Sommaire:

Introduction Générale

Ce TER s'inscrit dans le cadre du projet DOREMUS-ajouter note de bas de page avec lien-, qui est un projet de recherche financé par l'ANR en partenariat avec la BnF, Philharmonie de Paris, le laboratoire d'informatique, de robotique et de microélectronique et d'autres institutions.

DOREMUS, projet touchant la recherche sur le web sémantique, a comme objectif de développer des outils et des méthodes pour publier, partager, connecter, enrichir les catalogues d'oeuvres et d'événements musicaux dans le web de données. Son objectif global est de faciliter l'accès aux données et connaissances musicales via la construction d'un modèle de connaissance commun (une ontologie).

Les données sur lesquelles nous travaillons sont sous la forme de graphe RDF parfaitement structurées et sémantisées selon le modèle musicale DOREMUS.

Dans le but de créer une relation d'équivalence entre des œuvres provenant de deux institutions différentes, plusieurs approches et méthodes ont été mises en place. Naturellement la première approche, c'est de les comparer directement puisqu'ils ont été représentés suivant le même modèle. C'est-à-dire comparer d'une part les classes, qui sont des ressources ou des données brutes, entre elles et d'autres part les propriétés ou prédicats.
Ils s'avèrent que quelques problèmes ont été rencontrés dans cette étape, parmi lesquels la comparaison des classes de type string. Ces dernières sont des données brutes, non structurées donc difficilement comparable au niveau informatique.

Notre travail consiste à extraire ces données qui représentent une description importante de l'œuvre et de les structurer. Pour y parvenir, des méthodes de reconnaissance d'entité nommée, en partie, répondent à une telle problématique. Elles consistent à reconnaître puis extraire à partir des données textuelles, les termes utilisés dans un contexte spécifique. Nous devons en choisir les plus fiables et les appliquer sur nos données afin de choisir le plus performant ou efficace.

Partant de là, notre objectif consistera à étudier *dans un premier temps* le modèle DOREMUS incluant les technologies du web sémantique ainsi que les méthodes de reconnaissances et d'extraction d'entités nommées.
Suivi de l'extraction et du traitement de ces données (nettoyage, classification, …)
Ensuite, nous présenterons le processus expérimental suivi c'est à dire le Benchmark construit pour les tests, les techniques d'évaluations des outils avant de passer à l'analyse où nous décrirons les résultats obtenus.

Les outils choisis ont été développé certains en java et d'autres en python.
L'extraction aussi se fera avec le sparql en passant par soit du java ou du python.

1. Le modèle DOREMUS

Le modele DOREMUS **est un modèle de représentation d'oeuvres musicales, basé sur le modèle FRBRoo[] et CIDOC CRM[].**
Son objectif est de fournir un modèle de référence pour leur description, et le niveau d'information que les experts et d'autres institutions du patrimoine culturel, peuvent utiliser pour décrire leurs collections, et les entités commerciales connexes, afin d'améliorer le partage de l'information.

La création de ce modèle a été une importante phase du projet DOREMUS dont l'une des principales retombées est de permettre l'interconnexion et donc l'interopérabilité des données décrivant les œuvres musicales et les événements associés dans le web des données.
Avant de décrire le modèle, nous commencerons par définir les technologies du web sémantique sur lesquelles il est construit.

1. 1. Les technologies du web sémantique

Le web d'aujourd'hui est essentiellement syntaxique, c'est-à-dire, que son contenu est destiné à être affichée et lu que par des humains. Il n'est pas fait pour être manipulé par des programmes informatiques, qui sont incapables de caractériser les informations qu'ils parcourent.

La nouvelle génération de Web – Le Web sémantique – a pour ambition de structurer les données en vue d'en donner le sens et de relier ces données d'une source à une autre. Ainsi les ressources seront plus aisément accessibles aussi bien par l'homme que par la machine.

Il s'agit d'établir un format universel d'échange d'information et un langage permettant d'organiser les silos d'information par le biais d'une grammaire (standard RDF) et d'un vocabulaire commun (ontologies).

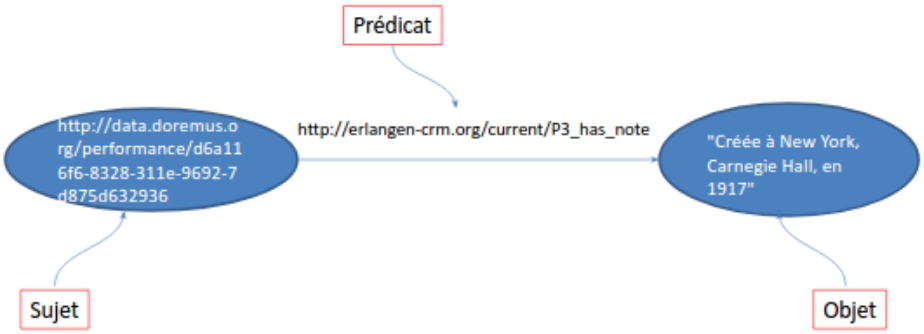
- RDF (Resource Description Framework) :

Le RDF est un langage de modélisation des données pour le Web sémantique. Toutes les informations sont stockées et représentées dans le RDF au moyen de propositions ou « triplets » de type (sujet, prédicat, objet). Le sujet représente la ressource à décrire, le prédicat une relation et l'objet une donnée ou une autre ressource : c'est la valeur de la propriété. Ainsi, avec ce format, on peut facilement indiquer qui est en relation avec quoi.

Le sujet et l'objet, dans le cas ou c'est une ressource, peuvent être identifiés par une URI ou êtres des noeuds anonymes. Le prédicat quant à lui est nécessairement identifié par une URI (Uniform Ressource Identifier), qui en plus d'identifier une

ressource, fournit un moyen de la localiser en décrivant son mécanisme d'accès principal. Le plus commun ou familier de ceux-ci étant le Universal Resource Locator (URL).

Figure 2:Graphe RDF (sujet, prédicat, objet)



Pour un accès simplifié à de telles informations, il existe un langage spécifique destinés à interroger les graphes RDF: le langage de requête SPARQL, qui possède une structure très similaire à celle employée dans le langage SQL :

SELECT * WHERE { ?sujet ?prédicat ?objet }

Si on adapte cette requête a nos données, nous aurons une requête de la forme ci-dessous :

```
SELECT *
WHERE { ?sujet
        <http://erlangen-crm.org/current/P3_has_note>  "Créée à New York, Carnegie Hall, en 1917" }
```

Les requêtes SPARQL fonctionne par correspondance du modèle de triplets dans la clause where avec les triplets du graphe RDF.
Dans cet exemple, le prédicat et l'objet du triplet sont des valeurs fixées de sorte que le modèle va correspondre seulement aux triplets avec ces valeurs. Le sujet est une variable sans aucune restriction.
Le modèle correspondant aux triplets avec ces valeurs d'objet et de prédicat, les fait correspondre avec les solutions pour toutes les variables dans ce cas ?sujet.

- Ontologie :

Pour comprendre ce que les données veulent dire, il faut passer par une composante essentielle du web sémantique, qui est un modèle ou une ontologie. Une ontologie code le sens, la signification en déclarant: des classes (des types d'objets) pour exprimer ce à quoi il est fait référence et des propriétés pour exprimer la nature des relations entre les objets ainsi définis.

Dans le modèle DOREMUS, les classes et les propriétés expriment les éléments importants dans la description des oeuvres musicales.

Le cœur du modèle est constitué de quelques composants de base : Œuvre, Expression, Événement, ainsi que quelques propriétés fondamentales permettant d'exprimer les relations de base: l'Identification (P1 est identifié par), la catégorisation (P2 est de type) et la description extensive (P3 a pour note).

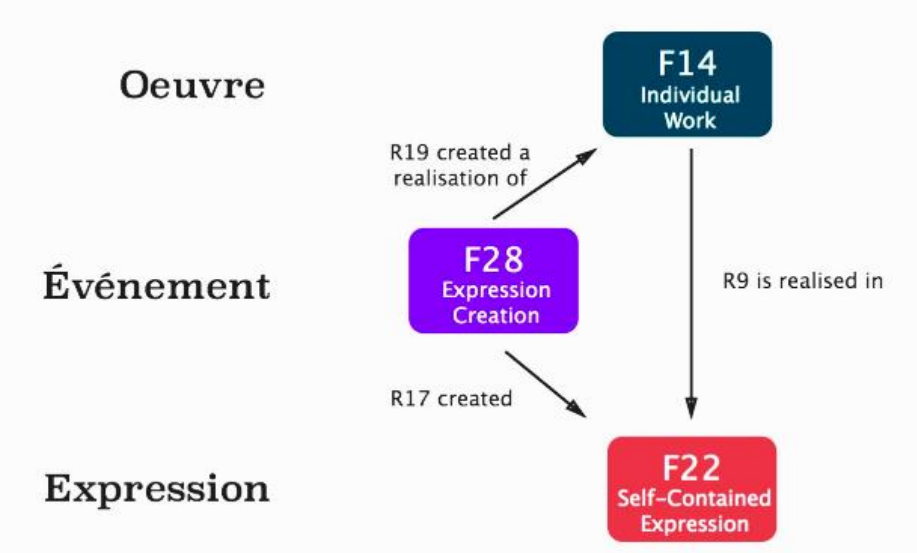


Figure 2: schéma illustrant le coeur du modèle DOREMUS.

Il présente une modélisation, depuis la création d'une œuvre jusqu'à son interprétation, son enregistrement et sa publication.

L'association des technologies du Web sémantique et la mise à disposition de ressources permettraient par exemple d'inférer une nouvelle connaissance à partir de la connaissance apportée par les ontologies.[K,Nebhi,2013] l'automatisation de ce processus d'annotation de documents pour le Web sémantique passe par l'utilisation de techniques issues du traitement automatique du langage naturel comme l'extraction d'information, la désambiguïsation ou la reconnaissance de termes. Notre étude étant centrée sur la reconnaissance d'entités nommées, nous ne définirons que celle ci.

1.2 L'extraction d'entités nommées

Après un aperçu des travaux sur la définition des technologies du web sémantique ainsi qu'une description du modèle de cette étude, nous introduisons la reconnaissance des entités dans les contenus textuels.

La reconnaissance d'entités nommées est une sous tâche du domaine d'extraction d'information consistant à rechercher des objets textuels (un mot ou des expressions) catégorisables dans des classes tels que : \subsection{La reconnaissance d'entités nommées et l'extraction de données}

Après un aperçu des travaux sur la définition des technologies du web sémantique ainsi qu'une description du modèle de cette étude, nous introduisons la reconnaissance des entités dans les contenus textuels.

La REN (reconnaissance des entités nommées) est une sous-tâche du domaine d'extraction d'information consistant à rechercher des objets textuels (un mot ou des expressions) catégorisables dans des classes tels que :
\begin{itemize}
\item noms de personnes,
\item noms d'organisations ou d'entreprises,
\item noms de lieux,
\item quantités,
\item distances,
\item valeurs,
\item dates,
\item etc.
\end{itemize}

Différentes approches existent[M.Boulaknadel et al, 2014]:
\begin{itemize}
\item L'extraction fondée sur des démarches linguistiques ou encore nommées symboliques, qui s'appuie sur l'utilisation de grammaires formelles construites (marqueurs lexicaux, des
\item dictionnaires de noms propres et parfois un étiquetage syntaxique).
\item
\item La seconde démarche fait usage de techniques statistiques ou encore dites à base d'apprentissage pour apprendre des spécifiés sur de larges corpus de textes où les entités-cibles ont été auparavant étiquetées nommés (corpus d'apprentissage), et par la suite adapter un algorithme d'apprentissage qui va permettre d'élaborer
\item automatiquement une base de connaissances à l'aide de plusieurs modèles numériques (CRF, SVM, HMM …).
\item
\item Les deux approches qu'on a cité auparavant ont fait l'apparition d'une troisième1 approche qui représente une combinaison de ses antécédents (l'approche Hybride),

elle utilise des règles écrites manuellement mais construisent aussi une partie de ses règles en se basant sur des informations syntaxiques et des informations sur le
\item discours extraites de données d'apprentissage grâce à des algorithmes d'apprentissage.
\end{itemize}

2. Etude technique

Notre objectif dans cette partie du mémoire sera d'examiner les entités nommées de type nom de person, des organisations, des lieux et des dates contenues dans un échantillon représentatif de données.

2.1 les données

Les données textuelles faisant l'objet du traitement sont identifiées dans les classes E62 de type String du modèle. Ce sont des notes ou commentaires rédigées indépendamment par différents experts du domaine et correspondent à la valeur d'une propriété P3_has_note du graphe. Cette derniere peut etre présente un peu partout dans le graphe mais la plupart d'entre elles étant déjà structurées. Nous nous intéresserons uniquement à celles des ressources de la classe F22_Self_Contained_Expression que nous avons extraite grâce à la requête SPARQL suivante :

```
PREFIX ecrm:  <http://erlangen-crm.org/current/>
PREFIX efrbroo: <http://erlangen-crm.org/efrbroo/>
SELECT * WHERE {?oeuvre a efrbroo:F22_Self-Contained_Expression;
                                    ecrm:P3_has_note ?note }
```

qui nous permet :

- de tester si la ressource est de type F22_Self-Contained_Expression,
- si elle a une propriete P3_has_note,
- et enfin d'extraire le sujet (ressource œuvre) et l'objet ou la valeur (donnée note)

2.2 Construction du benchmark

L'approche proposée pour l'évaluation des outils comporte notamment une constitution d'un corpus de référence qui nous servira de vérité de terrain pour évaluer la précision de nos outils. ce dernier a été construit à partir du résultat de la requête SPARQL ci dessus.

En apparence limpide, l'opération produit rapidement une quantité de questions et d'hésitations. Nous pourrions classer les problèmes rencontrés en trois grandes catégories décrites dans:

- Sens absolu et sens en contexte:

L’entité Faber peut être considéré comme de classe personne mais aussi une organisation.

<PERSON> Faber </PERSON>
<ORGANIZATION> Faber </ORGANIZATION>

- Frontière des entités :

ici il fallait considérer les informations importante qui se trouvent avant ou après l’entité.

dédiées \"à Madame <PERSON> Eugenia Errazuriz </PERSON>
dédiées \"à <PERSON> Madame Eugenia Errazuriz </PERSON>

- Entités imbriquées :

Dans certains cas, nous trouvions des entités imbriqué dans d’autres comme dans l’exemple suivant :

où le Vatican qui est un pays se retrouve à l'intérieur du nom d’une organisation.

<ORGANIZATION> la chapelle Giulia de Vatican </ORGANIZATION>.
<ORGANIZATION> la chapelle Giulia de <LOCATION> Vatican </LOCATION> </ORGANIZATION>.

Afin de lever toute ambiguïté, nous avons dû définir nos propres règles d’annotations pour chaque langage de programmation utilisée, et elles sont souvent divergentes entre elles.

Pour l’outil Stanford dont la librairie utilisé est en Java , nous avons du tagger chaque mots constituant l’entité:

Créé à Madrid/LOCATION, Cappella/ORGANIZATION di/ORGANIZATION San/ORGANIZATION Filippo/ORGANIZATION El/ORGANIZATION Real/ORGANIZATION, 1833/DATE (première version), et à Paris/LOCATION, Théâtre/ORGANIZATION des/ORGANIZATION Italiens/ORGANIZATION, le 7/DATE janvier/DATE 1842/DATE

Tandis que pour l’outil Polyglot dont la librairie disponible en Python, nous n’avions pas besoin de tagger chaque mots, un seul tag englobant l’entité suffisait.

Créé à <LOCATION> Madrid </LOCATION>, <ORGANIZATION> Cappella di San Filippo El Real </ORGANIZATION>, 1833 (première version), et à <LOCATION> Paris </LOCATION>, <ORGANIZATION> Théâtre des Italiens </ORGANIZATION>, le 7 janvier 1842}

L’annotation de ce corpus est accomplie de façon à fournir les informations nécessaires permettant d’évaluer la précision des outils.

Ce document de référence, comporte 223 oeuvres. Chaque oeuvre se présente selon le modèle illustré ci dessous et comporte donc :

- des informations relatives à son origine : URI
- des informations de catégorisation, sous forme de TAG ou mots-clés
- un numéro pour l'identifier au niveau de la sauvegarde
- un contenu sous forme de paragraphes.

Il convient d'observer que cette annotation a été réalisée de façon manuelle sur le contenu brut du résultat de l'extraction. Les résultats de cette première annotation ont ensuite été corrigés et complétés par ajout des éléments non repérés. Cette opération a été accomplie par deux personnes et validé par une troisième.

2.3 Approches d'extraction d'entités

Après avoir réalisé une étude comparative des différents outils NER, nous avons été amené à faire un choix des méthodes qu'on va tester parmi plusieurs existantes dans la littérature.

Le choix des outils retenus s'est fait par tris successifs. L'examen des principales applications citées dans la littérature, ainsi que de quelques répertoires d'APIs, nous a permis d'identifier une dizaine de services. un examen plus approfondi nous a conduit à en garder 6, que nous avons testés sommairement sur un échantillon de texte. Enfin, nous en avons retenu quatre selon les critères suivants :

- taux et qualité de reconnaissance manifestement raisonnables
- une offre gratuite minimale non limitée dans le temps.
- désambiguïsation des entités nommées basée sur des ressources exploitables et existence.
- Reconnaissance d'entité en langue française.
- Disponibilité des ressources et de la documentation ….

2.3.1 Polyglot-NER :

a - Principe de base:

Polyglot est un outil d'extraction d'entité nommé de langage naturel pour 40 langues principales. Au lieu de s'appuyer sur des connaissances spécialisés, c'est à dire, un ensemble de données annotés par l'human ou des ressources linguistiques

spécifiques - comme treebanks, corpus parallèles, et les règles orthographiques-, il est construit sur une approche d'apprentissage automatique (Word embedding). Cette approche est basée sur la quantification et cartographication des similitudes sémantiques entre les éléments linguistiques en fonction de leurs propriétés distributionnelles dans les données.

Cet outil utilise des modèles de corpus formés sur des ensembles de données extraites automatiquement à partir de Wikipédia pour entraîner le système et récupérer les étiquettes utilisées pour l'annotation. Ce dernier permet de reconnaître trois différentes étiquettes :

- **Emplacements** (Tag: `I-LOC`): villes, pays, régions, continents, quartiers, les divisions administratives ...
- **Organisations** (Tag: `I-ORG`): les équipes sportives, les journaux, les banques, les universités, les écoles, organismes sans but lucratif, les entreprises, ...
- **Personnes** (Tag: `I-PER`): les politiciens, les scientifiques, les artistes, les athlètes ...

b. Caractéristique de Polyglot :

L'outil Polyglot est conçu pour être très flexible et extensible. il intègre de nombreux outils de détection de la langue (196 langues), Tokenisation (165 langues), la reconnaissance d'entités nommées (40 langues), l'étiquetage morpho-syntaxique (part-of-speech tagging) (16 langues),l'analyse de Sentiment (136 langues), plongement lexical (137 langues), l'analyse morphologique (135 langues), et la translittération (69 langues).

c. Test avec l'outil Polyglot:

Un test facile pour la précision d'un outil d'extraction d'entité nommé est de comparer les entités extraites par les outils aux extractions annoté à la main. Avant de commencer, nous profitons de la fonctionnalité de chaque outil pour obtenir et catégoriser les noms de personnes, de lieux et d'organisations dans des classes.

Le script python parcours le corpus ligne par ligne, identifie les entités et les étiquettes en 3 catégories :

(`I-LOC` ([u'Madrid'])) (`I-ORG` ([u'Cappella'])) (`I-PER` ([u'Filippo','El', 'Real']))

2.3.2 Stanford :

a - Description

Stanford Named Entity Recognizer (NER) implémente en Java des modèles

de reconnaissance basés sur des champs conditionnels aléatoires linéaires (CRF). Ces modèles ont été entraînés sur les corpus de dépêches de ConLL 2003, MUC-6 et 7 et une partie d'ACE 2002. Ils peuvent détecter selon les cas de trois (Location, Person, Organization) à sept types d'entités (Location, Person, Organization, Money, Percent, Date, Time). D'autres modèles sont aussi disponibles selon la langue. il peut également être employé seul ou avec les systèmes Apache Tika, UIMA ou encore avec la bibliothèque Python NLTK.

b -Caractéristiques

- 3 Models de classification : 3 classe , 4 classe et 7 classe.
- Disponibles pour plusieurs langues : anglais, chinois, allemand, francais ...etc
- L'option désambiguïsation est non disponible.
- Format de sortie : slashtags, tabentities, xml, inlineXML, highlighted (interface)
- Il intégre une bonne gamme d'outils d'analyse grammaticale.
- Interfaces disponibles pour la plupart des principaux langages de programmation modernes.
- Possibilité de fonctionner comme un simple service Web(demo online)
- Possibilité de créer des modèles personnalisées.

2.3.3 TextRazor :

a- Description

TextRazor est un outil d'extraction d'entité nommée et de désambiguïsation qui réalise la reconnaissance d'entité en tirant parti d'une vaste base de connaissance extraite de diverses sources Web, y compris Wikipedia, DBPedia et Wikidata. son objectif est d'extraire et comprendre le qui, quoi, pourquoi et comment des contenus et dispose d'un dictionnaire de millions d'entités différentes possibles, que nous pouvons rechercher rapidement dans un texte à l'aide du moteur correspondant.

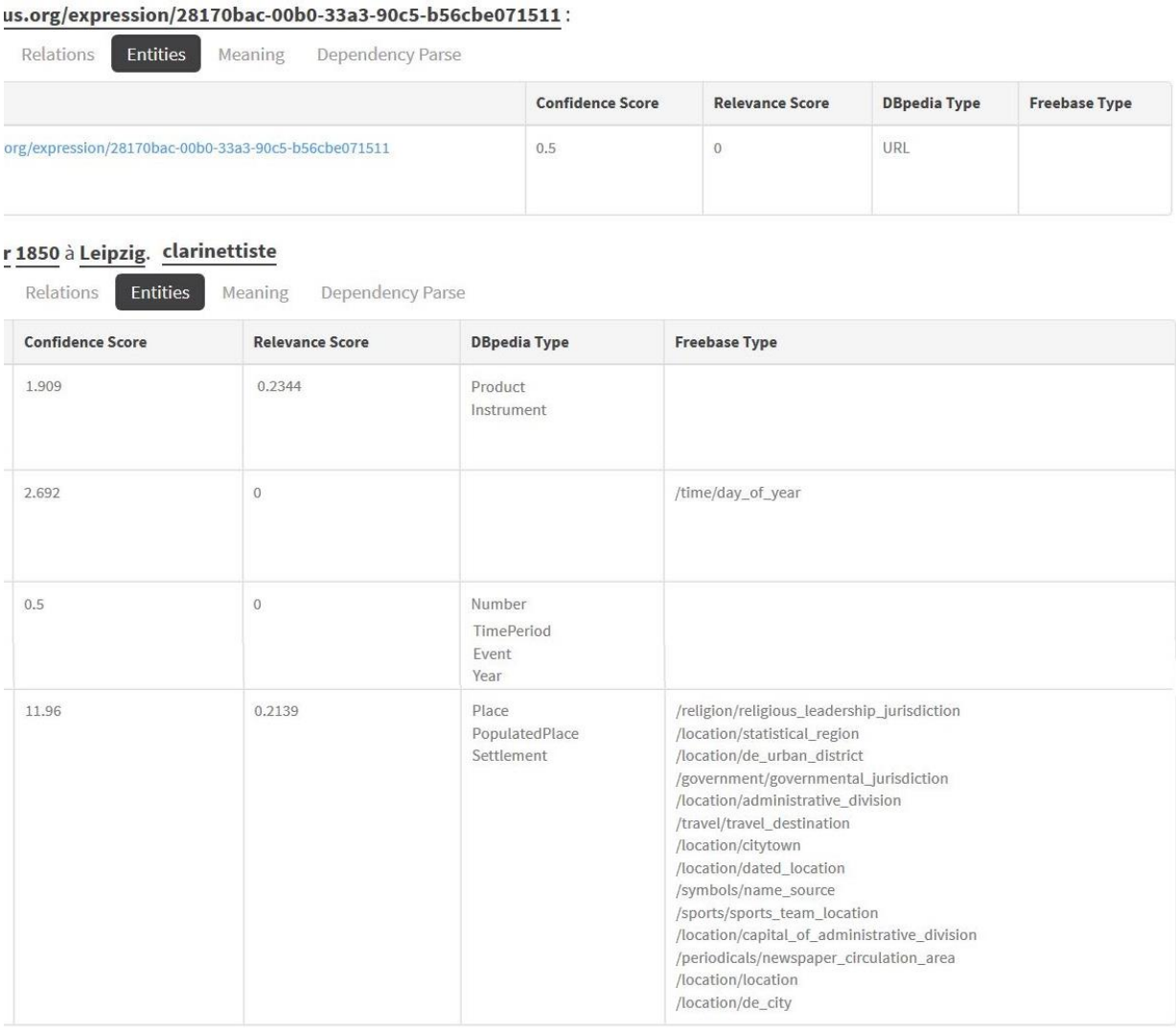
il utilise également un tagger statistique pour identifier les personnes, les lieux et les entreprises qui n'ont jamais été mentionnés auparavant et des expressions régulières pour repérer les mentions moins ambiguës telles que les adresses e-mail et les sites Web.
L'API de TextRazor peut être facilement intégrée à n'importe quelle langue qui peut envoyer une requête HTTP et analyser la réponse JSON, ce qui rend possible l'analyse de texte puissante avec seulement quelques lignes de code.

b- Caractéristiques

- L'extraction de la relation, et entailment.

- Enrichit des entités avec des informations telles que les données de localisation et les dates de naissance.
- Détecte automatiquement 142 langues et fournit la reconnaissance de l'entité et la détection de sujet pour 10 langues, dont l'anglais, l'espagnol, l'allemand, le français et le russe etc …
- Renvoie les réponses au format JSON, peut être consultée sur HTTP ou HTTPS, et prend en charge la compression GZIP facultative.
- les librairies officielles sont fournies pour Python, PHP et Java.
- Gratuite limitée à 500 requêtes par jour,
- modèle et classificateur personnalisé
- Près de 1500 types d'entités provenant de la BD wikidata y compris person/person, organisation/organisation, location/location ….
- Plus de 300 types d'entités provenant de la BD DBpedia y compris Person, Organisation, Place, Populated Place, Time, Year etc ….

demo :



us.org/expression/28170bac-00b0-33a3-90c5-b56cbe071511 :

Relations | Entities | Meaning | Dependency Parse

	Confidence Score	Relevance Score	DBpedia Type	Freebase Type
org/expression/28170bac-00b0-33a3-90c5-b56cbe071511	0.5	0	URL	

r 1850 à Leipzig. clarinettiste

Relations | Entities | Meaning | Dependency Parse

Confidence Score	Relevance Score	DBpedia Type	Freebase Type
1.909	0.2344	Product Instrument	
2.692	0		/time/day_of_year
0.5	0	Number TimePeriod Event Year	
11.96	0.2139	Place PopulatedPlace Settlement	/religion/religious_leadership_jurisdiction /location/statistical_region /location/de_urban_district /government/governmental_jurisdiction /location/administrative_division /travel/travel_destination /location/citytown /location/dated_location /symbols/name_source /sports/sports_team_location /location/capital_of_administrative_division /periodicals/newspaper_circulation_area /location/location /location/de_city

CATEGORIES

0.59 arts, c
entert
0.57 arts, c
entert
0.53 arts, c
entert
style>

TOPICS

1.00 Classi
1.00 Music
0.87 Comp
0.83 Classi
0.77 Classi
0.75 Male c
0.67 Male c
0.64 Perfor
0.63 Berlin
0.62 Rober
0.60 Comp
0.59 Arnold
0.59 Opera
0.58 Musici
0.58 People
0.56 Piano
0.56 Jewish
0.55 Berlin
0.55 Musica
0.54 Vocal
0.54 Classi
0.54 Austri

2.3.4 OpenNLP-NER

OpenNLP-NER est un outil d'extraction d'entité nommé de langage naturel (Traitement du langage naturel) pour 5 langues principales, basée sur des algorithmes d'apprentissage automatique (l'entropie maximale; maxEnt et Perceptron).ce sont une forme de régression logistique multinomiale qui utilise pour construire son modèle. De plus, cette bibliothèque a des modèles pré-construits pour certaines langues et des ressources textuelles annotées.
OpenNLP fournit des composants pour aborder des tâches PNL spécifiques telles que la tokenisation, segmentation de la phrase, marquage de la partie de la parole, l'extraction de l'entité nommée, découpage, l'analyse et la résolution de coréférence. Les composants peuvent être combinés pour créer un pipeline de traitement PNL.

2.4 Mesures d'évaluation des systèmes de REN :

L'évaluation des différentes méthodes d'extraction de connaissances fondée sur des corpus de référence utilise généralement des métriques qui permettent de mesurer la distance entre un ensemble de réponses correctes (la référence) et les hypothèses des systèmes de NER. Cette pratique vise à déterminer si un système offre un comportement attendu, en observant uniquement ses sorties.

La précision (P) et le rappel (R) sont les mesures les plus utilisées en évaluation des systèmes d'extraction d'informations. Définis à l'origine pour l'évaluation de la recherche documentaire [Salton et Buckley, 1988], ils sont applicables à toute tâche visant à identifier des éléments pertinents parmi un ensemble d'éléments candidats :

$$\mathbf{P} = \frac{VP}{(VP + FP)} \textbf{ et } \mathbf{R} = \frac{VP}{(VP+FN)}$$

La précision est donnée par le ratio entre les réponses correctes (VP) et toutes les réponses données par l'outil. Il permet d'estimer la fiabilité des hypothèses fournies par le système.

Alors que le rappel est donné par le ratio entre les réponses correctes et toutes les réponses attendues. Il permet d'estimer la capacité de l'outil à couvrir l'ensemble des réponses se trouvant dans le corpus de référence (Benchmark).

Aucune des deux métriques ne peut être considérée comme une métrique complète, puisque la formule de la précision ne prend pas en compte les erreurs de suppression, et que la formule du rappel ne prend pas en compte les erreurs

d'insertion. C'est leur moyenne harmonique, la F-mesure, qui est utilisée afin de comparer les performances des systèmes entre eux. sa formule est comme suit:

$$F = 2 \times \frac{P+R}{P \times R}$$

3. Implementation

Les formules de précision et rappel pouvant être traduites d'une autre manière par :

$$\mathbf{P} = \frac{VP}{(VP+FP)} = \frac{intersection\ (toutes\ les\ réponses\ données\ par\ l'outil\ ,toutes\ les\ réponses\ attendues)}{toutes\ les\ réponses\ données\ par\ l'outil}$$

$$\mathbf{R} = \frac{VP}{(VP+FN)} = \frac{intersection\ (toutes\ les\ réponses\ données\ par\ l'outil\ ,toutes\ les\ réponses\ attendues)}{toutes\ les\ réponses\ attendues}$$

Nous utiliserons les objets de type collections en java et les dictionnaires en python pour implémenter cette partie du code. Le choix du langage étant conditionné par les librairies et ou packages disponibles de l'outil.

4. Analyse

Les résultats obtenus sont synthétisés, dans les tableaux suivant:

Polyglot-NER	PRECISION	RECALL	FSCORE
PER	0,48	**0,92**	**0,60**
LOC	0,84	**0,94**	**0,89**
ORG	0,55	**0,93**	**0,69**

stanford3	PRECISION	RECALL	FSCORE
PER	**0,81**	**0,81**	**0,80**

LOC	0,84	0,94	0,89
ORG	0,76	0,76	0,76

stanford4	PRECISION	RECALL	FSCORE
PER	0,66	0,78	0,69
LOC	0,66	0,66	0,65
ORG	0,55	0,57	0,55

Stanford-7	PRECISION	RECALL	FSCORE
PER	0,74	0,68	0,69
LOC	0,68	0,69	0,67
ORG	0,66	0,65	0,66
DATE	0,94	0,93	0,94

Textrazor	PRECISION	RECALL	FSCORE
PER	0,51	0,46	0,47
LOC	0,63	0,62	0,62
ORG	0,81	0,81	0,81
DATE	0,46	0,46	0,46

OpenNLP	PRECISION	RECALL	FSCORE
PER	0.65	0.57	0.61
LOC	0.67	0.51	0.58
ORG	0.32	0.23	0.29
DATE	0.57	0.45	0.56

L’analyse des trois outils d’extraction d’entités selon nos critères d’évaluation nous a permis de choisir le TextRazor [2] car il répondait au mieux à nos attentes. Vous trouverez dans les tableaux ci-dessous les résultats obtenus.

Cette évaluation a pour objectif de vérifier la pertinence de l’algorithme d’extraction de relations à l’aide d’un corpus textuel et d’une classe de types de relations que nous avons définies. Nous voulons vérifier si l’algorithme détecte et retourne le type de relation correct en premier, c’est à dire avec le score de pertinence le plus élevé. Pour cela, nous avons utilisé la mesure de précision décrites précédemment. Tout d’abord, le calcul des statistiques est effectué sur le premier type de relation détecté pour chacune des relations (Top-1). Puis il est effectué sur les deux premiers types de relations détectés (Top-2) et ainsi de suite jusqu’à (Top-5).
Sur la Figure ci-dessous, nous constatons que lorsque l’utilisateur vérifie uniquement le premier type de relation détecté, celui-ci est précis à près de 80%. Tandis qu’en arrivant au Top-5, on remarque que 100% des types de relations détectés sont corrects.

Cette évaluation a pour objectif de vérifier la pertinence des outils d’extraction à l’aide d’un corpus textuel que nous avons définies. Nous voulons vérifier si l’outil détecte et retourne le type d’entité correct en premier, c’est à dire avec le score de pertinence le plus élevé. Pour cela, nous avons utilisé la mesure de précision, rappel et leurs moyennes harmonique décrites précédemment.

L’analyse par oeuvre montre des résultats correctement analysées et d’autres générant un fort taux d’échec. Le tableau suivant présente les meilleurs résultats obtenus pour chaque catégorie. Les catégories de person, location et date, par exemple, sont des catégories analysées avec une F-mesure supérieure à 80 %.

Meilleurs outils	PRECISION	RECALL	F-mesure
PER - Stanford-3	0,81	0,81	0,80
LOC- Polyglot	0,84	0,94	0,89
ORG- Stanford-3	0,76	0,76	0,76
DATE- Stanford-7	0,94	0,93	0,94

On constate également que Stanford est, de loin, les meilleures méthodes d'extraction. Cependant, ce résultat doit être nuancé car on s'aperçoit que pour les nom de lieux, la méthode Polyglot dépasse Stanford.

5. La partie conduite de projet:

5.1 Organisation du travail

Nous avons décomposé notre travail en cinq phases qui représentent le cycle de vie du projet :
Dans la phase 1, nous avons identifié le périmètre du projet. ce qui nous a permis de comprendre le contexte du travail et de faire des recherches sur les éléments clés du sujet.
Ensuite, nous avons formulés le projet (Phase 2), c'est à dire, définir les travaux à réaliser, préciser les objectifs à atteindre et les stratégies incluant les conclusions de l'étude de faisabilité à savoir le choix des langages de programmation, la durée d'apprentissage des technologies du web sémantique etc...
Pour la Phase de planification, nous avons identifié et ordonnancé les tâches à réaliser. Par la suite, nous avons déterminé les profils nécessaires à leur réalisation, en prenant en considération les compétences de chacune. Cela nous a permis d'établir un plan d'action permettant de déterminer les séquencements et le parallélismes de l'exécution des tâches précédemment identifiées. Nous avons aménagé une marge de flexibilité pour l'ordonnancement de chaque tâche en prévision des risques qu'on pourrait rencontrer dans l'évolution du projet. De plus, nous avons rajouté des contraintes, c'est à dire qu'on a associé à chaque tâche les dates au plus tôt et les dates au plus tard de l'exécution de la tâche.
Enfin, nous avons modélisé le réseau de dépendance entre tâches sous forme graphique (Planning GANTT).

Ce travail nous a permis d'acquérir certaines compétences en gestion de projet. Nous avons pu atteindre nos objectifs malgré les difficultés qu'on a pu rencontrer. il faudra reconnaître que c'est un projet ambitieux donc une bonne organisation était nécessaire. Nous avons décomposé la problématique en sous problèmes et estimé la durée pour chaque étape avec des ratio issus de l'expérience (la méthode de répartition proportionnelle). Au fur et à mesure que le projet évoluait, nous ajustions le planning en fonction de nos avancements. Vu que nous avons évalué les risques, et prévu des marges de flexibilité. nous avons pu maîtriser les désagréments rencontrés.

6 - Conclusion :

Le but de ce projet était de trouver un outil performant pour l'extraction de données. Pour cela, nous devrions mettre en place un programme permettant d'extraire des entités dans des documents textuels.

Pour mener à bien cette mission, nous avons tout d'abord construit notre jeu de données (corpus de référence). Ensuite, nous avons mené nos expérimentations en utilisant le corpus avec les quatres méthode choisies préalablement : TextRazor, Polyglot, Stanford et OpenNLP.

Les expérimentations menées sur les différentes méthodes séparément sur le jeu de données, nous ont permis de voir le comportement de celles ci sur des textes non structurés. Nous avons conclu que la méthode Stanford est la plus adaptée car elle permet de reconnaître la plupart des entités des trois catégories: nom de personnes, organisations et date. Et que Polyglot peut être aussi utilisée dans le cas des nom de lieux car elle a un rappel et une précision très élevés.

À la suite d'une série d'expérimentations, nous avons alors choisi le Stanford-7 polyglot et l'avons appliqué sur toutes les données DOREMUS.

Ce TER nous a permis de nous familiariser avec le monde de la recherche. Grâce à ce projet nous avons acquis un minimum d'expérience sur la recherche d'informations dans des documents textuels et des connaissances dans les technologies du web sémantique.

7 - Références :

Stanford Inconvenient

l’outil utilise des tags pour marquer la classe d’appartenance d’une entité. Ainsi les tokens n’appartenant à aucune classe vont être identifiés par les caractères « /O ». Ne fournissant pas de méthode pour afficher les entités individuellement et leur classe d’appartenance, nous avons fait appel aux expressions régulières pour extraire le texte qui se trouvent devant chaque tag correspondant à une entité. Par l’occasion on a pu remarquer que les mots pertinents précédés d’un signe de ponctuation sans espace ne seront pas bien parsés.
Par exemple pour ce texte « (Viktor Klemperer) » on aura comme output : « (/OViktor/PERSON Klemperer/PERSON)/O »
et le résultat avec le pattern regex sera « OVictor Klemperer » donc faudra bien prendre en compte la présence de certains caractères de ponctuation.