

Université Larbi Ben Mhidi

Niveau : 1ère année Master IA

Département MI

Module : TLN

Année : 2024-2025

Série de TP 5 : Analyse sémantique

Ontologies en Traitement du Langage Naturel avec Python

Énoncé

Vous devez développer un système qui permet de manipuler des ontologies dans le cadre du Traitement du Langage Naturel. Le système devra inclure les fonctionnalités suivantes :

1. Explorer l'ontologie lexicale WordNet
2. Créer et manipuler votre propre ontologie de domaine
3. Extraire automatiquement des relations sémantiques à partir de textes
4. Visualiser et interroger l'ontologie

Implémentez chaque fonction du code fourni et testez votre système sur le domaine de l'informatique.

Guide d'implémentation des ontologies en TLN avec Python

Étape 1: Implémentation de l'explorateur WordNet

1. Connexion à WordNet :

- Utilisez `nlk.corpus.wordnet` pour accéder à la base de données WordNet
- Pour un mot donné, récupérez ses synsets avec `wn.synsets(mot, lang='fra')`
- Si aucun résultat en français, essayez en anglais avec `wn.synsets(mot, lang='eng')`

2. Extraction des relations :

- Pour chaque synset, récupérez sa définition avec `synset.definition()`
- Récupérez les hyperonymes avec `synset.hypernoms()`
- Récupérez les hyponymes avec `synset.hyponymes()`
- Récupérez les méronymes avec `synset.part_meronyms()`

3. Structuration des résultats :

- Créez un dictionnaire pour stocker les informations sur chaque synset
- Incluez l'identificateur, la définition, et les listes de relations

Étape 2: Implémentation de la classe Concept

1. Définition des attributs :

- `nom`: le nom du concept (chaîne de caractères)
- `definition`: description du concept (chaîne de caractères)
- `hyperonymes`: liste des concepts parents (liste de chaînes)
- `hyponymes`: liste des concepts enfants (liste de chaînes)
- `attributs`: dictionnaire des propriétés du concept (clé-valeur)
- `relations`: dictionnaire des relations avec d'autres concepts (clé-liste)

2. Méthodes de sérialisation (optionnelles) :

- `to_dict()`: convertit le concept en dictionnaire pour la sauvegarde
- `from_dict()`: crée un concept à partir d'un dictionnaire chargé

Étape 3: Implémentation de la classe Ontologie

1. Méthodes de gestion des concepts :

- `ajouter_concept(nom, definition)`: vérifie si le concept existe déjà, sinon l'ajoute
- `definir_relation_est_un(hyponyme, hyperonyme)`: établit une relation hiérarchique
 - Vérifiez que les deux concepts existent dans l'ontologie
 - Ajoutez hyperonyme à la liste des hyperonymes du hyponyme
 - Ajoutez hyponyme à la liste des hyponymes de l'hyperonyme

2. Méthodes de gestion des attributs et relations :

- `ajouter_atribut(concept, attribut, valeur)`: ajoute une propriété à un concept
 - Vérifiez que le concept existe

- Ajoutez l'attribut et sa valeur au dictionnaire des attributs

➤ ajouter_relation(concept1, relation, concept2): crée une relation entre deux concepts

- Vérifiez que les deux concepts existent
- Ajoutez concept2 à la liste des cibles de la relation pour concept1

3. **Affichage de l'ontologie :**

- Parcourez chaque concept et ses relations
- Formatez l'information de manière lisible

Étape 4: Implémentation de l'extraction de relations

1. **Analyse du texte avec spaCy :**

- Chargez le modèle français: `nlp = spacy.load("fr_core_news_sm")`
- Analysez le texte: `doc = nlp(texte)`

2. **Extraction des entités nommées :**

- Parcourez les entités avec `doc.ents`
- Pour chaque entité, ajoutez un concept à l'ontologie

3. **Extraction des relations sujet-verbe-objet :**

- Identifiez les verbes racines (`token.dep_ == "ROOT" and token.pos_ == "VERB"`)
- Pour chaque verbe, trouvez son sujet (dépendance `nsubj`)
- Pour chaque verbe, trouvez son objet (dépendance `dobj` ou `obj`)
- Si sujet et objet existent, ajoutez une relation à l'ontologie

4. **Extraction des attributs :**

- Identifiez les adjectifs qui modifient les noms (`mod.dep_ == "amod"`)
- Ajoutez ces adjectifs comme attributs des concepts correspondants

Étape 5: Implémentation de l'interrogation de l'ontologie

1. **Analyse de la question :**

- Identifiez les mots-clés dans la question
- Déterminez le type de question (type, relation, attribut...)

2. **Recherche de réponses :**

- Pour les questions sur les types, consultez les hyponymes du concept
- Pour les questions sur les parents, consultez les hyperonymes
- Pour les questions sur les attributs, consultez le dictionnaire des attributs
- Pour les questions sur les relations, consultez le dictionnaire des relations

3. **Formatage de la réponse :**

- Construisez une réponse claire en fonction des informations trouvées

Étape 6: Utilisation pratique

1. Création d'une ontologie de domaine :

- Définissez une ontologie pour l'informatique
- Ajoutez des concepts comme "Ordinateur", "Logiciel", "Matériel", etc.
- Établissez des relations hiérarchiques (par exemple, "Ordinateur portable" est un "Ordinateur")
- Ajoutez des attributs (par exemple, "Ordinateur portable" a un attribut "poids")
- Définissez des relations sémantiques (par exemple, "Ordinateur" exécute "Logiciel")

2. Test de l'extraction :

- Utilisez un texte technique sur l'informatique
- Extrayez les relations et enrichissez votre ontologie

3. Test des requêtes :

- Posez différentes questions à votre ontologie
- Vérifiez que les réponses sont correctes et pertinentes

Astuces d'implémentation

1. Gestion des erreurs :

- Vérifiez toujours l'existence des concepts avant de manipuler leurs relations
- Gérez les cas où les ressources linguistiques ne sont pas disponibles

2. Extensions possibles :

- Ajoutez des méthodes pour fusionner des ontologies
- Implémentez un mécanisme d'inférence pour déduire de nouvelles relations
- Ajoutez des fonctionnalités de recherche avancée

3. Optimisations :

- Utilisez des structures de données efficaces pour les recherches fréquentes
- Considérez l'utilisation de bases de données pour les ontologies volumineuses