RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

Université Dr. Tahar Moulay SAIDA

Faculté : Technologie

Département : Informatique

MEMOIRE DE MASTER

Option:MICR

Vers la construction des graphes de connaissances à partir des textes

Présenté par:

- **DAOUDI Asma**

Encadré par:

- **Dr ZAHAF Ahmed**

Année universitaire :2020/2021

Remerciement:

Je tien à saisir cette occasion et adresser mes profonds remerciements et mes profondes reconnaissances à :

- *Mon encadrant de mémoire de fin d'étude* ***Dr ZAHAF Ahmed****, pour ses précieux conseils et son orientation ficelée tout au long de ma recherche.*
- *Les responsables de* ***SONALGAZ*** *pour leur contribution dans la réussite de ce projet notamment* ***MOKHTAR Cherif****.*
- *A ma famille et mes amis qui par leurs prières et leurs encouragements, on a pu surmonter tous les obstacles.*
- *Je voudrais remercier aussi toutes les personnes qui ont participé de près ou de loin à mes recherches et à l'élaboration de ce mémoire.*

À tous ces intervenants, je présente mes remerciements, mon respect et ma gratitude.

Dédicace:

Au meilleur des pères

A ma très chère maman
Qu'ils trouvent en moi la source de leur fierté
A qui je dois tout

A mon frère CHERIF
qui m'a vraiment encouragé durant toute mes années d'étude

A ma sœur AICHA et mes frères MUSTAPHA et AMINE
A qui je souhaite un avenir radieux plein de réussite

A HOUDA;
Pour votre compréhension et patience

A mes Amis A tous ceux qui me sont chers

A toutes les personnes qui sont chères à mon cœur,

Table de matières

Table des figures

Table des tableaux:

Introduction générale

Introductiongénérale:

Les entreprises accumulent de grandes quantités de texte sous forme de notes de service, de contrats, d'articles, de courriels, de critiques de produits, de manuels, de rapports, d'actualités, etc., qui sont une mine de connaissances. Il n'est pas surprenant, puisque les humains sont les ultimes producteurs et consommateurs de connaissances, que les connaissances se manifestent dans la langue parlée et imprimée (texte) .

Cependant, les solutions de traitement de la langue dominantes sont incomplètes et ne conviennent pas à un déploiement plus large.

Lancé en 2012 sur la version française de Google, le knowledge graph, littéralement « graphe de connaissances », est une innovation du célèbre moteur de recherche dans l'affichage des pages de résultats. Il s'agit d'une compilation structurée et particulièrement détaillée de toutes les informations détenues par Google ainsi que des éléments sémantiques provenant d'autres sources, comme Wikipédia, Wikidata ou CIA World Factbook, concernant l'objet de la recherche. En d'autres termes, le knowledge graph fournit une réponse organisée à une requête lancée par l'utilisateur. Les graphes de connaissances sont devenus une abstraction convaincante, pour organiser les connaissances structurées du monde sur Internet, et un moyen pour intégrer les informations extraites à partir de plusieurs sources d'information. Les graphes de connaissances ont également commencé à jouer un rôle très important dans l'apprentissage automatique (machine learning) en tant que méthode d'intégration des connaissances mondiales, et pour expliquer ce qui est appris.

Un graphe de connaissances est constitué d'entités, de propriétés et de relations stockées dans une base de données Graph sous forme de nœuds et d'arêtes**[1]**. Les connaissances, c'est-à-dire les entités, les propriétés et les relations extraites de sources de données telles qu'un texte, peuvent être structurées sous forme de nœuds et d'arêtes et chargées dans des bases de données graphiques avec une impédance minimale.

Il existe deux types d'approches dans le développement des KG **[2]**: l'approche descendante se concentre sur le schéma de connaissances tels que les ontologies de domaine et l'approche ascendante se concentre sur les instances de connaissances telles que les ensembles de données liées (en anglais, Linked Open Data). Dans l'approche descendante mettant l'accent sur les ontologies de domaine bien définies, les ontologies de domaine et leur schéma doivent d'abord être définis, puis des instances de connaissances sont ajoutées à la base de connaissances. L'approche ascendante extrait des instances de connaissances à partir de ressources de connaissances. Après la fusion des connaissances des instances peuplées, les ontologies de niveau supérieur sont construites au moyen d'instances de connaissances pour créer l'ensemble des KG. Le travail présenté dans ce mémoire suit l'approche ascendante pour la construction des graphes de connaissances à partir des textes. Nous utilisons des techniques du

traitement du langage naturel (NLP) pour traiter les sources textuelles et créer des graphes de connaissances pouvant prendre en charge une variété d'analyses.

Il y a trois tâches NLP qui sont directement pertinentes à la construction d'un graphe de connaissances : l'extraction d'entités, l'extraction de relations et la résolution d'entités. L'extraction d'entité est la tâche d'identifier les entités clés d'intérêt (par exemple, les organisations, les personnes, les lieux, etc.) à partir du texte. Ces entités constituent généralement les nœuds d'un graphe de connaissances. L'extraction de la relation est la tâche dans laquelle, étant donné deux entités d'intérêt et du texte, nous extrayons les relations entre elles (par exemple, les ventes nettes, les informations sur l'équipe de gestion, etc.) du texte. Parfois, l'extraction de relation est également utilisée pour extraire les propriétés d'une entité donnée. Les relations et propriétés extraites deviendront généralement des relations ou des propriétés de nœud dans notre graphe de connaissances. La résolution d'entité consiste à identifier si plusieurs mentions dans un texte font référence à la même entité.

Le mémoire est organisé sous forme trois chapitres. Le premier chapitre introduit les graphes de connaissances en générale et quelques graphes de connaissances open source et commerciale. Le deuxième chapitre est dédiéà la présentation de quelques techniques de la construction des graphes de connaissances. Dans le chapitre trois nous présentons notre application pour l’extraction des connaissances à partir des textes.

Chapitre1:

Les graphes de connaissances

1 Les graphes de connaissances

1.1 Bases de connaissances

1.1.1 Définition :

La notion de base de connaissances vient des sciences informatiques pour décrire une base de données regroupant des connaissances spécifiques à un domaine spécialisé précis, sous une forme exploitable par un ordinateur. Elle contient des règles, des faits ou d'autres représentations nécessaires à l'exercice d'une activité donnée pour laquelle cette base de connaissances a été développée **[1]**.

Pour rappel, un graphe est un ensemble d'entités reliées entre elles, composé de nœuds qui représentent les entités et d'arcs ou arêtes qui représentent les relations. Les relations ou arcs peuvent être enrichis par des attributs ou encore une valeur quantitative représentant le poids de la relation**[3]**.

Le Knowledge Graph, terme introduit par Google, est une représentation de la connaissance relative à un domaine ou une entreprise sous une forme qui est facilement exploitable par la machine. Il est constitué d'entités et de relations, les entités étant des noms de personnes, des concepts, ou des objets**[3]**.

Le graphe de connaissance est une base de connaissance sémantique qui permet de décrire la sémantique des sources d'information et rendre ainsi le contenu explicite**[1]**. En effet, une entité ou un mot seul ne porte pas beaucoup de sens en soi, celui-ci se révèle quand l'entité est prise dans son contexte qui est défini par les propriétés de l'entité et les relations avec des entités auxiliaires. Chaque entité participe donc à la compréhension des entités auxquelles elle est liée. On dit alors que la base de connaissance est sémantique car elle encode le sens des données afin de mieux permettre aux machines de comprendre une requête introduite par l'être l'humain. On peut donc utiliser le langage naturel pour interroger une base de connaissance et générer une réponse en langage naturel car les mots sont reliés aux concepts. C'est ce qui permet aux assistants vocaux tels que Google Assistant, Siri ou Alexa de comprendre la requête de l'utilisateur et de répondre avec précision**[1]**.

1.1.2 Regain d'intérêt et nouvelles applications :

Bien que cette représentation graphique de la connaissance ne soit pas récente, elle a gagné une popularité et elle est un élément clé pour des applications d'intelligence artificiel lié à la recherche rapide et contextuelle d'information ainsi qu'à la prise de décision. Les données sont stockés dans des bases de données de type graphe et peuvent être structurées ou non structurées **[3]**.

Il existe de nombreuses applications des graphes de connaissances à la fois dans la recherche et l'industrie. Dans l'informatique, nous trouvons plusieurs manières d'utilisations d'une représentation

graphique, par exemple, le flux de donnéesgraphiques, diagrammes de décision binaires, cartes d'état, etc.

Dans cequi suit nous présentons deux applications concrètes qui ont conduit à la récente augmentation de la popularité des graphes de connaissances :organisation de l'information sur Internet et intégration des données**[3]**.

1.1.3 Graphes pour organiser les connaissances sur Internet

Nous expliquerons l'utilisation d'un graphe de connaissances sur le web en reprend l'exemple deWikidatacité dans **[4]**.. Wikidataprésente le stockage central pour les données structurées de Wikipédia, Pour montrer l'interaction entre les deux et pour motiver l'utilisation du graphe de connaissances Wikidata,Considérons la ville de Winterthur en Suisse qui a une page dans Wikipedia La page Wikipédia de Winterthour répertorie ses villes jumelles : deux en Suisse, une en République tchèque et une en Autriche La ville deL'Ontario en Californie qui a une page Wikipédia intitulée Ontario, Californie, répertorie Winterthur comme son ville sœur. Les relations entre villes sœurs et villes jumelles sont identiques et réciproques Ainsi, sila ville A est une ville sœur d'une autre ville B, alors B doit être une ville sœur de A. Cette inférence doit être automatique, mais parce que cette information est indiquée en anglais dans Wikipedia, il n'est pas facile de détecter cet écart.En revanche, dans la représentation Wikidata de Winterthur, il existe une relation appelé organisme administratif jumelé qui répertorie la ville de l'Ontario.Comme cette relation est symétrique,la page Wikidata de la ville d'Ontario inclut automatiquement Winterthur. Ainsi, lorsque le graphe de connaissances de Wikidatasera entièrement intégré à Wikipédia, de telles divergences seront naturellementdisparaître**[4]**.

Wikidata inclut des données de plusieurs fournisseurs indépendants, par exemple, la Bibliothèque du Congrès qui publie des données contenant des informations sur Winterthur. En utilisant l'identifiant Wikidata pour

Winterthur, les informations publiées par la Bibliothèque du Congrès peuvent être facilement liées à des informations disponibles auprès d'autres sources. Wikidata facilite l'établissement de tels liens en publiant les définitions des relations utilisées dans [Schema.Org]**[4]**.

Le vocabulaire des relations dans Schema.Org nous donne, au moins, trois avantages. Premièrement, il est possible d'écrire des requêtes qui s'étendent sur plusieurs ensembles de données qui n'auraient pas été possibles autrement. Un exemple d'une telle requête est : Afficher sur une carte les villes de naissance des personnes décédées en Heure d'hiver Deuxièmement, avec une telle capacité de requête il est possible de générer facilement des boîtes d'informations structurées dans Wikipedia.

Troisièmement, les informations structurées renvoyées par les requêtes peuvent égalementapparaissent dans les résultats de recherche qui est désormais une fonctionnalité standard pour les principaux moteurs de recherche. Une version récente de Wikidatacontient plus de 80

millions d'objets, avec plus d'un milliard de relations parmi ces objets. Wikidata établit des connexions à travers plus de 4872 catalogues différents dans 414 différentes langues publiées par des fournisseurs de données indépendants. Selon l'estimation récente, 31 % des sites Web, et plus de 12 millions de fournisseurs de données publient des annotations utilisent actuellement le vocabulaire de Schema.Org**[3]**.

1.1.4 Graphes pour l'intégration de données dans les entreprises

L'intégration de données est le processus consistant à combiner des données provenant de différentes sources et à fournir à l'utilisateur une vue unifiée des données. Une grande partie des données des entreprises réside dans les bases de données relationnelles. Une approche de l'intégration des données repose sur un schéma global qui capture les interrelations entre les éléments de données représentés dans ces bases de données. La création d'un schéma global est un processus extrêmement difficile car il existe de nombreuses tables et attributs, les experts qui ont créé ces bases de données ne sont généralement pas disponibles ; et en raison du manque de documentation, il est difficile de comprendre le sens des données. En raison des défis liés à la création d'un schéma global, il est pratique de contourner ce problème et de convertir les données relationnelles en une base de données avec le schéma générique des triplets, c'est-à-dire un graphe de connaissances. Les mappages entre les attributs sont créés en fonction des besoins, par exemple, en réponse à des questions commerciales spécifiques, et peuvent eux-mêmes être représentés dans un graphe de connaissances. Nous illustrons ce processus à l'aide d'un exemple concret**[3]**.

De nombreuses institutions financières souhaitent créer un graphe de connaissances d'entreprise qui combine les données internes des clients avec les données sous licence de tiers. Un exemple d'utilisation d'un graphed'entreprise est d'évaluer le risque tout en prenant des décisions de prêt. Les données externes contiennent des informations telles que les fournisseurs d'une entreprise. Si une entreprise reconnue des difficultés financières, cela augmente le risque d'accorder un prêt aux fournisseurs de cette entreprise. Pour combiner ces données externes avec les données internes, il faut relier les schémas externes avec le schéma interne de l'entreprise. De plus, les noms de sociétés utilisés dans les sources externes doivent être liés aux identifiants clients correspondants utilisés par les institutions financières. Lors de l'utilisation d'une approche de graphes de connaissances pour l'intégration de données, la détermination de telles relations peut être retardée jusqu'à ce qu'elles soient réellement requises.

1.2 Modèles de graphes de connaissances : RDF et Graphes de propriétés

Il existe deux modèles de graphes de connaissances : le modèle de description des ressources Web RDF et les graphes de propriétés

1.2.1 RDF (Resource Description Framework)

RDF n'est pas à proprement parler un langage. Il s'agit plutôt d'un modèle de données pour décrire des ressources sur le web**[3]**, **[5]**.On entend par ressource toute entité que l'on veut décrire sur le web mais qui n'est pas nécessairement accessible sur le web. Par exemple, on pourrait fournir des informations sur l'auteur d'un tel document, bien que la personne décrite ne soit pas accessible par le web. On trouve plutôt des ressources, comme sa page personnelle, ou une photo, qui peuvent être obtenues à partir de leur URL (Universal Resource Locator). Ces ressources sont reliées à cette personne, mais ne sont pas cette personne.

Pour désigner cette personne, on utilisera une URI (Universal Resource Identifier), un nom unique qui ressemble syntaxiquement à une URL, mais à la différence près qu'il n'est pas nécessaire que celle-ci soit accessible sur le web. D'une certaine manière l'ensemble des URL est inclus dans l'ensemble des URI (voir figure 1).

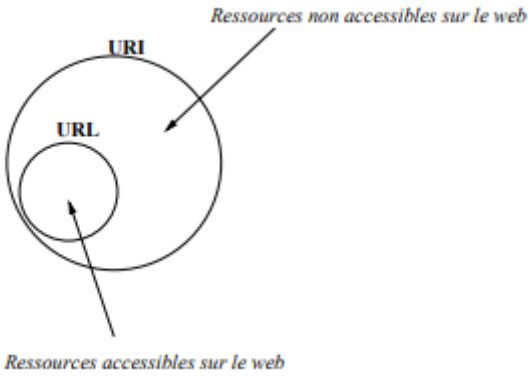


Figure 1:Ensembles des URI et des URL

La raison d'être de RDF est de permettre que les informations sur les ressources soient manipulées par des applications, plutôt que d'être simplement affichées aux utilisateurs du web. C'est entre autres pour cette raison qu'une syntaxe XML a été proposée pour véhiculer des informations modélisées en RDF**[5][6].** Il existe aussi une autre notation plus facile à lire pour un humain : Turtle.

Parmi les caractéristiques importantes de RDF, on retrouve sa flexibilité et son extensibilité. N'importe qui peut ajouter des informations sur une ressource, en autant que l'on connaisse son URI. Ainsi, deux documents distincts situés à des endroits différents peuvent fournir des descriptions d'une

même ressource. Rien n'empêche une application d'extraire les descriptions fournies par ces documents et de le fusionner. Aussi, rien n'oblige qu'une ressource décrite existe réellement. En fait, RDF impose peu de contraintes sur les descriptions possibles. Supposons par exemple que nous voulions décrire une personne qui s'appelle Michel Gagnon, qui travaille au département de génie informatique de l'École polytechnique de Montréal et dont la page personnelle se trouve à l'URL suivante : http://www.professeurs.polymtl.ca/michel.gagnon. Pour ce faire, il faut d'abord reconnaître qu'il y a quatre entités référées par la description : la personne en question, son nom, l'endroit où elle travaille et sa page personnelle. Pour chacune, sauf le nom, il faut donc une URI pour la représenter. Le nom est un cas spécial, puisqu'il s'agit en fait d'une chaîne de caractères. Nous pourrrons dans ce cas utiliser la chaîne directement, sans passer par l'intermédiaire d'une URI pour la désigner. Supposons donc que les quatre entités de notre description sont désignées de la manière suivante :

La personne décrite http://www.polymtl.ca/Profs#MichelGagnon le nom de la personne "Michel Gagnon" le lieu de travail : http://www.polymtl.ca/Vocabulary#dgi

La page personnelle http://www.professeurs.polymtl.ca/michel.gagnon

Veuillez noter que même si les URI désignant la personne et son lieu de travail ont la forme http://www..., cela ne signifie pas nécessairement qu'il s'agisse d'une URL correspondant à un document auquel on peut accéder sur le web. Il s'agit tout simplement d'une convention utilisée pour uniformiser les URI. L'intérêt de cette convention est qu'elle laisse la possibilité de traiter cette URI comme une URL et de mettre sur le web un document correspondant à cette URL. Un document, par exemple, qui expliquerait de manière informelle l'entité représentée par l'URI en question. Mais rappelons encore une fois qu'il n'est pas nécessaire qu'il y ait un document sur le web pour chaque URI**[5]**.

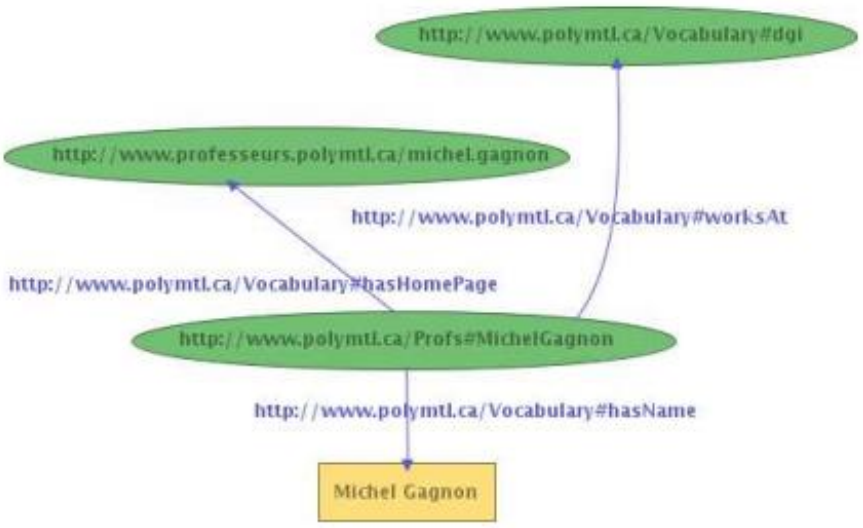


Figure 2:Un premier exemple

Il nous faut maintenant représenter les relations entre ces entités. RDF prévoit l'existence de propriétés. Il s'agit d'entités dont le rôle est d'établir un lien entre deux autres entités. Dans notre exemple, il faudra utiliser trois propriétés pour relier la personne avec son nom, son lieu de travail, et sa page personnelle.

1.2.2 Graphes de propriétés

Le modèle de graphes de propriétés est utilisé par de nombreux systèmes de bases de données de graphes populaires. Contrairement à RDF qui était explicitement motivé par un besoin de modéliser des informations sur le Web, les systèmes de base de données gèrent des données de graphes générales. Les systèmes des graphes de bases de données se distinguent des bases de données relationnelles traditionnelles en s'appuyant peu sur un schéma prédéfini et l'optimisation des opérations qui impliquent des parcours de graphes. Dans cette section, nous considérerons le modèle de données du graphe de propriétés et le langage (Cypher) chiffrement utilisé pour l'interroger **[3]**.

1.2.2.1 Modèle de données de graphe de propriétés

Un modèle de données de graphe de propriétés se compose de nœuds, de relations et de propriétés.Chaque nœud a une étiquette et un ensemble de propriétés sous la forme de paires clé-valeur arbitraires. Les clés sont des chaînes et les valeurs sont des types de données arbitraires. Une relation est une arête dirigée entre deux nœuds, a une étiquette et peut avoir un ensemble de propriétés**[4]**.

Dans le graphe de propriétés ci-dessous, ART et BEA sont deux nœuds. Le nœud pour BEA a deux propriétés : Age et localité de voisinage. Ces deux nœuds sont connectés par une arête étiquetée. Cette limite a la propriété d'indiquer l'année depuis laquelle ART et BEA se sont connus**[4]**.

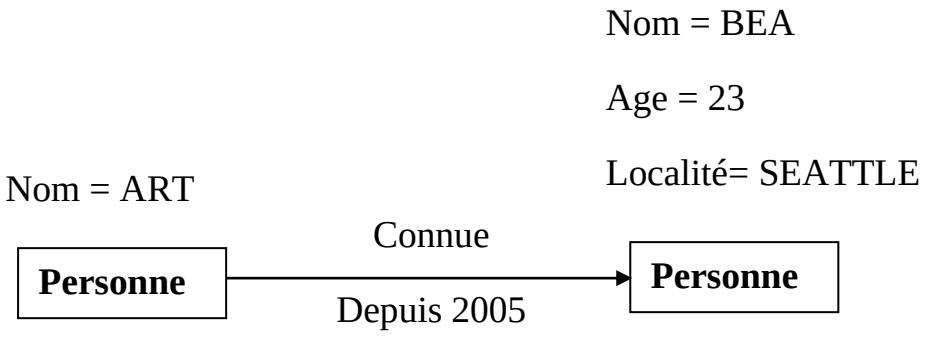


Figure 3 : Modèle d'un graphe de propriété

1.2.2.2 Langage de CypherQuery

Cypher est un langage permettant d'interroger des données représentées dans un modèle de données de graphe de propriétés. Les concepts de conception de Cypher sont envisagés pour être adoptés dans une norme ISO pour un langage de requête graphique. En plus de l'interrogation, Cypher peut également être utilisé pour créer, mettre à jour et supprimer des données d'une base de données graphique. Dans cette section, nous présenterons uniquement les capacités de requête de Cypher.

L'exemple ci-dessous montre la requête Cypher par rapport au graphique de données considéré précédemment et les requêtes pour les personnes connues par ART. La requête Query se compose de deux parties : la clause MATCH spécifie un modèle de graphique qui doit correspondre au graphe de données et la clause RETURN spécifie ce que la requête doit renvoyer. Le modèle de graphe est spécifié dans une notation ASCII pour les graphes : chaque nœud est écrit entre parenthèses, et chaque arête est écrite sous forme de flèche. Les spécifications de nœud et de relation incluent leurs types respectifs et toutes les propriétés supplémentaires qui doivent être mises en correspondance**[4]**.

```
MATCH (p1:Person {name: art}) -[:knows]-> (p2: Person)RETURN p2
```

Dans l'exemple ci-dessous, nous montrons la requête Cypher qui demande tous les amis d'une personne qui existent depuis 2010

```
MATCH (p1:Person {name:art}) -[:knows {since: 2010}]-> (p2: Person)
```

À partir de la requête ci-dessus, nous pouvons voir qu'il est tout aussi facile d'associer des propriétés à des relations qu'à des nœuds. Une personne peut avoir des amis depuis des années avant 2010, et si nous voulons que la requête inclue également ces amis, cela peut être fait en ajoutant une clause WHERE

```
 MATCH(p1:Person name:art})[:knows{since:Y}]->(p2: Person)
WHERE Y <= 2010
RETURN p2
```

Grâce à la clause WHERE, il est possible de spécifier une variété de contraintes de filtrage ainsi que des modèles qui peuvent être utilisés pour restreindre les résultats de la requête. En outre, Cypher fournit des constructions de langage pour compter les résultats, regrouper les données par valeurs et rechercher des valeurs minimales/maximales, ainsi que d'autres opérations mathématiques et d'agrégation.

Chapitre 2 :

Construction des graphes de connaissance

2 Construction des graphes de connaissances

2.1 Architecture d'un graphe de connaissance

En général, l'architecture des graphes de connaissances peut être dérivée comme illustré **à la figure 3**.Du point de vue des graphes de connaissances basés sur l'ontologie, il existe deux approches principales pour créer des graphes de connaissances L'un est descendant et l'autre ascendant**[2]**

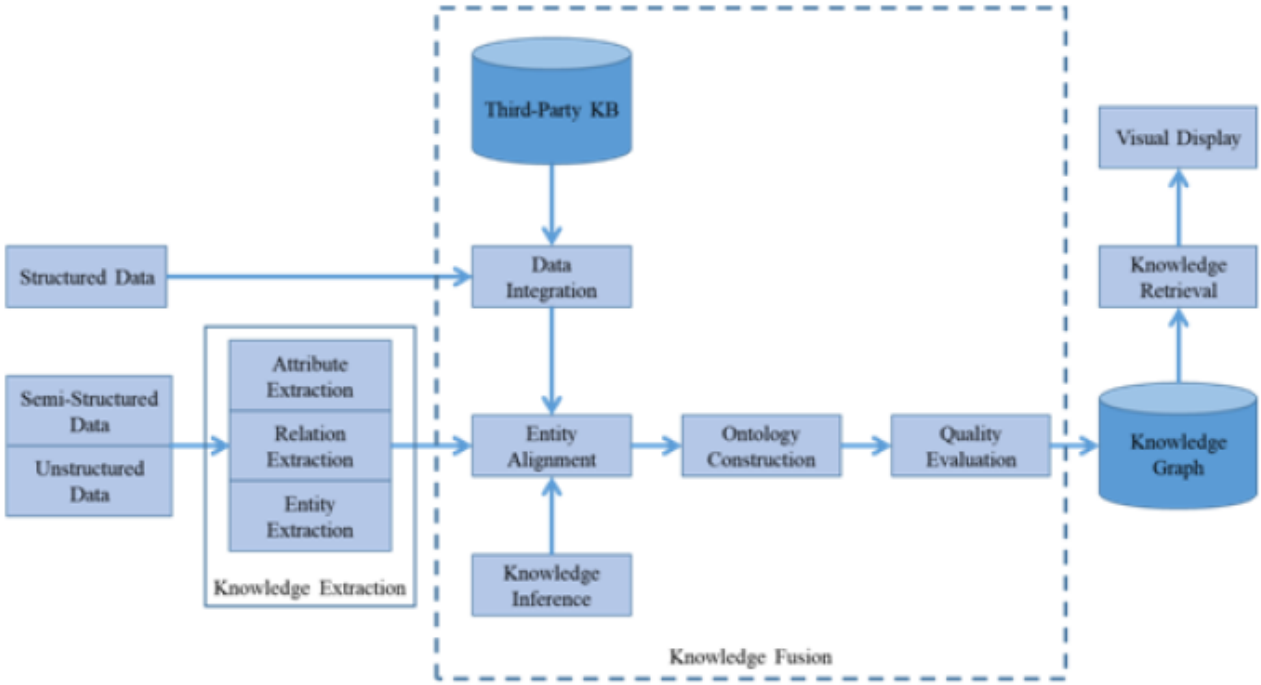


Figure 3:Architecture d'un graphe de connaissances

2.2 Construction des graphes de connaissances

Il existe deux types d'approches pour la construction des graphes de connaissance [5]s : l'approche descendante se concentre sur le schéma de connaissances tels que les ontologies de domaine et l'approche ascendante se concentre sur les instances de connaissances telles que les ensembles de données liées (en anglais, Linked Open Data). Dans l'approche descendante mettant l'accent sur les ontologies de domaine bien définies, les ontologies de domaine et leur schéma doivent d'abord être définis, puis des instances de connaissances sont ajoutées à la base de connaissances. L'approche ascendante extrait des instances de connaissances à partir de ressources de connaissances. Après la

fusion des connaissances des instances peuplées, les ontologies de niveau supérieur sont construites au moyen d'instances de connaissances pour créer l'ensemble des KG. Le travail présenté dans ce mémoire suit l'approche ascendante pour la construction des graphes de connaissances à partir des textes. Nous utilisons le traitement du langage naturel (NLP) pour traiter les sources textuelles et créer des graphes de connaissances pouvant prendre en charge une variété d'analyses.

Il y a trois tâches NLP qui sont directement pertinentes à la construction d'un graphe de connaissances : l'extraction d'entités, l'extraction de relations et la résolution d'entités. L'extraction d'entité est la tâche d'identifier les entités clés d'intérêt (par exemple, les organisations, les personnes, les lieux, etc.) à partir du texte. Ces entités constituent généralement les nœuds d'un graphe de connaissances. L'extraction de la relation est la tâche dans laquelle, étant donné deux entités d'intérêt et du texte, nous extrayons les relations entre elles (par exemple, les ventes nettes, les informations sur l'équipe de gestion, etc.) du texte. Parfois, l'extraction de relation est également utilisée pour extraire les propriétés d'une entité donnée. Les relations et propriétés extraites deviendront généralement des relations ou des propriétés de nœud dans notre graphe de connaissances. La résolution d'entité consiste à identifier si plusieurs mentions dans un texte font référence à la même entité**[2]**

2.3 Procédures d'approche ascendante des Graphes de connaissances

Cette section décrit les procédures globales de l'approche ascendantepour construire des graphes de connaissances. La figure 3 montre l'ensemble de l'architecture de cette méthode. Chacune des sous-sections suivantes présentes les technologies utilisées dans chaque phase de cette architecture**[2]**

2.3.1 Extractionde connaissances

L'extraction de connaissances est analysée sous quatre aspects : sources de données, types, approches et outils.

2.3.1.1 Sources de données

Pour l'extraction de connaissances, il existe principalement trois types de sources de données : données structurées comme les bases de données relationnelles, semi-structurées telles que des données HTML, XML et JSON des données non structurées telles que texte libre, image et documents. Différentes sources de données sont les résultats de différentes méthodes et stratégies d'extraction de connaissances.Les connaissances extraites sont généralement représentées dans un format lisible par machine, tels que RDF et JSON-LDformat.

La plupart des premiers graphes de connaissances n'extraient des connaissances qu'à partir de source de données spécifique et unique. Par exemple, les premiers graphes de connaissance de l'industrie médicale extraient généralement les connaissances à partir des articles médicaux en ligne. Concernent les graphes de connaissances industriels, une grande échelle des connaissances approfondies sont extraient à partir d'une grande variété de sources de données Pour exemple, Google

KnowledgeVault acquiert des connaissances à partir de documents texte, Arbres HTML, tableaux HTML et pages annotées humaines.

Outre, certains ensembles de données et sites Web fournissent des instances de connaissances de qualité pour les graphes de connaissances Par exemple,Wikipédia, l'encyclopédie Web typique, est devenue la source des instances les plus importantes pour les graphes de connaissances. À l'heure actuelle, la plupart des graphes de connaissances extraient les instances de Wikipédia telles que YAGO et DBpedia. Certains graphes de connaissance de l'industrie ont également obtenir des connaissances de Wikipédia. Par exemple, le système médical CRF (ConditionalRandom Fields) capture les connaissances médicales de Wikipédia. De plus, WordNet, GeoNames, ConceptNet, IMDB (Internet MovieDatabase) et MusicBrainz sont également des bonnes ressources de connaissances.

Hétérogène, inter-domaine et multilingue sont des caractéristiques typiques des ressources de connaissances actuelles pour une large diffusion des graphes de connaissances qui apportent de grands défis pour les techniques d'extraction connaissance**[2].**

2.3.1.2 Types d'extraction de connaissances

Les types d'extraction de connaissances sont en général divisés en troistypes : extraction d'entité, extraction de relation et extraction d'attribut. Cette dernière, l'extraction d'attributs peut être considérée comme une sorte d'extraction de relations spéciales.

Extraction d'entités, y compris la reconnaissance d'entités nommées (NER),est de découvrir des entités à partir d'une grande variété de ressources de connaissanceset essayez de les classer dans des catégories prédéfinies telles que personne,emplacement, organisation, titre de l'actualité, service, heure, date, etc.La qualité de l'extraction d'entités influence généralement grandement l'efficacitéet la qualité de l'acquisition ultérieure des connaissances, c'est donc l'un desparties la plus fondamentale et la plus importante de l'extraction des connaissances.

Après l'extraction des entités, les relations entre les entités sontanalysées aux relations d'extrait conceptuel. L'extraction de la relation estpour trouver les relations entre les entités et obtenir des informations sémantiques afin de construire des graphes de connaissances.

L'extraction d'attributs consiste à définir la sémantique intentionnelledes entités tandis que l'extraction de la relation consiste à spécifier la dénotation sémantique des entités.L'extraction d'attribut est importante pour définir plus clairement le concept d'entité**[2].**

2.3.1.3 Approches des extractions de connaissances

Les approches d'extraction de connaissancesexigent l'utilisation des techniques du traitement du langage naturel(NLP),des techniques dutextmining et des techniques dumachine learning. Au début l'extraction des connaissances utilise généralement des corpus annotés manuellement.

A cette époque, les approches d'extraction de connaissances utilisaient principalement des méthodes basées sur des règles et de dictionnaires.Les approches d'apprentissage automatique utilisent

principalement les algorithmes d'apprentissage supervisé qui construisent un modèled'apprentissage à partir d'un ensemble de données annoté manuellement. Cependant, les corpus annotés manuellement nécessitent un grand nombre d'experts du domaine et doivent passer beaucoup de temps pour générer l'information. L'échelle du corpus résultant est généralement petite, ce qui est difficile à répondre au besoin d'extraction de connaissances. De plus, les approches supervisées ont des limitations importantes même si elles peuvent obtenir une grande précision. Les approches supervisées s'appuient trop sur le corpus d'origine car elles ne permettent pas d'identifier de nouvelles entités nommées et ne peuvent pas généraliser à des relations différentes.

Actuellement, certains algorithmes semi-supervisés et non supervisés ont été proposés pour éviter une partie de l'effort d'annotation. De nombreux types différents de classificateurs d'apprentissage automatique ont été appliqués à l'extraction de connaissances, tels que les modèles de Markov cachés (HMM), les champs aléatoires conditionnels (CRF), les k-Nearest-Neighbors (KNN), les modèles d'entropie maximale et les machines à vecteurs de support (SVM. Les performances de ces algorithmes de machine learning dépendent des fonctionnalités utilisées.

Pour l'évaluation de la qualité des algorithmes d'extraction, la précision, le rappel et la F-mesure sont généralement utilisés. La précision, également appelée valeur prédictive positive, est la proportion de classifications correctes où les instances sont jugées positives. Le rappel, également connu sous le nom de sensibilité, est la proportion où les positifs sont jugés positifs. La précision est utilisée pour mesurer la qualité des résultats de classification et le rappel est utilisé pour mesurer la capacité du classificateur à trouver la bonne correspondance. La F-mesure est la moyenne harmonique de la précision et du rappel, est également un indicateur d'évaluation complet. La définition de ces facteurs critères est la suivante**[2]** :

$$\text{Précision} = \frac{TP}{TP + FF} \text{Rappel} = \frac{TP}{TP + FN}$$

$$F - mésure = 2 \times \left(\frac{pécision \times Rappel}{précision + Rappel}\right)$$

TP (True Positive) : une instance positive qui devrait également être positive.

FP (False Positive) : une instance négative qui devrait être positive

FN (False Negative) : une instance positive qui est supposée être négative.

2.3.1.4 Outils d'extractions de connaissances

Au fil des années, de nombreux outils d'extraction de connaissances ont été publiés, ce qui a grandement facilité le développement de technologies en relation. **Le tableau 1** suivant regroupe certains outils d'extraction de connaissances et leurs d'utilisation. Différents outils présentent des performances et des fonctions différentes. Il est nécessaire de sélectionner l'outil approprié en fonction des tâches d'extraction de connaissances et des caractéristiques des ressources de connaissances**[2]**

Nom	Usage
Stanford NER	Extraction des entités
OpenNLP	Extraction des entités
AIDA	Extraction des entités
CiceroLite	Extraction des entités, marquage de sens, étiquetage de rôle sémantique
FOX	Extraction des entités, extraction des termes et des relations
Open Calais	Extraction des entités, extraction des faits et des relations
ReVerb	Identification et extraction des relations binaires
Wikimeta	Reconnaissance d'entités nommées multilingues et étiquetage de sens

Tableau 1:Comparaison des outils d'extraction de connaissances

2.3.2 Fusion des connaissances

L'objectif de la fusion des connaissances est de réaliser l'alignement d'entités et la construction d'ontologies, qui est un processus itératif. La construction de l'ontologie ne se terminera pas tant que les résultats de l'évaluation de la qualité ne répondront pas aux exigences.

2.3.2.1 L'alignement d'entités

L'alignement d'entités, également connu sous le nom de résolution d'entités ou d'appariement d'entités, est le processus permettant de juger si différentes entités se réfèrent ou non aux mêmes objets du monde réel.L'alignement d'entités utilise généralement une variété de techniques d'appariement d'entités combinées aux caractéristiques des graphes de connaissances pour identifier et aligner les entités qui se réfèrent aux mêmes objets. **La figure 4** montre le processus d'alignement d'entité en détail**[2]**

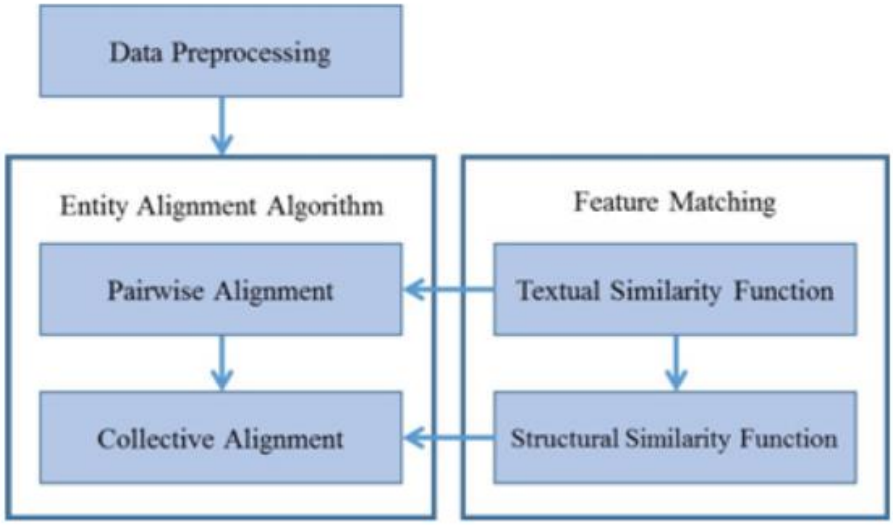


Figure 4: Le processus d'alignement des entités

Le prétraitement des données est également appelé normalisation des données,qui est une étape importante de la tâche d'alignement d'entités. Le prétraitement des données doit traiter les problèmes d'intégrité et de cohérence, tels que l'hétérogénéité de plusieurs sources, l'incohérence des définitions des données et la diversité des représentations des données.L'intégration des données dans la figure **(Figure 4)** appartient également au prétraitement des données.

L'alignement des entités comporte deux étapes : aligner par paires, puis aligner les entités collectives.La correspondance des caractéristiques est généralement utilisée pour réaliser l'alignement des entités.La mise en correspondance des caractéristiques provient d'applications NLP qui consistent en une fonction de similarité d'attribut et une fonction de relation de similarité. Quand la fonction de similarité textuelle est utilisée pour faire correspondre et comparer les attributs, la fonction de similarité structurelle est utilisée pour faire correspondre et comparer les relations. Le processus d'alignement des entités repose généralement sur des informations externes telles que des liens Wikipédia ou des informations sur les entités dans un corpus développé manuellement. Ces dernières années, l'inférence de connaissances a été proposée pour appliquer l'alignement d'entités.L'idée principale est d'utiliser des règles logiques obtenues à partir du troisième graphe ou corpus de connaissances afin d'identifier et d'aligner**[2]**

2.3.2.2 Construction et évaluation d'ontologies

Le but de la fusion des connaissances est de créer une ontologie et de construire un graphe de connaissances. À l'exception de l'entité et des relations alignées, la création d'une ontologie et d'un graphe de connaissances nécessite davantage de travaux : construction de la taxonomie et de la structure hiérarchique, ajout de métadonnées et d'autres sources de données. Afin d'assurer la qualité du graphe de connaissances, en utilisant une ontologie générale comme FOAF et les métadonnées

générales du schéma.org sont nécessaires. Si les évaluations de qualité de l'ontologie et du graphe de connaissances ne satisfont pas aux exigences, le processus de construction et de fusion du graphe de connaissances sera itéré **[2]**

2.3.3 Stockage des graphes de connaissances

Les graphes de connaissances sont généralement stockés dans des bases de données NoSQL. Il existe deux types de stockage principaux : l'un est un magasin basé sur RDF (Resource Description Framework) et l'autre utilise un magasin de base de données de graphes.

RDF est une représentation de graphes de connaissances, qui utilise triple (sujet, prédicat et objet) et IRI/URI pour décrire la structure du graphe. Le premier stockage RDF n'est pas natif : des approches basées sur des SGBD qui utilisent une triple table, une table de propriétés et un partitionnement vertical pour stocker et interroger les données RDF. Plus tard, une série de systèmes de stockage natifs ont été proposés tels que Jena2, 3store, RDF-Store, 4Store, TripleT, RDF3X, Virtuoso, etc. La plupart des systèmes de stockage natifs fournissent une requête SPARQL ou de type SPARQL. Actuellement, de nouvelles approches de stockage combinant des bases de données NoSQL et des systèmes de stockage RDF ont été proposées. Par exemple, les approches de stockage hybride telles que Jena+HBase, Hive+HBase, Cas-sandra+Sesame ont été testées et appliquées pour le grand volume de connaissances**[2]**.Certaines bases de données NoSQL comme Couchbase ont également été appliquées pour stocker des connaissances modélisées en RDF. L'avantage du stockage de graphes de connaissances basé sur RDF est que l'efficacité de la requête et de la jointure par fusion de motifs triples est bonne. Cependant, l'efficacité de la requête est améliorée par l'indexation, les meilleurs résultats de requête demandent un coût d'espace de stockage énorme.

Les bases de données de graphes sont un autre moyen important de stockage, elles stockent les nœuds, les arêtes et les propriétés des graphes. Les avantages de cette approche sont que les bases de données de graphes elles-mêmes fournissent les langages de requête de graphes parfaits et prennent en charge une variété d'algorithmes d'exploration de graphes. Cependant, le stockage distribué des bases de données graphiques pose quelques problèmes de gestion : mise à jour lente des connaissances, coût de maintenance élevé et incohérence des connaissances distribuées. La base de données de graphes typique Neo4j est populaire, c'est donc un projet open source et fournit un stockage de graphes natif.

Il n'y a pas de schéma de stockage approprié proposé pour les graphes de connaissances. Les éléments suivants sont les principales exigences pour le stockage de graphiques de connaissances à grande échelle.

- ✓ Le stockage sous-jacent doit être garanti comme étant évolutif et hautement disponible, ce qui peut utiliser des bases de données relationnelles, des bases de données NoSQL et des bases de données en mémoire.
- ✓ La segmentation des données peut être effectuée selon les besoins
- ✓ Le cache et l'index sont utilisés en temps opportun.

- ✓ Les systèmes de stockage peuvent gérer efficacement le grand volume de graphiques de connaissances.

2.3.4 Récupération et visualisation des graphes de connaissances

SPARQL est largement utilisé comme langage de requête standard du graphe de connaissances. Presque tous les systèmes de graphes de connaissances à grande échelle fournissent un point de terminaison de requête SPARQL. Il existe de nombreux types de formats de sortie de résultat de requête SPQRAL tels que JSON, JSON-LD, XML, RDF/XML, RDF/N3, CSV, TSV et HTML. Cependant, presque tous les formats de sortie sont lisibles par machine, pas par l'homme. Par conséquent, la visualisation de graphes de connaissances est l'un des sujets de recherche les plus importants. Certains formats de résultats de requête des graphes de connaissances sont basés sur du texte. La visualisation à l'aide des navigateurs sont les approches les plus courantes, comme illustré dans IsaViz, RDF Gravity, DBpedia Mobile, Fenfire et OpenLink Data Explorer. Ces deux types d'approches ont leurs propres avantages et inconvénients. Les formats de résultats de requête basés sur du texte fournissent une analyse fine du graphe de connaissances. La visualisation graphique des graphes de connaissances permet à la vue d'ensemble de naviguer et de découvrir les connaissances associées.

La récupération de connaissances est aussi une récupération sémantique. La récupération des connaissances n'est plus une simple correspondance de caractères. Il utilise généralement des règles logiques sous un modèle sémantique et un modèle d'inférence pour réaliser des récupérations, car l'ontologie est basée sur la logique de description. Par conséquent, la récupération de connaissances a la capacité de raisonnement. Actuellement, le graphe de connaissance a été largement appliqué à la recherche intelligente, aux systèmes Q/A et aux systèmes de recommandation**[2]**

Chapitre 3 : Implémentation

3 Implémentation

3.1 Introduction :

Dans ce chapitre, nous allons développer une application pour établir une solution "end-to-end" qui sertextraire un graphe de connaissances à partir des textes.

En utilisant des techniques NLP telles que la segmentation de phrases, l'analyse des dépendances.

3.2 Extraction des entités :

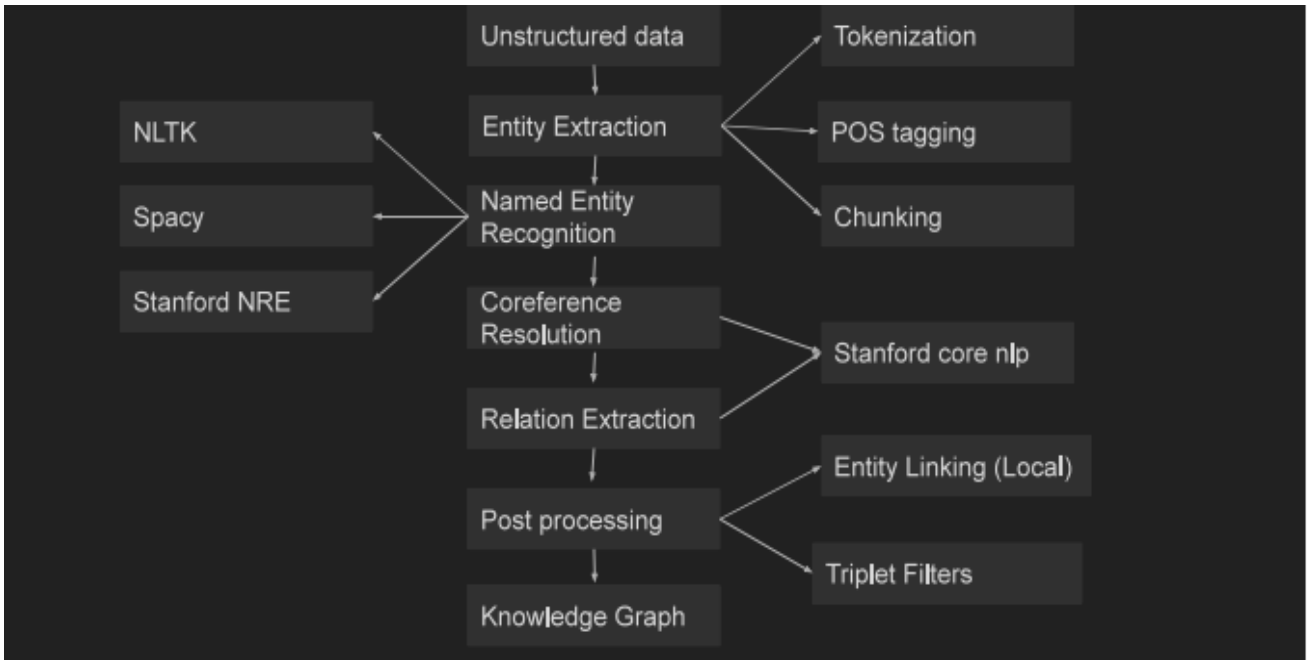


Figure 5:Présentation de processus

3.2.1 Segmentation des phrases:

La première étape de la construction d'un graphe de connaissances consiste à diviser le document texte ou l'article en phrases. Ensuite, nous ne sélectionnerons les phrases dans lesquelles il y a exactement 1 sujet et 1 objet.

Regardons un exemple de texte ci-dessous :

“Indian tennis *player*SumitNagalmoved up six places from 135 to a career-best 129 in the latestmen’s singles ranking. The 22-year-old recently won the *ATP* Challenger tournament. He made his Grand Slamdebutagainst Federer in the 2019 US OpenNagal won the first set.”

Séparons le paragraphe ci-dessus en phrases :

- Indian tennis playerSumitNagalmoved up six places from 135 to a career-best 129 in the latestmen's singles ranking
- The 22-year-old recently won the ATP Challenger tournament
- He made his Grand Slamdebutagainst Federer in the 2019 US Open
- Nagal won the first set

Sur ces quatre phrases, nous allons présélectionner la deuxième et la quatrième car chacune d'elles contient 1 sujet et 1 objet. Dans la deuxième phrase, "22-year-old "est le sujet et l'objet est "ATP Challenger tournament".Dans la quatrième phrase, le sujet est "Nagal" et "first set"est l'objet :

Sentence	Subject	Object
The 22-year-old recently won ATP Challenger tournament.	22-year-old	ATP Challenger tournament
Nagal won the first set.	Nagal	first set

Tableau 2:Sentence de segmentation

L'enjeu est de faire comprendre le texte à votre machine, notamment dans le cas d'objets et de sujets multi-mots.Par exemple, extraire les objets des deux phrases ci-dessus est un peu délicat. Pouvez-vous penser à une méthode pour résoudre ce problème?

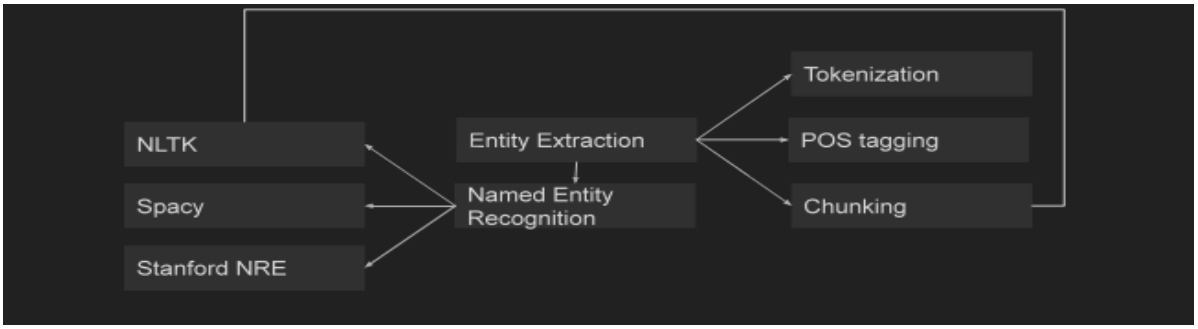


Figure 6:Reconnaissance d'entité nommée

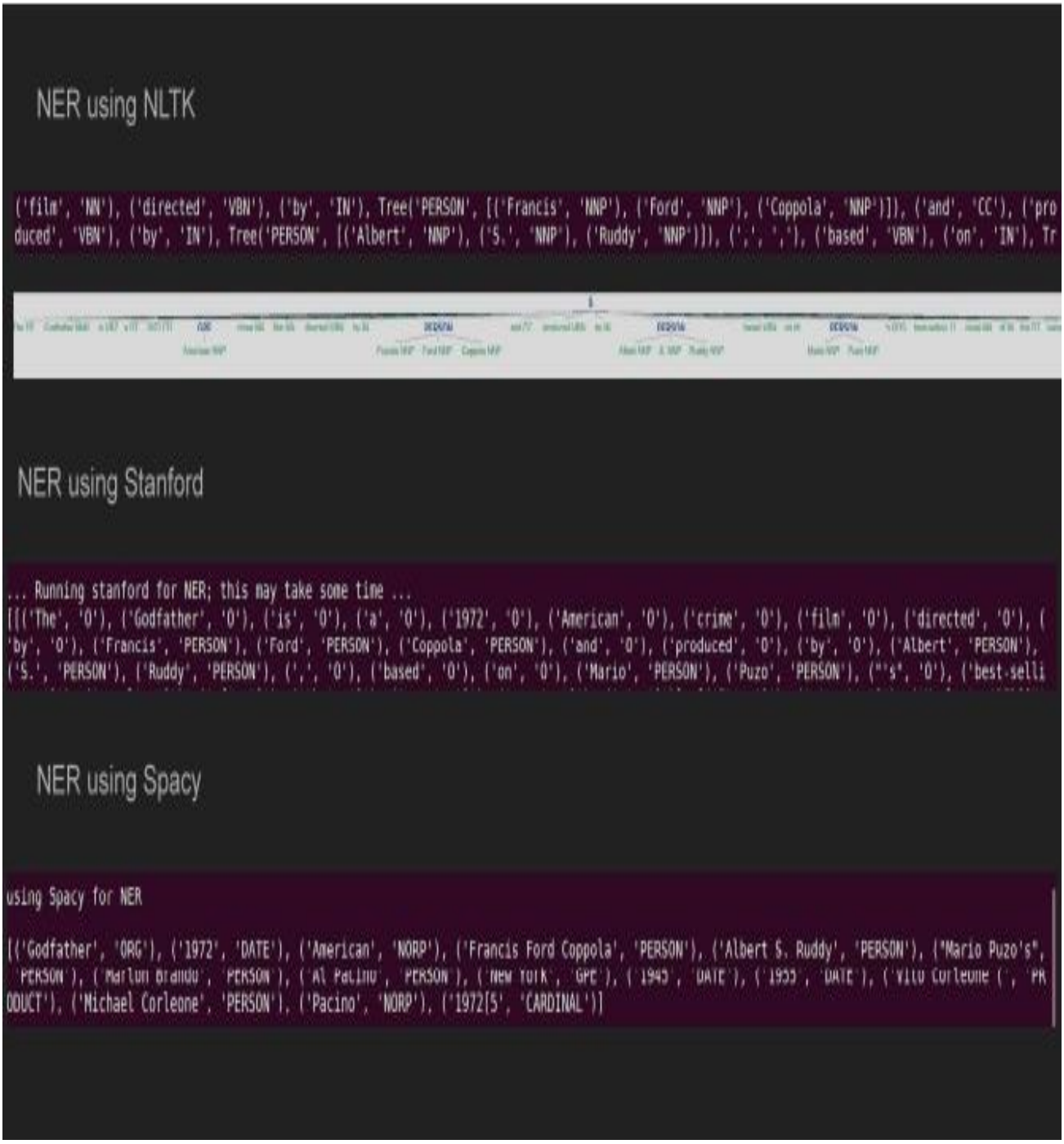


Figure 7:Reconnaissance d'entité nommée 2

Résolution de coréférence :

Résolution de coréférence

Quoi?

"D. Trumpispresident of United States. He waselected to office in 2016"

* Résoudre : He = Trump

Pourquoi?

* La plupart des textes ne font référence qu'une seule fois aux entités principales

* Différents nœuds dans l'extraction de la relation rendront cet aspect inutilisable

Comment?

* Stanford Core NLP donne des expressions de mots connexes
* Identifier la coréférence à remplacer pour chaque liste
* Utilise des entités nommées reconnues
* Déterminer quelle entité doit être remplacée dans la phrase
* Remplacer si le remplacement est valide.

3.3 Introduction à l'extraction d'informations avec Python et spaCy et nltk et stanford et verbos et obtimized :

Python est le **langage de programmation** open source le plus employé par les informaticiens. Ce langage s'est propulsé en tête de la gestion d'infrastructure, d'analyse de données ou dans le domaine du développement de logiciels. En effet, parmi ses qualités, Python permet notamment aux développeurs de se concentrer sur ce qu'ils font plutôt que sur la manière dont ils le font. Il a libéré les développeurs des contraintes de formes qui occupaient leur temps avec les langages plus anciens. Ainsi, développer du code avec Python est plus rapide qu'avec d'autres langages.

Il reste aussi accessible pour les débutants, à condition de lui consacrer un peu de temps pour la prise en main. De nombreux tutoriels sont d'ailleurs disponibles pour l'étudier sur des sites Internet spécialisés ou sur des comptes Youtube. Sur les forums d'informatique, il est toujours possible de trouver des réponses à ses questions, puisque beaucoup de professionnels l'utilisent.

Les principales utilisations de Python par les développeurs sont :

la programmation d'applications

- la création de services web
- la génération de code
- la métaprogrammation.

On différencie deux versions : Python 2 et Python 3. Python 2, l'ancienne version propose des mises à jour jusqu'en 2020. Python 3 est la version actuelle. Son interpréteur est plus efficace, ainsi que son contrôle de concurrence[7]

spaCy est une bibliothèque open source gratuite pour le traitement avancé du langage naturel (NLP) en Python.

Si vous travaillez avec beaucoup de texte, vous voudrez éventuellement en savoir plus à ce sujet. Par exemple, de quoi s'agit-il ? Que signifient les mots dans leur contexte ? Qui fait quoi à qui ? Quelles entreprises et quels produits sont mentionnés ? Quels textes se ressemblent ?

spaCy est spécialement conçu pour une utilisation en production et vous aide à créer des applications qui traitent et « comprennent » de gros volumes de texte. Il peut être utilisé pour créer des

systèmes d'extraction d'informations ou de compréhension du langage naturel, ou pour pré-traiter du texte pour un apprentissage en profondeur.

En savoir plus sur ce texte source .Vous devez indiquer le texte source pour obtenir des informations supplémentaires.

spaCy fournit une variété d'annotations linguistiques pour vous donner un aperçu de la structure grammaticale d'un texte. Cela inclut les types de mots, comme les parties du discours, et la façon dont les mots sont liés les uns aux autres. Par exemple, si vous analysez du texte, cela fait une énorme différence si un nom est le sujet d'une phrase, ou l'objet - ou si "google" est utilisé comme un verbe, ou fait référence au site Web ou à l'entreprise dans un le contexte.

Natural LanguageToolkit (NLTK) est une plate-forme utilisée pour créer des programmes Python qui fonctionnent avec des données de langage humain pour une application dans le traitement statistique du langage naturel (NLP).

Il contient des bibliothèques de traitement de texte pour la tokenisation, l'analyse, la classification, le radicalisme, le balisage et le raisonnement sémantique. Il comprend également des démonstrations graphiques et des exemples d'ensembles de données, ainsi qu'un livre de recettes et un livre qui explique les principes sous-jacents aux tâches de traitement du langage sous-jacentes prises en charge par NLTK.

StanfordNLP est la combinaison du progiciel utilisé par l'équipe de Stanford dans la tâche partagée CoNLL 2018 sur l'analyse de dépendance universelle et de l'interface Python officielle du groupe avec le logiciel Stanford CoreNLP.

StanfordNLP est une collection de modèles de pointe pré-entraînés. Ces modèles ont été utilisés par les chercheurs lors des concours CoNLL 2017 et 2018. Tous les modèles sont construits sur PyTorch.

3.3.1Extraction d'entités :

L'extraction d'une seule entité de mot à partir d'une phrase n'est pas une tâche difficile. Nous pouvons facilement le faire à l'aide de balises de parties de discours (POS). Les noms et les noms propres seraient nos entités.

Cependant, lorsqu'une entité s'étend sur plusieurs mots, les balises POS seules ne suffisent pas. Nous devons analyser l'arbre de dépendance de la phrase. Vous pouvez en savoir plus sur l'analyse des dépendances dans l'article suivant.

3.3.2 Extraction de relations :

* Relation Extraction est au centre de notre pipeline

* Ici, nous formons des triplets bruts de la forme ->« Entité-1, Relation, Entité-2 ».

* Pour y parvenir, nous utilisons un wrapper python pour Stanford OpenlE et enregistrons la sortie sous forme de fichier csv

* Comme vous pouvez le voir ci-dessous, la sortie est redondante et brute, nous effectuons donc du post-traitement

Godfather	is	1972 American crime film directed by Francis Ford Coppola
Godfather	is	1972 American crime film
Godfather	is	1972 American crime film directed
Godfather	is	1972 crime film
Godfather	is	1972 crime film directed
Mario Puzo	on	best-selling novel of same name
Godfather	is	1972 crime film directed by Francis Ford Coppola
Godfather	is	American
It	stars	Marlon Brando
It	stars Marlon Brando as	leaders of fictional New York crime family
It	stars Marlon Brando as	leaders
story	focusing on	transformation from reluctant family outsider to ruthless mafia boss
story	chronicles family under	patriarch Vito Corleone
story	focusing on	transformation of Michael Corleone from reluctant family outsider to ruthless mafia boss
story	focusing on	transformation of Michael Corleone from reluctant family outsider

Figure 8:Post-traitement

* Le format de la sortie structurée attendue (illustré ci-dessous) pour la visualisation graphique est différent de ce qui est obtenu à partir de l'extraction de relation

```
FromType, FromName, Edge, ToType, ToName
Person, John Doe, FRIENDS, Person, Emily
Person, John Doe, MEMBER OF, Organization, Acme
```

Figure 9:Le format de la sortie

* FromName et ToName représentent respectivement entity1 et entity2

* Edge représente la relation entre les deux entités nommées

Sortie de post-traitement :

PERSON	Godfather	is	GPE	1972 crime film directed by Francis Ford Coppola
ORG	Paramount Pictures	obtained	O	rights for price of $ 80
PERSON	Godfather	is	NORP	1972 American crime film directed by Francis Ford Coppola
PERSON	Godfather	is	DATE	1972 crime film directed by Francis Ford Coppola
PERSON	Jack Woltz	Rendina as	ORG	Philip Tattaglia
PERSON	Godfather	is	DATE	1972 American crime film directed
PERSON	Godfather	is	GPE	1972 American crime film directed by Francis Ford Coppola
ORG	Connie	has	ORG	husband Corleone

Figure 10:Sortie de post-traitement

3.4 Construire un graphe de connaissances :

Nous allons enfin créer un graphe de connaissances à partir des entités extraites et des

prédicats(relation entre entités).

Créons un data frame d'entités et de prédicats :

```
# extract subject
source = [i[0] for i in entity_pairs]

# extract object
target = [i[1] for i in entity_pairs]

kg_df = pd.DataFrame({'source':source, 'target':target, 'edge':relations})
```

Figure 11:data frame d'entités et de prédicats

Ensuite, nous utiliserons la bibliothèque networkx pour créer un réseau à partir de cette trame de données. Les nœuds représenteront les entités et les arêtes ou connexions entre les nœuds représenteront les relations entre les nœuds.

Ce sera un graphe orienté. En d'autres termes, la relation entre toute paire de nœuds connectés n'est pas bidirectionnelle, elle est uniquement d'un nœud à un autre. Par exemple, « John eatspasta »

```
# create a directed-graph from a dataframe
G=nx.from_pandas_edgelist(kg_df, "source", "target",
                          edge_attr=True, create_using=nx.MultiDiGraph())
```

Figure 12:Création d'un réseau

Traçons le réseau :

```
plt.figure(figsize=(12,12))


pos = nx.spring_layout(G)
nx.draw(G, with_labels=True, node_color='skyblue', edge_cmap=plt.cm.Blues, pos = pos)
plt.show()
```

Figure 13:tracer le réseau

Sortir:

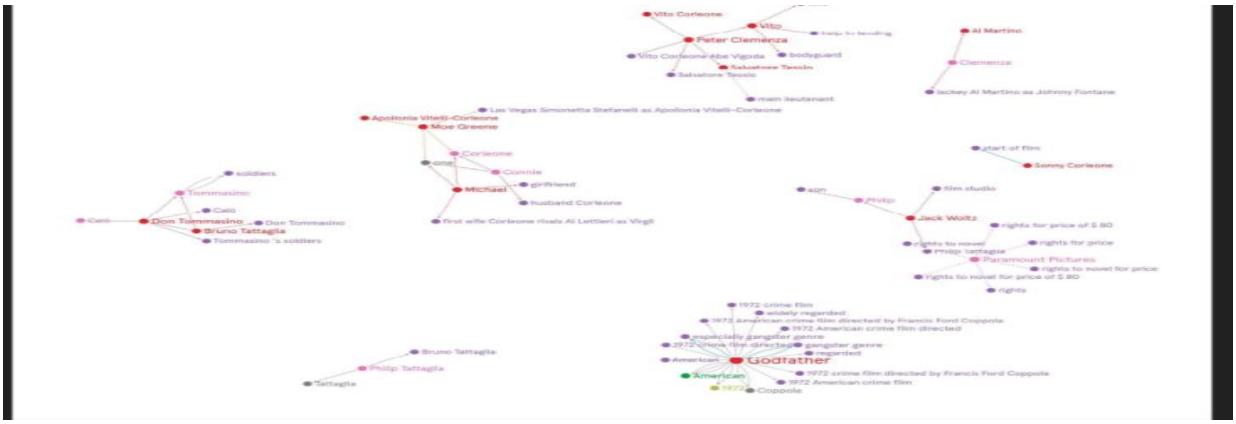


Figure 14: Sortie

3.5 NetBeans IDE :

NetBeans IDE est un environnement de développement intégré (EDI), permet également de supporter différents autres langages, comme java, C, C++, JavaScript, XML, Groovy, PHP et HTML de faon native ainsi que bien d'autres (comme Python ou Ruby) par l'ajout de greffons. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refacturant, éditeur graphique d'interfaces et de pages Web). NetBeans constitue par ailleurs une plateforme qui permet le développement d'applications spécifiques (bibliothèque Swing (Java)). L'IDE NetBeans s'appuie sur cette plateforme.

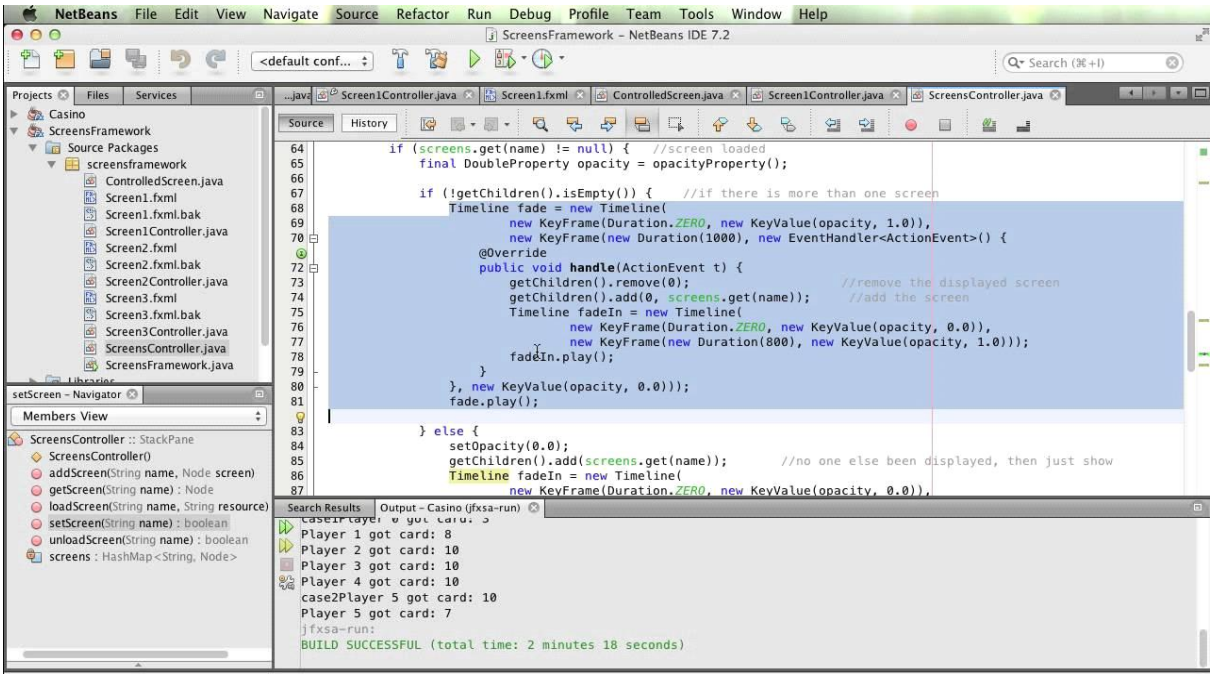


Figure 15:fenêtre de programmation Sur Netbeans

3.6 Langage de programmation (Java):

Java est un langage de programmation et une plateforme informatique qui ont été créés par Sun Microsystems en 1995. Beaucoup d'applications et des sites Web ne fonctionnent pas si Java n'est pas installé et leur nombre ne cesse de croître chaque jour. Java est rapide, sécurisé et fiable. Des ordinateurs portables aux centres de données, des consoles de jeux aux super ordinateurs scientifiques, des téléphones portables à Internet, la technologie Java est présente sur tous les fronts.

C'est un langage de programmation à usage général, évolué et orienté objet dont la syntaxe est proche du C.

Ses caractéristiques ainsi que la richesse de son écosystème et de sa communauté lui ont permis d'être très largement utilisé pour le développement d'applications de types très disparates. Java est notamment largement utilisée pour le développement d'applications d'entreprises et mobiles.

3.6.1 Environnement Java

Java est un langage interprété, ce qui signifie qu'un programme compilé n'est pas directement exécutable par le système d'exploitation mais il doit être interprété par un autre programme, qu'on appelle interpréteur.

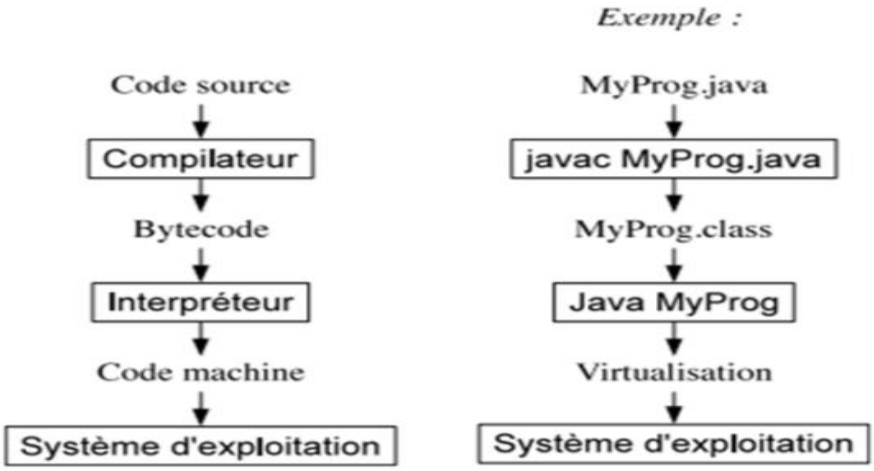


Figure 16:architecture exécutable Code java

3.6.2 JavaFX :

3.6.2.1 JavaFX Intégration :

JavaFX est une bibliothèque graphique intégrée dans le JRE et le JDK de Java. Oracle la décrit comme « The Rich Client Platform », c'est-à-dire qu'elle permet de réaliser des interfaces graphiques évoluées et modernes grâce à de nombreuses fonctionnalités, telles que les animations, les effets, la 3D, l'audio, la vidéo, etc. Elle a de plus l'avantage d'être dans le langage Java, qui permet de réaliser des architectures avec des paradigmes objet, et aussi de pouvoir utiliser le typage statique. Dans ce premier tutoriel, nous allons voir ensemble un rapide historique de la bibliothèque pour ensuite découvrir les fondamentaux qui sont les classes « Stage », « Scene », « Application » et le « threading » associé, pour finir nous verrons les « Node » avec un exemple d'utilisation du « scene graphe ». Cette présentation ne fait pas dans le bling-bling, même si JavaFX est doué pour cela, en préférant se focaliser sur les concepts primordiaux d'une telle bibliothèque.

Bien comprendre ces basiques vous aidera bien à commencer pour ensuite pouvoir faire des interfaces de qualité et peut-être spectaculaires

Figure 17:projet Java FX Main

3.7 Démonstration de l'application :

3.7.1 Interfaces de l'application :

L'exécution de notre application étais développé sous une plateforme de l'environnement NeatBeans IDE suivant les étapes qui se suit :

3.7.2 Interface d'accueil

L'utilisateur dois chargé la plateforme NeatBeans et par suite exécuter le MAIN contenant le programme appelant l'application et qui nous donne la possibilité de charger le texte en question à partir d'une bouton load file.

Après choisir le où les textes sous forme lignes et colonnes suivant le nombres de textes

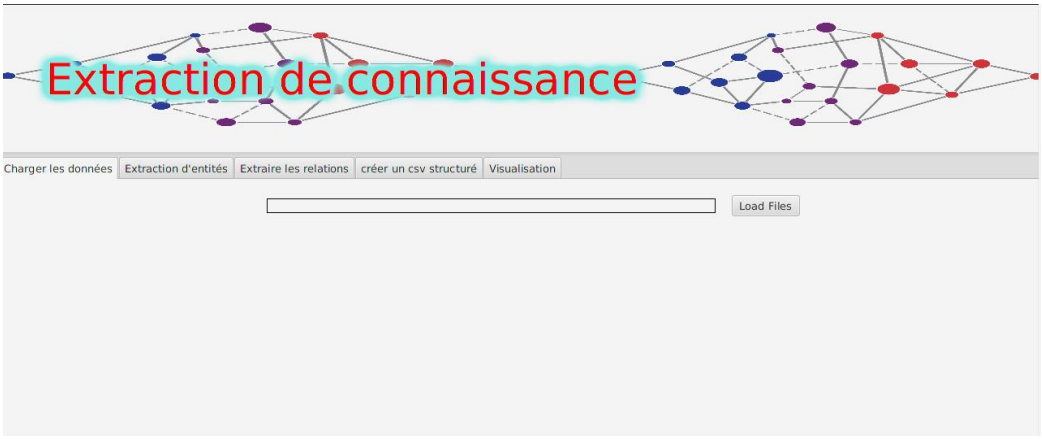


Figure 18:Interface d'accueil

Il est a signalé que cette première étape nous montre toutes les boites de dialogues nécessaire pour continué l'exécution tel que :

3.7.2.1 Extraction d'entités :

A partir du bouton « extraction d'entités » l'utilisateur dois coucher et choisirai par suite les bibliothèques Spacy, NLTK, Stanford, Verbose, Optimized déjà s'exprimer dans le contexte ci-après, Il reste juste d'appuyer sur OK

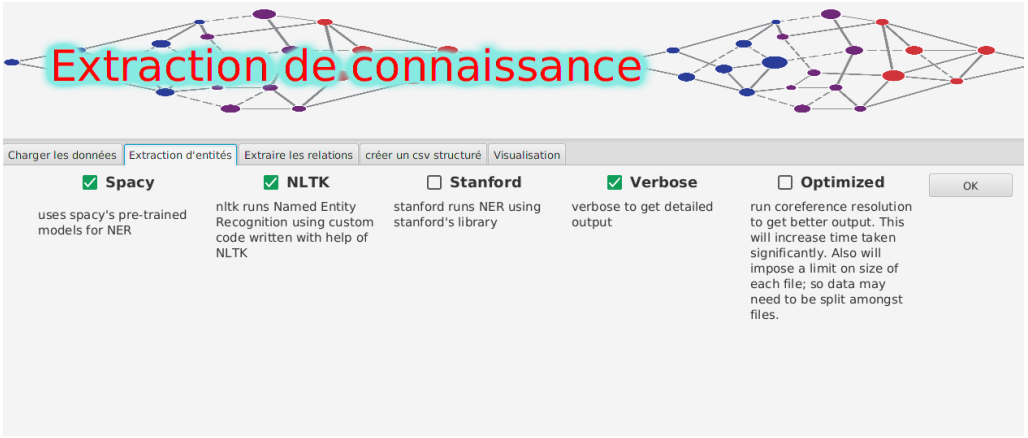
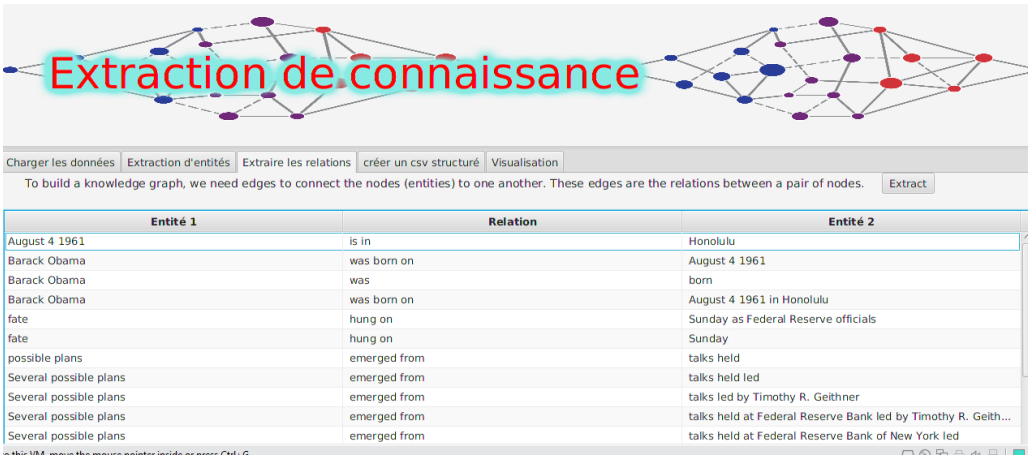


Figure 19:Extraction d'entités

3.7.2.2 Extraire les relations :

L'étape suivante aide notre programme de reconnaitre les détails du texte tel que : nom, personne, organisation…..

La boite de dialogue extraire les relations nous permettre à faire cette étape



Entité 1	Relation	Entité 2
August 4 1961	is in	Honolulu
Barack Obama	was born on	August 4 1961
Barack Obama	was	born
Barack Obama	was born on	August 4 1961 in Honolulu
fate	hung on	Sunday as Federal Reserve officials
fate	hung on	Sunday
possible plans	emerged from	talks held
Several possible plans	emerged from	talks held led
Several possible plans	emerged from	talks led by Timothy R. Geithner
Several possible plans	emerged from	talks held at Federal Reserve Bank led by Timothy R. Geith...
Several possible plans	emerged from	talks held at Federal Reserve Bank of New York led

Figure 20:Extraire les relations

3.7.2.3 Crées un csv structure :

Après avoir faire sortir les relations entre les entités, nous passons à l'étape suivante à travers la création d'un CSV qui à pour rôle de bien identifié chaque entité par type et faire sortir en détails les relations nonseulement entre ces entités lais aussi entre leurs types.

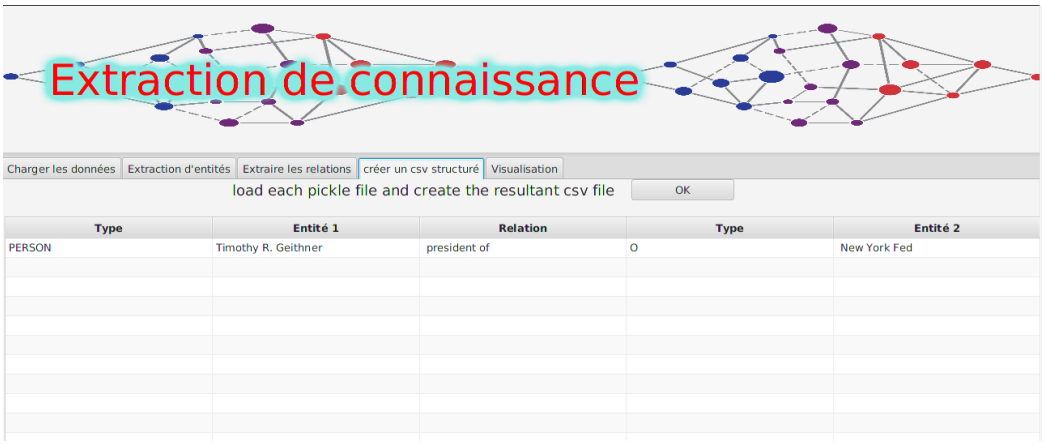


Figure 21:Crées un csv structure

3.7.2.4 Visualisation :

Toujours à partir des boites de dialogues, le bouton visualisation nous permettre d'obtenir les relations sous forme graphe :

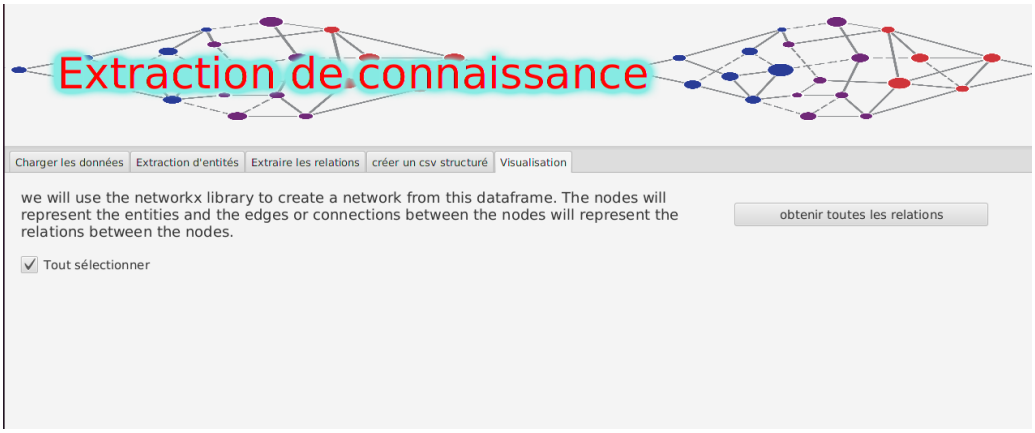


Figure 22: La visualisation

Figure 23:Le graphe de connaissance

3.8 Conclusion :

Ce chapitre a abordé l'aspect de l'implémentation de notre application. Dans ce chapitre, nous avons exposé et présenté les différentes phases suivies pour la conception et la réalisation de notre application En effet, nous avons explicité comment Construire un Graph de connaissance à partir de données textuelles Ensuite nous avons présenté les aspects techniques utilisés dans notre travail. Nous avons commencé par présenter les aspects d'implémentation qui sont utilisés dans notre travail. L'implantation repose essentiellement sur le langage JAVA avec NetBeans comme environnement de développement,

Ensuite, nous avons présenté et commenter les différentes parties de notre application. Avec les premiers tests notre système a réalisé des résultats encourageants.

Conclusion générale

Conclusion générale :

Un graphe de connaissances est essentiellement un réseau sémantique, qui est une structure de données basée sur des graphes qui stocke les connaissances sous forme de graphiques et renvoie les connaissances traitées et sensibles aux utilisateurs. Il se compose de « nœuds » et de « bords », les nœuds représentent des « entités » dans le monde réel et les bords représentent des « relations » entre les entités.

En général, les graphes de connaissances sont divisés en graphes de connaissances générales et graphes de connaissances de domaine. Parmi eux, le graphe de connaissances générales est principalement recherché par les principaux moteurs de recherche pour améliorer la précision de la recherche et s'efforcer de donner directement la réponse cible ; Alors que le graphe de connaissance du domaine peut fournir plusieurs applications cibles en fonction de la situation spécifique du domaine.

Spécifique au domaine financier, car elle inclut tous les horizons de la vie, y compris l'économie, les industries, les entreprises et bien d'autres aspects de la connaissance, la carte de la connaissance financière est relativement particulière par rapport aux autres domaines. Plus précisément, les entités impliquées dans le graphique des connaissances financières comprennent : les entreprises, les produits, les titres et les employés. La relation entre les entités comprend : la relation d'équité, la relation de travail, la relation de garantie, la relation fournisseur, la relation concurrent, la relation de production, la relation d'achat et les relations amont et aval . Parmi eux, certaines entités et relations peuvent être extraites et générées automatiquement, telles que les relations d'équité, les relations de travail, etc., et des informations générales peuvent être obtenues sur la plate-forme d'enregistrement du bureau industriel et commercial. Les relations en amont et en aval entre les produits nécessitent des sources de données systématiques, ce qui pose d'énormes défis pour l'acquisition et l'identification d'informations.

Bibliographie

Bibliographie

[1] ALEXANDRE FAURE, LA BASE DE CONNAISSANCES ELEMENT CENTRAL DE LA GESTION DES COLLECTIONS ELECTRONIQUES DES BIBLIOTHEQUES UNIVERSITAIRES, SUBMITTED ON 12 FEB 2014, P 35-36.

[2] ZHANFANG ZHAO. ARCHITECTURE OF KNOWLEDGE GRAPH CONSTRUCTION TECHNIQUES INTERNATIONAL JOURNAL OF PURE AND APPLIEDMATHEMATICS/VOLUME 118 NO. 19 2018, 1869-1883
ISSN: 1311-8080 (PRINTED VERSION); ISSN: 1314-3395 (ON-LINE VERSION)
URL: HTTP://WWW.IJPAM.EU
SPECIAL ISSUE

[3] UKSG KBART. KBART: BASES DE CONNAISSANCES ET OUTILS ASSOCIES/RAPPORT PREPARE PAR LE GROUPE DE TRAVAIL NISO/UKSG KBART JANVIER 2010 [EN LIGNE]. [S.L.] : UNITED KINGDOM SERIALS GROUP, [S.D.]. DISPONIBLE SUR : < HTTP://WWW.COUPERIN.ORG/IMAGES/STORIES/KBART/KBART_COUPERIN_FR.PDF > (CONSULTE LE 23 MAI 2013)

[4] Whatis a KnowledgeGraph? CS520
https://web.stanford.edu/~vinayc/kg/notes/What_is_a_Knowledge_Graph...

[5] HTTP://WWW.SMALSRESEARCH.BE/UN-FRAUDEUR-NE-FRAUDE-JAMAIS-SEUL-PARTIE-2/

[6] SCALESEM : MODEL CHECKING ET WEB SEMANTIQUE, MAHDI GUEFFAZ, THESE DE DOCTORAT EN INFORMATIQUE, ECOLE DOCTORAL SCIENCE POUR L'INGENIEUR ET MICROTECHNIQUES (BESANÇON ; DIJON ; BELFORT

[7]HTTPS://WWW.JOURNALDUNET.FR/WEB-TECH/DICTIONNAIRE-DU-WEBMASTERING/1445304-PYTHON-DEFINITION-ET-UTILISATION-DE-CE-LANGAGE-INFORMATIQUE/#PYTHON-DEFINITION