Recherche par mots clés dans un graph RDF

Web semantique

Table of contents

01

Methode naive

Problématique

Etant donné un graph rdf et un ensemble de mots clés on veut extraire les ressources, propriétés et valeurs qui match le mieux avec ces mots clés et ensuite construire un graph à partir des éléments qui match.

Concepts

Recherche par mot clé

Extraction des noeuds et propriétés qui correspondent le mieux avec les mots clés

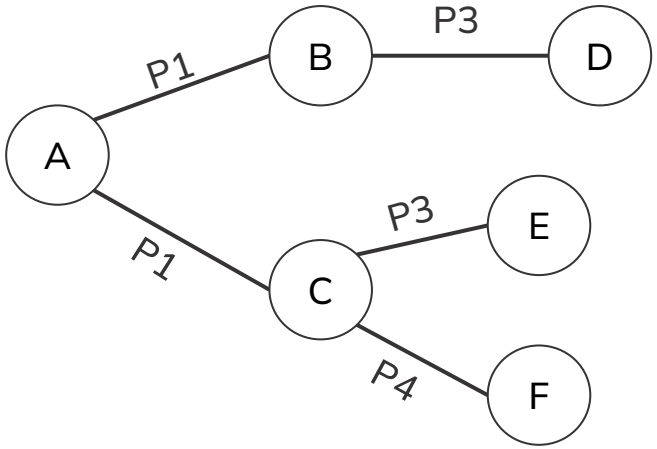
Construction du graph

Construire un graph qui réunit les éléments extrait dans l'étape 1 en se basant sur le graphe de base

Recherche par mots clés

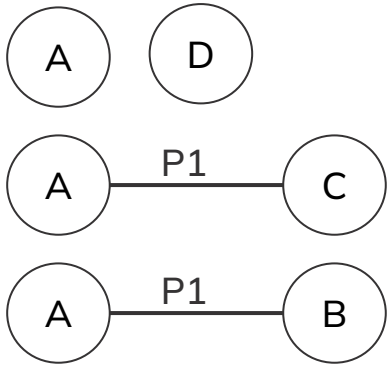
- On parcourt les triplets du graph RDF.
- Si l'un des mots clés de notre liste est présent en tant que Ressource ou valeur, on enregistre le noeud concerné.
- Si le mot clé correspond à un prédicat, on enregistre les noeud sources et destination ainsi que le predicat.

Recherche par mots clés



Mots clés: {A, D, P1}

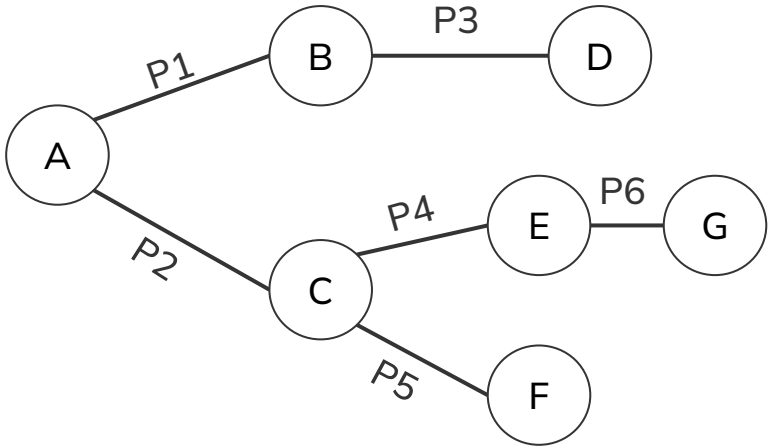
Résultats:



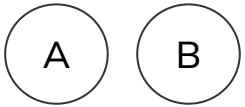
Construction du graph

- On choisit un nœud de départ aléatoirement parmi la liste des nœuds tirés des mots clés.
- Tant que tous les noeuds clés n'ont pas été visités:
  - si le noeud en cours fait partie d'un prédicat alors:
    - On cherche le plus court chemin entre le nœud source et les autres nœuds clés qui n'ont pas encore été visités.
    - On fait de même avec le noeud destination du prédicat.
    - À l'issue, on choisit le plus petit court chemin des deux.
    - On ajoute le plus court chemin a notre graph résultat.
    - On ajoute les nœuds source et destination à la liste des nœuds clés visités.

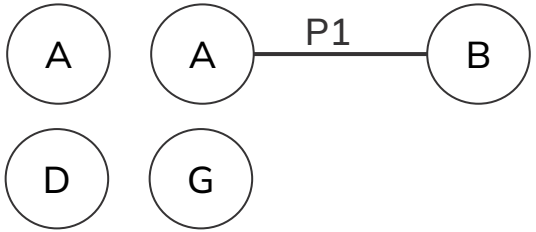
Construction du graph



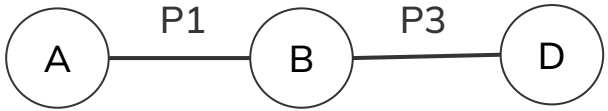
Visités:



Mots clés:



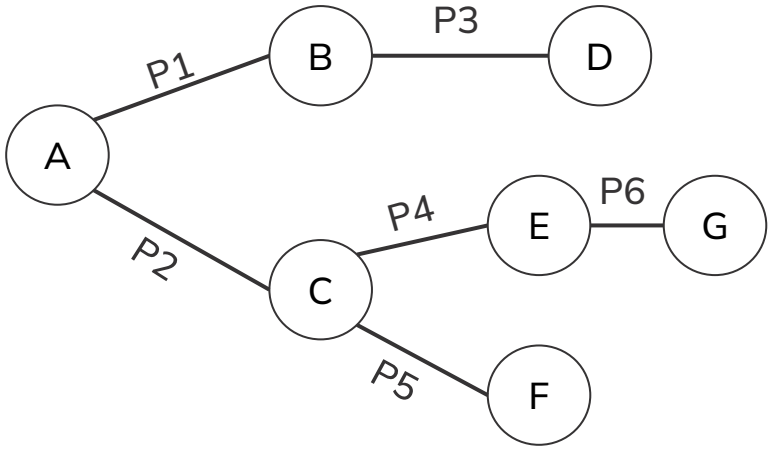
Résultat:



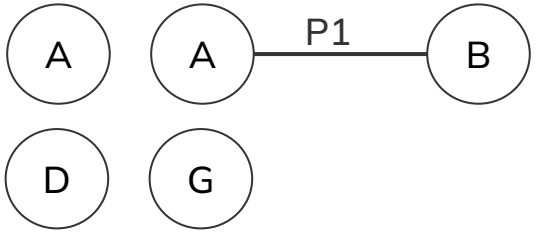
Construction du graph

- Tant que tous les noeuds clés n’ont pas été visités:
  - Si le noeud en cours ne fait pas partie d’un prédicat clé alors:
    - On cherche le plus court chemin entre le nœud et les autres nœuds clés qui n’ont pas encore été visités.
    - On ajoute le noeud en cours à la liste des nœuds clés visités.

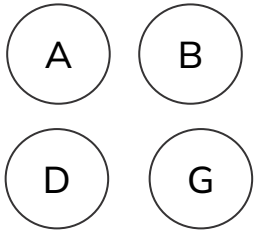
Construction du graph



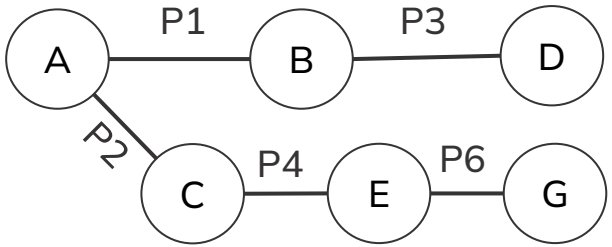
Mots clés:



Visités:



Résultat:



02

Etat de l'art

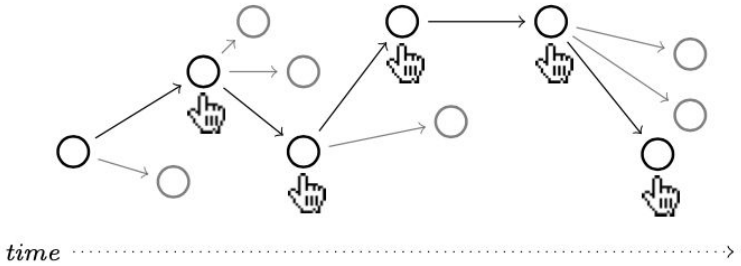
Keyword Searching and Browsing in Databases using BANKS

L'article propose une approche permettant **l'extraction de sous graphe dans un graph** à partir de mots clés.

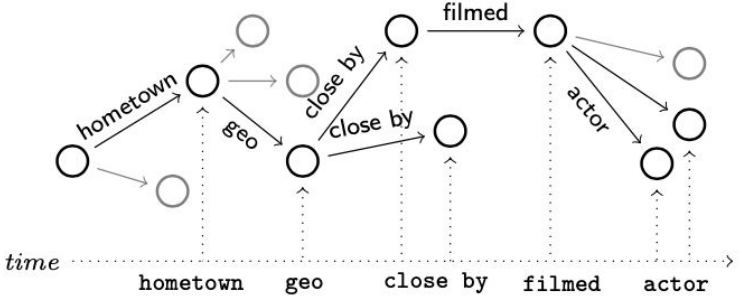
- Comme le mot clé Ki avec : nom de table, colonne, vue, … Nous nous sommes inspirés de leur méthode de comparaison de mots clés.
- Construction du graph réponse : trouver un sommet commun à partir duquel un chemin direct existe vers au moins un nœud pour chaque mot clé.
- Ranking : pour avoir la réponse la plus pertinente.

Keyword-Based Navigation and Search over the Linked Data Web

La problématique posée concerne l'utilisation des approches de recherche par mots-clés sur les graphes dans le contexte du Web de données liées (Linked Data web).



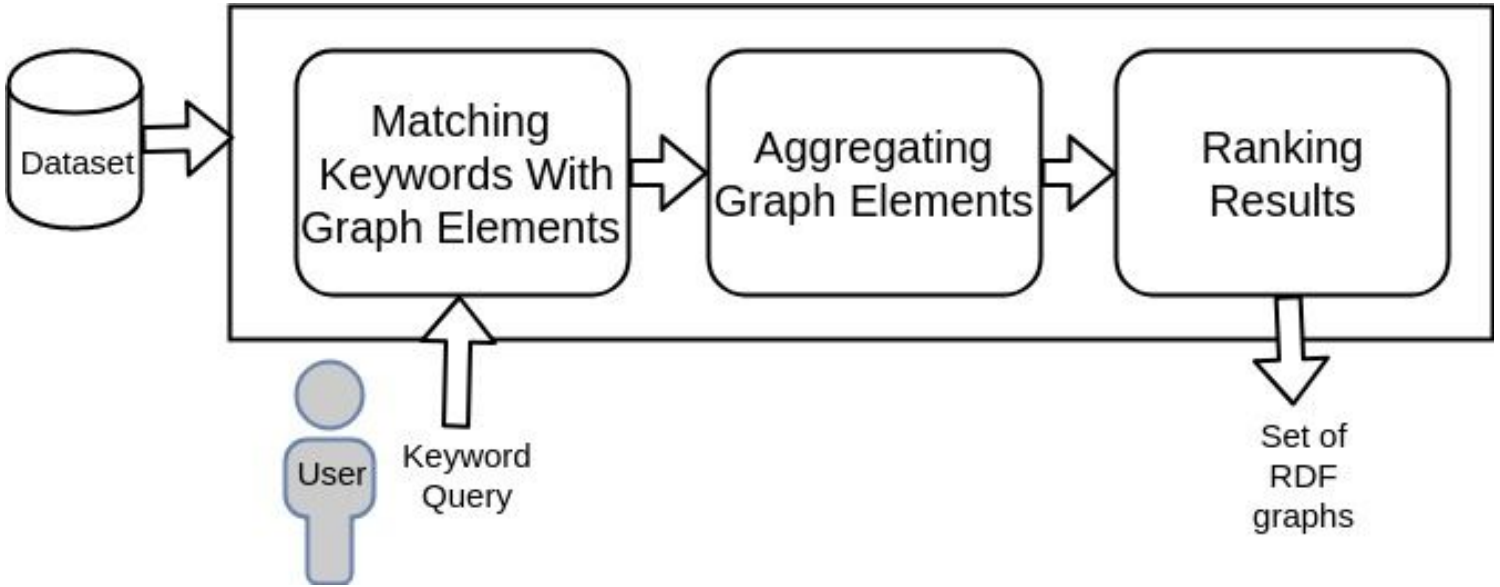
Navigation courante



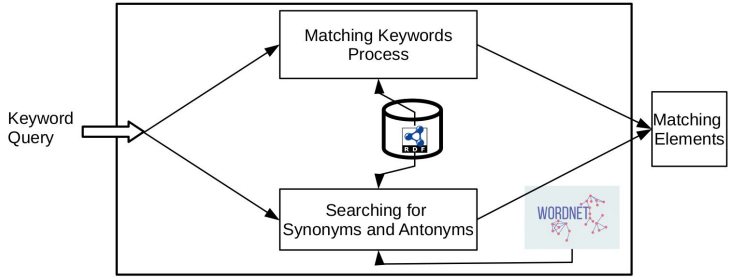
Leur approche

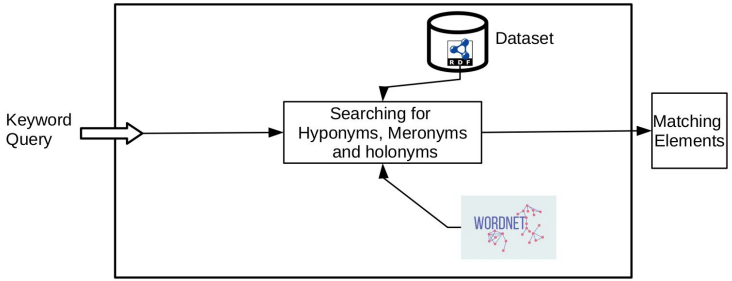
Source : l'article concerné

Keyword search over graph rdf using WordNet



Keyword search over graph rdf using WordNet





03

Propositions d’amelioration

Recherche de synonymes pour les mots clés

- Pour les mots clés qui n’ont pas de correspondance dans le graph, on propose de chercher des synonymes à ces mots qui pourrait donner des résultats lors de l'étape 1 de recherche.
- Afin d’avoir une méthode qui soit pertinente et qui prenne en considération le contexte du graph rdf, on propose d’utiliser une méthode basée sur les LLM.

Recherche de synonymes pour les mots clés

Algorithme 3 : Recherche de synonyme

Inputs: keyword (mot clé)
graph (fichier rdf ou ontologie)
client (client OpenAi pour utiliser l'api)

Output: result (synonyme du mot clé)

```
1  assistant = "you use only the following context to answer.\nHere's an rdf graph in xml
   format: "+ str(graph)

   prompt_template = """give me a keyword synonyme of {} so that when i search for node,
   relation, literal coresponding to that keyword in the rdf graph given it return me some
2  results.
   return me only the keyword synonyme not rdf elements.
   the keyword must correspond to an element in rdf graph.""".format(keyword)

   messages = [{'role': 'system', 'content': 'print only the synonyme word.'},
3              {"role": "user", "content": prompt_template},
               {"role": "assistant", "content": assistant},
               ]

   result = client.chat.completions.create(
4      model="gpt-3.5-turbo",
       messages=messages,
       temperature=0
   )
```

Recherche de synonymes pour les mots clés

- Cette approche de génération de synonyme permet de prendre en considération le context dans la proposition du synonyme.
- Elle permet aussi de corriger les fautes d'orthographe que pourrait commettre l'utilisateur.
- Elle permet aussi à l'utilisateur de donner une description de la ressource recherché au lieu de donner juste un mot clé.

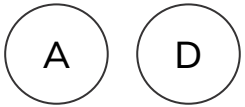
Produit cartésien des éléments extraits

Une autre amélioration qu'on propose est d'utiliser la méthode du produit cartésien:

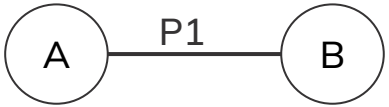
- Pour chaque mot clé on va parcourir tous les noeuds/prédicats qui ont été extraits à l'étape 1.
- On fait la combinaison de chaque nœud/prédicat avec tous les autres nœuds/prédicats des autres mots clés.
- Si pour chaque mot clé on a Nj noeuds/prédicats alors le nombre de combinaisons possible sera égale à ∏Nj

Produit cartésien des éléments extraits

Mot clé 1:

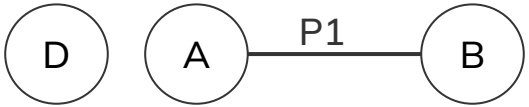


Mot clé 2:



Résultat 1:

Résultat 2:



Produit cartésien des éléments extraits

- Contrairement à la méthode de base qui aura tendance à retourner plus d'information que nécessaire, cette méthode permet d'avoir des graph moins dense mais avec moins de nœuds non clés.
- Cette méthode va permettre de répondre aux requêtes de recherche ou on sait d'avance que pour chaque mot clé on veut avoir seulement 1 nœud/prédicats.

Méthode de ranking

- Vu qu'on aura plusieurs réponses potentielles à la requête, on doit pouvoir choisir le meilleur graph à travers une méthode de ranking.
- Pour le ranking nous avons opté pour le score suivant:

$$Score = \frac{A}{N}$$

Ou A va représenter le nombre de nœuds/prédicats extrait après l'étape 1.

N va représenter le nombre de nœuds/prédicats après l'étape 2

Méthode de ranking

- Plus le score est élevé meilleur est le résultat.
- Cette méthode de ranking favorise les graph ou on introduit le moins de nœuds/prédicats non clés.

Méthode finale

1. Recherche de noeuds/prédicats qui correspondent à des mot clés en utilisant potentiellement la fonction synonyme.
2. Générer toutes les combinaisons possible avec le produit cartésien.
3. Pour chaque combinaison:
    a. Construire le graph final
    b. Calculer le score de ranking pour décider ou non de le garder comme graph de réponse finale.

04
Comparaison avec l'état de l'art

Keyword Searching and Browsing in Databases using BANKS

Aspect	Notre méthode	BANKS
type de données	Graph RDF	données relationnelles transformées en graph
Construction de l'arbre de réponse	Prend un noeud aléatoire et cherche le plus court chemin vers le prochain et ainsi de suite	Cherche le noeud le plus proche d'au moins un noeud pour chaque mot clé (Djisktra)
Ranking des réponses	graph score = nombre de noeuds pertinents / total de noeuds dans le graph	graph score : **combinaison du score du noeud (Ns) et score des arrêtes (Es).** Ne tient pas compte du score des noeuds intermédiaire

Keyword-Based Navigation and Search over the Linked Data Web

Aspects	Notre méthode	Keyword navigation, search
Type de données	base de données RDF	Données web liées
comparaison entre mots clés et éléments du graph	• Recherche de valeur exacte • Recherche de synonymes	• Recherche de valeur exacte dans les documents • Pas de gestion d synonymes

Keyword search over graph rdf using WordNet

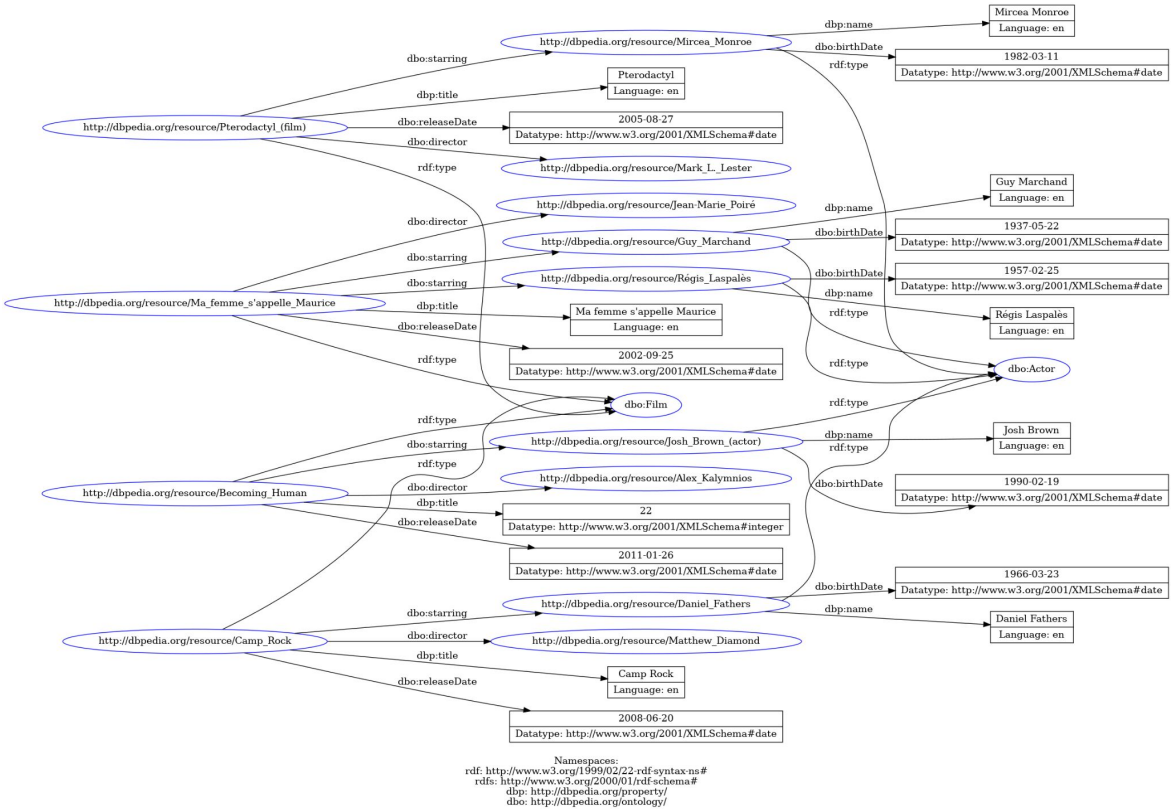
Aspects	Notre approche	cette approche
comparaison des mots clés Ki	similaire	similaire
Recherche des synonymes	Utilisation de LLM Recherche de synonyme (proche ou pas) sémantique facile **Gestion efficaces des erreurs d'orthographe et de grammaire** Extension facile à la recherche d'antonyme, Hyponyme et autres	Utilise WordNet pour la recherche **Large gestion des relations avec le mots clé (antonyme, Hyponyme, ...)**

Keyword search over graph rdf using WordNet

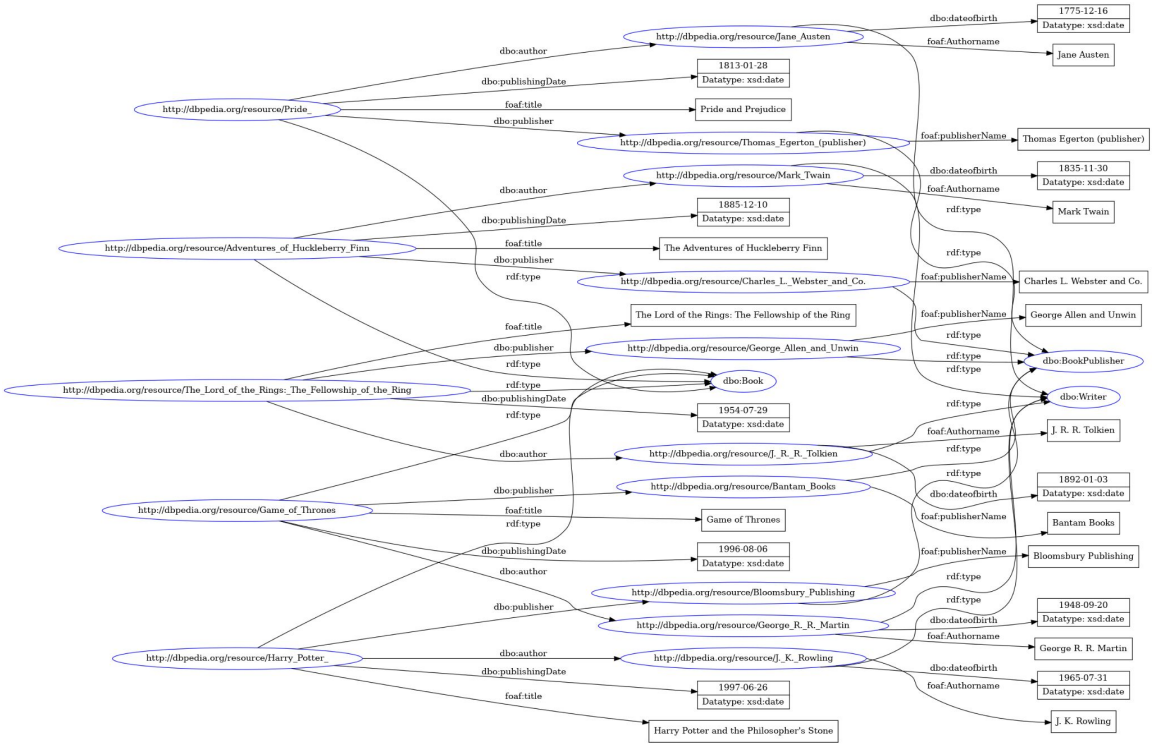
Aspects	Notre approche	cette approche
Construction du graph de réponse	Cherche le plus court chemin vers le prochain noeud • plus rapide • Chemin global final pas optimal	Cherche le chemin le plus court vers tous les autres nœuds du sous graph. • plus coûteux • chemin global optimal
Méthode de ranking	graph score = nombre de noeuds pertinents / total de noeuds dans le graph	introduit la notion de poids (Wa) pour chaque noeud faisant parti du graph $\text{Score} = 1 - \frac{[w_a * A + (1 - w_a) * L]}{N}$

05
Data sets

Graph sur les films



Graph sur les Livres



06

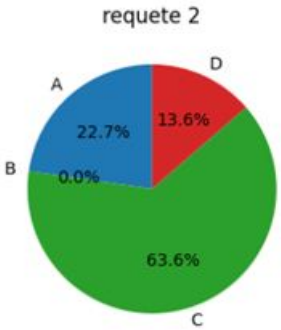
Test de performance

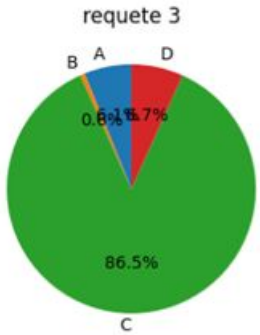
Expérimentations

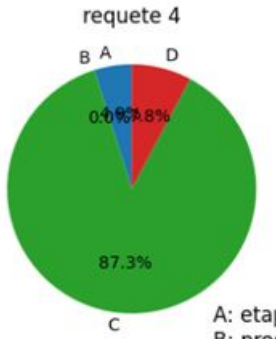
Les requêtes testées sont:

- 1- (["josh","starring"])
- 2- (["Daniel","starring"])
- 3- (["2005","film","starring","director"])
- 4- (["name","birthDate","Camp Rock"])
- 5- (["releaseDate","starring","director","Becoming_Human"])
- 6- (["title","starring", "Guy"])
- 7- (["film","starring","2002","name","birthDate"])
- 8- (["director","film","2002"])

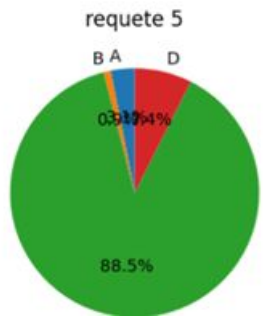
Résultat de chaque requête

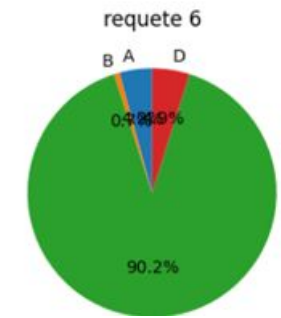


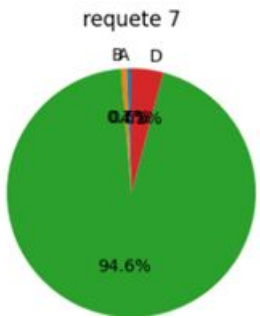


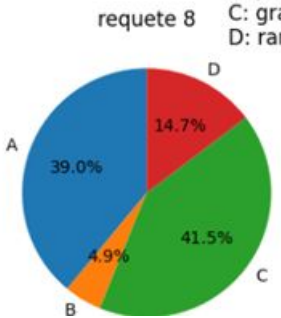


A: etape1
B: produit_cartesien
C: graph_construct
D: ranking

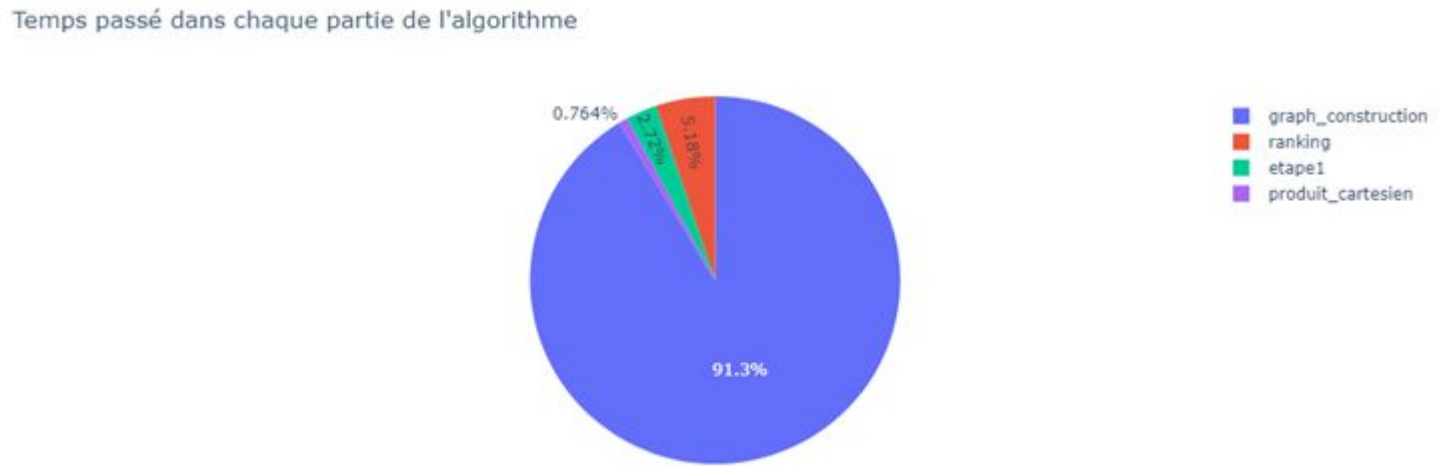








Moyen des résultats



07

Test de Precision

Protocole

1. Définition de 10 requêtes n'impliquant pas l'utilisation de la fonction synonyme par dataset et de leurs équivalents nécessitant son utilisation par dataset
2. Pour chaque chaque requête, définition de la réponse idéale manuellement
3. Execution des requêtes
4. Comparaison des résultats fourni par notre système avec les résultats attendus
   Métriques: edit distance, precision, recall, f1-score
5. Calcul de la moyenne, de l'écart-type, du minimum et du maximum par dataset

Requêtes

Requêtes ne nécessitant pas l'utilisation de la fonction synonyme

```
requetes = []
1.  requetes.append(["josh brown","starring"])
2.  requetes.append(["film","actor","name","2011"])
3.  requetes.append(['Daniel Fathers','starring','name'])
```

Requêtes utilisant la fonction synonyme

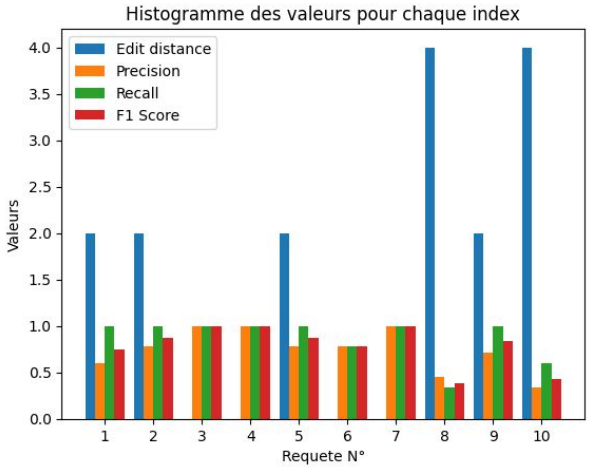
```
requetes_syn = []
1.  requetes_syn.append(["josh brown","played in"])
2.  requetes_syn.append(["movie","actor","name of actor","2011"])
3.  requetes_syn.append(['Daniel Fathers','acted in','name'])
```

Exemple de réponse idéale pour la requête 1

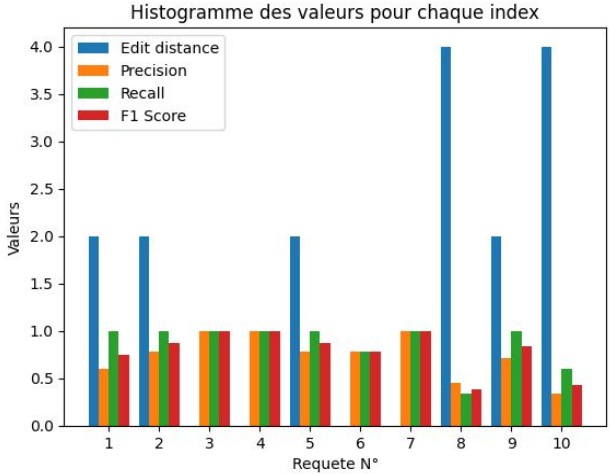
```
Réponse idéale pour la requête 1
graphs_ideal = []
graph1 =
graph_mots_cle([{'node_src':rdflib.term.URIRef('http://dbpedia.org/resource/Josh_Brown_(actor)
'),
                 'node_dest':rdflib.term.URIRef('http://dbpedia.org/resource/Becoming_Human'),
                 'predicate':rdflib.term.URIRef('http://dbpedia.org/ontology/starring')}],

[rdflib.term.URIRef('http://dbpedia.org/resource/Josh_Brown_(actor)'),rdflib.term.URIRef('http
://dbpedia.org/resource/Becoming_Human')
```

Résultats Dataset Films

Requêtes ne nécessitant pas l'utilisation de la fonction synonyme

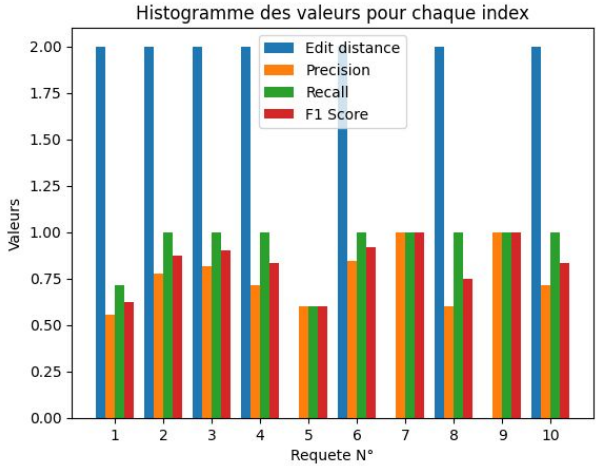


Requêtes utilisant la fonction synonyme

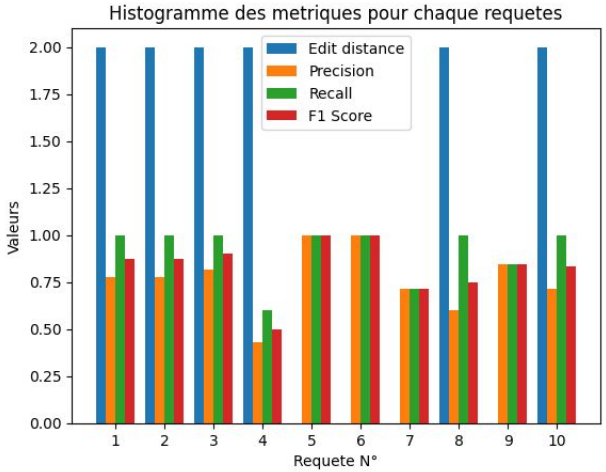


Résultats Dataset Livres

Requêtes ne nécessitant pas l'utilisation de la fonction synonyme



Requêtes utilisant la fonction synonyme



Résultats

Films Dataset

Evaluation des requêtes exactes

```
Statistiques:
              Moyenne    Ecart-type   Min         Max
ED          : [1.6        1.49666295 0.         4.        ]
Precision   : [0.74354978 0.2177335  0.33333333 1.        ]
Recall      : [0.87111111 0.22084015 0.33333333 1.        ]
F1 Score    : [0.79242979 0.21101787 0.38461538 1.        ]
```

Evaluation des requêtes utilisant la fonction synonyme

```
Statistiques:
              Moyenne    Ecart-type   Min         Max
Edit distance: [1.6        1.49666295 0.         4.        ]
Precision   : [0.74354978 0.2177335  0.33333333 1.        ]
Recall      : [0.87111111 0.22084015 0.33333333 1.        ]
F1 Score    : [0.79242979 0.21101787 0.38461538 1.        ]
```

Livre Dataset

Evaluation des requêtes exactes

```
Statistiques:
              Moyenne    Ecart-type   Min         Max
Edit distance: [1.4        0.91651514 0.         2.        ]
Precision   : [0.76262404 0.14947639 0.55555556 1.        ]
Recall      : [0.93142857 0.13950349 0.6        1.        ]
F1 Score    : [0.83333333 0.13170885 0.6        1.        ]
```

Evaluation des requêtes utilisant la fonction synonyme

```
Statistiques:
              Moyenne    Ecart-type   Min         Max
Edit distance: [1.2       0.9797959 0.        2.       ]
Precision   : [0.76770341 0.1629941  0.42857143 1.        ]
Recall      : [0.91604396 0.13957619 0.6        1.        ]
F1 Score    : [0.82937729 0.13989567 0.5        1.        ]
```

Passons maintenant aux tests !