

Données semi structurées et sémantique

Mini-projet - de la modélisation à la visualisation

LIU Taoran

Source de données : [TripAdvisor European restaurants](#)

Git: [LIUTaoran-0214/LIUTaoran-Europe-Restaurant](#)

1. Introduction aux données

1.1 Source du jeu de données

Les données proviennent de l'ensemble de données TripAdvisor European Restaurants sur Kaggle. TripAdvisor est un site web de voyage qui répertorie les adresses des restaurants (texte et coordonnées), leurs descriptions, les notes et avis des utilisateurs, ainsi que leurs styles culinaires. L'identifiant unique du restaurant est `restaurant_link`, qui est également utilisé comme IRI.

- Informations de base sur le restaurant: `restaurant_name`, `address`, e.t.c.
- Informations géographiques: `latitude`, `longitude`, `country`, `city`, e.t.c.
- Informations d'évaluation:
 - Note globale: `avg_rating`
 - Notes pour chaque critère: `food`, `service`, `value`, `atmosphere`.
 - Nombre d'avis: `excellent`, `veryGood`, `average`, `poor`, `terrible`.
- Informations sur les repas: `cuisines`, `meals`, `features`
 - Options spéciales: `vegetarian_friendly`, `vegan_options`, `gluten_free`.
- Niveau de prix: `price_level`, `price_range`
- Heures d'ouverture: `original_open_hours`, e.t.c.
- Autres champs: `awards`, `keywords`, e.t.c.

1.2 Nettoyage et conversion des données en RDF

1. Les champs obligatoires sont les suivants :

- Informations de base sur le restaurant: `restaurant_name`, `address`.
- Champs liés à la localisation: `latitude`, `longitude`, `country`, `city`.
- Champ d'évaluation principal: `avg_rating`
- Nombre d'avis: `excellent`, `veryGood`, `average`, `poor`, `terrible`.
- Notes pour chaque critère: `food`, `service`, `value`.
- Informations sur les repas: `cuisines`, `meals`, `features`.
- Niveau de prix : `price_level`.

Les enregistrements ne comportant pas ces champs obligatoires seront supprimés. Cela permettra de réduire la taille des données de plusieurs millions à environ 30,000.

2. Standardisation du format des chaînes de caractères

Les données d'origine contenant des valeurs de type caractère au format JSON, plusieurs ajustements de format ont été apportés aux valeurs des champs, notamment:

- Suppression des espaces superflus et des caractères invisibles: suppression des espaces en début et en fin de chaîne et remplacement des espaces vides.

- Unification des types de données: conversion des champs tels que la latitude et la longitude, les notes et le nombre d'avis en types numériques.
- Normalisation des champs booléens: Unification des types Y/N, yes/no, true/fault, etc., en valeurs booléennes.
- Fractionnement des champs à valeurs multiples : division des champs, tels que cuisines, meals, features, awards, reliés par des délimiteurs.

Configuration RDF:

<code>R: restaurant_link</code>	<code>>-.originalLocation→</code>	<code>L: original_location</code>
<code>* Restaurant</code>		<code>Add object...</code>
<code>Add type...</code>		
	<code>>-.country→</code>	<code>L: country</code>
		<code>Add object...</code>
	<code>>-.region→</code>	<code>L: region</code>
		<code>Add object...</code>
	<code>>-.province→</code>	<code>L: province</code>
		<code>Add object...</code>
	<code>>-.city→</code>	<code>L: city</code>
		<code>Add object...</code>
	<code>>-.isClaimed→</code>	<code>L: claimed</code>
		<code>Add object...</code>
	<code>>-.awards→</code>	<code>L: awards</code>
		<code>Add object...</code>
	<code>>-.popularityDetailed→</code>	<code>L: popularity_detailed</code>
		<code>Add object...</code>
	<code>>-.popularityGeneric→</code>	<code>L: popularity_generic</code>
		<code>Add object...</code>
	<code>>-.topTags→</code>	<code>L: top_tags</code>
		<code>Add object...</code>
	<code>>-.priceLevel→</code>	<code>L: price_level</code>
		<code>Add object...</code>
	<code>>-.priceRange→</code>	<code>L: price_range</code>
		<code>Add object...</code>
	<code>>-.meals→</code>	<code>L: meals</code>
		<code>Add object...</code>
	<code>>-.cuisines→</code>	<code>L: cuisines</code>
		<code>Add object...</code>
	<code>>-.specialDiets→</code>	<code>L: special_diets</code>
		<code>Add object...</code>
	<code>>-.features→</code>	<code>L: features</code>
		<code>Add object...</code>
	<code>>-.isVegetarianFriendly→</code>	<code>L: vegetarian_friendly</code>
		<code>Add object...</code>
	<code>>-.isVeganOptions→</code>	<code>L: vegan_options</code>
		<code>Add object...</code>
	<code>>-.isGlutenFree→</code>	<code>L: gluten_free</code>
		<code>Add object...</code>
	<code>>-.originalOpenHours→</code>	<code>L: original_open_hours</code>
		<code>Add object...</code>
	<code>>-.openDaysPerWeek→</code>	<code>L: open_days_per_week</code>
		<code>Add object...</code>
	<code>>-.openHoursPerWeek→</code>	<code>L: open_hours_per_week</code>
		<code>Add object...</code>
	<code>>-.workingShiftsPerWeek→</code>	<code>L: working_shifts_per_week</code>
		<code>Add object...</code>
	<code>>-.avgRating→</code>	<code>L: avg_rating</code>
		<code>Add object...</code>
	<code>>-.reviewsCount→</code>	<code>L: total_reviews_count</code>
		<code>Add object...</code>
	<code>>-.defaultLanguage→</code>	<code>L: default_language</code>
		<code>Add object...</code>
	<code>>-.reviewsCountInDefaultLanguage→</code>	<code>L: reviews_count_in_default_language</code>
		<code>Add object...</code>
	<code>>-.keywords→</code>	<code>L: keywords</code>
		<code>Add object...</code>
	<code>>-.restaurantName→</code>	<code>L: restaurant_name</code>
		<code>Add object...</code>
	<code>>-.address→</code>	<code>L: address</code>

3. Fusion des informations similaires

- La latitude et la longitude sont fusionnées en coordonnées.
- Le nombre de commentaires est fusionné en une composante d'évaluation.
- Les scores des différents aspects sont fusionnés en une composante de score.

Configuration RDF:



Exemples de tuples:

```

:restaurant_g503974-d6886701
  :rdf:type          :Restaurant;
  :address           "Lakeside Kings Road, Cleethorpes DN35 0AG England";
  :avgRating         "3.5"^^xsd:double;
  :awards            "Certificate of Excellence 2016";
  :city              "Cleethorpes";
  :coordinate         [ :latitude   "53.54695"^^xsd:double;
                      :longitude  "-0.012405"^^xsd:double
                    ];
  :country            "England";
  :cuisines           "Cafe" , "British";
  :defaultLanguage    "English";
  :features            "Highchairs Available" , "Takeout" , "Outdoor Seating" , "Seating" , "Wheelchair Accessible";
  :isClaimed          true;
  :isGlutenFree       false;
  :isVeganOptions     false;
  :isVegetarianFriendly false;
  :meals              "Lunch" , "Brunch" , "Breakfast";
  :openDaysPerWeek    "7"^^xsd:double;
  :openHoursPerWeek   "49"^^xsd:double;
  :originalLocation    "Europe, United Kingdom (UK), England, Lincolnshire, Cleethorpes";
  :originalOpenHours   "Mon:10:00-17:00,Tue:10:00-17:00,Wed:10:00-17:00,Thu:10:00-17:00,Fri:10:00-17:00,Sat:10:00-17:00,Sun:10:00-17:00";
  :popularityDetailed "#37 of 93 Restaurants in Cleethorpes";
  :popularityGeneric  "#56 of 140 places to eat in Cleethorpes";
  :priceLevel          "€";
  :region              "Lincolnshire";
  :restaurantName      "Cleethorpes Discovery Centre Cafe";
  :reviewsComponents   [ :averageCount   "21"^^xsd:int;
                        :excellentCount  "42"^^xsd:int;
                        :poorCount        "18"^^xsd:int;
                        :terribleCount   "12"^^xsd:int;
                        :veryGoodCount   "48"^^xsd:int
                      ];
  :reviewsCount         "133"^^xsd:int;
  :reviewsCountInDefaultLanguage "133"^^xsd:double;
  :scoreComponents      [ :atmosphereScore  "4.5"^^xsd:double;
                        :foodScore        "4"^^xsd:double;
                        :serviceScore     "4"^^xsd:double;
                        :valueScore       "4"^^xsd:double
                      ];
  :topTags              "Cheap Eats" , "British" , "Cafe";
  :workingShiftsPerWeek "7"^^xsd:double .

```

3. Configuration RDFS

3.1 Class and Properties

- :Restaurant (Restaurant)

- :restaurantName (Nom)
 - :address (Adresse)
 - :originalLocation (Lieu d'origine)
 - :country (Pays)
 - :city (Ville)
 - :province (Province)
 - :region (Région)
 - :coordinate (Coordonnées)
 - :avgRating (Note globale)
 - :reviewsCount (Nombre total d'avis)
 - :defaultLanguage (Langue par défaut)
 - :reviewsCountInDefaultLanguage (Nombre d'avis dans la langue par défaut)
 - :reviewsComponents (Distribution des avis)
 - :scoreComponents (Scores détaillés)
 - :cuisines (Cuisine)
 - :meals (L'heure du repas)
 - :features (Spécialités / services)
 - :specialDiets (Régimes spéciaux)
 - :topTags (Top tags)
 - :priceLevel (Niveau de prix)
 - :priceRange (Fourchette de prix)
 - :originalOpenHours (Horaires d'ouverture)
 - :openDaysPerWeek (Jours d'ouverture par semaine)
 - :openHoursPerWeek (Heures d'ouverture par semaine)
 - :workingShiftsPerWeek (Plages/Shifts par semaine)
 - :awards (Récompenses)
 - :keywords (Mots-clés)
 - :popularityGeneric (Aperçu de la popularité)
 - :popularityDetailed (Détails de la popularité)
 - :isClaimed (Statut revendiqué)
 - :isVegetarianFriendly (Options végétariennes)
 - :isVeganOptions (Options vegan)
 - :isGlutenFree (Options sans gluten)
- :Coordinate (Coordonnées / Coordinate): :latitude (Latitude); :longitude (Longitude)
- :ReviewsComponents (Nombre de commentaires)
 - :excellentCount (Nombre de commentaires "Excellent")
 - :veryGoodCount (Nombre de commentaires "Très bien")
 - :averageCount (Nombre de commentaires "Moyen")
 - :poorCount (Nombre de commentaires "Médiocre")
 - :terribleCount (Nombre de commentaires "Horrible")
- :ScoreComponents (Scores détaillés)
 - :foodScore (Note de la nourriture)
 - :serviceScore (Note du service)
 - :valueScore (Note du prix)
 - :atmosphereScore (Note d'atmosphère)

3.2 Lien Wikidata

Pour doter le graphe de fonctionnalités de liaison de base, j'ai créé un lien entre « chaînes de restaurants » et Wikidata. Plus précisément, j'ai ajouté owl:sameAs à l'entité Wikidata correspondante (wd:Qxxxx) pour les instances de restaurant du jeu de données appartenant à des chaînes connues, permettant ainsi l'alignement et l'enrichissement des bases de connaissances croisées. Le numéro Q de Wikidata pour la marque de la chaîne se trouve dans le fichier res_brand.csv du dépôt, obtenu par une recherche manuelle.

Le script `generate_wikidata_link.py`, généré par LLM, lit le champ :`restaurantName` de chaque instance de restaurant et le compare au nom de la marque dans `res_brand.csv`. Pour chaque instance de restaurant correspondante, le script génère un fragment Turtle (fichier de sortie nommé `wikidata_brand_links.ttl`), chaque lien étant composé de deux parties. Exemple Turtle :

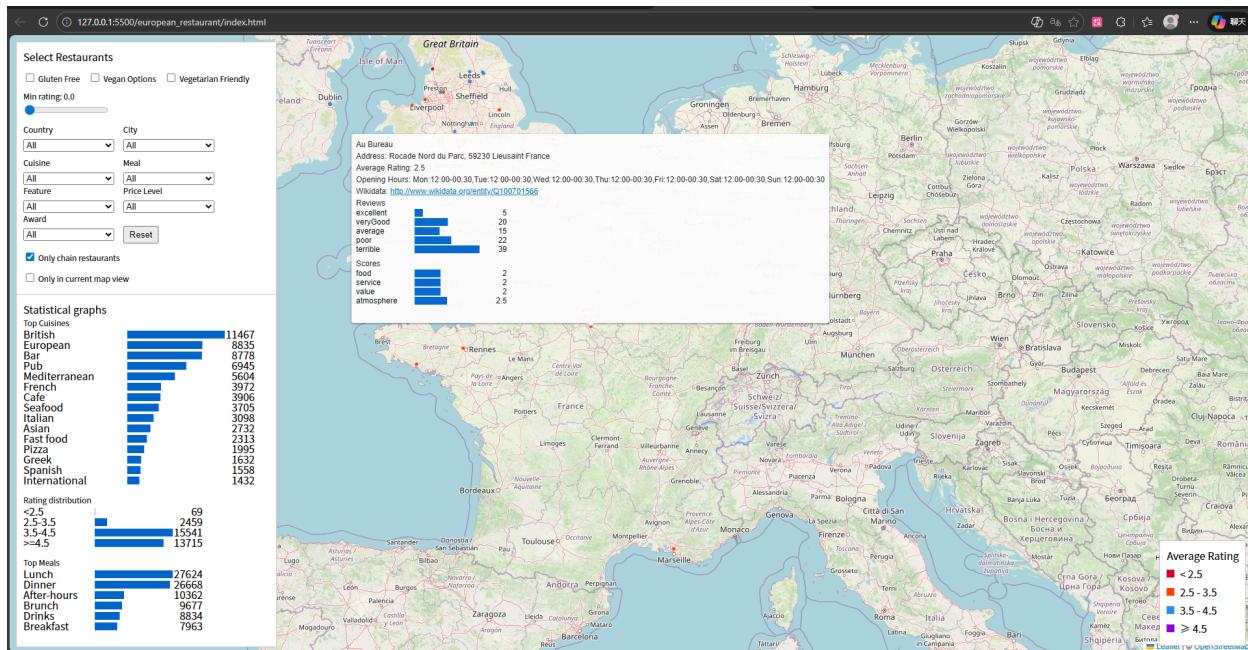
```
:restaurant_g315927-d8809797 rdfs:label "Starbucks"@en ;
owl:sameAs wd:Q37158 .
```

Le fragment de lien généré est ajouté à la fin du fichier TriG (en tant que triplet supplémentaire), ce qui permet d'ajouter à l'instance de restaurant correspondante dans l'ensemble de données à la fois des attributs d'ontologie et des liens externes.

4. Développement web

4.1 Technologie utilisée

- Stockage des données : Apache Jena Fuseki.
- Affichage et marqueurs de la carte : Leaflet.
- Visualisation graphique : D3.js
- Requêtes SPARQL : Envoi des requêtes au point de terminaison via d3sparql.



4.2 Logique d'interaction principale

- L'interface utilisateur propose un curseur de note minimale pour filtrer les données.
- L'interface utilisateur propose plusieurs menus déroulants pour filtrer les données par Country, City, Price Level, Cuisine, Meal, Feature, Award.
- L'interface utilisateur propose une option permettant de sélectionner les chaînes de restaurants pour filtrer les données.
- Repérez le restaurant sur la carte à l'aide de ses coordonnées GPS. En survolant le repère avec la souris, vous verrez des informations détaillées le concernant (nom, adresse, note moyenne,

horaires d'ouverture, nombre d'avis et graphique des notes). Si le restaurant possède un lien vers Wikidata, le lien est affiché dans la fenêtre contextuelle.

L'interface utilisateur propose une option de sélection permettant aux chaînes de restaurants de filtrer les données. Logique de base:

- Au chargement de la page, toutes les options disponibles sont récupérées et affichées dans la liste déroulante.
- Après la sélection d'une condition par l'utilisateur, la fonction `refresh()` est appelée. La logique de requête pour les attributs `cuisines`, `meals`, `features` et `awards` est plus complexe en raison du fractionnement des cellules. Afin d'éviter un temps de chargement initial excessif, ces attributs ne sont interrogés via SPARQL qu'après sélection dans la liste déroulante, récupérant ainsi les valeurs des attributs correspondants dans l'instance du restaurant.
- Utilisez SPARQL pour interroger les coordonnées géographiques (`latitude/longitude`) des restaurants en fonction des critères de filtrage actuels.
- Mettez à jour les marqueurs sur la carte.

4.3 Requêtes SPARQL clés

1. Informations sur le restaurant Demande principale

```
const queryMainBase = `
PREFIX : <http://ltr.european-restaurants.org/>
PREFIX owl: <http://www.w3.org/2002/07/owl#>

SELECT ?r ?name ?address ?lat ?lon ?avgRating ?openHours ?keywords
      ?isGlutenFree ?isVeganOptions ?isVegetarianFriendly
      ?country ?city ?priceLevel
      ?averageCount ?excellentCount ?veryGoodCount ?poorCount ?terribleCount
      ?atmosphere ?service ?food ?value
      ?sameAs
WHERE {
  ?r a :Restaurant ;
    :restaurantName ?name ;
    :coordinate [ :latitude ?lat ; :longitude ?lon ] ;
    :address ?address ;
    :avgRating ?avgRating ;
    :isGlutenFree ?isGlutenFree ;
    :isVeganOptions ?isVeganOptions ;
    :isVegetarianFriendly ?isVegetarianFriendly ;
    :country ?country ;
    :city ?city ;
    :priceLevel ?priceLevel ;
    :reviewsComponents [
      :averageCount ?averageCount ;
      :excellentCount ?excellentCount ;
      :veryGoodCount ?veryGoodCount ;
      :poorCount ?poorCount ;
      :terribleCount ?terribleCount
    ] ;
    :scoreComponents [
      :atmosphereScore ?atmosphere ;
      :serviceScore ?service ;
      :foodScore ?food ;
      :valueScore ?value
    ] .
  OPTIONAL { ?r :originalOpenHours ?openHours . }
  OPTIONAL { ?r :keywords ?keywords . }
  OPTIONAL { ?r owl:sameAs ?sameAs . }
}
LIMIT 5000`
```

2. Requête du panneau de sélection

```

function querySelect(uris) {
  const values = uris.map(u => `(${u})`).join(" ");
  return `
PREFIX : <http://ltr.european-restaurants.org/>

SELECT ?r
  | (GROUP_CONCAT(DISTINCT ?cuisine; separator="||") AS ?cuisines)
  | (GROUP_CONCAT(DISTINCT ?meal; separator="||") AS ?meals)
  | (GROUP_CONCAT(DISTINCT ?feature; separator="||") AS ?features)
  | (GROUP_CONCAT(DISTINCT ?award; separator="||") AS ?awards)
WHERE {
  VALUES (?r) { ${values} }

  OPTIONAL { ?r :cuisines ?cuisine . }
  OPTIONAL { ?r :meals ?meal . }
  OPTIONAL { ?r :features ?feature . }
  OPTIONAL { ?r :awards ?award . }
}
GROUP BY ?r
`;
}

```

4.4 Instructions de génération de code LLM

Les pièces générées avec l'aide de LLM comprennent:

- Script de génération de liens Wikidata.
- Cadres de graphiques dans les pages web.
- Implémentation de JavaScript pour afficher l'emplacement des restaurants et les barres de sélection ville/pays en fonction de la latitude et de la longitude.

Outre le script de génération de Wikidata, les modifications suivantes ont été apportées à la sortie LLM:

- Le filtrage des prix a été remplacé par priceLevel ; l'utilisation de priceRange dans LLM entraînait une barre de requête excessivement longue, ne permettant pas de couvrir tous les points.
- Ajout de barres de sélection pour les plats, les spécialités et les récompenses.
- Ajustement de la mise en page et du texte des graphiques pour éviter les chevauchements, et modification des schémas de couleurs.
- Ajout de graphiques de visualisation scoreComponents à la fenêtre contextuelle.
- Ajout de requêtes pour openHours et des mots-clés afin d'enrichir les informations sur les restaurants.
- Ajout de la possibilité de fixer la fenêtre contextuelle en cliquant sur les signes de ponctuation, permettant ainsi l'interaction avec le lien wikidata.