



Faculté des Sciences de Montpellier



Master AIGLE & DECOL

Rapport de TER
HMIN201

Extraction de Relations Sémantiques
dans Wikipédia

Réalisé par:

Miassa Abboute
Thierno Barry
Karim Dahdouh

Encadrant:

M. Mathieu Lafourcade

Rapporteur

Nancy Rodriguez

Mai 2020

Sommaire

Contexte.....	3
Problématique.....	3
I. Plate-forme JeuxDeMots.....	4
II. Etat de l'art.....	5
II.1 Approche d'extraction des relations sémantiques.....	5
II.2 Approche statistique.....	5
II.3 Approche linguistique.....	5
II.4 Approche Contributive.....	5
II.5 Terminologie des relations sémantiques.....	6
II.5.1 Synonymie.....	6
II.5.2 Antonymie.....	6
II.5.3 Hypéronymie/hyponymie:.....	6
II.5.4 Homonymie.....	7
II.5.5 Méronymie.....	7
II.5.6 Paronymie.....	7
II.5.7 Polysémie.....	7
II.6 Etude de quelques méthodes d'extraction d'extraction de patrons et de nouveaux couples8	
II.6.1 Processus d'extraction de patrons et de nouveaux couples, proposé par Hearst (1992)[11].....	8
II.7 Méthode d'extraction des relations sémantiques proposée.....	12
II.7.1 Processus d'extraction de relations.....	13
II.7.2 Relations sémantiques et règles d'extraction :.....	13
II.7.3 Les règles.....	14
III. Réalisation.....	18
III.1 Langage de programmation et environnement.....	18
Python.....	18
Jupyter.....	18
III.2 Algorithme d'extraction.....	19
III.3 Test.....	19
IV. Conduite de projet.....	22
IV.1 Planification du projet.....	22
IV.2 Tâches.....	24
IV.3 Ressources.....	24
IV.4 Planing avant confinement.....	25
IV.5 Planning après le confinement.....	26
IV.5.1 Diagramme de Gantt.....	26
IV.5.2 Charges de travail des ressources.....	26
IV.6 Statistiques du projet TER.....	27
IV.6.1 Diagramme de Perte.....	28

Contexte

Notre TER s'inscrit dans le cadre de l'extraction automatique de relations sémantiques qui fait partie intégrante du domaine de Traitement automatique du langage naturel. Notre objectif est de trouver des relations entre les termes d'un texte collecté à partir de Wikipédia. Plus précisément, nous nous intéressons au domaine médical en se basant sur des articles médicaux.

Pour un début, nous récupérerons des pages qui contiennent des données non structurées à partir de wikipedia ou orphanet et nous définirons des règles et les appliquerons pour extraire des relations sémantiques bien spécifiques qu'on puisse y trouver et les stocker dans notre base de données.

Ce TER a Pour objectif de renforcer le réseau lexico-sémantique du domaine médical, en occurrence enrichir le la base de connaissance JeuxDeMots.

Problématique

Le traitement automatique du langage naturel (TALN) est un domaine multidisciplinaire impliquant la linguistique, l'informatique et l'intelligence artificielle[1]. Il s'agit de l'ensemble des recherches et outils et algorithmes visant à modéliser et reproduire, à l'aide de machines, la capacité humaine à produire et à comprendre des énoncés linguistiques dans des buts de communication.

En fait, l'extraction automatique de relations sémantiques entre les termes peut servir à plusieurs finalités. Aider à indexer des documents rapidement afin de réaliser des systèmes de recherches d'information, d'améliorer la qualité des bases de connaissances existantes en ajoutant de nouvelles informations extraites à partir des textes analysés [2].

Parmi les grandes problématiques de TALN, c'est l'extraction de relations sémantiques entre les termes afin de bien comprendre de sens des phrases. Le niveau de complexité de cette extraction dépend des spécificités de la langue du texte à analyser, surtout qu'il y a plusieurs types des relations(synonymie, hyperonymie, causatif, caractéristique, etc). Le besoin de trouver des informations sémantiques à partir des données structurées est un enjeu primordial dans le domaine de TALN.

Le traitement automatique du langage naturel qui consiste à manipuler en particulier des données textuelles pour ressortir de l'information, est un sujet très prisé par les chercheurs. L'informatisation des services de santé et le développement du dossier médical personnalisé ont accéléré la production de données. Les services de santé se servent des outils informatiques pour garder ou archiver des informations médicales de tout type (les dossiers des patients, les résultats des examens, des résumés de radiographie...) en vue de faciliter et bien organiser la prise en charge des patients.

En effet, vu le volume de ses données qui ne cessent d'augmenter à une grande vélocité, il est très important de savoir comment ressortir des informations potables, utiles et adaptées au besoin du personnel de santé. Ces données sont en grande partie sous forme textuelle et c'est pourquoi donc le TALN s'avère très utile pour l'aide à la décision. Nous allons donc extraire des données sémantiques à partir de ses informations textuelles médicales et les structurer afin de faciliter la recherche de l'information et donc l'aide à la décision.

I. Plate-forme JeuxDeMots

JeuxDeMots est une plate-forme en ligne s'inscrit dans le cadre d'un programme de recherche en Traitement Automatique du Langage Naturel (TALN). Il est développé au sein de l'équipe TEXTE du LIRMM par Mathieu Lafourcade et lancé en juillet 2007, dont l'objectif de construire un réseau lexical[3].

JeuxDeMots est un GWAP (Game With A Purpose), c'est à dire une outil ludique conçue pour amener le public à générer des relations entre les mots. En fait, Le projet JeuxDeMots est un ensemble de jeux annexes qui consolident, précisent, enrichissent, ou valident certaines relations du réseau lexical créé par les joueurs, voire en créent d'autres[4]. Parmi ces jeux annexes on trouve: Tierxical, Totaki, Askit, Asku, LikeIt, SexIt, Politit, ColorIt, Emot.

JeuxDeMots est donc un instrument d'acquisition de données lexico-sémantiques générales. Il s'agit de réseau lexical le plus performant de la langue française que ce soit au niveau de son ampleur, de sa structure, et de sa diversité en incluant des ressources plus spécialisées et accessibles séparément, comme les relations de couleurs, ou les connotations politiques ou sexuelles des mots d'usage courant, ou encore les relations termes-sentiments. Le grand avantage de JeuxDeMots c'est que toutes les données soient libres de droit et gratuitement accessibles.

JeuxDeMots développé par Monsieur Lafourcade est un réseau de connaissance générale. Nous nous en servons donc pour notre travail qui concerne une spécialité bien précise à savoir le domaine médical.

II. Etat de l'art

II.1 Approche d'extraction des relations sémantiques

Dans la littérature, on peut distinguer plusieurs approches, comme l'approche contributive proposée par Lafourcade (2015) [5], celle automatique (BabelNet, Navigli and Ponzetto, 2010), et bien sûr l'approche manuelle pour WordNET (Miller, 1995). En effet, de nombreux travaux ont été consacrés à l'extraction des relations sémantiques. On peut les organiser en plusieurs familles.

II.2 Approche statistique

Cette approche exploite la régularité des termes dans un texte déterminé. Autrement dit, la technique statistique observe les fréquences de mots afin d'extraire les termes des documents. Le principe de cette approche est le suivant: "L'emploi de deux termes en cooccurrence est l'expression d'une relation sémantique entre eux" [5].

L'avantage de l'approche statistique est la facilité de sa mise en oeuvre, ainsi que son indépendance du corpus. Parmi les travaux de recherche basés sur cette approche, Hindle (1990), qui ont proposé une méthode statistique permettant d'extraire des relations sémantiques. Cette méthode détermine la similitude des noms en utilisant une métrique de similitude dérivée de la distribution des sujets, des verbes et des objets dans un corpus de texte[6].

II.3 Approche linguistique

Cette approche sur l'analyse des informations linguistiques trouvées dans un texte. En d'autres termes, l'approche linguistique cherche à découvrir des relations sémantiques entre concepts d'un corpus, tels que synonymie, Hyperonymie, hyponymie, etc.

II.4 Approche Contributive

Un des moyens de mettre en place une ressource lexicographique consiste à la construire collaborativement. l'idée est de mettre en place un système d'extraction automatique constitué de plusieurs de contributeurs au sein du réseau. Les joueurs via leurs contributions (activité ludique) valident ou invalident les termes proposées par les autres participants [7].

II.5 Terminologie des relations sémantiques

Les relations sémantiques peuvent être des relation ontologiques (hyperonymes, hyponymes, parties/tout), lexicales (synonymes). Elles jouent un rôle primordial dans mise en oeuvre les applications de TALN, tels que la traduction automatique, la création de résumé, ou encore la détection de textes similaires [8].

Une relation sémantique est une relation mettant en évidence les liens de signification entre les termes ou les classes d'une langue donnée. On constate que dans un texte, les mots sont liés entre eux par des rapports sémantiques. Ces relation peuvent être: synonymie, antonymie, etc.

II.5.1 Synonymie

Désigne la relation que deux ou plusieurs formes différentes (deux ou plusieurs signifiants) ayant le même sens (un seul signifié) entretiennent entre elles. Le principe de synonymie se base sur une procédure de substitution qui consiste à remplacer un mot par un autre dans un même contexte [9]. En effet, la synonymie est une relation qu'entretiennent entre eux divers termes ou expressions ayant le même sens ou un sens voisin.

- Exemple : permission synonyme de autorisation

II.5.2 Antonymie

C'est le contraire de la synonymie. Elle désigne une relation entre deux termes de sens contraires. La relation d'antonymie existe surtout dans les mots qui représentent des qualités ou des valeurs (bon/mauvais, vrai/faux, etc.), des quantités (peu/beaucoup, aucun/tout), des dimensions (grand/petit, long/court), des déplacements (haut/bas, droit/gauche, devant/derrière), des rapports chronologiques (jeune/vieux, après/avant).

- Exemple : heureux antonyme de triste

II.5.3 Hypéronymie/hyponymie:

Hyponymie est le terme plus général qui accompagne les hyperonymes. Autrement dit, le mot vedette est englobé dans le sens de son hyperonyme (s). Par exemple, fruit/ aliment sont des hyperonymes de cerise.

Alors que, hyponymie est un terme moins généraux qui accompagnent les hyperonymes. Par exemple, le lampadaire et néon sont des hyponymes de lampe.

Ce sont deux relations réciproques, antonymes, donc l'hyponymie : désignant la relation du genre à l'espèce . Tandis que l'hyponymie désigne la relation de l'espèce au genre. Ainsi, un animal est l'hyperonyme d'un chat, alors que un chat est l'hyponymes d'un animal.

II.5.4 Homonymie

Relation entre plusieurs formes linguistiques ayant le même signifiant, graphique ou phonique, et des signifiés entièrement différents. En effet, l'homonymie est une relation d'identité entre mots qui ne concerne que la forme, alors que toute affinité de sens est exclue (Qualifie un mot qui a la même orthographe (homographe) ou la même prononciation (homophone) qu'un autre)

Exemple

- est et est sont homonymes
- Un "livre" de pain et un "livre" de rhétorique.

II.5.5 Méronymie

Relation sémantique entre mots d'une même langue. Est une relation partitive hiérarchisée. Par exemple: "bras" est un méronyme de "corps".

II.5.6 Paronymie

Relation lexicale qui porte entre deux mots paronymes, c'est-à-dire dont les sens sont différents mais dont l'écriture et/ou la prononciation sont fort proches.

- Exemple: repartir et répartir.

II.5.7 Polysémie

La polysémie est la propriété d'un signifiant de renvoyer à plusieurs signifiés présentant des traits sémantiques communs. Il s'agit d'un mot peut avoir deux ou plusieurs sens différents. Par exemple, lit (lire) et lit (pour se coucher) un instrument.

II.6 Etude de quelques méthodes d'extraction d'extraction de patrons et de nouveaux couples

II.6.1 Processus d'extraction de patrons et de nouveaux couples, proposé par Hearst (1992)[11].

Hearst a travaillé sur l'extraction de patrons ainsi que de nouveaux couples en **1992**. Il a opté pour un processus itératif permettant d'extraire des patrons à partir de quelques couples dits « sources » (seed). Ils sont alors *Patrons*. Ainsi, à partir d'eux, trouver ou découvrir de nouveaux couples. En effet comme le but c'est de construire des couples de manière itérative, à chaque itération, les nouveaux couples découverts sont ajoutés aux couples sources de départ. ci-dessus, une image de son processus d'extraction de patrons et de nouveaux couples.

1. Définir la relation d'intérêt, par exemple l'hyponymie
2. Fournir un ensemble de couples qui respectent cette relation, par exemple « *England-country* ». Ces couples « sources » (*seed*) peuvent être définis manuellement ou bien extraits d'une base de connaissance existante.
3. Recueillir les phrases contenant les deux termes de ces couples sources
4. Chercher des environnements communs parmi ces phrases, ceux-ci seront alors considérés comme patrons pour notre relation d'intérêt
5. Les patrons émergents permettent de trouver de nouvelles instances pour la relation d'intérêt. Il s'agit alors de recommencer le processus à l'étape 2.

Figure 1: Hearst(1992) [11] méthode d'extraction de pattern et de nouveaux couples

Hearst a donc misé sur les patrons. Il estime qu'ils nécessitent peu de connaissances préalables et peuvent facilement se retrouver dans plusieurs types de texte.

Cependant, sa méthode présente un manque d'automatisation, puisque la recherche de patrons (étape 4) est réalisée manuellement.

D'autres chercheurs s'étaient alors intéressés aux travaux de Hearst(1992)[11] comme Pennacchiotti and Pantel (2006)[12], etc. Parmi ces chercheurs, Morin(1999)[13] a travaillé particulièrement sur la relation d'hyponymie du français. Comme Hearst, il s'est penché les patrons lexico-syntaxiques. Il proposa alors le système appelé **PROMÉTHÉE**.

En effet partant des faiblesses de Hearst, il a automatisé sa méthode du mieux qu'il pouvait néanmoins pas totalement. La tâche quatre(4) au cours de laquelle on extrait l'environnement commun a été automatisée mais nécessite l'intervention d'un analyste pour valider la phase automatisée pour chaque itération. ci-dessous la méthode d'extraction proposée par Morin(1999) [13].

1. Définir la relation d'intérêt, ici l'hyponymie
2. Fournir un ensemble de couples qui respectent cette relation, par exemple (*glycérol, polyol*). Ces couples « sources » (*seed*) peuvent être définis manuellement ou bien extraits d'une base de connaissance existante.
3. Recueillir les phrases contenant les deux termes lemmatisés de ces couples sources. Par exemple, le couple (*glycérol, polyol*) sélectionne la phrase « *L'hydrolyse des substrats est activée par le glucose et les polyols tels que le sorbitol et le glycérol.* »
4. Chercher des environnements communs parmi ces phrases, sous la forme d'expressions lexico-syntaxiques. Des schémas candidats sont donc extraits, ici : « NP tel que LISTE »
5. Valider les schémas candidats les plus pertinents
6. Utiliser les nouveaux schémas pour extraire de nouveaux couples de termes candidats
7. Valider les couples de termes candidats les plus pertinents. Ces nouveaux couples sont ajoutés à la liste de couples sources initiale, et le processus est réitéré à partir de l'étape 3.

Figure 2: Morin(1999) [13] méthode d'extraction de pattern et de nouveaux couples

La méthode proposé par Morin(1999) utilise des textes bruts de départ, il y a donc naturellement une première analyse qui est effectuée. cette méthode proposée par Morin à savoir **PROMÉTHÉE** effectue une première étape de pré syntaxe (prédiction, segmentation en phrases et segmentation en occurrences de 8 formes), puis un étiquetage morpho-syntaxique puis une lemmatisation. Des sigles sont ensuite extraits, puis les syntagmes nominaux (SN) sont formés, et enfin les successions de syntagmes nominaux (autrement dit, les listes). Ci-dessous, le système **PROMÉTHÉE**.

	<i>SN (LISTE) manuellement raffiné en :</i>	
1ère itération	{certain quelque plusieurs...} SN (LISTE)	(1)
	{deux trois quatre... 2 3 4...} SN (LISTE)	(2)
<hr/>		
	<i>SN : LISTE manuellement raffiné en :</i>	
	{certain quelque plusieurs...} SN : LISTE	(3)
2nde itération	{deux trois quatre... 2 3 4...} SN : LISTE	(4)
	SN , particulièrement SN ,	(5)
	{de autre}? SN tel que LISTE	(6)
<hr/>		
	SN {et ou} de autre SN	(7)
3ème itération	{de autre}? SN comme LISTE	(8)
	SN tel LISTE	(9)
<hr/>		
	SN et notamment SN	(10)
4ème itération	chez le SN , SN ,	(11)

Figure 3: méthode d'extraction de patrons par le système PROMÉTHÉE. (Morin,1999)

En effet, Comme la majorité des travaux réalisés dans l'extraction de patterns lexico-syntaxiques, Morin a expérimenté sa méthode sur des corpus spécifiques. Ces corpus ne sont en général pas assez fournis. Du fait que ces corpus ne soient pas assez riches, il est difficile d'y extraire certains patrons car ils apparaissent trop peu. Les corpus extraits d'internet règlent donc indéniablement ce problème. Ils permettent d'augmenter l'utilisation des patrons et des couples extraits.

D'ailleurs, Ravichandran and Hovy (2002) [14], deux autres chercheurs Américains qui ont utilisé la même logique avec les corpus et ils ont prouvé que cette approche donnait de très bons résultats.

Leurs travaux reposent particulièrement sur un système de **question-réponse**. Leur méthode utilise un arbre de suffixes pour trouver les patrons récurrents en fonction du contexte. ci-dessous la méthode proposée par Ravichandran et Hovy.

1. Choisir un exemple pour un type de question. Par exemple, pour une date de naissance, la question est « *Mozart* » et la réponse est « *1756* »
2. Soumettre la question et la réponse à un moteur de recherche
3. Récupérer les 1 000 premiers documents retournés
4. Segmenter ces documents en phrases
5. Ne garder que les phrases qui contiennent la question et la réponse, les tokéniser
6. Construire un arbre de suffixes pour trouver toutes les sous-chaînes de toutes longueurs
7. Ne garder que les sous-chaînes qui contiennent et la question et la réponse
8. Remplacer « *Mozart* » et « *1756* » par <NAME> et <ANSWER>

Pour calculer la précision des patrons :

1. Soumettre seulement la question au moteur de recherche, ici « *Mozart* »
2. Récupérer les 1 000 premiers documents
3. Segmenter ces documents en phrases
4. Ne garder que les phrases contenant la question
5. Chercher tous les patrons obtenus précédemment. Compter ceux qui contiennent la bonne réponse dans le champ <ANSWER>, et ceux qui contiennent un autre terme
6. Ne garder que les patrons qui apparaissent suffisamment (plus de 5 fois)

Figure 4: méthode d'extraction de patrons pour un système de question-reponse.

(Ravichandran et Hovy,2002)

L'efficacité de ce système est estimée en fonction de l'exactitude d'une réponse sur un type de relation donné. Cette méthode n'était donc pas très efficace pour les relations de type sémantique comme l'hyperonymie.

II.7 Méthode d'extraction des relations sémantiques proposée

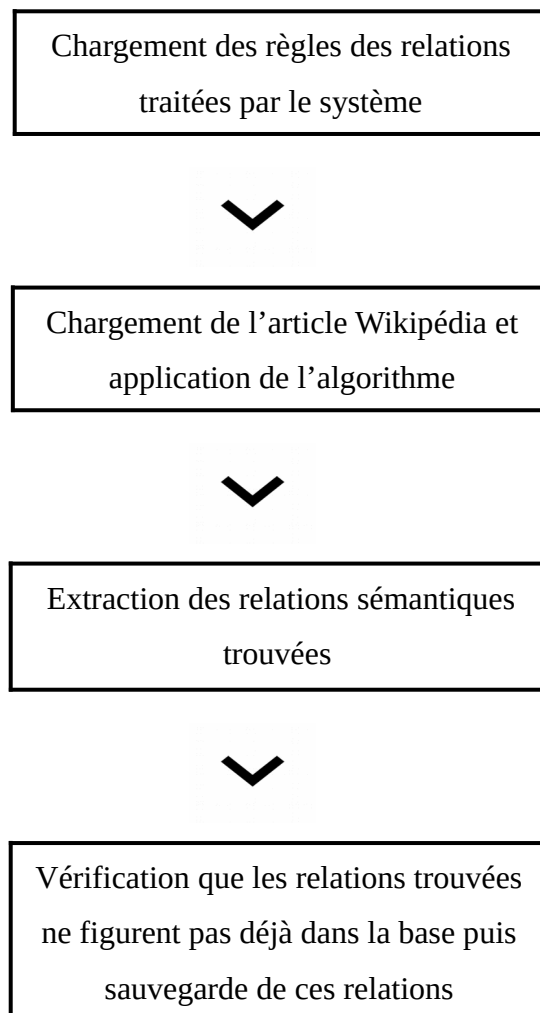
La problématique d'extraction de relations sémantiques a depuis longtemps suscité l'intérêt chez plusieurs chercheurs et de spécialités différentes. Ainsi, plusieurs méthodes d'extraction de relations sémantiques ont été proposées.

En effet, certaines méthodes se basent sur des travaux déjà existants et certaines essaient de réinventer le processus d'extraction. Aujourd'hui, la méthode la plus simple serait d'utiliser les couples déjà extraits dans **jeuxDeMots**. Dans ce cas, le processus serait donc entièrement automatique parce que le concept c'est d'utiliser des données textuelles ou autres ressources déjà existantes. Cela pourrait s'avérer très pratique et pourrait ainsi alléger la quantité énorme de travail à faire.

Cependant, nous avons opté pour une approche assez originale en accord avec notre encadreur M.Lafourcade. Nous partirons donc du niveau le plus bas à savoir la définition des règles à partir desquelles, nous nous baserons pour extraire des relations identifiées dans une phrase ou un texte donné. Cette approche nécessite donc assez de connaissances en linguistique. c'est pourquoi nous avons d'ailleurs profité de notre UE en NLP(Natural Language processing) pour construire nos bases en linguistique informatique.

La méthode que nous avons proposée est donc assez originale et c'est à nous de construire nos propres couples, les rechercher de manière itérative dans un article Wikipédia. Nous avons donc défini des règles permettant de construire des couples de mots comme(**chat** | **animal**) de la relation **r_isa** encore appelée (**r_est_Un**). Nous expliquerons un peu plus bas notamment dans la présentation de notre modèle ainsi que son fonctionnement, comment ce processus d'extraction fonctionne.

II.7.1 Processus d'extraction de relations



II.7.2 Relations sémantiques et règles d'extraction :

Pour extraire des relations sémantiques à partir de wikipédia, nous nous sommes focalisés dans un premier temps, sur l'analyse des types de relations suivantes:

r_isa (X est une sorte de Y)

Il est demandé d'énumérer les génériques/hyperonymes du terme. Par exemple, 'animal' et 'mammifère' sont des génériques de 'chat'.

r_own (X possède Y)

Que POSSÈDE le terme suivant ? Par exemple, un étudiant possède un stylo, une cavalière des bottes, (étudiant r_own stylo).

r_carac

Pour un terme donné, souvent un objet, il est demandé d'en énumérer les CARACTéristiques (adjectifs) possibles/typiques. Par exemple, 'liquide', 'froide', 'chaude', pour 'eau'.

Pour cela, on a défini un ensemble de règles associées à chaque type de relation. Ci-dessous un extrait du fichier des règles de la relation **r_isa**.

II.7.3 Les règles

Ci-dessous quelques règles d'extraction des relations sémantiques.

r_isa :

```
nc1 être det nc2 => nc1 r_isa nc2
nc1 être det nc2 punc => nc1 r_isa nc2
det nc1 être det nc2 => nc1 r_isa nc2
det nc1 être det nc2 punc => nc1 r_isa nc2
nc1 nc2 être det nc3 => nc1 nc2 r_isa nc3
nc1 nc2 être det nc3 punc => nc1 nc2 r_isa nc3
det nc1 nc2 être det nc3 => nc1 nc2 r_isa nc3
det nc1 nc2 être det nc3 punc => nc1 nc2 r_isa nc3
det nc1 nc2 être p nc3 punc => nc1 nc2 r_isa nc3
det nc1 p nc2 être det nc3 => nc1 r_isa nc3
det nc1 p nc2 être p nc3 => nc1 r_isa nc3
det nc1 p nc2 être det nc3 punc => nc1 r_isa nc3
det nc1 p nc2 être p nc3 punc => nc1 r_isa nc3
det nc1 p det nc2 être det nc3 => nc1 r_isa nc3
det nc1 p det nc2 être det nc3 adj => nc1 r_isa nc3
det nc1 p det nc2 être p nc3 => nc1 r_isa nc3
det nc1 p det nc2 être p nc3 adj => nc1 r_isa nc3
```

Quelques règles d'extraction de relations sémantiques pour la relation (**r_isa**)

r_own :

```
nc1 avoir det nc2 => nc1 r_own nc2
nc1 avoir det nc2 punc => nc1 r_own nc2
det nc1 avoir det nc2 => nc1 r_own nc2
det nc1 avoir det nc2 punc => nc1 r_own nc2
nc1 nc2 avoir det nc3 => nc1 nc2 r_own nc3
nc1 nc2 avoir det nc3 punc => nc1 nc2 r_own nc3
nc1 nc2 avoir det nc3 punc => nc1 nc2 r_own nc3
det nc1 nc2 avoir det nc3 => nc1 nc2 r_own nc3
det nc1 nc2 avoir det nc3 punc => nc1 nc2 r_own nc3
det nc1 p nc2 avoir det nc3 => nc1 r_own nc3
det nc1 p nc2 avoir det nc3 punc => nc1 r_own nc3
det nc1 p det nc2 avoir det nc3 => nc1 r_own nc3
det nc1 p det nc2 avoir det nc3 punc => nc1 r_own nc3
```

Quelques règles d'extraction de relations sémantiques pour la relation (**r_own**)

r_carac :

```
nc1 être adj => nc1 r_carac adj
nc1 être adj punc => nc1 r_carac adj
det nc1 être adj => nc1 r_carac adj
det nc1 être adj punc => nc1 r_carac adj
nc1 nc2 être adj => nc1 nc2 r_carac adj
nc1 nc2 être adj punc => nc1 nc2 r_carac adj
det nc1 nc2 être adj => nc1 nc2 r_carac adj
det nc1 nc2 être adj punc => nc1 nc2 r_carac adj
```


det nc1 p nc2 être adj => nc1 r_carac adj
det nc1 p nc2 être adj punc => nc1 r_carac adj
det nc1 p det nc2 être adj => nc1 r_carac adj
det nc1 p det nc2 être adj punc => nc1 r_carac adj
det nc1 p det nc2 être adj => nc1 r_carac adj

Quelques règles d'extraction de relations sémantiques pour la relation (**r_carac**)

En effet, afin d'extraire les relations sémantiques, on a appliqué plusieurs techniques et outils comme NLTK. NLTK fournit un ensemble de fonctions permettant d'effectuer une tâche de TALN (pos_tag() pour le POS-Tagging, sent_tokenize() pour la segmentation des phrases, word_tokenize() pour la tokenisation,...).

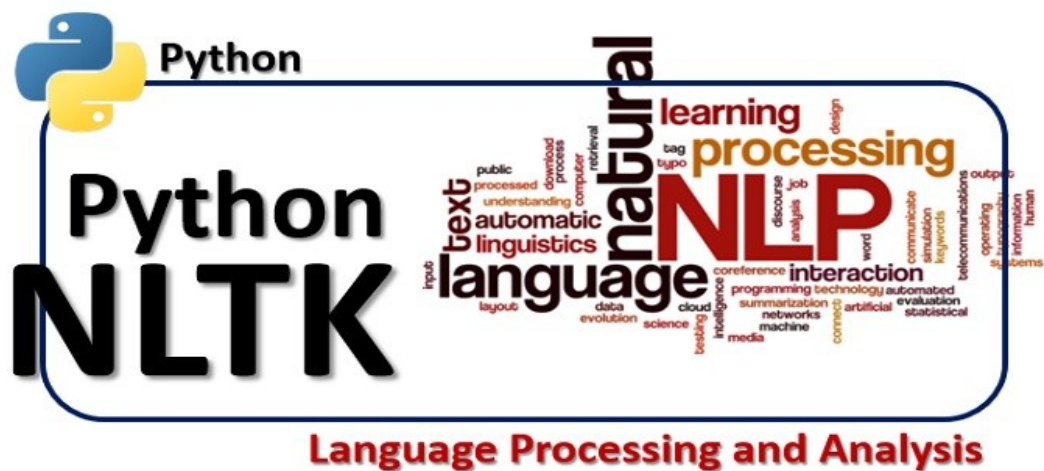


Figure 5: NLTK(Natural Language ToolKit)

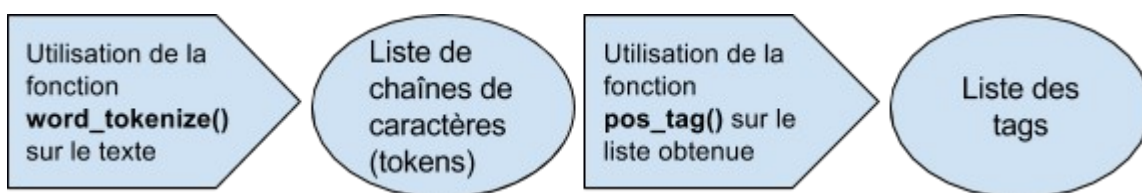


Figure 6: processus d'étiquetage

L'objectif était de transformer un texte en une série de tokens individuels, puis d'appliquer un pos-tagging pour générer le type de chaque mot(verbe, nom, adjectif, etc.).

Dans un premier temps, nous avons commencé par des phrases simples . Nous avons fait de l'étiquetage mot par mot de la phrase. Pour réaliser cette tâche, nous allons utiliser la bibliothèque NLTK(Natural language ToolKit). Pour faire ce travail d'étiquetage, nous allons commencer par découper la phrase par mot encore appelé (tokenization) et ensuite appliquer notre fonction d'étiquetage.

Exemple : la phrase : “*Jean mange la pomme*”

```
['jean', 'mange', 'la', 'pomme']  
[('jean', 'NC'), ('mange', 'ADJ'), ('la', 'DET'), ('pomme', 'N')]
```

Figure 7: résultat de la tokenization et de l'étiquetage

A noter que nous avons bien sûr utiliser un french_tagger parce que nous serons amenés à travailler sur des articles wikipédia en Français.

Après cette étape, nous allons procéder à l'application des règles pour procéder à l'extraction . Comme indiqué, nous nous focaliserons dans un premier temps à l'extraction des relations sur de simples phrases.

Exemple :

Nous allons essayer d'extraire une relation dans la phrase suivante :

<le chat est un animal.>

et le résultat donné par notre algorithme est

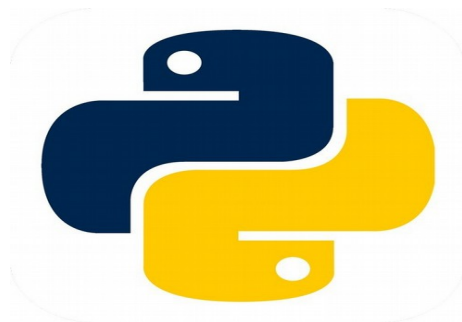
```
la phrase <le chat est un animal.> contient une relation semantique de type:  
chat R_isa animal
```

Figure 8: résultat d'une simple extraction de démo

III. Réalisation

III.1 Langage de programmation et environnement

Python



Après prospection de plusieurs langages de programmation, nous avons opté au final pour python qui nous offre plusieurs bibliothèques pour faire du NLP(Natural Language Processing). Il n'y a pas lieu de vanter ici les prouesses de ce langage tant connu aujourd'hui et qui s'avère efficace dans plusieurs domaines.



Nous n'allons pas aussi donner trop de détails sur ce très bon environnement de développement. Nous allons juste retenir que le projet Jupyter existe pour développer des logiciels open source, des standards ouverts et des services pour l'informatique interactive à travers des dizaines de langages de programmation.

III.2 Algorithme d'extraction

Dans cette partie, nous avons procédé de manière itérative. Nous présenterons donc les trois(2) dernières versions de nos travaux. Ces différents travaux ont été discutés avec notre encadreur M. Lafourcade et il n'a pas manqué de nous apporter des suggestions.

En effet, dans la version qui précédait celle-ci, nous avons juste présenté un similaire de prototype pour être sûr que c'était la bonne voie qu'on a emprunté. Nous avons défini des règles minimalistes qui s'adaptent à un type de phrase comme <<Le chat est un animal>> une phrase très simple. Il faut noter que c'était cette version qu'on devait présenter après le retour des vacances de Février. Après présentation, naturellement l'encadreur avait des suggestions. ce sont ces suggestions qui nous a permis d'aborder une version assez lucide et concrète de notre travail, en intégrant de nouvelles méthodes comme la prise en charge d'un fichier qui contiendra les règles d'extractions ou encore les données textuelles à partir desquelles nous appliquerons notre algorithme pour extraire des relations sémantiques.

Lors de la précédente version, on était en vue de sortir les règles du code et de les mettre dans un fichier à part et c'est ce que nous avons fait. Dans la prochaine version, nous prévoyons aussi de sortir la phrase/texte wikipédia à analyser pour que ça soit un fichier en entrée.

En effet, il est important de rappeler que nos règles sont en fonction de l'étiquetage que nous retourne le PostTagger(français). ci-dessous un exemple de règles:

<<det nc1 p det nc2 être p nc3 => nc1 r_isa nc3.>>

Jusque là donc on a une version de l'algorithme qui interprète bien nos règles.

Nous allons maintenant passer un petit paragraphe de wikipedia pour tester notre modèle d'extraction de relation sémantique avec un vrai exemple.

III.3 Test

Texte wikipedia :

<< Le cancer est une maladie provoquée par la transformation de cellules qui deviennent anormales et prolifèrent de façon excessive. Ces cellules dérégées finissent parfois par former une masse qu'on appelle tumeur maligne. Les cellules cancéreuses ont tendance à envahir les tissus voisins et à se détacher de la tumeur initiale. Elles migrent alors par les vaisseaux sanguins et les vaisseaux lymphatiques pour aller former une autre tumeur (métastase). >>

Nous pouvons remarquer que les phrases sont assez longues et très différentes de nos petites phrases comme <<le chat est malade.>> Après un premier test naturellement notre algorithme était un peu perdu et comme on l'a dit précédemment c'est un travail itératif donc on a adapté nos règles d'extraction pour ne pas limiter le nombre de mots dans la phrase.

Cependant on avait déjà géré tout ce qui est mot composé par exemple.

<< Le chat de la voisine est malade.>> ou *<< Le coronavirus de Wuhan....>*. c'est donc un grand pas.

La dernière version venait donc à point régler le problème de phrases trop longues et régler les derniers détails majeurs de notre modèle. En effet, on ne peut couvrir tous les cas possibles et même avec les règles qu'on a eu à définir, il se peut qu'il y est des erreurs ou des manquements. C'est pourquoi nous avons mis nos règles dans ce fichier pour faciliter la maintenance et l'évolution de notre modèle, tout en respectant les recommandations techniques ou de bonnes pratiques pour qu'on puisse améliorer notre travail et aller plus en profondeur.

Lors de notre dernier rendez-vous avec notre encadreur, on avait reçu des suggestions comme toujours et nous avons su en faire bon usage. Parmi elles : quelques modifications notamment sur les règles en nommant par exemple les NC(Noms Communs) que nous donnait notre PosTagger en

nc1,nc2,nc3, etc, pour mieux les distinguer. chose qu'on a faite. l'autre point c'était d'adapter notre algorithme pour qu'il prenne enfin un texte de wikipédia pour voir s'il arrive à extraire de nouvelles relations.

Petite présentation du nouveau format des règles:

//r_isa

nc1 être det nc2 => nc1 r_isa nc2

nc1 être det nc2 punc => nc1 r_isa nc2

det nc1 être det nc2 => nc1 r_isa nc2

det nc1 être det nc2 punc => nc1 r_isa nc2

nous avons là appliqué les recommandations.

Ensuite, il y a par exemple l'auxiliaire être qui est en brut(infinitif) dans la règle mais nous savons que nous pouvons le retrouver sous plusieurs temps(conjugaison) dans le texte à traiter. pour donc alléger notre fichier de règles ainsi que notre algorithme, nous avons créé un fichier qui contient la conjugaison des auxiliaires (être et avoir). ce qui nous facilitera la maintenance de notre modèle.

Pour tester notre algorithme qui traite jusque là trois(3) relations à savoir:

- r_isa
- r_carac
- r_own

Nous avons pris des phrases qu'on a déjà testées plusieurs fois et dont nous connaissons déjà le résultat, et nous avons maintenant pris un petit paragraphe sur un article médical de wikipédia pour voir ce que ça allait donner aussi.

Texte wikipedia :

<< *Le cancer est une maladie provoquée par la transformation de cellules qui deviennent anormales et prolifèrent de façon excessive. Ces cellules dérégées finissent parfois par former une masse qu'on appelle tumeur maligne. Les cellules cancéreuses ont tendance à envahir les tissus voisins et à se détacher de la tumeur initiale. Elles migrent alors par les vaisseaux sanguins et les*

vaisseaux lymphatiques pour aller former une autre tumeur (métastase). >>

resultat :

une realation extraite : de type r_isa:

<<cancer r_isa maladie>>

En effet, nous pouvons faire un premier constat à savoir donc le format de la phrase dans laquelle la relation a été extraite. Elle contient (19) mots. ce qui est loin de nos petites phrases comme : *<< Le chat de la voisine est malade.>>*

En effet, nous nous sommes dit qu'on devait adopter une logique dans laquelle il s'agira de trouver notre règle dans une phrase quelque soit sa longueur. C'est sûr qu'on traite pas tous les cas possibles, mais nous arriverons à extraire un maximum de relations possibles que si nous essayons de limiter le nombre de mots contenus dans une phrase.

Bien évidemment, notre texte est dans un fichier d'entrée comme les règles ainsi que la conjugaison des auxiliaires. L'autre point aussi c'est qu'on a commencé la construction de la base de connaissances de nos trois(3) relations traitées jusque là, qui représente de dernier point à faire dans le sujet de TER. Nous avons donc créé des fichiers(r_isa, r_own, r_carac) où nous enregistrons les nouvelles relations.

En résumé voilà où nous en sommes et nous avons en perspective de traiter de nouvelles relations dans l'avenir parceque c'est un sujet très très intéressant, tester et retester notre algorithme pour le rendre plus efficace et optimal, améliorer et trouver de nouvelles règles sur les types de relations déjà traitées, extraire au maximum de nouvelles relations sur des articles médicaux de wikipédia. Nous prévoyons aussi de déployer notre modèle en ligne pour vulgariser son utilisation et essayer d'obtenir plus de retour et l'améliorer encore et encore.

IV. Conduite de projet

IV.1 Planification du projet

Afin de réaliser le projet TER dans le délai fixé (20 Mai), il a fallu définir les tâches essentielles et estimer le temps à consacrer pour chacune en fonction de nos ressources. Pour cela, nous avons tracé, avec notre encadrant Mr. Lafourcade une liste des objectifs dès le début du projet (5 Février).

La mise en place de notre planning se fait à l'aide de du logiciel open source ProjectLibre. Il offre une large palette de fonctionnalités, et permet de créer tout type de diagramme. Cela nous a permis de présenter notre diagramme de gantt, de PERT, et de visualiser la charge de travail de chaque ressource durant tout le cycle de vie de ce projet.






























Dans notre conduite de projet, nous avons commencé par saisir des tâches, établir des liens entre celles-ci, puis la création des ressources et les affecter à ces tâches. En effet, nous avons opté pour la méthode de répartition proportionnelle afin de faire l'estimation des charges. En partant de ce principe nous avons vingt-sept (27) boîtes à saisir. Celles-ci sont regroupées en trois grandes tâches, à savoir:

- ★ Etudes des besoins
- ★ Analyse du projet
- ★ Conception
- ★ Développement et test
- ★ Clôture (Rédaction de rapport et présentation).

Comme nous avons vu dans l'UE conduite de projet, nous avons fait une répartition temporelle de 10% (Etude des besoins), presque 40% (Analyse projet et conception), 50%(Développement, Test et clôture). Par conséquent, on a consacré 6 jours pour l'étude des besoins, 11,75 pour l'analyse du projet, 10 jours pour la conception, 31 jours pour le développement et test et enfin 11 jours pour la rédaction du rapport et présentation.

Le tableau suivant donne une description détaillé des différents tâches effectuées dans le cadre de notre TER.

IV.2 Tâches

		Name	Perce...	Duration	Start	Finish	Resource Names
1		☐Projet TER	100%	69.25 da...	2/5/20, 8:00 AM	5/12/20, 10:00 AM	
2		☐Etude des besoins	100%	6.25 days	2/5/20, 8:00 AM	2/13/20, 10:00 AM	
3		Accord sur les objectifs	100%	3 days	2/5/20, 8:00 AM	2/7/20, 5:00 PM	Miassa Abboute;Encad...
4		Spécification des besoins	100%	2 days	2/10/20, 8:00 AM	2/11/20, 5:00 PM	Miassa Abboute;Encad...
5		Définition des contraintes	100%	1.25 days	2/12/20, 8:00 AM	2/13/20, 10:00 AM	Miassa Abboute;Encad...
6		☐Analyse du projet	100%	10.75 days	2/13/20, 10:00 AM	2/27/20, 5:00 PM	
7		Recherche de travaux existants	100%	5 days	2/13/20, 10:00 AM	2/20/20, 10:00 AM	Miassa Abboute
8		Etude de relations sémantiques	100%	4.125 days	2/13/20, 12:00 PM	2/19/20, 2:00 PM	Zainoul Barry
9		Exploration de la base de connaissance	100%	5 days	2/21/20, 8:00 AM	2/27/20, 5:00 PM	Karim Dahdouh
10		☐Conception	100%	10.125 d...	2/28/20, 8:00 AM	3/13/20, 9:00 AM	
11		Identification de la gouvernance	100%	2 days?	2/28/20, 8:00 AM	3/2/20, 5:00 PM	Encadrant
12		Listing des tâches	100%	1.5 days?	3/3/20, 8:00 AM	3/4/20, 1:00 PM	
13		Identification des moyens matériels et l...	100%	3 days?	3/3/20, 9:00 AM	3/6/20, 9:00 AM	Miassa Abboute;Encad...
14		Choix du langage de développement	100%	2.625 da...	3/3/20, 9:00 AM	3/5/20, 3:00 PM	Zainoul Barry;Karim D...
15		Aclimatation des moyens	100%	5 days	3/6/20, 9:00 AM	3/13/20, 9:00 AM	Miassa Abboute
16		☐Développement et Tests	100%	30.875 d...	3/13/20, 9:00 AM	4/24/20, 5:00 PM	
17		Proposition d'une première version de l'...	100%	5 days	3/13/20, 9:00 AM	3/20/20, 9:00 AM	Zainoul Barry;Karim D...
18		Création fichier de règles	100%	3 days?	3/20/20, 9:00 AM	3/25/20, 9:00 AM	Miassa Abboute
19		Implémentation + Test de la première v...	100%	2 days	3/25/20, 9:00 AM	3/27/20, 9:00 AM	Zainoul Barry
20		Création d'une base de données lexico-s...	100%	2 days	3/25/20, 9:00 AM	3/27/20, 9:00 AM	Karim Dahdouh
21		Retour de l'encadrant + Recueil des su...	100%	6 days	3/30/20, 8:00 AM	4/6/20, 5:00 PM	Miassa Abboute;Encad...
22		Améliorations de l'algorithme + Tests	100%	2.5 days?	3/30/20, 8:00 AM	4/1/20, 1:00 PM	Karim Dahdouh
23		Modification et remplissage du fichier d...	100%	3 days	4/3/20, 8:00 AM	4/7/20, 5:00 PM	Miassa Abboute
24		Remplissage de la base de données lexi...	100%	7 days	4/9/20, 8:00 AM	4/17/20, 5:00 PM	Zainoul Barry;Karim D...
25		Echange avec une doctorante sur le sujet	100%	5 days	4/20/20, 8:00 AM	4/24/20, 5:00 PM	Miassa Abboute
26		☐Clôture	100%	11.25 days	4/27/20, 8:00 AM	5/12/20, 10:00 AM	
27		Rédaction rapport final	100%	9.125 days	4/27/20, 8:00 AM	5/8/20, 9:00 AM	Miassa Abboute;Zaino...
28		Préparation slides présentation	100%	2.125 days	5/8/20, 9:00 AM	5/12/20, 10:00 AM	Miassa Abboute;Zaino...
Projet TER - Organisation							

Après avoir tracé les tâches, nous avons établi des connexion entre celles ci suivant les priorités et recommandations de notre encadrant.

IV.3 Ressources

Concernant les ressources, nous sommes trois étudiants qui travaillent sur ce projet, plus notre encadrant qui nous propose de collaborer avec sa doctorante pour de profiter de son expérience, parce qu'elle fait des travaux de recherche sur l'extraction des relations sémantiques. La liste des ressources sont illustrées dans le tableau ci-dessous.

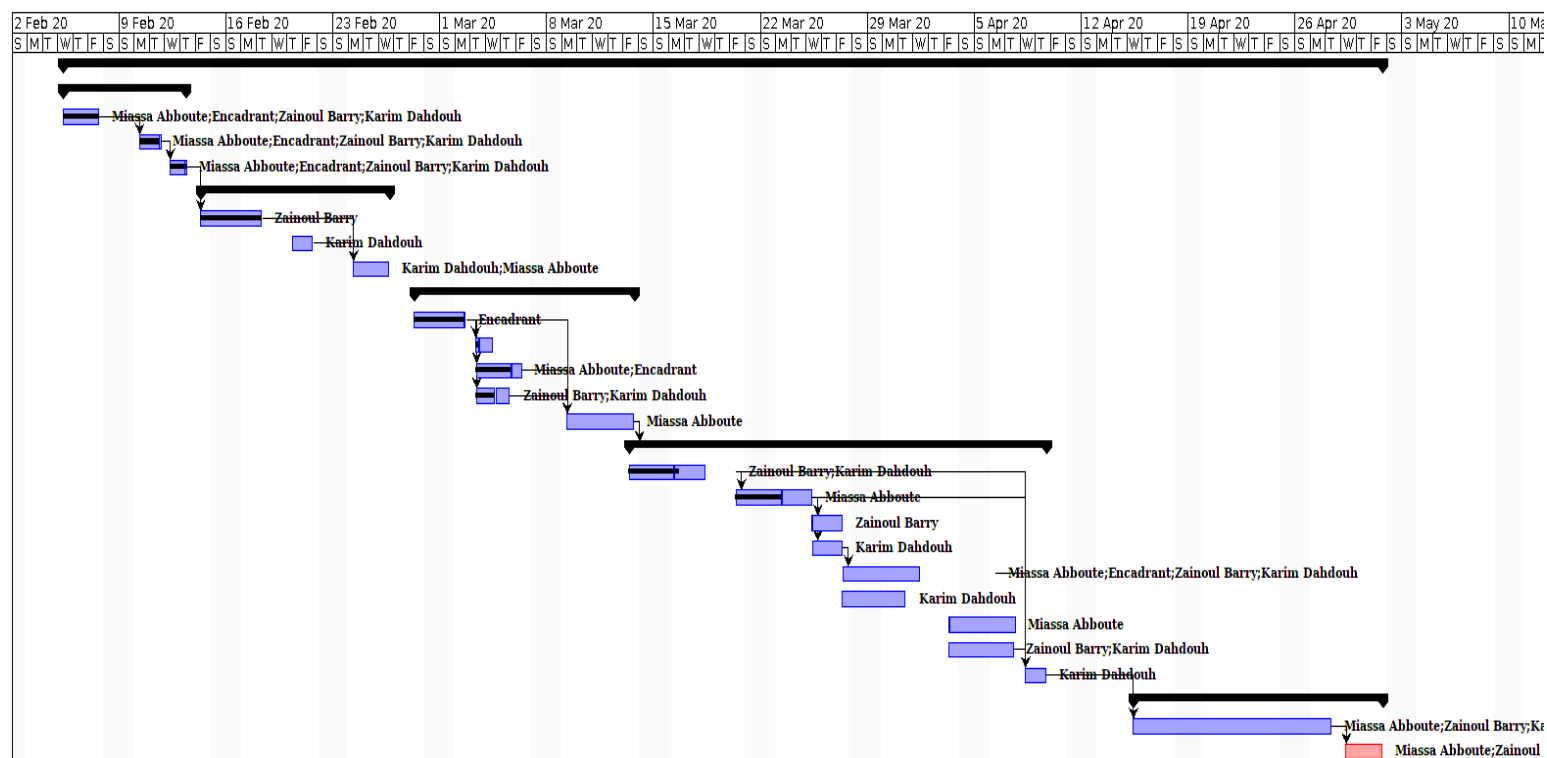
		Name	Type	Initials	Max. Units	Standard Rate	Overtime Rate	Cost Per Use
1		Miassa Abboute	Work	M	100%	\$0.00/hour	\$0.00/hour	\$0.00
2		Zainoul Barry	Work	Z	100%	\$0.00/hour	\$0.00/hour	\$0.00
3		Karim Dahdouh	Work	K	100%	\$0.00/hour	\$0.00/hour	\$0.00
4		Encadrant	Work	K	100%	\$0.00/hour	\$0.00/hour	\$0.00
5		Doctorante	Work	M	100%	\$0.00/hour	\$0.00/hour	\$0.00

IV.4 Planing avant confinement

D'abord, un premier planning a été établi. Cependant, depuis, une situation très grave (covid-19) a atteint le pays, ainsi que beaucoup d'autres dans le monde. Durant cette période difficile, nous sommes obligés de vivre dans

des conditions exceptionnelles, et respecter les gestes barrières et de mesures de distanciation physique à cause de confinement. Malheureusement, cette pandémie dure plus longtemps, ce que nous oblige à travailler à distance et de planifier à nouveau nos rendez-vous avec l'encadrant et la répartition des tâches entre nous. Le TER étant un projet à effectuer en groupe, ce confinement durcit le travail. Néanmoins, nous essayons de communiquer très souvent entre nous, puis avec notre encadrant, pour le respect des délais.

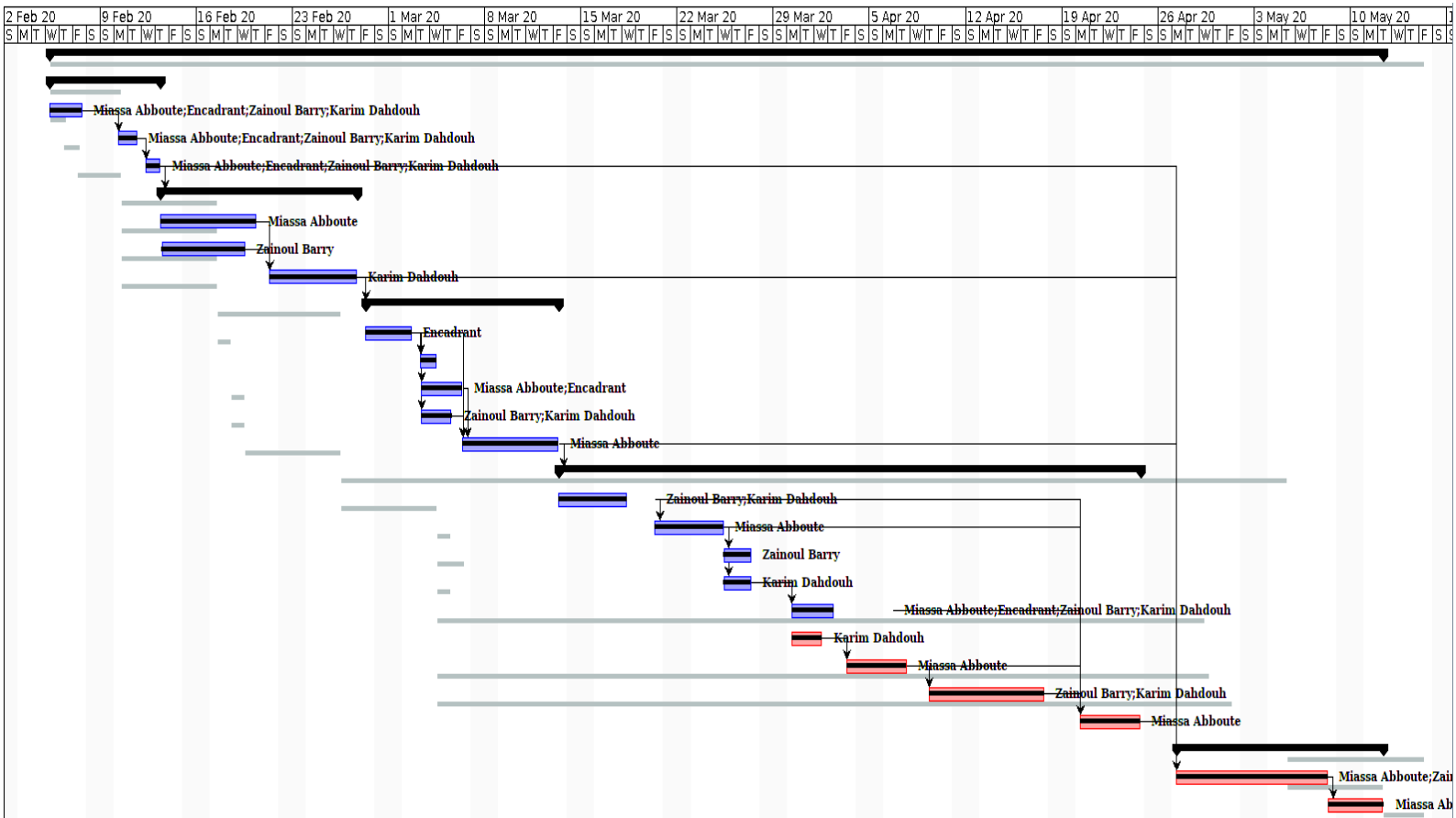
Le planning avant le confinement, proposé ci-dessous, a été mis en place, ainsi que le diagramme de Gantt qui nous permet de visualiser dans le temps les diverses tâches composant le projet.



IV.5 Planning après le confinement

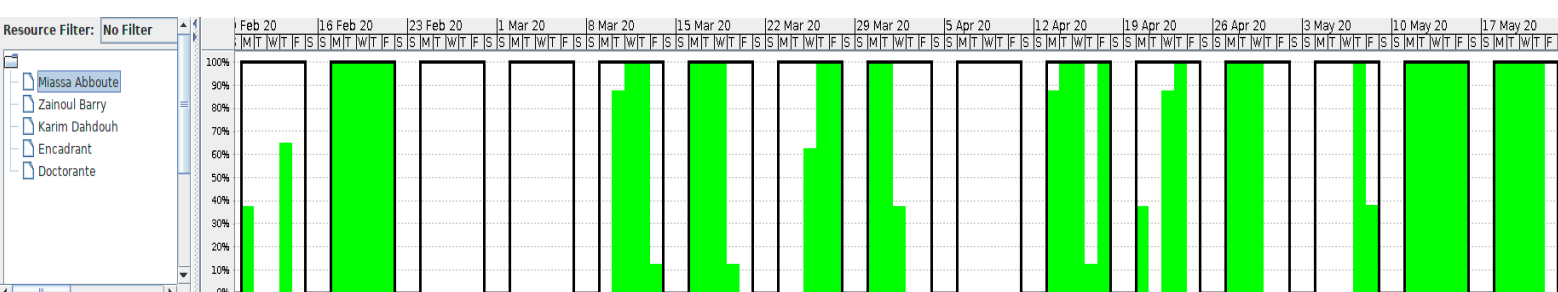
IV.5.1 Diagramme de Gantt

Le diagramme suivant représente le déroulement dans le temps de notre TER adapté à la situation du confinement, à partir du démarrage le 2 avril jusqu'à la date de fin le 12 mai. C'est une version sauvegardée de notre planning qui montre le lien entre les tâches ainsi que le chemin critique.



IV.5.2 Charges de travail des ressources.

Pour pouvoir répartir d'une façon homogène nos ressources dans le temps, nous sommes basés sur deux stratégies, qui sont le lissage (en déplacement les tâches) et le nivellement (en affectant les tâches aux ressources libres). Le schéma ci-dessous donne un exemple de la répartition des charges de travail de notre collègue Miassa Aboutte entre le 2 avril et le 12 mai (fin projet).



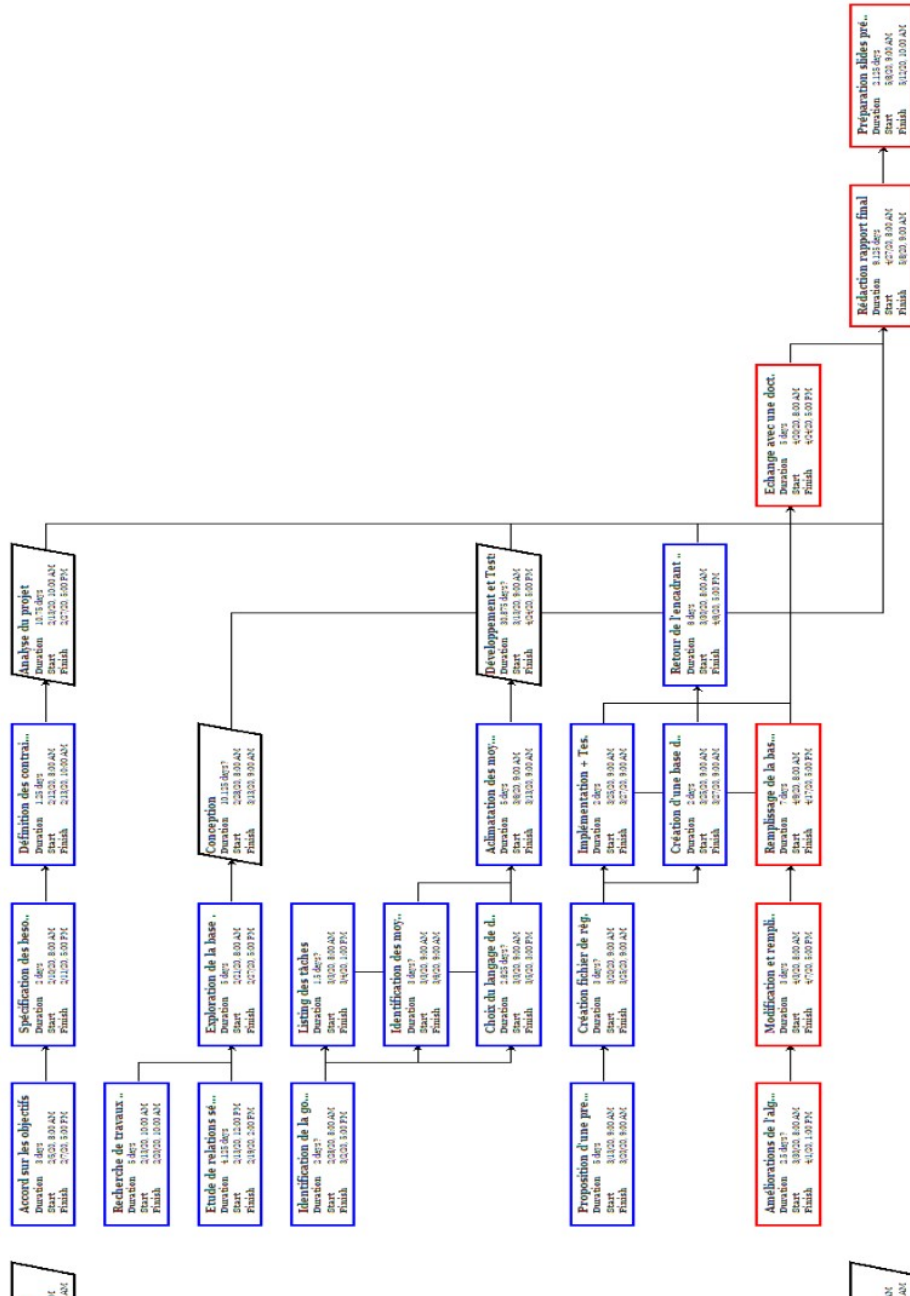
IV.6 Statistiques du projet TER

Le schéma ci-dessous donne un récapitulatif de l'organisation de notre TER.

General		Statistics		Notes	
Name: <input type="text" value="Projet TER - Organisation"/>					
Start:	2/5/20, 8:00 AM	Finish:	5/12/20, 10:00 AM		
Baseline Start:	2/5/20, 8:00 AM	Baseline Finish:	5/15/20, 10:00 AM		
Actual Start:	2/5/20, 8:00 AM	Actual Finish:	5/12/20, 10:00 AM		
Duration:	69.25 days	Baseline Duration:	72.25 days		
Actual Duration:	0 days	Remaining Duration:	69.25 days		
Work:	854.678 hours	Baseline Work:	548 hours		
Actual Work:	854.678 hours	Remaining Work:	0 hours		
Cost:	\$0.00	Baseline Cost:	\$0.00		
Actual Cost:	\$0.00	Remaining Cost:	\$0.00		

IV.6.1 Diagramme de Perte

Le diagramme suivant décrit les différentes tâches et l'interconnexion entre celles-ci. Il montre également le chemin critique.



Conclusion

L'objectif de ce projet était l'extraction automatique de relations sémantiques dans Wikipédia. Nos connaissances et compétences dans ce vaste domaine, bien qu'elles soient limitées, nous ont permis de gouverner chaque étape. De l'étude des besoins, en passant par l'analyse puis la conception et le développement du projet en nous appuyant sur différentes méthodes existantes, nous avons pu respecter les modalités définies. Nous avons comme perspectives, l'amélioration et l'optimisation de l'algorithme et l'enrichissement de notre fichier de règles, afin de permettre une extraction maximale de nouvelles relations sémantiques présentes dans tout texte étudié. Nous souhaiterions aussi déployer notre modèle perfectionné en ligne pour vulgariser son utilisation.

A son terme, nous avons consolidé les connaissances précédemment acquises dans le domaine du TALN (Traitement Automatique du Langage Naturel), une spécialité du passé, enracinée dans le présent et qui s'ancre dans le futur. Ce fut par ailleurs, une occasion de mettre à l'épreuve nos capacités de gestion, qui semblent avoir été à la hauteur, l'organisation de notre groupe ayant permis d'outrepasser toute difficulté rencontrée.

References

- [1] “Traitement automatique du langage naturel,” *Wikipédia*. 08-Dec-2019.
- [2] L. Ramadier and M. Lafourcade, “Patrons sémantiques pour l’extraction de relations entre termes - Application aux comptes rendus radiologiques,” in *TALN: Traitement Automatique des Langues Naturelles*, Paris, France, 2016.
- [3] “infosJeuxDeMots.” [Online]. Available: <http://imaginat.name/infosJDM.html>. [Accessed: 07-Mar-2020].
- [4] “Bénéfices et limites de l’acquisition lexicale dans l’expérience JeuxDeMots | Mathieu Lafourcade et Alain Joubert.” [Online]. Available: <https://benjamins.com/catalog/lis.30.06laf>. [Accessed: 07-Mar-2020].
- [5] G. Grefenstette, “Use of syntactic context to produce term association lists for text retrieval,” in *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, Copenhagen, Denmark, 1992, pp. 89–97, doi: 10.1145/133160.133181.
- [6] D. Hindle, “Noun Classification From Predicate-Argument Structures,” in *28th Annual Meeting of the Association for Computational Linguistics*, Pittsburgh, Pennsylvania, USA, 1990, pp. 268–275, doi: 10.3115/981823.981857.
- [7] “Jeux et intelligence collective - résolution de problèmes et acquisition de données sur le Web. | Request PDF,” *ResearchGate*. [Online]. Available: https://www.researchgate.net/publication/271214294_Jeux_et_intelligence_collective_-_resolution_de_problemes_et_acquisition_de_donnees_sur_le_Web. [Accessed: 07-Mar-2020].
- [8] M. Lafourcade and N. L. Brun, “Parcourir, reconnaître et réfléchir. Combinaison de méthodes légères pour l’extraction de relations sémantiques,” p. 9.
- [9] V. Quasimodo Substituables, “Les relations sémantiques - ppt télécharger.” [Online]. Available: <https://slideplayer.fr/slide/13877370/>. [Accessed: 07-Mar-2020].
- [11] Hearst, M. A. (1992). Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th Conference on Computational Linguistics - Volume 2, COLING '92*, pages 539–545, Stroudsburg, PA, USA. Association for Computational Linguistics.
- [12] Pennacchiotti, M. and Pantel, P. (2006). Ontologizing semantic relations. In *Proceedings of the*

21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics, ACL44, pages 793–800, Stroudsburg, PA, USA. Association for Computational Linguistics.

- [13] Morin, E. (1999). Extraction de liens sémantiques entre termes à partir de corpus de textes techniques. PhD thesis, Nantes, Grenoble. Th. : informatique.

- [14] Ravichandran, D. and Hovy, E. (2002). Learning surface text patterns for a question answering system. In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, ACL '02, pages 41–47, Stroudsburg, PA, USA. Association for Computational Linguistics.