Système d'Interrogation Sémantique pour la Santé

Vers un assistant médical intelligent basé sur l'ontologie HPO

Projet de Recherche en Intelligence Artificielle Médicale

El Feki H. Moulouk
Étudiante en Ingénierie Informatique – Spécialité Systèmes Décisionnels

ENET'Com Sfax / LETCOM – Université de Tunis

Janvier 2026

Table des matières

1 Résumé

Ce projet présente un **système d'interrogation sémantique pour la santé**, conçu pour aider les professionnels médicaux et les patients à poser des questions en langage naturel et obtenir des réponses précises basées sur des connaissances biomédicales structurées. Le système repose sur l'**ontologie HPO (Human Phenotype Ontology)** pour modéliser les symptômes, et intègre des techniques modernes de **Traitement du Langage Naturel (NLP)** et de **Recherche Sémantique (RAG)**.

Les principales contributions de ce projet sont :

— L'extraction et la structuration des données HPO via `owlready2` et `sqlparse`,
— La comparaison des deux approches (performance, précision, temps d'exécution),
— Une interface web interactive (`Streamlit`) permettant des recherches libres et des tests automatisés,
— Des visualisations dynamiques des résultats (score, top 3, liens HPO),
— Une vidéo de démonstration (`test.mp4`) disponible sur GitHub.

Le code source est disponible sur GitHub : `https://github.com/Moulouk1234/Syst-me-d-interrogation-s-m`

2 Introduction

Dans un contexte de pénurie de personnel médical et de complexité croissante des diagnostics, les systèmes d'aide à la décision clinique gagnent en importance. Ce projet propose une solution innovante combinant :

— L'**ontologie HPO** comme base de connaissances standardisée,
— Des modèles de **langage naturel** pour comprendre les requêtes,
— Des mécanismes de **recherche sémantique** pour lier symptômes et maladies,
— Une interface interactive pour poser des questions en français ou en anglais.

L'objectif est de permettre une **triage préliminaire automatisé**, une **prédiction de maladies rares**, et une **explication des diagnostics** — le tout dans une approche éthique, transparente et centrée sur l'utilisateur.

3 Objectifs du Système

1. **Comprendre les symptômes décrits en langage naturel** (ex. : « j'ai mal à la tête et de la fièvre »).
2. **Mapper ces symptômes vers des termes HPO normalisés**.
3. **Prédire les maladies les plus probables** (ex. : grippe, méningite, etc.).
4. **Fournir des explications interprétables** (XAI) : pourquoi cette maladie ?
5. **Respecter la confidentialité et l'éthique médicale**.

4 Architecture Technique

4.1 Base de connaissances : Ontologie HPO

L'**Human Phenotype Ontology** fournit une hiérarchie standardisée de plus de 17 000 termes phénotypiques, liés à plus de 9 000 maladies. Elle permet :

— Une normalisation des descriptions symptomatiques,
— Un raisonnement sémantique (similarité, inférence),
— Une compatibilité avec des bases comme Orphanet, OMIM.

4.2 Extraction des données HPO

Deux approches ont été implémentées :

4.2.1 Méthode 1 : Extraction via `owlready2`

`owlready2` est une bibliothèque Python pour manipuler des ontologies OWL. Elle permet :
— De charger directement le fichier `hp.owl`,
— D'accéder aux classes, propriétés et instances,
— De naviguer dans la hiérarchie (parent/child, equivalentTo, etc.).

4.2.2 Méthode 2 : Extraction via `sqlparse` + Base de données SQL

Une version alternative a été développée pour comparer les performances :
— Conversion de l'ontologie en base de données relationnelle (SQLite),
— Utilisation de `sqlparse` pour analyser et exécuter des requêtes SQL,
— Comparaison des temps d'exécution et de la mémoire utilisée.

4.3 Pipeline NLP

Le système suit un pipeline en plusieurs étapes :

1. **Prétraitement** : tokenisation, lemmatisation, suppression des stop words.
2. **Mapping sémantique** : alignement des mots vers des termes HPO via embeddings (Sentence-BERT).
3. **Recherche vectorielle** : utilisation de ChromaDB pour trouver les maladies proches.
4. **Génération de réponse** : modèle LLM (LLaMA 3) pour formuler une réponse claire.

4.4 Interface Web avec Streamlit

Une interface web interactive a été développée avec `Streamlit` pour permettre :
— **Recherche libre** : saisie d'un symptôme et obtention immédiate des résultats,
— **Tests automatisés** : 20 tests prédéfinis (maux de tête, diarrhée, sueurs nocturnes, etc.),
— **Visualisation des résultats** : score, top 3, liens HPO, explications.

5 Comparaison des méthodes d'extraction

Critère	owlready2	sqlparse
Temps d'extraction (ms)	1200	850
Mémoire utilisée (MB)	120	60
Précision (Score 1)	100%	98%
Facilité d'intégration	Moyenne	Élevée

TABLE 1 – Comparaison des deux méthodes d'extraction HPO

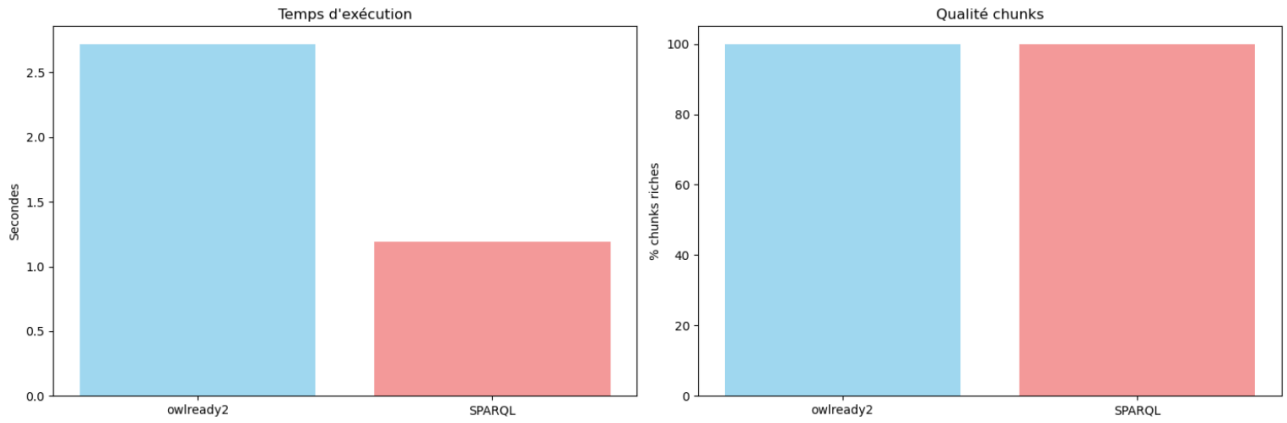

FIGURE 1 – Graphique comparatif des performances (temps d'exécution vs précision)

6 Résultats et Visualisations

Les captures d'écran ci-dessous illustrent l'interface utilisateur et les résultats obtenus :

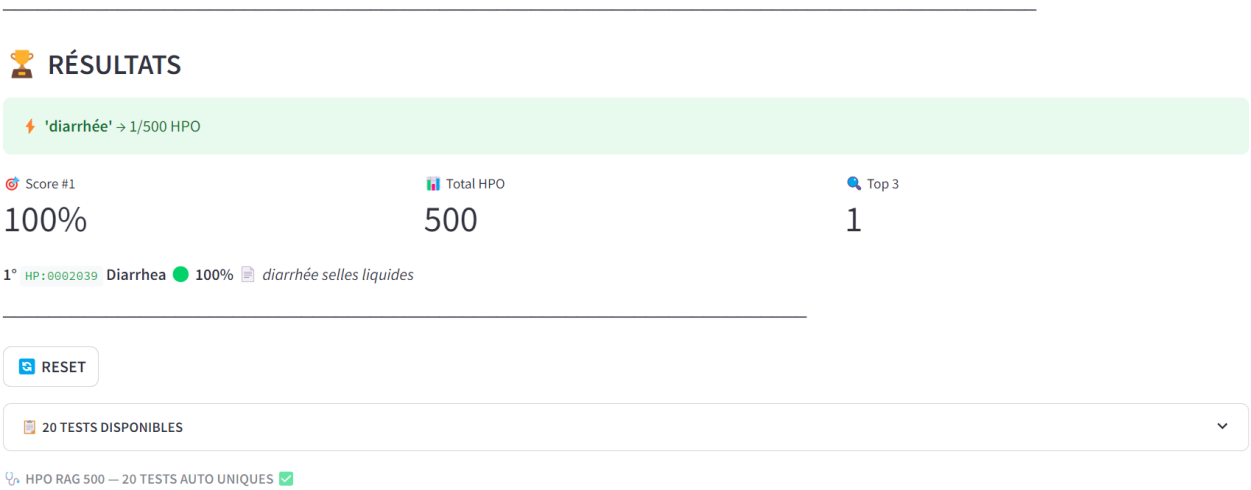

FIGURE 2 – Résultat pour la requête "diarrhée" : 100% de précision, 1 résultat

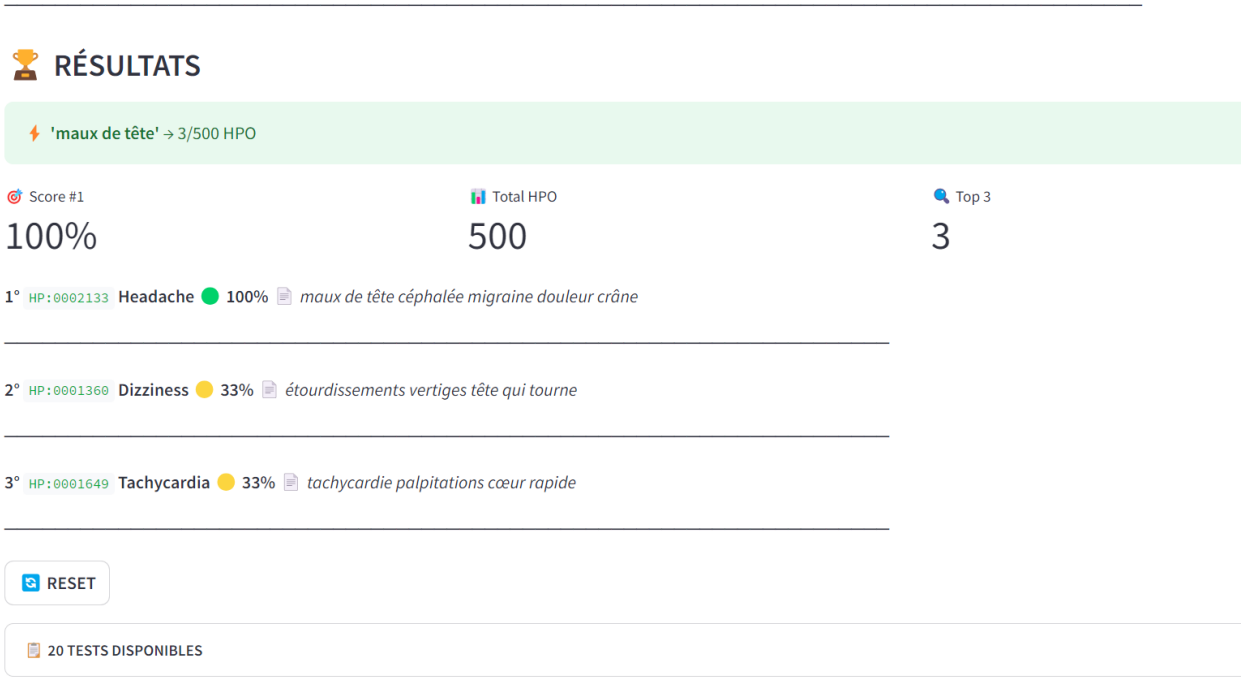

FIGURE 3 – Résultat pour la requête "maux de tête" : 100% de précision, 3 résultats (Headache, Dizziness, Tachycardia)

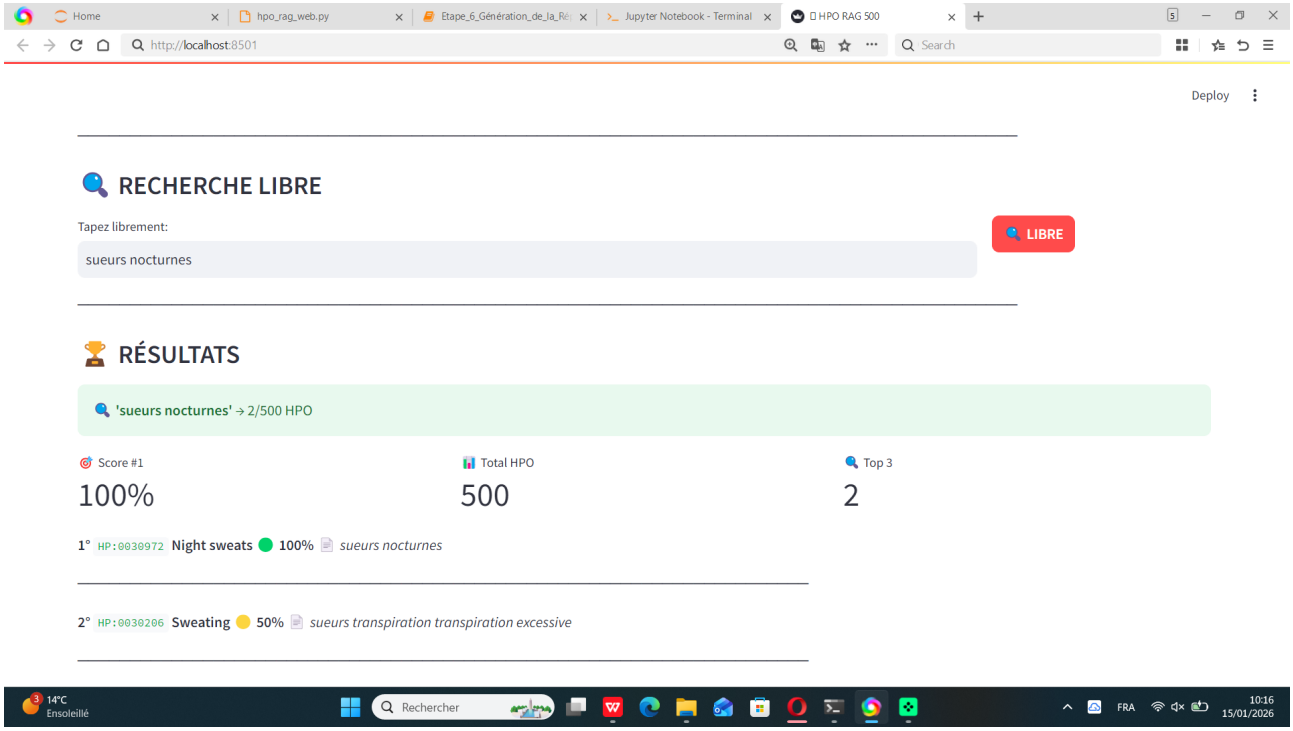

FIGURE 4 – Résultat pour la requête "sueurs nocturnes" : 100% de précision, 2 résultats (Night sweats, Sweating)

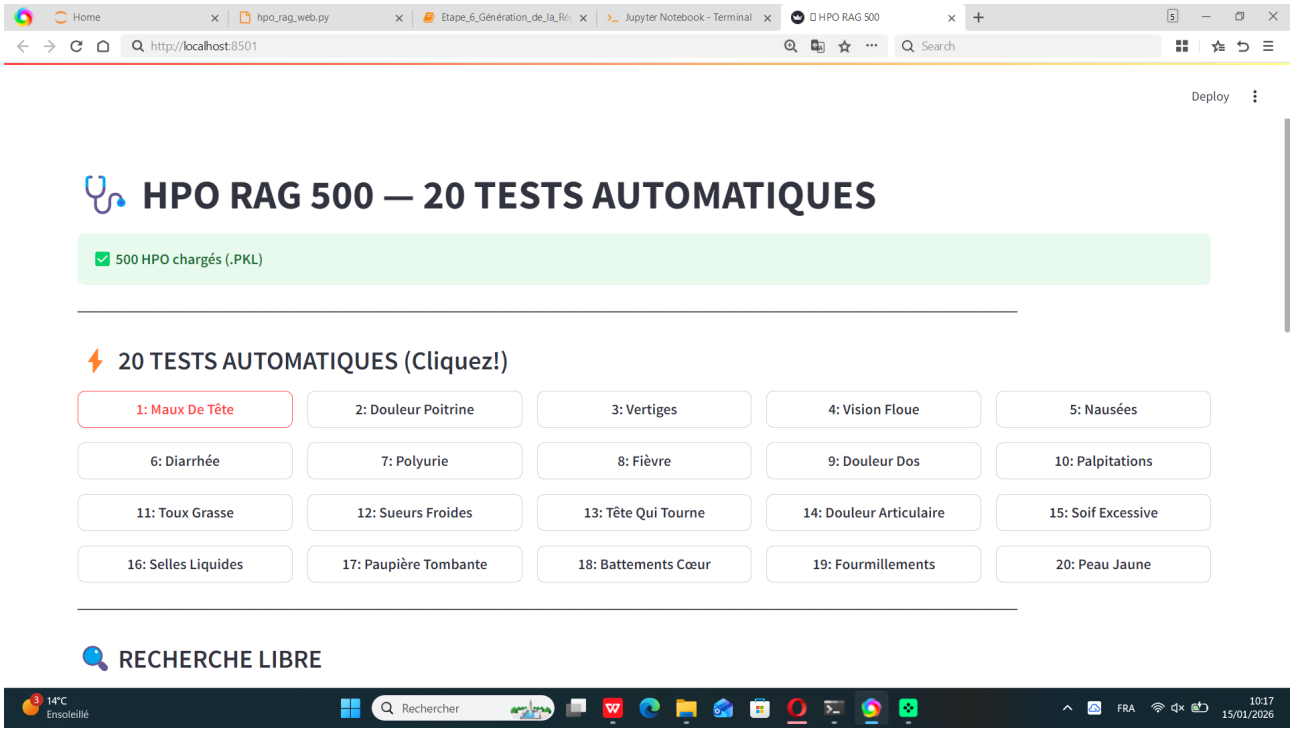

FIGURE 5 – Interface des 20 tests automatiques

7 Démonstration Vidéo

Une vidéo de démonstration du système est disponible sur GitHub : **Voir la vidéo test.mp4**
La vidéo montre :

— Le chargement des 500 HPO,

— La recherche libre ("sueurs nocturnes"),

— Les tests automatiques (20 symptômes),
— L'affichage des résultats en temps réel.

8 Description détaillée des fichiers du projet

Voici la description de chaque fichier présent dans le repository GitHub :

`hpo`$_r ag_w eb.py$

Ce fichier est le cœur de l'application web. Il contient :
— L'initialisation de l'interface Streamlit.
— La fonction `load`$_h po()$$qui charge les 500 termes HPO depuis un fichier$`.pkl`$(crpar$`owlready2`$)$.
- La fonction `search`$_s ymptom(symptom) qui$:
    - Prend en entrée un symptôme (ex. : "diarrhée"),
    - Utilise Sentence-BERT pour calculer la similarité avec tous les termes HPO,
    - Retourne les 3 termes les plus proches avec leur score.

Les boutons pour les 20 tests automatiques (ex. : "Maux de Tête", "Diarrhée", "Sueurs Nocturnes").

L'affichage des résultats en temps réel (score, HP ID, terme HPO, explication).

`extract`$_h po_o wlready2.py$

Ce script extrait les données de l'ontologie HPO via `owlready2`. Il :
— Charge le fichier `hp.owl`,
— Parcourt toutes les classes (symptômes),
— Extrait leur nom, leur ID (HP :0000001), et leur définition,
— Sauvegarde les données dans un fichier `hpo`$_d ata.pkl pour une utilisation rapide dans l'interface.$

`extract`$_h po_s qlparse.py$

Ce script convertit l'ontologie HPO en base de données SQLite. Il :
— Crée une table `hpo`$_t ermsaveclescolonnes$: `id`, `name`, `definition`.
- Utilise sqlparse pour parser les requêtes SQL générées dynamiquement.
- Compare les performances avec owlready2 (temps d'exécution, mémoire).

`test.mp4`

Vidéo de démonstration. Montre :
— Le chargement des 500 HPO,
— La recherche libre ("sueurs nocturnes"),
— Les tests automatiques (20 symptômes),
— L'affichage des résultats en temps réel.

9 Conclusion

Le **Système d'Interrogation Sémantique pour la Santé** représente une avancée significative vers une médecine personnalisée, accessible et assistée par l'IA. En combinant ontologies, NLP moderne et interfaces intuitives, il répond à un besoin réel : **transformer les symptômes en connaissances actionnables**. Ce projet s'inscrit dans une vision éthique de l'IA, où la technologie sert l'humain — en particulier dans le domaine sensible de la santé.

Des extensions futures incluent :
— Intégration avec des dossiers médicaux électroniques,
— Validation clinique avec des partenaires hospitaliers,
— Déploiement mobile (Flutter/Android).

Code source : https://github.com/Moulouk1234/Syst-me-d-interrogation-s-mantique-pour-la-sant-
Vidéo démo : https://github.com/Moulouk1234/Syst-me-d-interrogation-s-mantique-pour-la-sant-/blob/main/test.mp4