TP + FN) = TP / P.
- La **spécificité** est : TN / (TN + FP) = TN / N.

Principe de la courbe ROC :

Si le test donne un résultat numérique avec un seuil t tel que la prédiction est positive si x>t, et la prédiction est négative si x<t, alors au fur et à mesure que t augmente :

- La spécificité augmente.
- Mais la sensibilité diminue.

La courbe ROC représente l'évolution de la sensibilité (taux de vrais positifs) en fonction de la spécificité (taux de faux positifs) quand on fait varier le seuil t.

- C'est une courbe croissante entre le point (0, 0) et le point (1, 1) et en principe au-dessus de la première bissectrice.
- Une prédiction random donnerait la première bissectrice.
- Meilleure est la prédiction, plus la courbe est au-dessus de la première bissectrice.
- Une prédiction idéale est l'horizontale y = 1 sur]0, 1] et le point (0, 0).

- L'aide sous la courbe ROC (AUC, Area Under the Curve) donne un indicateur de la qualité de la prédiction (1 pour une prédiction idéale, 0.5 pour une prédiction random).

On arrive à un autre type de classification ; qui est le classification non supervisée.

2) La classification non supervisée :

La classification non supervisée désigne un corpus de méthodes ayant pour objectif de dresser ou de retrouver une typologie existante caractérisant un ensemble de n observations, à partir de p caractéristiques mesurées sur chacune des observations. Par typologie, on entend que les observations, bien que collectées lors d'une même expérience, ne sont pas toutes issues de la même population homogène, mais plutôt de K populations. Exemple :

- L'ensemble des clients d'une banque est une collection de n observations étant caractérisée par la nature des p transactions bancaires qu'elle réalise. Il existe certainement différents k « profils types » de clients. L'objectif est alors d'une part de retrouver ces profils types à partir de l'information sur les transactions bancaires, et d'autre part de déterminer, pour chaque observation, à quel profil type elle correspond.

2. 1. Les méthodes de partitionnement :

- **Mesure de similarité et de dissimilarité, distances :**

Afin de définir l'homogénéité d'un groupe d'observations, il est nécessaire de mesurer une ressemblance entre deux observations. On introduit aussi les notions de di similarité et de similarité :

- **Définition 1 :** une di similarité est une fonction d qui à tout couple (x_1, x_2) associe une valeur dans $\mathbb{R}+$, et telle que
  - $\star\ d(x_1, x_2) = d(x_2, x_1) \geq 0,$
  - $\star\ d(x_1, x_2) = 0 \Rightarrow x_1 = x_2.$

Autrement dit, moins les unités x_1 et x_2 se ressemblent, plus le *score est élevé*. Remarquons qu'une distance est une di similarité, puisque toute distance possède les deux propriétés précédentes ainsi que l'inégalité triangulaire. Toutes les distances connues, en particulier la distance euclidienne, sont donc des exemples de di similarité.

A l'inverse, une autre possibilité consiste à mesurer la ressemblance entre observations à l'aide d'une similarité.

- **Définition 2 :**

Une similarité est une fonction s qui à tout couple (x_1, x_2) associe une valeur dans $\mathbb{R}^+$, et telle que

$\star\ s(x_1, x_2) = s(x_2, x_1) \geq 0,$

$\star\ s(x_1, x_1) \geq s(x_1, x_2).$

Contrairement à la di similarité, plus les unités x_1 et x_2 se ressemblent plus le score est élevé. On peut citer comme exemple de similarité la valeur absolue en coefficient de corrélation :

$$|\rho(x_1, x_2)| = \left| \frac{\sum_{j=1}^{p}(x_{1j} - x_{1\bullet})(x_{2j} - x_{2\bullet})}{\sqrt{\sum_{j=1}^{p}(x_{1j} - x_{1\bullet})^2 \sum_{j=1}^{p}(x_{2j} - x_{2\bullet})^2}} \right|$$

2. 2. Formalisation du problème :

L'objectif de la classification non supervisée étant de déterminer des groupes- on désignera ces groupes dans la suite par « classes »- homogènes et distincts, il est nécessaire de formaliser ces deux notions du point de vue géométrique. Pour cela, nous repartons de la définition de l'inertie d'un nuage de points.

L'inertie totale est définie de la manière suivante :

$Inertie_{totale} = 1/N \sum_{I=1,n} d_{ig}^2$: la distance moyenne d'un spectre au centre de gravité du nuage de points.

K groupes G_1 … G_k d'effectifs N_1 … N_k de centres de gravité g_1 … g_k

Centre de gravité du nuage g

$Inertie_{inter} = 1/N \sum_{K=1,k} N_k\ d^2_{gkg2}$: la distance moyenne des centre de gravité des Classes au centre de gravité de nuage de points.

$Inertie_{intra} = 1/N \sum\sum_{K=1,k} d^2_{igk}$: c'est la distance moyenne d'un individu au centre de gravité de sa propre classe.

Alors inertie totale est :

$Inertie_{totale}$ = inertie inter + inertie intra.

2 .3. Métode des K-means :

On suppose qu'il existe K classes distinctes. On commence par désigner K centres de classes $\mu 1 \ldots \mu k$ parmi les individus. Ces centre peuvent être soit choisis par l'utilisateur pour leur « représentativité », soit désignés aléatoirement. On réalise ensuite itérativement les deux étapes suivantes :

- Pour chaque individu qui n'est pas un centre de classe, on regarde qui est le centre de classe le plus proche. On définit ainsi K classes C_1 …. C_k, ou
  C_i = {ensemble des points les plus proches du centre μ_i}.
- Dans chaque nouvelle classe C_i, on définit le nouveau centre de classe μ_i comme étant le barycentre des points de C_i.

L'algorithme s'arrête suivant un critère d'arrêt fixé par l'utilisateur qui peut être choisi parmi les suivants : soit le nombre limite d'itération est atteint, soit l'algorithme a convergé, c'est-à-dire qu'entre deux itérations les classes formées restent les mêmes, soit l'algorithme a « «presque » convergé, c'est-à-dire que l'inertie intra-classe ne s'améliore quasiment plus entre deux itérations.

2 .4. Classification ascendante hiérarchique :

La classification ascendante hiérarchique, notée CAH, a pour objectif de construire une suite de partitions emboitées des données en n classes, n-1 classes, …, 1 classe. Ces méthodes peuvent être vues comme la traduction algorithmique de l'adage « qui se ressemble s'assemble ».

- A l'étape initiale, les n individus constituent des classes à eux seuls.
- On calcule les distances deux à deux entre individus, et les deux individus les plus proches sont réunis en une classe.
- La distance entre cette nouvelle classe et les n – 2 individus restants sont ensuite calculée, et à nouveau les deux éléments (classes ou individus) les plus proches sont réunis.

Ce processus est réitéré jusqu'à ce qu'il ne reste plus qu'une unique classe constituée de tous les individus. On constate que la nouveauté ici vient de la nécessité de définir deux distances : la distance usuelle entre deux individus, et une distance entre classes. Le choix d'une distance entre individus ayant déjà été discuté, il reste à choisir une distance entre classes, en gardant à l'esprit que l'objectif est de trouver la partition en K classes des observations dont l'inertie intra-classe est minimale.

IV. La différence entre classification supervisée et non supervisée :

La différence entre une classification supervisée et classification non supervisée. En classification non supervisée, l'appartenance des observations à l'une des K populations n'est pas connue. C'est justement appartenance qu'il s'agit de trouver à partir des p descripteurs disponibles. En classification supervisée au contraire, l'appartenance des n observations aux différentes populations est connue, et l'objectif est de construire une règle de classement pour prédire la population d'appartenance de nouvelles observations.

Conclusion :

On a vu pour l'instant, les différents types de classification, avec différents modèles associés pour chaque type. En ce qui suit, on verra un chapitre sur le wordNet.

CHAPITRE 3

WordNet

Introduction :

Wordnet est une ressource lexicale de large couverture, développée depuis plus de vingt ans pour la langue anglaise. Elle est utilisable librement, ce qui en a favorisé une diffusion très large. Plusieurs autres ressources linguistiques ont été constituées à partir de wordnet. Des programmes issus du monde de l'Intelligence Artificielle ont également établi des passerelles avec wordnet.

L'ensemble constitue un « écosystème » complet couvrant des aspects lexicaux, syntaxiques et sémantiques. Ces ressources fournissent un point de départ intéressant pour des développements sémantique en TAL ou dans le cadre du web sémantique, tel que la recherche d'information, l'inférence pour la compréhension automatique de texte.

Alors WordNet en fait c'est quoi ?

I. Définition de WordNet :

C'est une base de données lexicale développée depuis 1985 par des linguistiques 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 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.

WordNet utilise la notion de synset. [François-Régis Chaumartin]

Exemple d'explication d'un mot « dog » en wordnet :

The noun dog has 7 senses (first 1 from tagged texts)

1. (42) **dog**, domestic dog, Canis familiaris -- (a member of the genus Canis (probably descended from the common wolf) that has been domesticated by man since prehistoric times; occurs in many breeds; "the dog barked all night")
2. frump, **dog** -- (a dull unattractive unpleasant girl or woman; "she got a reputation as a frump"; "she's a real dog")
3. **dog** -- (informal term for a man; "you lucky dog")
4. cad, bounder, blackguard, **dog**, hound, heel -- (someone who is morally reprehensible; "you dirty dog")
5. frank, frankfurter, hotdog, hot dog, **dog**, wiener, wienerwurst, weenie -- (a smooth-textured sausage of minced beef or pork usually smoked; often served on a bread roll)
6. pawl, detent, click, **dog** -- (a hinged catch that fits into a notch of a ratchet to move a wheel forward or prevent it from moving backward)
7. andiron, firedog, **dog**, dog-iron -- (metal supports for logs in a fireplace; "the andirons were too hot to touch")

The verb dog has 1 sense (first 1 from tagged texts)

1. (2) chase, chase after, trail, tail, tag, give chase, **dog**, go after, track -- (go after with the intent to catch; "The policeman chased the mugger down the alley"; "the dog chased the rabbit")

II. Notion de synset :

Le synset (ensembles 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 contient avec les sens voisins.

Les noms et les verbes sont organisés en hiérarchies. Des relations d'hyperonymie (« est un ») et d'hyponymie reliant les « ancêtres » des noms et des verbes avec leurs « spécialisations ». [François-Régis Chaumartin]

III. Les relations : [F Gayral]

- Vocabulaire = ensemble de mots.

- Mot = (f, s)

  f : forme

  s : sens

- Catégories syntaxiques : Noms, Adj, Verbes, etc.

- Morphologie : c'est la relation entre les formes de mots. En anglais : inflection, derivation, composition.

- Sémantique lexical : c'est la relation entre les sens des mots, qui détermine la définition d'un mot.

IV. Exemples de relations :

1. La polysémie:

C'est un mot qui a plusieurs sens. Ici le mot 'table' qui est un nom féminin peut avoir le sens d'un meuble ou bien le sens d'un ensemble de tuples.

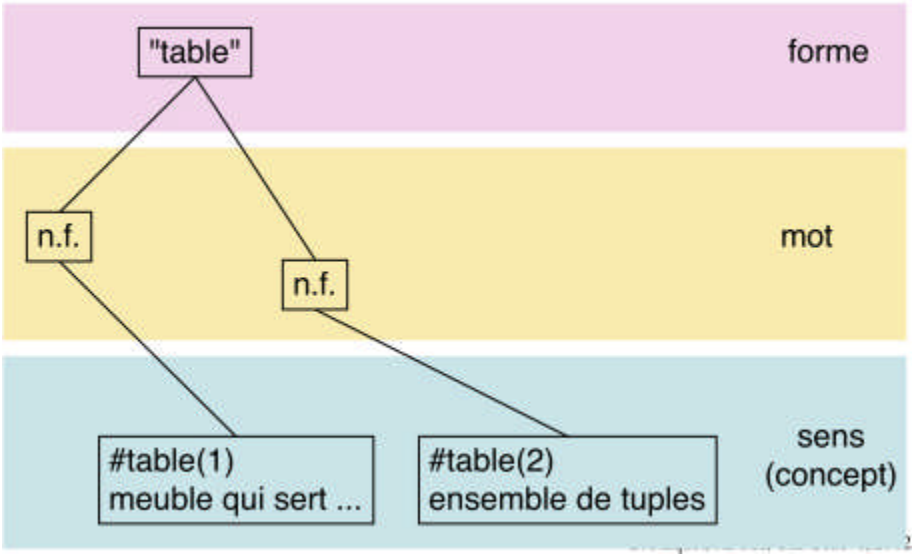

Figure 15 : La polysémie.

2. Synset :

Qui veut dire synonymie ; deux mots différents qui portent le même sens. Ici le mot ‘table’ qui veut dire ‘ensemble de tuples’ est un synset du mot ‘relation’.

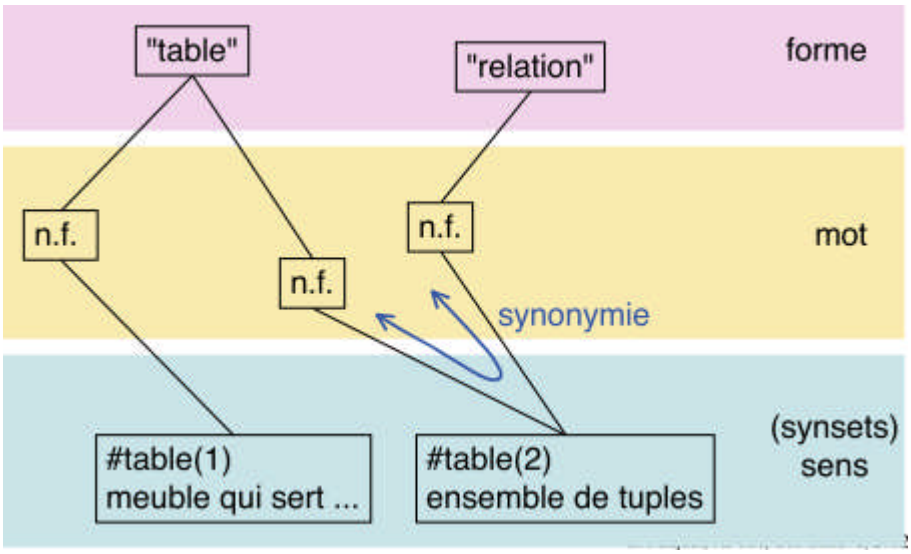

Figure 16 : Le synset.

3. Relations morphologiques :

La relation entre les formes d'un mot. Ici le verbe 'to go' son passé est 'went'. C'est une morphologie.

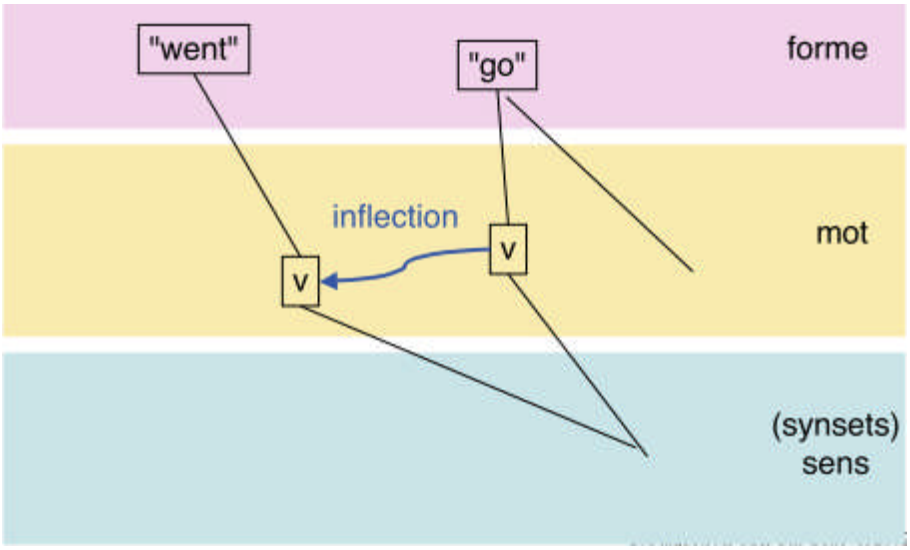

Figure 17 : Les relations morphologiques.

4. Relation sémantique hyperonymie :

Il s'agit de donner la définition du mot, en utilisant l'expression « est un ». Dans notre exemple « la table est un meuble ».

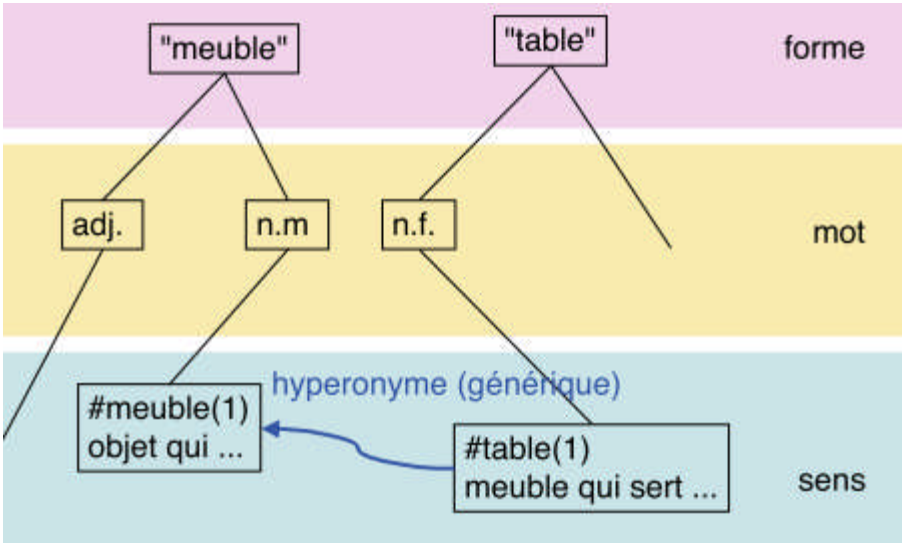

Figure 18 : Relation sémantique hyperonymie.

5. Méronymie (holonymie) :

C'est le fait de donner les composants de l'objet qu'on est en train de définir. Dans notre exemple « la table se compose de pieds ».

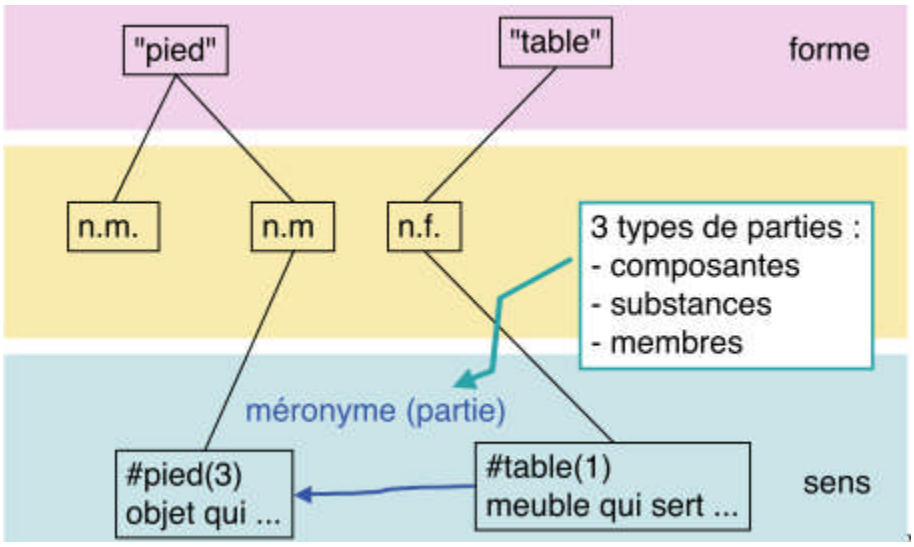

Figure 19 : Méronymie.

6. L'antonymie :

Un mot est un antonyme d'un autre mot, s'il désigne son sens contraire. Exemple : petit ≠ grand. Dans notre exemple le mot « mou » est le contraire du mot « dur », qui sont tous les deux des adjectifs.

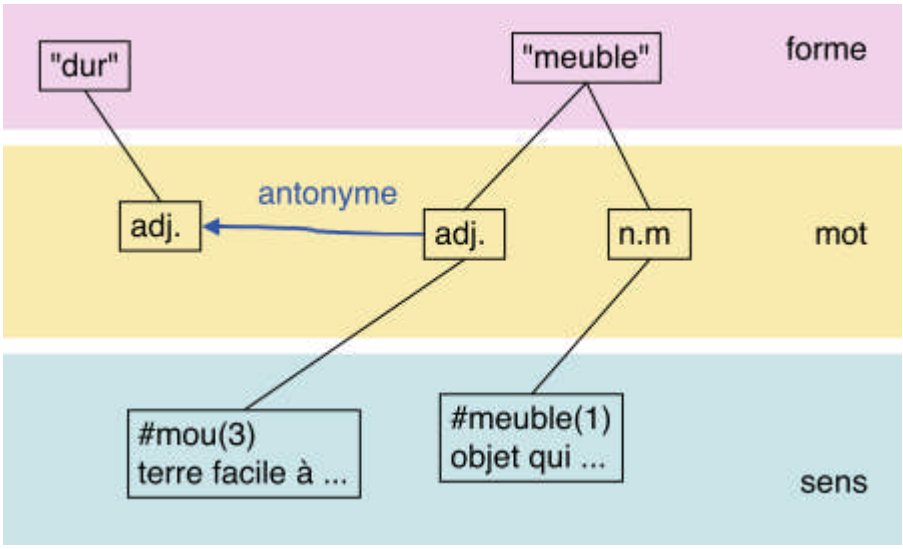

Figure 20 : L'antonymie.

7. Troponymie:

Dans notre exemple il ya une troponymie entre « «manger » et « déguster ». Tel que déguster est de manger de façon à gouter ce que l'on mange. Donc le sens de manger est inclus à l'intérieur.

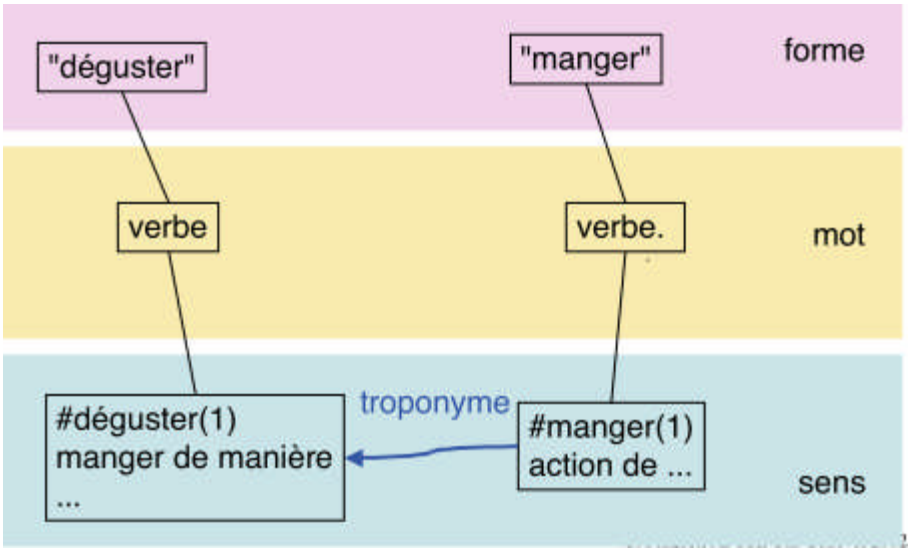

Figure 21 : Troponymie.

8. Implication (entailment) :

C'est-à-dire. Si on a un sens, ce sens automatiquement implique un autre sens. Dans notre exemple : si on a le verbe voler. Automatiquement on déduit le verbe « décoller » qui veut dire le départ du vol. Ce qui fait le verbe « voler » implique le verbe « décoller ».

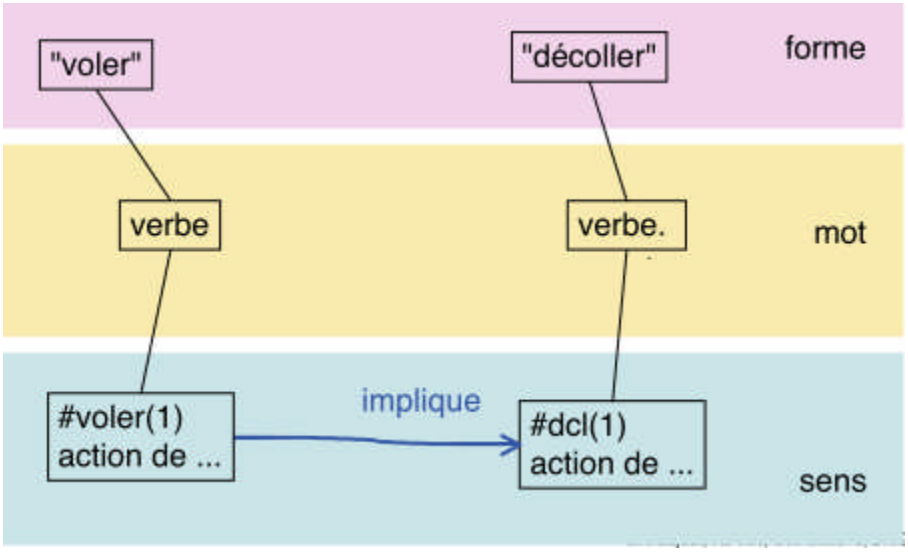

Figure 22 : L'implication.

On distingue aussi dans le wordnet ce qu'on appelle l'ontologie. Alors que veut dire l'ontologie.

V. L'ontologie :

A l'instar d'un dictionnaire traditionnel, wordNet offre ainsi, pour chaque mot, une liste de synsets correspondant à toutes ses acceptions répertoriées. Mais les synsets ont également d'autres usages : ils peuvent représenter des concepts plus abstraits, de plus haut niveau que les mots et leurs sens, qu'on peut organiser sous forme d'ontologies.

Une ontologie est un système de catégories permettant de classifier les éléments d'un univers. Le système de catégorisation correspond aux relations sémantiques vues précédemment. Ceci permet de regrouper de manière cohérente toutes les composantes d'un univers linguistique telles que les mots, les sens ou bien les concepts. Exemple :

Car, auto, automobile, machine, motorcar

- Motor vehicle, automotive vehicle
  - Vehicle
    - Conveyance, transport
      - Instrumentality, instrumentation

- Artifact, artefact
  - Object, physical object
    - Entity, something

Dans cet exemple, le dernier concept, « entité quelque chose », est le plus général, le plus abstrait. Il pourrait ainsi être le super-concept d'une multitude de concepts plus spécialisés. [wikipédia]

VI. Les limites de wordNet :

- **Informations manquantes :**

wordNet ne précise pas l'étymologie, la prononciation, les formes de verbes irréguliers et ne contient que des informations limitées sur l'usage des mots.

- **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 tache de désambigüisation lexicale.

- **Absence de relations pragmatiques :**

WordNet ne matérialise pas de 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 trouve formalisée dans aucune relation. De même, des relations pragmatiques telles que savon/bain (soap/bath) sont absentes de WordNet.

VII. Fréquence des lemms :

WordNet donne une fréquence d'apparition pour chaque lemme définissant un synset. Ce nombre indique combien de fois un mot apparait dans un sens spécifique. Pour un nom ou un verbe, la somme cumulée des fréquences d'un synset et de ses hyponymes au sein d'un sous-arbre de la hiérarchie permet de calculer son Contenu Informationnel.

VIII. 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. Elle permet de quantifier la similarité de deux concepts. Elle peut également servir dans le cadre de désambigüisation lexicale.

IX. verbNet :

VerbeNet est un lexique des classes de verbes anglais. C'est un projet mené sous l'impulsion de Martha Palmer (d'abord à l'université de Pennsylvanie, puis à Boulder au Colorado). VerbNet regroupe par classe les verbes partageant les mêmes comportements syntaxiques et sémantiques. C'est un prolongement des travaux de (Levin, 1993). (Chaumartin, 2006) décrit comment mettre en œuvre WordNet et VerbNet pour implémenter une interface syntaxe-sémantique.

X. Structure d'une description de classe de verbes :

Chaque fichier de VerbNet décrivant une classe de verbes est représenté en XML, et découpé en sections balisées selon une structure arborescente :

- <MEMBERS> : décrit les verbes membres qui appartiennent à la classe, en précisant l'identifiant vers le(s) synset(s) correspondant(s) de wordNet,
- <THEMROLES> : indique des rôles thématiques de la classe :
  - <SELRESTRS> : précise leurs éventuelles contraintes de sélections.
- <FRAMES> : indique chacune des constructions typiques, en donnant à chaque fois :
  - <SYNTAX> : sa syntaxe
  - <SEMANTICS> : sa sémantique
  - <EXAMPLES> : un ou plusieurs exemples
- <SUBCLASSES> : regroupe éventuellement en sous-classes :
  - <VNSUBCLASS> : les cas particuliers d'une classe de verbes.

X.1. Un exemple : la classe de verbe « murder » :

Par exemple, le fichier murder.xml décrit trois constructions typiques :

- Agent élimine patient.
- Agent élimine patient avec instruments.
- Instruments éliminent patient.

Chaque description de classe de verbes déclare des contraintes de sélection sur les rôles thématiques. Par exemple, pour « murder », l'agent et le patient doivent avoir un trait animé (Humain ou Organisation) et l'instrument doit être concret.

X.2. Description de la syntaxe :

Le deuxième frame de la classe de verbes « murder » décrit :

- <SYNTAX>
  <NP value= « agent »/>
  <VERB/>
  <NP value= « patient »/>
  <PREP value= « with »/>
  <NP value= « instrument »/>
  <SYNTAX/>

- <EXAMPLES>
  <EXAMPLE> « Brutus killed Caesar with a knife »</EXAMPLE>.

X.3. Description de la sémantique:

Par exemple, pour « murder » :

- Au démarrage de l'évènement, Patient est vivant : alive(start(E), Patient).
- A la fin de l'évènement, Patient n'est plus vivant : ! alive (result(E), Patient).

XI. Prise en compte de l'héritage entre classes :

La balise <SUBCLASSES> déclare les éventuelles sous-classes qui spécialisent une classe de verbe donnée. Une sous classe permet :

- De raffiner les contraintes de sélection portant sur les rôles thématiques.
- De déclarer de nouveaux rôles thématiques.
- D'associer des verbes à la sous classe.
- De créer de nouveaux frames.

XII. FrameNet :

FrameNet, projet mené à Berkeley à l'initiative de Charles Fillmore, est fondé sur la sémantique des cadres (« frame semantics »). FrameNet a pour objectif de documenter la combinatoire syntaxique et sémantique pour chacun des sens d'une entrée lexicale à travers une annotation manuelle d'exemples choisis dans des corpus sur des critères de représentativité lexicographique. Les annotations sont ensuite synthétisées dans des tables, qui résument pour chaque mot les cadres avec leurs actants sémantiques et arguments syntaxiques.

XIII. Exemple de FrameNet : description du cadre « Crime_scenario » :

A crime is committed and comes to the attention of the Authorities. In response, there is a criminal investigation (often) Arrest and criminal court proceedings. The investigation, Arrest, and other parts of the Criminal Process are pursued in order to find a Suspect (who then may enter the Criminal process to become the Defendant) and determine if this Suspect matches the perpetrator of the Crime, and also to determine if the Charges match the Crime. If the Suspect is deemed to have committed the Crime. Then they are generally given some punishment commensurate with the Charges.

Frame Elements:

Authorities [] The group which is responsible for the maintenance of law and order, and such have been given the power to investigate Crime. Find Suspects and determine if a Suspect should be submitted to the Criminal process.

Charge[] A description of a type of act that is not permissible according to the law of society.

Crime[] An act, generally intentional, that matches the description that belongs to an official Charge.

perpetrator The individual that commits a Crime.

Suspect The individual which is under suspicion of having committed the Crime.

XIV. Lien entre Wikipédia et WordNet:

Wikipédia est une encyclopédie libre et multilingue écrite de façon collaborative sur Internet avec la technologie wiki. Plusieurs projets visent à établir automatiquement des liens entre la Wikipédia et WordNet.

Un algorithme rapide permet de réaliser la correspondance entre wordnet et wikipédia. Si aucun synset n'a de lemme en commun avec le titre de l'article, ce

dernier est ignoré. Si un seul synset de WordNet a un lemme égal au titre, l'article y est lié sans analyse. Le système analyse les définitions de WordNet, et construit pour chacune d'entre elles un vecteur booléen (contenant « 1 » pour chaque terme en commun avec l'article et « 0 » pour chaque mot en disjonction). L'algorithme calcule alors une mesure de type cosinus entre les vecteurs, et retient le meilleur article. Les auteurs revendiquent une précision de 91,11% de similarité.

Conclusion :

Nous avons présenté WordNet, ainsi que plusieurs autres ressources de nature lexicale, syntaxique et sémantique, qui s'y rattachent. Le fait de mettre en commun plusieurs ressources de large couverture permet d'espérer des progrès dans les applications de TAL. Dans le chapitre qui suit, on parlera de la conception et de la réalisation de notre projet.

CHAPITRE 4

conception et réalisation

Introduction :

Le temps joue un rôle important dans toute espèce d'information, et peut être très utile dans le traitement du langage naturel, et tache de récupération d'informations, telles que l'ordre naturel des évènements, l'exploitation des documents, la compréhension de requêtes temporelles ……. Etc. Le temps est aussi important dans le processus du langage naturel (NLP) et en recherche d'information aussi (IR).

Dans ce chapitre, on va essayer d'implémenter notre classifieur qui permet de faire la classification temporelle du texte. Et essayer de réaliser une application similaire à TempoWordNet. Avant cela, on essaiera de voir quelques notions ; tel que TempoWordNet, python, …….etc. et par la suite essayer de représenter notre petite application.

I. Les objectifs et défis :

Dans ce mémoire, nous étudions comment faire une classification temporelle du texte en utilisant une ontologie (WordNet). Pour identifier les expressions temporelles, c'est-à-dire des expressions qui sont marqueurs du passé du présent ou de l'avenir dans le texte de la langue naturelle.

En outre, nous étudions comment utiliser WordNet pour certains taches telles que la classification temporelle de la phrase.

Cette tache est difficile, et il existe plusieurs explications pour cette difficulté. Les informations temporelles qui dénotent le passé le présent et l'avenir dans le texte linguistique naturel peuvent être transmises par une large gamme. Ces propriétés doivent être correctement identifiées interprétées et combinées pour dériver les informations temporelles appropriées. Un autre défi est le fait que l'information temporelle n'est pas toujours exprimée explicitement mais plutôt implicitement et nécessite une inférence dérivée de la connaissance du monde. Exemples :

- John est tombé. Mary le poussa.
- John est tombé. Mary a demandé de l'aide.

Ces deux exemples ont des similarités syntaxiques mais les évènements qu'ils décrivent ne sont pas dans le même ordre temporel.

L'information temporelle dans ces exemples est implicite, car les évènements décrits ne sont ni ancrés à des points précis dans le temps, ni spécifiquement commandés en ce qui concerne l'évènement voisin.

Pour dériver l'interprétation temporelle correcte pour ces exemples, il faut compter sur le contenu sémantique,

Bien que leur structure et leur syntaxe soit si similaire. Dans le premier exemple l'évènement de chute temporaire après l'évènement de la poussée. Tandis que

dans le second exemple, l'évènement de chute précède l'évènement demandant de chute.

II. WordNet temporel :

II.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 synset 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.

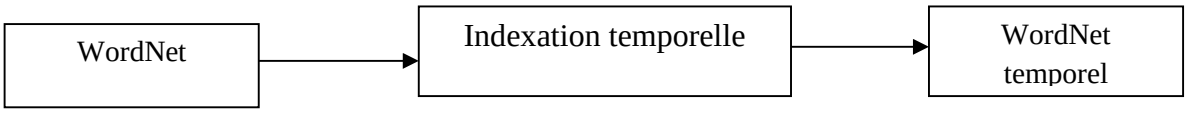

Figure 23 : schéma descriptif de notre approche.

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

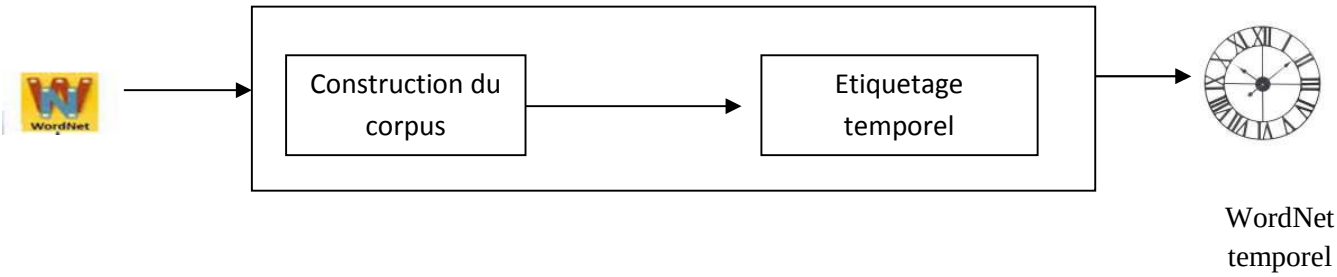

Figure 24 : Processus d'indexation temporelle.

II.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 les jeux 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 25 : Matrice de confusion.

II.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.

II. 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 présent 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 langue 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 assurer la présence 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).

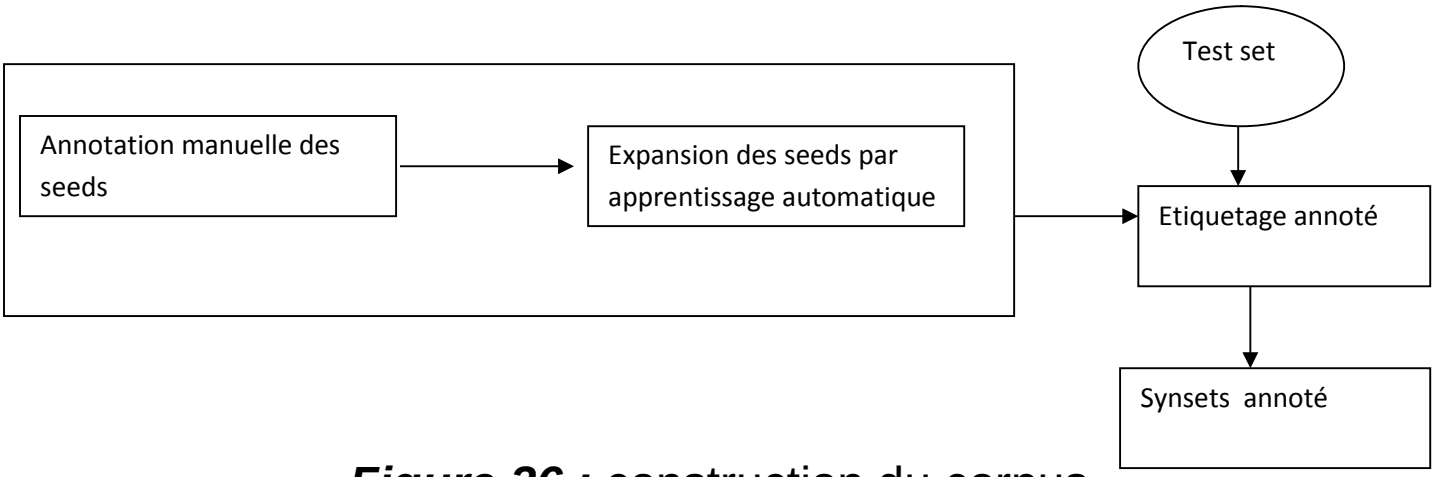

Figure 26 : construction du corpus.

```
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']
```

Figure 27 : liste des seeds extraits de notre programme.

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

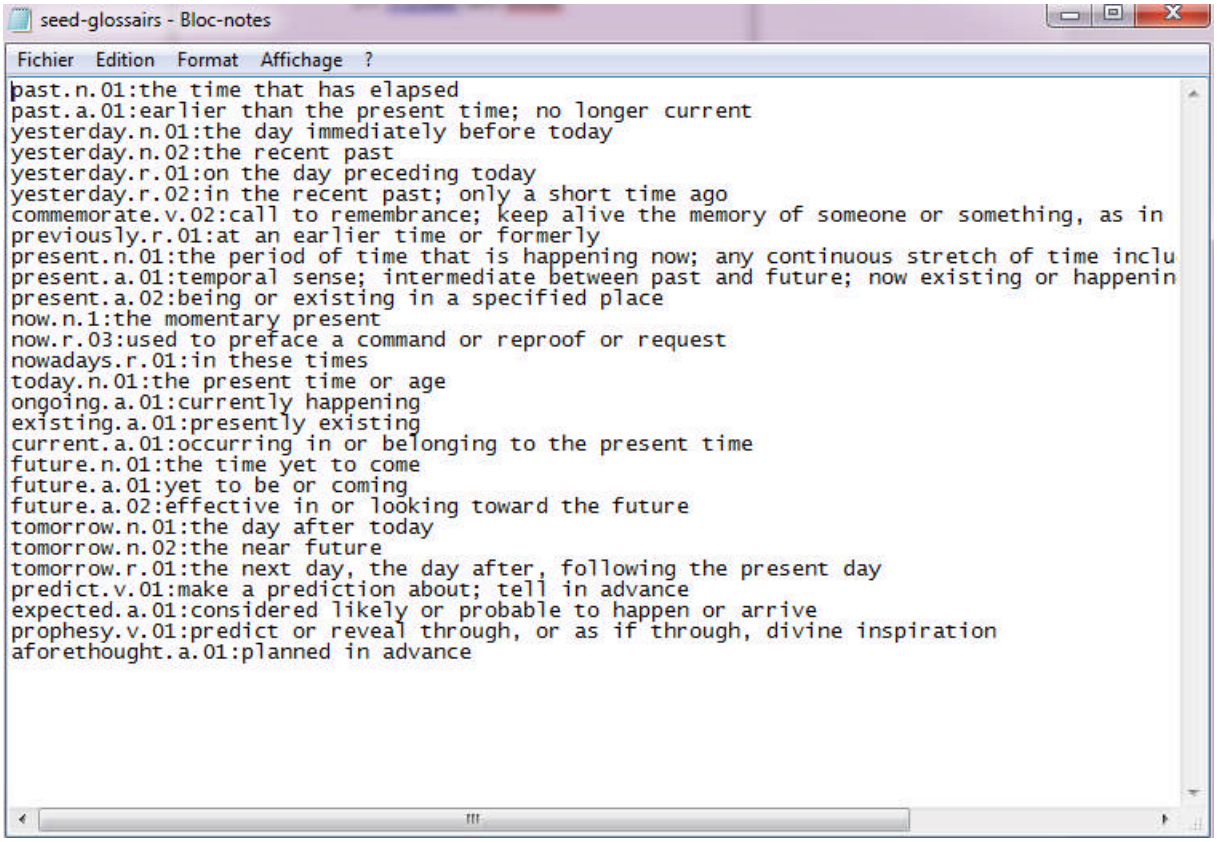

Figure 28 : fichier « seed-glossaire.txt ».

En suite, nous avons filtré l'ensemble des glossaires en enlevant les stopsword (mots d'arrêt : 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 29 : 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 30 : Table des synset 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 31 : Table des synsets 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 32 : Table des synsets 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 avec les 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 probabilité 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 probabilité des mots dans chaque classe, on utilise la formule s uivante :

P (attribut/classe)=(nk+1)/n+|vocabulaire|

Sachant : **nk**: nombre d'occurrence de la caractéristique (features qui seront utilisés dans le reste de notre étude), 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 33 : probabilités conditionnelles des features.

II.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 précédent :

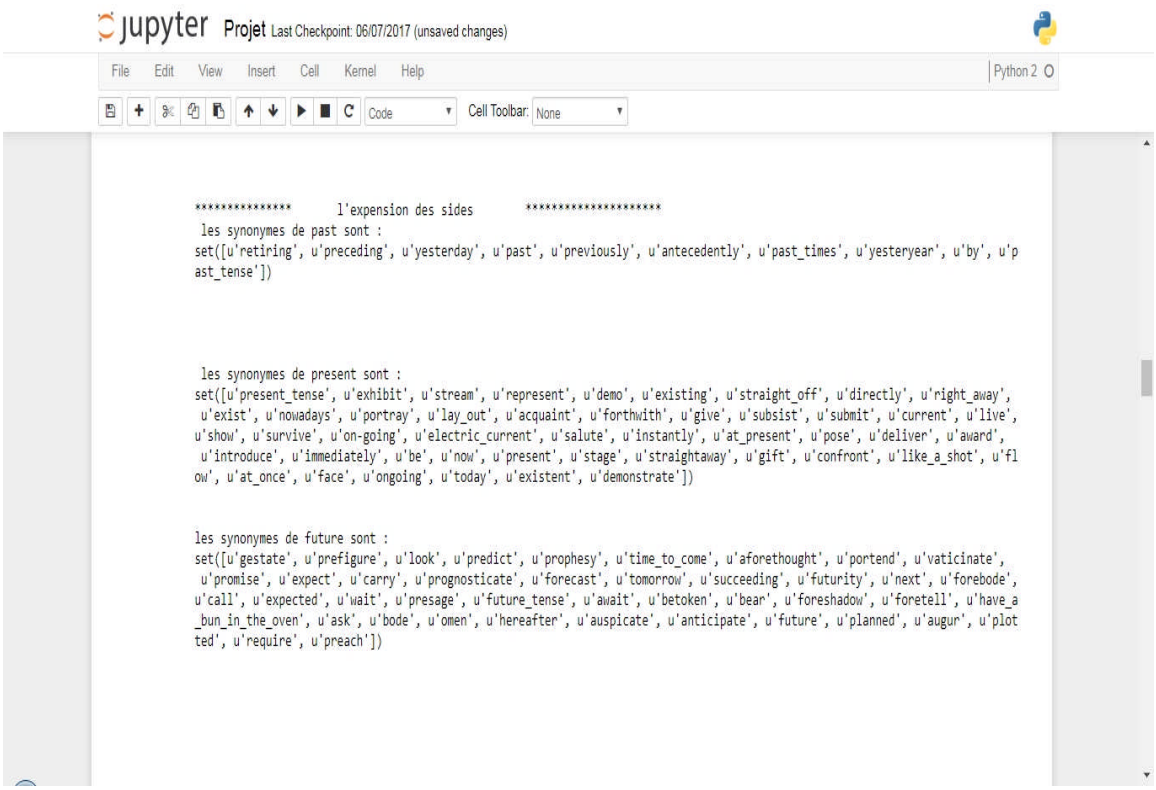

Figure 34 : Résultat d'expansion en appliquant la « synonymie ».

Le problème qui se pose : il est souvent coûteux d'obtenir les labels (coté financié ou coté temporel), alors qu'en général il est peu coûteux d'obtenir les données sans label qui stipule de propager les labels de l'ensemble des seeds vers l'ensemble non étiqueté.

Exemples : – appétence (oui / non) pour une offre commerciale en fonction du profil client => nécessite une campagne sur des centaines ou milliers de clients – réaction à un nouveau médicament (OK /pas OK) en fonction de paramètres du diagnostic => nécessite des dizaines de tests sur des volontaires ❯ Par contre : il est souvent peu coûteux d'obtenir les données descriptives (sans le "résultat", le label) par apprentissage automatique.

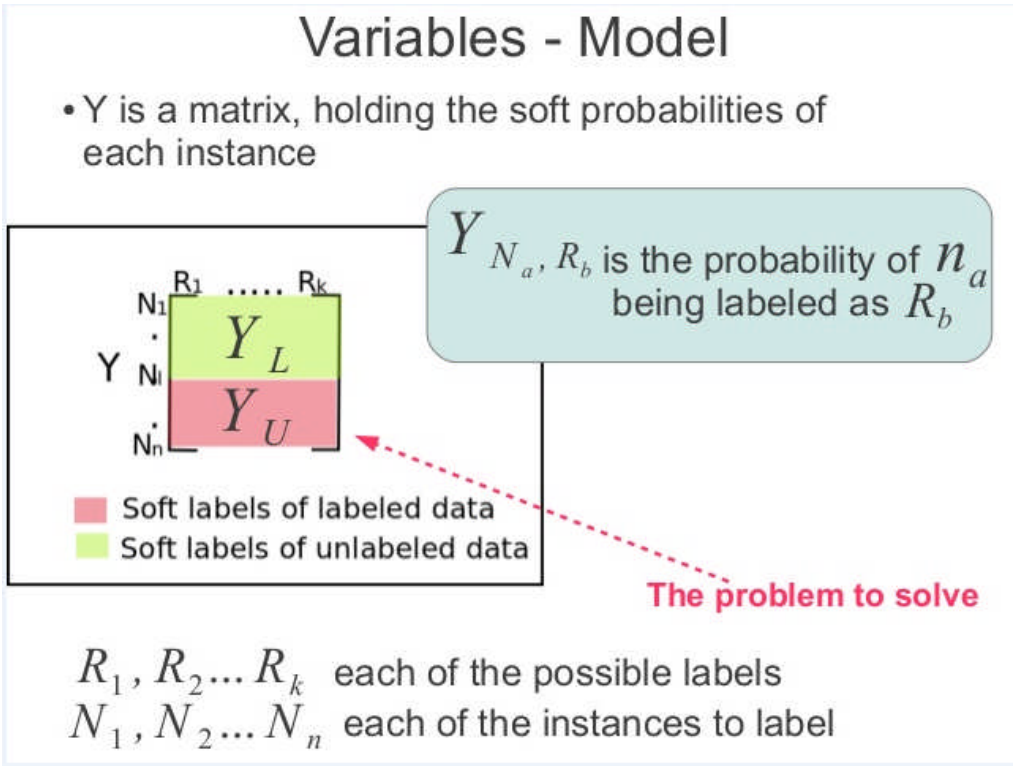

Figure 35 : Une matrice Y avec des lignes unlabeled.

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 :

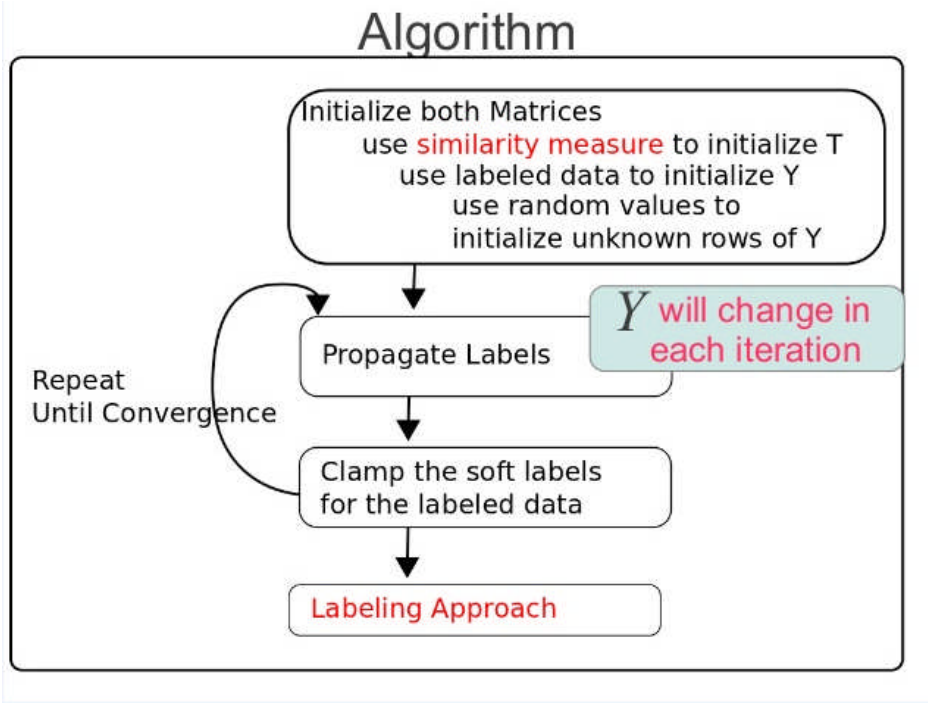

Figure 36 : Notre approche en inclure LabelPropagation.

III. <u>Implémentation de notre approche :</u> Nous allons décrire maintenant l'environnement de programmation de notre approche.

<u>III.1.Environnement et outils d'implémentation :</u>

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.

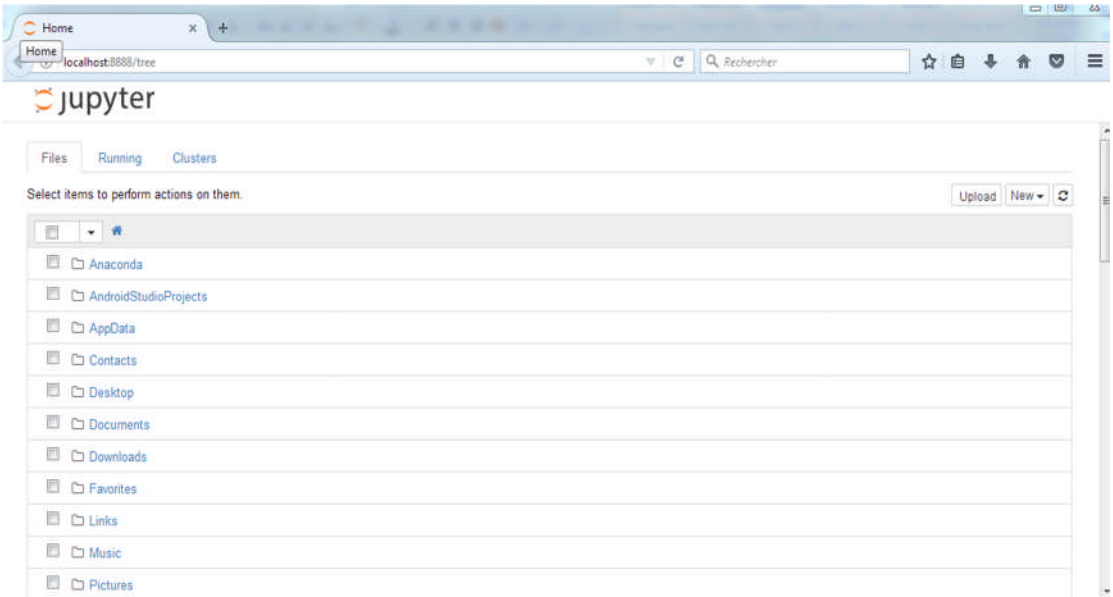

Figure 37 : représentation de l'interface d'environnement de travail.

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

En informatique, une fois avoir trouvé l'algorithme, il faut l'implémenter. Pour cela, il est nécessaire d'utiliser un langage de programmation. Certainement, python est un langage qui nous permet d'implémenter des algorithmes d'apprentissage.

Python est un langage interprété, c'est-à-dire chaque ligne de code est lue puis interprétée afin d'être exécutée par l'ordinateur.

Son origine est le langage de script du système d'exploitation Amoeba. Il a été développé par Guido Von Rossum au CWI, à l'université d'Amsterdam et nommé par rapport au Monthy Python's Flying Circus.

Depuis python est devenu un langage de programmation généraliste. Il offre un environnement complet de développement comprenant un interpréteur performant et de nombreux modules. Un atout indéniable est sa disponibilité sur la grande majorité des plates formes courantes (BeOS, MAC OS X, Unix, Windows).

Python est un langage open source supporté, développé et utilisé par une large communauté : 300.000 utilisateurs et plus de 500.000 téléchargements par ans.

- ✓ **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).

- ✓ ***Extensible Markup Language*** (**XML**) « langage de balisage extensible» en français) est un métalangage informatique de balisage générique qui dérive du SGML. Cette syntaxe est dite « extensible » car elle permet de définir différents espaces de noms, c'est-à-dire des langages avec chacun leur vocabulaire et leur grammaire, comme XHTML, XSLT, RSS, SVG… Elle est reconnaissable par son usage des chevrons (<, >) encadrant les balises. L'objectif initial est de faciliter l'échange automatisé de contenus complexes (arbres, texte riche…) entre systèmes d'informations hétérogènes (interopérabilité). Avec ses outils et langages associés, une application XML respecte généralement certains principes :

- la structure d'un document XML est définie et validée par un schéma.
- un document XML est entièrement transformable dans un autre document XML.

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

Notre projet se situe dans le domaine de traitement automatique des langues, il porte sur l'implémentation et l'évaluation d'une annotation 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 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 base de connaissance lexicale wordNet ainsi que ces composants. Enfin nous avons défini et suivi, l'annotation temporelle de wordNet, le langage de programmation 'python' 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 de la base de connaissance wordNet.
- Découvrir le langage de programmation python.
- Mettre l'accent sur la manière dont le système fonctionne.

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

Bibliographie

- Abeille A, 1993.
- Allen J, 1998.
- Dutoit R., Bourlard H. et al , 2000
- Pierrel J.-M, 2000.
- Mitkov R, 2002.
- Delsarte Philippe, Thayse André, 2001.
- Habert B. et alii, 1997.
- C. Biernacki, G.Celeux, and G.Govaert, 2000.
- G.Celeux and D. Diebolt, 1985.
- G. Celeux and G, 1992.
- P Dempster, N.M. Laird, and D. B. Rubin, 1999-2000.
- Philippe Michel, 1999-2000.
- D. Reinhardt, 2005.
- Y.C. Yao, 1988.
- S. Tufféry, 2007.

Webliographie

www.tsi.enst.fr/-compedel/

http://www.math.univ-toulouse.fr/~besse/Wikistat/

http://cedric.cnam.fr/~saporta/DM.pdf/

https://fr.wikipedia.org/wiki/WordNet/

https://wordnet.princeton.edu/

https://books.google.dz/

https://books.google.dz/