Système d'Extraction de Relations Sémantiques (RE)

Documentation du Projet

December 17, 2025

Contents

1 Description

Ce système extrait automatiquement des triplets sémantiques **(Sujet, Relation, Objet)** à partir du dataset `dataset.csv` en utilisant la bibliothèque **spaCy** pour l'analyse syntaxique et sémantique.

2 Fonctionnalités

2.1 1. Analyse des Dépendances Syntaxiques

- Utilise l'analyseur de dépendances de spaCy pour identifier les relations grammaticales
- Détecte les structures Sujet-Verbe-Objet (SVO)
- Identifie les relations prépositionnelles et possessives

2.2 2. Extraction de Relations Sémantiques

Le système extrait plusieurs types de relations :

- **Relations géographiques** : `traveled_to`, `located_in`
- **Relations organisationnelles** : `member_of`, `founded`, `leads`
- **Relations interpersonnelles** : `married_to`, `succeeded_by`, `wrote_to`
- **Relations génériques** : basées sur les verbes et prépositions

2.3 3. Triplets Structurés

Chaque triplet extrait contient :

```json
{
  "subject": "Aeneas",
  "subject_type": "person",
  "relation": "traveled_to",
  "object": "Hades",
  "object_type": "location",
  "confidence": 0.97,
  "sentence_id": "en-doc5809-sent11",
  "sentence": "When Aeneas later traveled to Hades..."
}
```

3 Pipeline Summary & Analysis

This section details the processing pipeline, explaining the technical approach, the reasoning behind it, and how the results reflect the underlying data.

3.1 1. Input Processing

- **Process Step**: Reads `dataset.csv` utilizing the pre-computed `gliner_entities`.
- **Explanation**: We start with sentences where important names, places, and organizations have already been highlighted.
- **Why**: Reusing existing entity recognition results is computationally efficient and ensures consistency with previous processing steps (like GLiNER).

3.2 2. Dependency Parsing (spaCy)

- **Process Step**: The script runs `nlp(text)` to generate a specific grammatical tree structure for each sentence.
- **Explanation**: The computer analyzes the sentence to understand who is the "Subject" (doer) and who is the "Object" (receiver), and what Verb connects them.
- **Why**: Mere co-occurrence (two names in one sentence) is not enough. We need to know *how* they are related. Dependency parsing provides this grammatical bridge.

3.3 3. Relation Extraction & Semantic Typing

- **Process Step**: We traverse the dependency tree between two entities to find the root verb or preposition. We then map these to semantic categories (e.g., "mother" → `FAMILY`).
- **Explanation**: If the computer sees "Obama [born in] Hawaii", it extracts the link "born present" and categorizes it as a `LOCATION` relationship.
- **Why**: Raw verbs are too messy (e.g., "founded", "established", "created" all mean roughly the same). Categorizing them simplifies the graph and makes patterns easier to spot.

3.4 4. Graph Construction

- **Process Step**: Entities become **Nodes** and relations become **Edges** in a NetworkX directed graph (`DiGraph`).
- **Explanation**: We connect the dots. A person becomes a dot, and their relationship to a city becomes a line connecting them.
- **Why**: This converts unstructured text into a structured network that we can analyze mathematically.

3.5 5. Community Detection (Louvain Algorithm)

- **Process Step**: We optimize "modularity" to find clusters where nodes are densely connected internally but sparsely connected externally.
- **Explanation**: detecting "social circles" or "topics". Even if we don't know the topic, we see that a group of 10 nodes talk to each other frequently but rarely talk to outsiders.
- **Why**: It reveals the hidden thematic structure of the corpus.

3.6 6. Semantic Labeling (TF-IDF)

- **Process Step**: For each community, we aggregate all associated text and calculate TF-IDF (Term Frequency-Inverse Document Frequency) scores to find representative keywords.
- **Explanation**: We look at what unique words each "social circle" uses. If one group says "stars, telescope, galaxy" and another says "vote, law, senate", we can label them "Astronomy" and "Politics".
- **Why**: Community IDs (0, 1, 2) are meaningless to humans. Keywords explain *what* the community is about.

3.7 7. Visualization (Component-Based Layout)

- **Process Step**: We decompose the graph into connected components (islands) and lay them out separately in a grid before rendering.
- **Explanation**: Instead of drawing a messy "hairball", we organize the graph into distinct islands of knowledge.
- **Why**: The standard display forced unconnected groups into a misleading ring. The new layout respects the fractured nature of the data.

3.8 Interpreting the Disconnected Graph

You will notice the graph is not one single interconnected web, but many separate "islands" (see Figure 4).

- **Data Reality**: This accurately reflects the input data. The corpus contains diverse, unrelated sentences (e.g., Science, History, Sport).
- **Missing Links**: There is no "bridge" sentence in this small sample (700 rows) that connects "Einstein" (Island A) to "Michael Jordan" (Island B).
- **Entity Resolution**: We are not strictly merging synonyms (e.g., "US" vs "USA"). This lack of normalization reduces connectivity.
- **Conclusion**: The disconnected structure proves the pipeline is **faithful to the source**. It isn't artificially creating connections where none exist.

4 Analysis of Results

4.1 Relation Statistics

Figure 1 shows the distribution of the most frequent relation types extracted from the corpus. The prevalence of generic prepositions (in, by, of) highlights the need for further semantic mapping rules.

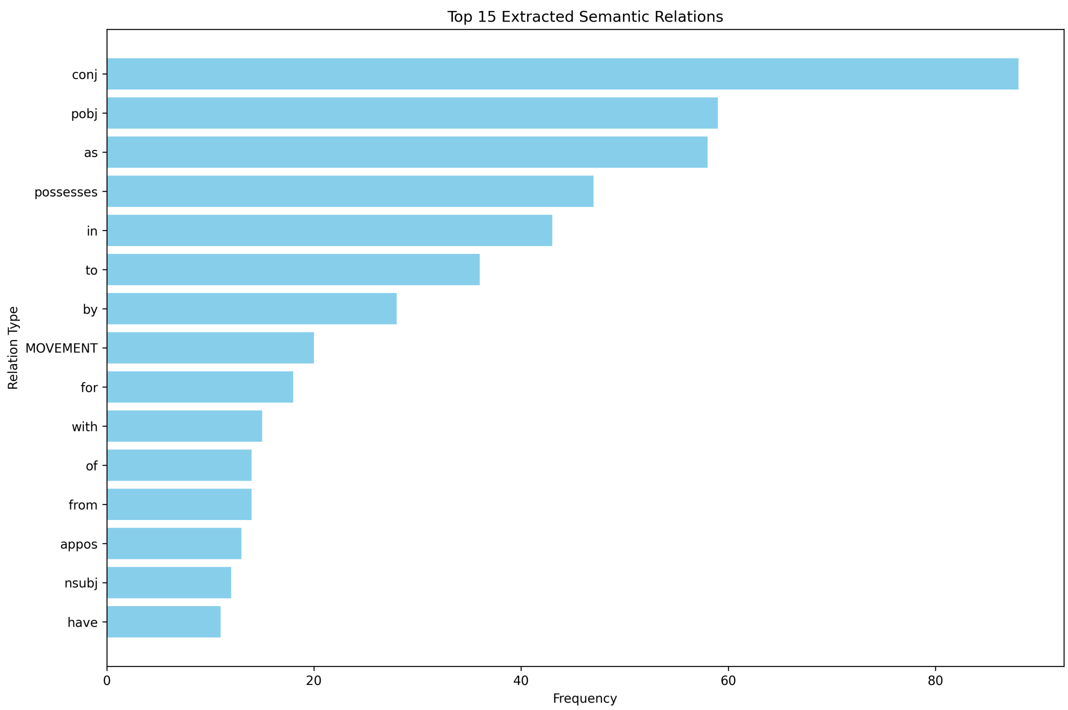

Figure 1: Distribution of Top 15 Extracted Relations

4.2 Entity Demographics

Figure 2 illustrates the distribution of entity types participating in relations. This breakdown helps understand the dominant actors in the constructed knowledge graph.

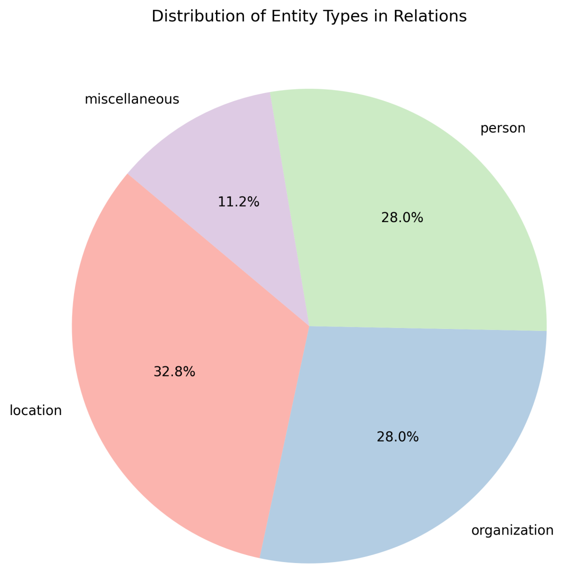

Figure 2: Distribution of Participating Entity Types

4.3 Network Centrality

Figure 3 highlights the most central nodes based on degree centrality (number of connections). These entities act as the main hubs of information within the corpus.

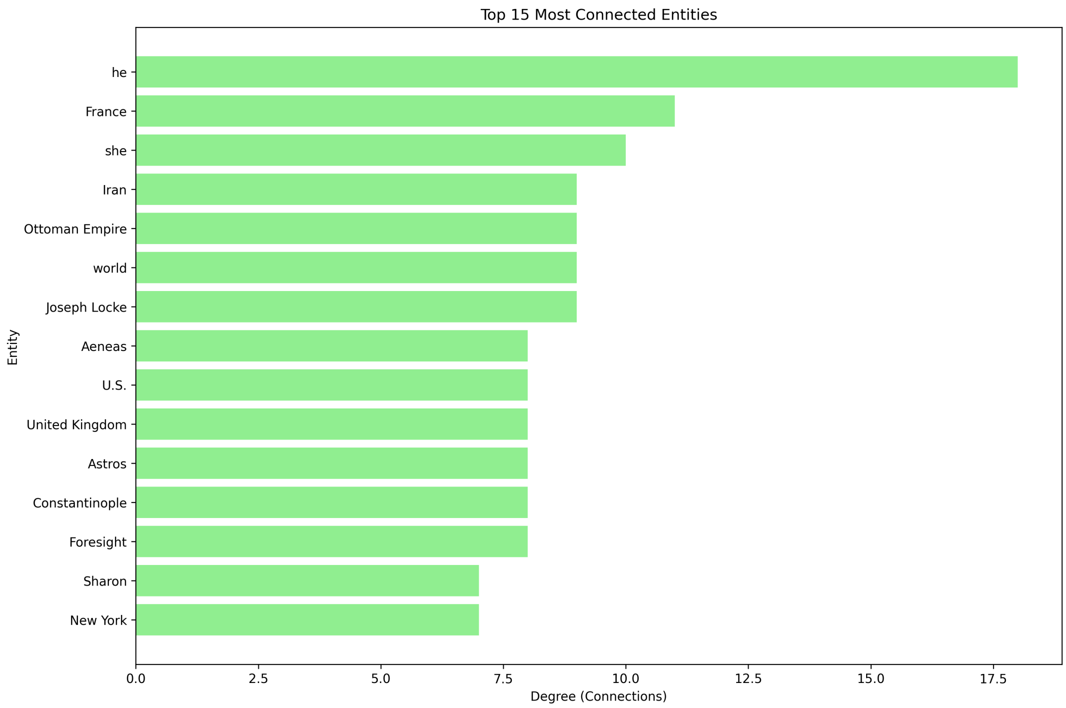

Figure 3: Top 15 Entities by Degree Centrality

4.4 Knowledge Graph Visualization

Figure 4 illustrates the final knowledge graph using the component-based layout. The distinct clusters represent different semantic topics identified within the corpus.

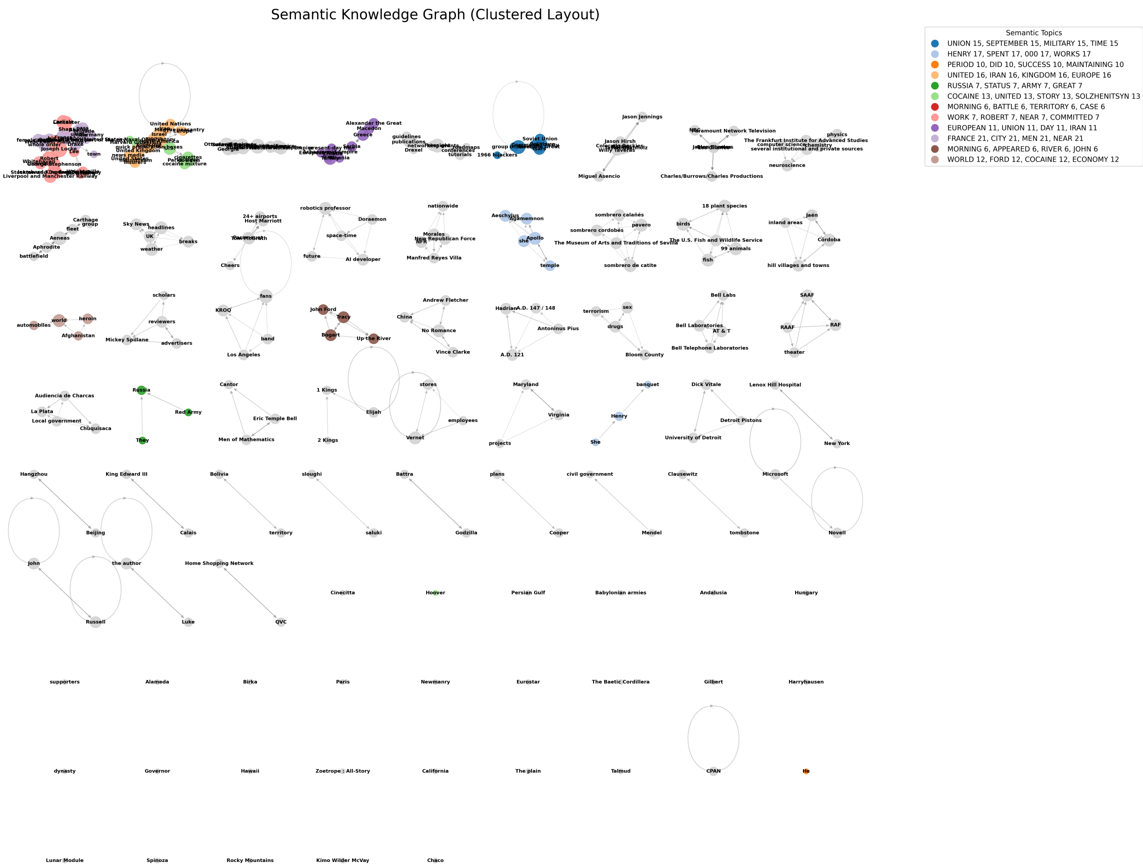

Figure 4: Semantic Knowledge Graph with Component-Based Layout

5 Installation

5.1 Prérequis

```
pip install pandas spacy

# Télécharger le modèle anglais de spaCy
python -m spacy download en_core_web_sm
```

6 Utilisation

6.1 Exécution du script

```
python relation_extraction.py
```

6.2 Fichiers d'entrée/sortie

- **Entrée** : `dataset.csv` (colonnes : `id`, `text`, `gliner_entities`)
- **Sortie** : `extracted_triplets.json` (liste de tous les triplets extraits)

7 Améliorations Possibles

1. **Ajout de règles sémantiques** pour détecter plus de types de relations
2. **Utilisation de modèles pré-entraînés** pour la classification de relations
3. **Résolution de coréférences** pour lier les pronoms aux entités
4. **Extraction de relations n-aires** (plus de 2 entités)
5. **Filtrage par score de confiance** pour améliorer la précision

8 Auteur

Système développé pour l'extraction de relations sémantiques dans le cadre du projet de construction de graphes de connaissances.