Projet Web Sémantique

Sujet 1 :

Extraction d’un graphe de connaissances à partir d’un texte en langage naturel

Présenté par **Thivagini SUGUMAR, Nadine AL HAJJ, Lynda ZAIDI, Mohamed OUMEZZAOUCHE**

Encadrée par **Zoubida KEDAD**

6 Mars 2024

PLAN

- Introduction
- Description de la méthode implémentée
- Evaluation de la méthode implémentée
- Comparaison de méthodes
- Conclusion

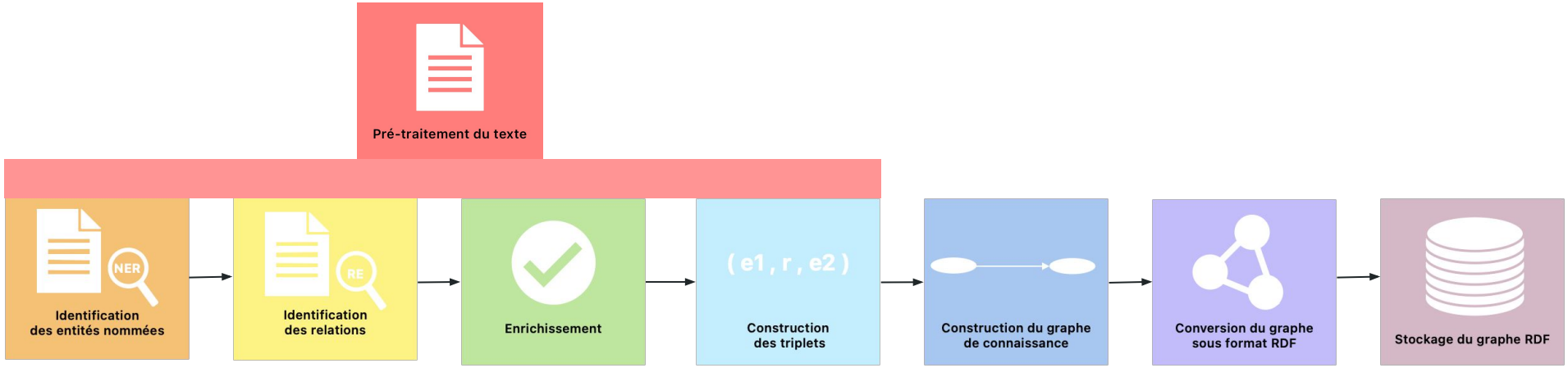
INTRODUCTION

- La montée en puissance des données non structurées sur le Web et dans divers domaines de l'information a suscité un besoin croissant de méthodes et d'outils permettant de traiter et d'exploiter ces données de manière efficace. Dans ce contexte, les **technologies du Web sémantique et de l'analyse du langage naturel** jouent un rôle crucial en permettant de donner un sens et une structure aux données non structurées.

- L'objectif principal de ce projet est de **présenter notre approche d'extraction de graphes de connaissances à partir de textes non structurés**, ainsi que de discuter d'autres méthodes disponibles dans la littérature. Nous mettrons en avant les **bases théoriques, les techniques d'extraction d'entités nommées, et la modélisation des relations entre celles-ci**, tout en explorant les applications concrètes de cette approche dans divers domaines.

DESCRIPTION

DESCRIPTION GÉNÉRALE DE LA MÉTHODE UTILISÉE



IDENTIFICATION DES ENTITÉS NOMMÉES

- Librairies SpaCy

```
i=0
entites = []
label = []
doc = nlp(texte)

# Ajouter les entités avec Spacy
for ent in doc.ents:
    ent_text = ent.text
    ent_label = ent.label_
    entites.append(ent_text)
    label.append(ent_label)
    i+=1
```

Alice PER est née à Paris LOC le 10 mai 1990. Elle a étudié l' Informatique LOC à l' Université de Stanford ORG . Après avoir obtenu son diplôme, elle a travaillé pour une entreprise de technologie à Washington LOC . Actuellement, elle réside à New York LOC . Pendant des années, Alice PER a acquis une réputation dans le domaine de la technologie et a été reconnue pour son expertise. Cependant, elle a ressenti le besoin de changement et a décidé de déménager sur la côte Est LOC , à New York LOC . Elle consacre son temps à découvrir de nouveaux quartiers, à se immerger dans la diversité culturelle de la ville et à se impliquer dans des projets communautaires. Aujourd'hui, Alice PER continue de repousser les limites dans le domaine de la technologie, tout en restant attachée sa quête de connaissances. Elle a également commencé à s'impliquer activement dans la communauté technologique locale, participant à des conférences. Sa passion pour le mentorat lui a également conduite à rejoindre des initiatives visant à encourager les jeunes talents à poursuivre une carrière dans les STEM (Science, Technology MISC , Engineering MISC , Mathematics PER). Jean PER , un brillant médecin rencontré lors d'une conférence internationale sur les nouvelles technologies à Paris LOC , est devenu un ami proche d' Alice au fil du temps MISC , partageant avec Alice PER une passion commune pour l'innovation et la recherche. Jean PER est né à Lyon LOC le 15 juin 1985. Leur amitié florissante était tissée de conversations stimulantes sur les avancées technologiques et médicales, nourrissant ainsi leur esprit de curiosité et d'exploration.

```
*** Il y a 20 entités nommées. ***
['Alice', 'Paris', 'Université de Stanford', 'San Francisco', 'New York', 'Alice', 'Californie', 'côte Est', 'New York', 'New York', 'Alice', 'Alice', 'New York', 'Alice', 'STEM', 'Science', 'Technology', 'Engineering', 'Mathematics', 'Alice']
['PER', 'LOC', 'ORG', 'LOC', 'LOC', 'PER', 'LOC', 'LOC', 'LOC', 'LOC', 'PER', 'PER', 'LOC', 'PER', 'MISC', 'ORG', 'ORG', 'ORG', 'ORG', 'PER']
```

IDENTIFICATION DES ENTITÉS NOMMÉES

- Règles heuristiques

```
doc = nlp(" ".join([token.lemma_ for token in doc if not token.is_stop and not token.is_punct]))

# Ajouter les entités avec des règles heuristiques
for token in doc:
    ent_text = token.lemma_
    ent_label = None

    # Ajouter la catégorie pour les jours
    if token.like_num and token.nbor(1).text.lower() in ["janvier", "février", "mars", "avril", "mai", "juin", "juillet", "août", "septembre", "octobre",
        ent_label = 'DAY_OF_YEAR'

# Ajouter la catégorie pour les mois
if token.text.lower() in ["janvier", "février", "mars", "avril", "mai", "juin", "juillet", "août", "septembre", "octobre", "novembre", "décembre"]:
    ent_label = 'MONTH_OF_YEAR'

# Ajouter la catégorie pour les années
if token.like_num and len(token.text) == 4:
    ent_label = 'YEAR'
```

```
if ent_label:
    entites.append(ent_text)
    label.append(ent_label)
    i+=1
```

IDENTIFICATION DES ENTITÉS NOMMÉES

- Règles heuristiques

```python
# Ajouter la catégorie pour les domaines d'étude
if token.i > 0 and token.nbor(-1).text.lower() == "étudier":
    ent_label = 'STUDY_DOMAIN'
```

```
*** Il y a 42 entités nommées. ***
['Alice', 'Paris', 'Université de Stanford', 'San Francisco', 'New York', 'Alice', 'Californie', 'côte Est', 'New York', 'New York', 'Alice', 'Alice', 'New York', 'Alice', 'STEM', 'Science', 'Technology', 'Engineering', 'Mathematics', 'Alice', '10', 'mai', '1990', 'informatique', 'diplôme', 'entreprise', 'technologie', 'solide réputation', 'technologie', 'leadership', 'succès professionnel', 'expertise', 'besoin changement', 'équilibr carrière', 'nouveau quartier', 'projet communautaire', 'technologie', 'éducation', 'technologie', 'activement communauté technologique local', 'idée startup', 'santé numérique']
['PER', 'LOC', 'ORG', 'LOC', 'LOC', 'PER', 'LOC', 'LOC', 'LOC', 'LOC', 'PER', 'PER', 'LOC', 'PER', 'MISC', 'ORG', 'ORG', 'ORG', 'ORG', 'PER', 'DAY_OF_YEAR', 'MONTH_OF_YEAR', 'YEAR', 'STUDY_DOMAIN', 'GAIN', 'NON_REAL_ENTITY', 'DOMAIN', 'GAIN', 'DOMAIN', 'QUALITY', 'QUALITY', 'QUALITY', 'FEELINGS', 'GAIN', 'RESEARCH', 'OCCUPATION', 'DOMAIN', 'DOMAIN', 'DOMAIN', 'OCCUPATION', 'RESEARCH', 'DOMAIN']
```

IDENTIFICATION DES RELATIONS

- Les relations extraites sont représentées par les verbes identifiés.

	repousser
naître	rester
étudier	attacher
obtenir	commencer
travailler	impliquer
résider	participer
acquérir	rejoindre
reconnaître	viser
ressentir	encourager
décider	poursuivre
déménager	rencontrer
consacrer	devenir
découvrir	partager
immerger	naître
impliquer	tisser
continuer	nourrir

VALIDATION DES FAITS

- Validation des entités par Wikidata

```
L'entité 'Informatique' est valide sur Wikidata.
Description et propriété :
  Label: computer science
  Description: study of computation
  ID Wikidata: Q21198
```

- Validation des relations par Wikidata

```
la relation 'naître' est valide sur Wikidata.
Description et propriété :
  Label: A Child Is Born
  Description: book by Lennart Nilsson
  ID Wikidata: Q3059379
```

ANNOTATION DES ENTITÉS NOMMÉES

- **Ontologies**

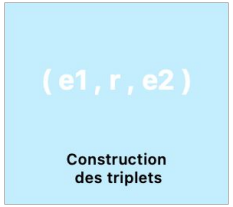
- Utilisation de l'**API de Geonames et Wikidata**
  - pour extraire des informations sur les entités dans le texte
- Pour chaque **entité de type LOC** détectée :
  - Interrogation de l'API de géolocalisation
  - Récupération du nom, du pays (si disponible), de la latitude et de la longitude
- Pour les **autres types d'entités** :
  - Interrogation de l'API de Wikidata
  - Récupération d'information générale
- Stockage des informations dans un **fichier JSON**
- Utilisation de ces informations pour enrichir la compréhension du texte, en tant que propriétés pour le graphe

```
Processing Entity 2: Paris (Label: loc)
Information for Paris:
Name: Paris
Country: France
Latitude: 48.85341, Longitude: 2.3488
```

```
Résultats pour l'entité : Alice
Alice : prenom feminin
```

```
{
    "Alice": "prenom feminin",
    "Paris": {
        "description": "Paris (France)",
        "latitude": "48.85341",
        "longitude": "2.3488"
    },
```

CONSTRUCTION DES TRIPLETS



- **Prétraitement du texte**
  - Gestion des coréférences

AVANT: (**elle**, étudier, Informatique)

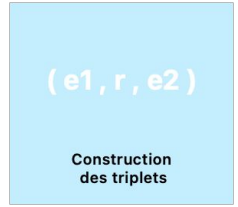
```
Alice est née à Paris le 10 mai 1990. Elle a étudié l'Informatique à l'Université de Stanford. Après avoir obtenu son diplôme, elle a travaillé pour une
```

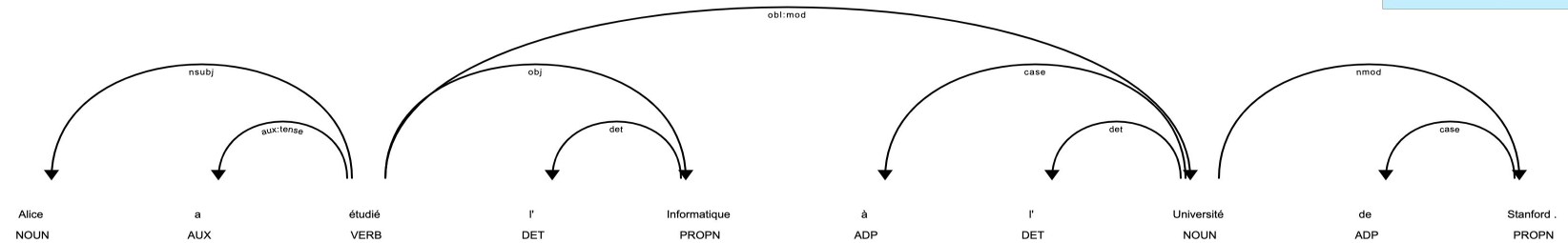
APRES: (**Alice**, étudier, Informatique)

```
Alice est née à Paris le 10 mai 1990 . Alice a étudié l' Informatique à l' Université de Stanford . Après avoir obtenu son diplôme , Alice a travaillé pour une
```

CONSTRUCTION DES TRIPLETS

- Analyse grammaticale



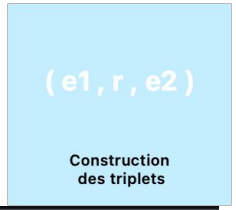



```
1 Alice a étudié l ' Informatique à l ' Université de Stanford .
2 nsubj aux:tense ROOT det case obj case det det obl:mod case nmod punct
3 NOUN AUX VERB DET DET PROPN ADP DET DET NOUN ADP PROPN PUNCT
4 étudié étudié étudié Informatique Informatique étudié Université Université Université étudié Stanford Université étudié
```

1. **token.text** : texte tokenisé (coupé mot par mot)
2. **token.dep_** : relation de dépendance grammaticale du token par rapport à son token parent
3. **token.pos_** : catégorie grammaticale du token (POS)
4. **token.head.text** : texte du token parent du token actuel

CONSTRUCTION DES TRIPLETS

- Extraction du triplet (sujet, verbe, objet)



Sujet

```
# Recherche du sujet
if token.head.text==verbe and "nsubj" in token.dep_:
    sujet=token.text
    print("[S]: "+sujet)
```

Verbe

```
# Verbe trouvé : à mettre à l'infinitif
elif token.text==verbe:
    verbeInfinitif=token.lemma_+"_1"
    if verbeInfinitif not in listeRelations:
        listeRelations.append(verbeInfinitif)
    else:
        match = re.search(r'\d+$', verbeInfinitif)
        num = int(match.group())
        verbeInfinitif = verbeInfinitif[:match.start()] + str(num + 1)
        listeRelations.append(verbeInfinitif)
    print("[V]: "+verbeInfinitif)
```

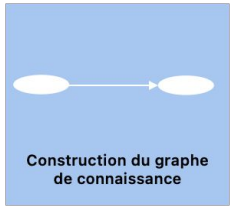
Objet

```
# Recherche de l'objet
elif token.head.text==verbe and ("obl" in token.dep_ or "obj" in token.dep_):
    # Cas d'un seul objet pour un (sujet,verbe)
    if nbObjets == 0:
        objet=token.text
        print("[O]: "+objet)
        nbObjets+=1
```

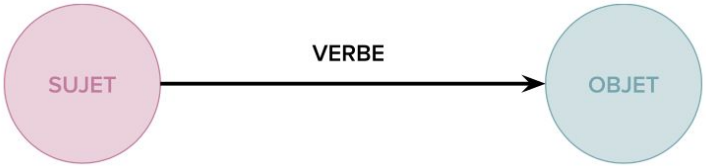
(sujet, verbe, objet)

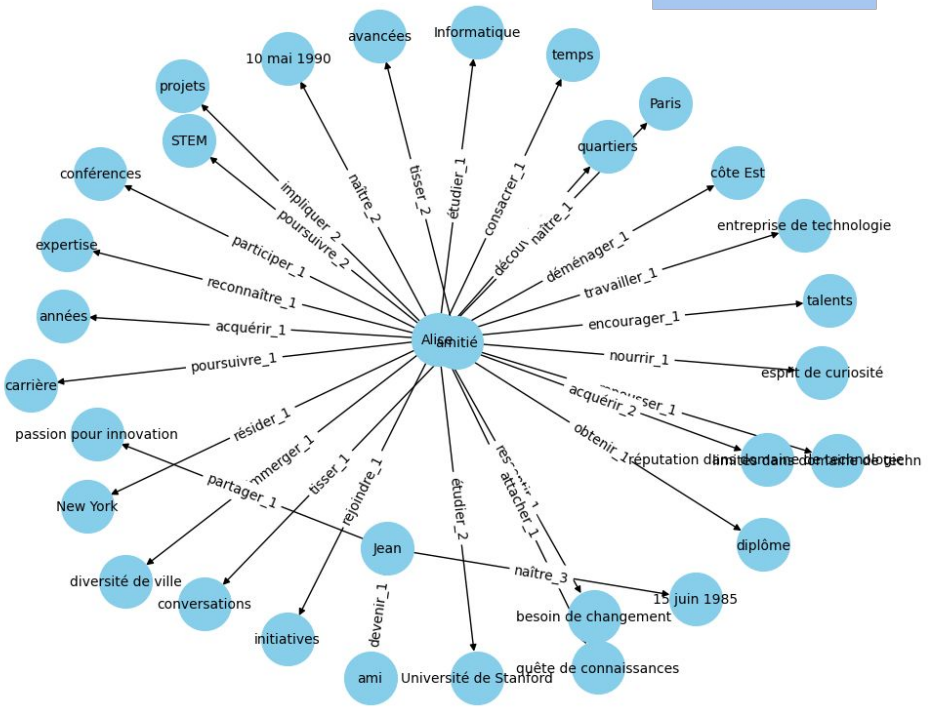
```
2 : ('Alice', 'reconnaître_1', 'expertise')
3 : ('Alice', 'immerger_1', 'diversité de ville')
4 : ('Alice', 'ressentir_1', 'besoin de changement')
5 : ('Alice', 'décider_1', 'New York')
6 : ('Alice', 'repousser_1', 'limites dans domaine de technologie')
7 : ('Alice', 'étudier_2', 'Université de Stanford')
8 : ('Alice', 'étudier_1', 'Informatique')
9 : ('Alice', 'naître_2', '10 mai 1990')
10 : ('Alice', 'obtenir_1', 'diplôme')
11 : ('amitié', 'tisser_1', 'conversations')
12 : ('Alice', 'acquérir_2', 'réputation dans domaine de technologie')
13 : ('Alice', 'naître_1', 'Paris')
14 : ('Jean', 'naître_3', '15 juin 1985')
15 : ('Alice', 'attacher_1', 'quête de connaissances')
16 : ('amitié', 'tisser_2', 'avancées')
17 : ('Jean', 'partager_1', 'passion pour innovation')
18 : ('Alice', 'consacrer_1', 'temps')
19 : ('Alice', 'impliquer_1', 'projets')
20 : ('Alice', 'participer_1', 'conférences')
21 : ('Alice', 'résider_1', 'New York')
22 : ('Alice', 'encourager_1', 'talents')
23 : ('Alice', 'rejoindre_1', 'initiatives')
24 : ('Alice', 'impliquer_2', 'projets')
25 : ('Alice', 'déménager_1', 'côte Est')
26 : ('Alice', 'acquérir_1', 'années')
27 : ('Alice', 'poursuivre_2', 'STEM')
28 : ('Alice', 'travailler_1', 'entreprise de technologie')
29 : ('Alice', 'poursuivre_1', 'carrière')
30 : ('amitié', 'nourrir_1', 'esprit de curiosité')
31 : ('Jean', 'devenir_1', 'ami')
```

GRAPHE DE CONNAISSANCE

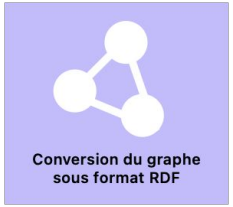


(sujet, verbe, objet)





GRAPHE SOUS FORMAT RDF/XML



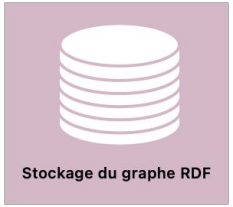
Alice

Jean

amitié

```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF
   xmlns:ns1="http://www.semanticweb.org/thivani/ontologies/2024/1/"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
>
  <rdf:Description rdf:about="http://www.semanticweb.org/thivani/ontologies/2024/1/untitled-ontology-4Alice">
    <ns1:untitled-ontology-4découvrir_1>quartiers</ns1:untitled-ontology-4découvrir_1>
    <ns1:untitled-ontology-4reconnaître_1>expertise</ns1:untitled-ontology-4reconnaître_1>
    <ns1:untitled-ontology-4immerger_1>diversité de ville</ns1:untitled-ontology-4immerger_1>
    <ns1:untitled-ontology-4ressentir_1>besoin de changement</ns1:untitled-ontology-4ressentir_1>
    <ns1:untitled-ontology-4décider_1>New York</ns1:untitled-ontology-4décider_1>
    <ns1:untitled-ontology-4repousser_1>limites dans domaine de technologie</ns1:untitled-ontology-4repousser_1>
    <ns1:untitled-ontology-4étudier_2>Université de Stanford</ns1:untitled-ontology-4étudier_2>
    <ns1:untitled-ontology-4étudier_1>Informatique</ns1:untitled-ontology-4étudier_1>
    <ns1:untitled-ontology-4naître_2>10 mai 1990</ns1:untitled-ontology-4naître_2>
    <ns1:untitled-ontology-4obtenir_1>diplôme</ns1:untitled-ontology-4obtenir_1>
    <ns1:untitled-ontology-4acquérir_2>réputation dans domaine de technologie</ns1:untitled-ontology-4acquérir_2>
    <ns1:untitled-ontology-4naître_1>Paris</ns1:untitled-ontology-4naître_1>
    <ns1:untitled-ontology-4attacher_1>quête de connaissances</ns1:untitled-ontology-4attacher_1>
    <ns1:untitled-ontology-4consacrer_1>temps</ns1:untitled-ontology-4consacrer_1>
    <ns1:untitled-ontology-4impliquer_1>projets</ns1:untitled-ontology-4impliquer_1>
    <ns1:untitled-ontology-4participer_1>conférences</ns1:untitled-ontology-4participer_1>
    <ns1:untitled-ontology-4résider_1>New York</ns1:untitled-ontology-4résider_1>
    <ns1:untitled-ontology-4encourager_1>talents</ns1:untitled-ontology-4encourager_1>
    <ns1:untitled-ontology-4rejoindre_1>initiatives</ns1:untitled-ontology-4rejoindre_1>
    <ns1:untitled-ontology-4impliquer_2>projets</ns1:untitled-ontology-4impliquer_2>
    <ns1:untitled-ontology-4déménager_1>côte Est</ns1:untitled-ontology-4déménager_1>
    <ns1:untitled-ontology-4acquérir_1>années</ns1:untitled-ontology-4acquérir_1>
    <ns1:untitled-ontology-4poursuivre_2>STEM</ns1:untitled-ontology-4poursuivre_2>
    <ns1:untitled-ontology-4travailler_1>entreprise de technologie</ns1:untitled-ontology-4travailler_1>
    <ns1:untitled-ontology-4poursuivre_1>carrière</ns1:untitled-ontology-4poursuivre_1>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.semanticweb.org/thivani/ontologies/2024/1/untitled-ontology-4Jean">
    <ns1:untitled-ontology-4naître_3>15 juin 1985</ns1:untitled-ontology-4naître_3>
    <ns1:untitled-ontology-4partager_1>passion pour innovation</ns1:untitled-ontology-4partager_1>
    <ns1:untitled-ontology-4devenir_1>ami</ns1:untitled-ontology-4devenir_1>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.semanticweb.org/thivani/ontologies/2024/1/untitled-ontology-4amitié">
    <ns1:untitled-ontology-4tisser_1>conversations</ns1:untitled-ontology-4tisser_1>
    <ns1:untitled-ontology-4tisser_2>avancées</ns1:untitled-ontology-4tisser_2>
    <ns1:untitled-ontology-4nourrir_1>esprit de curiosité</ns1:untitled-ontology-4nourrir_1>
  </rdf:Description>
</rdf:RDF>
```

STOCKAGE DU GRAPHE RDF



- Stockage du graphe

GraphDB

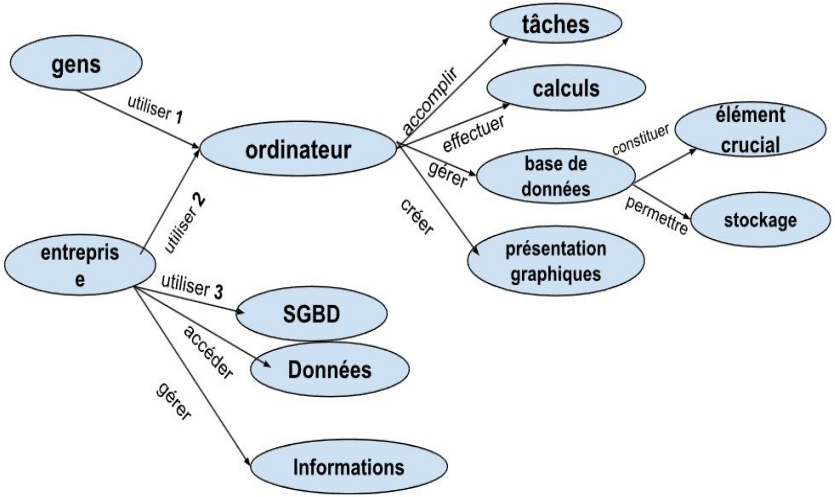
- Interrogation de la base de données via des requêtes SPARQL

```
SELECT ?person ?date_naissance
WHERE {
    ?person <http://www.semanticweb.org/thivani/ontologies/2024/1/untitled-ontology-4naître_2> ?date_naissance .
}
```

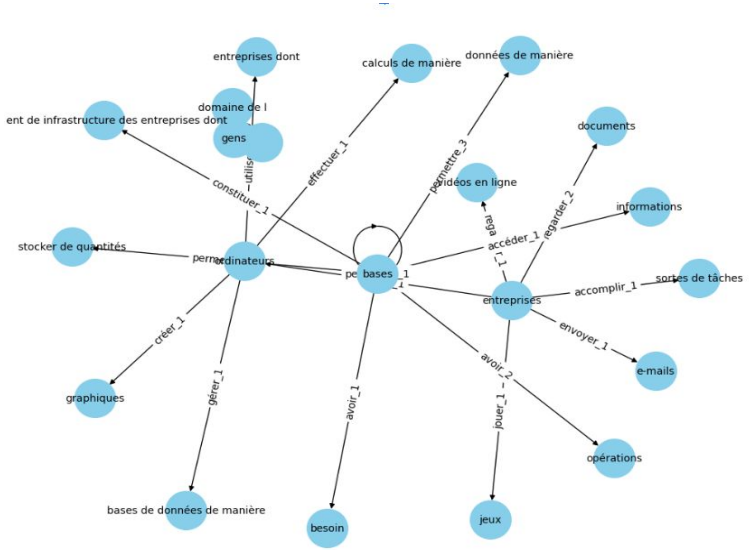
	person	date_naissance
1	http://www.semanticweb.org/thivani/ontologies/2024/1/untitled-ontology-4Alice	10 mai 1990

EVALUATION

EVALUATION



Graphe fait à la main



Graphe fait par notre méthode

COMPARAISON DE METHODES

ARTICLE D1-1

- **Problématique** : Construction de graphes de connaissances à partir de textes non structurés

- **Solution** : plateforme dslr
  - Extraction des entités et relations par stanford coreNLP à partir de documents stockés dans Apache Solr
  - Enrichissement à partir de Wikidata: faits de haute qualité pour compléter le graphe de connaissances.
  - vérification des faits réalisée par l'alignement des sous graphe ,comparer des relations extraites avec les faits de Wikidata à l'aide de requêtes.

- **Comparaison avec notre approche :**
  - Points commun :utilisation de bibliothèques de NLP, l'annotation des entités et la validation avec Wikidata.
  - Règles heuristiques pour l'extraction des entités et des relations / Extraction automatisée
  - Validation manuelle / Enrichissement automatique

ARTICLE D1-3

- **Problématique** : Construction d'un graphe de connaissances à partir des documents non structurés dans le domaine juridique en Inde

- **Solution** : Méthode basée sur des règles
  - Sélection du jeu de données
  - Prétraitement de données : logiciel GATE
  - Extraction des NER et ER : règles en JAPE et NyOn
  - Construction de triplets pour former un graphe de connaissances
  - Chargement & interrogation dans un triple store utilisant SPARQL

- **Comparaison avec notre approche** :
  - Reprise des étapes principales de l'article
  - Logiciel GATE / librairie SpaCy
  - Différentes utilisations des ontologies (NyOn/GeoNames)

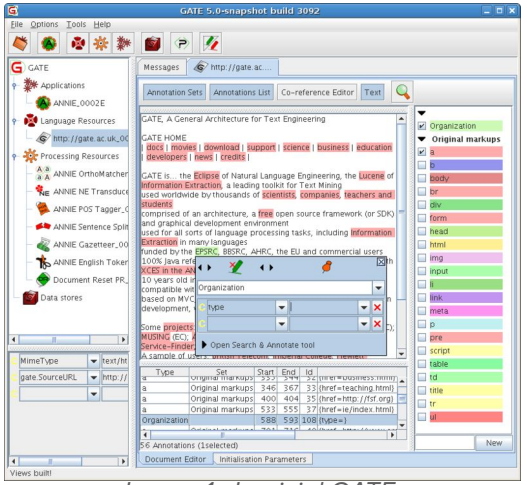


Image 1 : Logiciel GATE

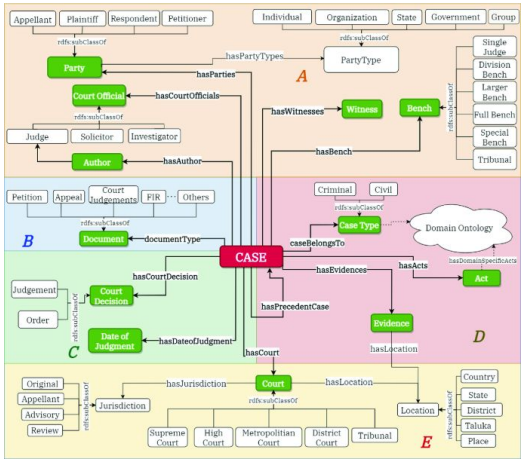


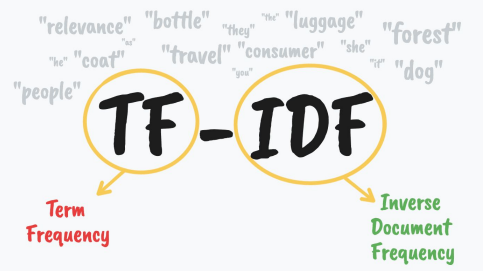
Image 2 : Ontologie NyOn

ARTICLE D2-1

- **Problématique** : Construction de graphes de connaissances à partir de textes non structurés dans l'éducation à la cybersécurité

- **Solution :** Méthode pour la construction des graphes de connaissances
  - Acquisition des connaissances.
  - Collecte de données.
  - Intégration de connaissances.
  - Développement de l’ontologie.
  - Correspondance des entités.
  - Stockage de connaissances :
    - Neo4J comme système de gestion de bases de données

ARTICLE D3-2



- Méthode automatique pour construire une hiérarchie de concepts de domaine à partir d'un corpus textuel.
  - Pour aider à la construction d'ontologies et à l'indexation sémantique des collections de documents.
- Importance des stop word et du stemming.
- Avantages de l'utilisation de la C-Value ainsi que le poids “tf.idf”
  - pour sélectionner les termes candidats dans le processus de construction d'ontologies de domaine.
- **Principe** : la sélection des termes candidats dans le processus d'extraction de termes
  - 2 types de sélections :
    - Les termes candidats idéaux doivent servir à discriminer entre les documents pertinents et non pertinents lors de la requête de la collection (évalué grâce au poids “tf.idf”).
    - Les bons termes candidats pour une ontologie de domaine doivent également être des termes fréquents, car ces termes fréquents sont généralement représentatifs d'un domaine (évalué grâce à la valeur C-Value).
- En conclusion, l'utilisation d'un schéma de pondération spécifique et la structuration des termes peuvent être appliquées indépendamment de la langue, ce qui rend la méthode adaptable à différents contextes linguistiques.

CONCLUSION

- **Conclusion** : Bien que notre méthode ait démontré des résultats prometteurs, les évaluations ont souligné la nécessité d'améliorations pour augmenter la précision et la pertinence des résultats.

- **Suggestion** : se concentrer sur des domaines plus spécifiques
  - enrichir les entités et les relations en fonction de contextes restreints,
  - conduire à l'obtention de graphes de connaissances plus précis et plus pertinents.

QUESTIONS ?