Université Claude Bernard Lyon 1

Cahier des charges

Intégration spatiale entre Geonames et OpenStreetMap

Amaia NAZÁBAL et Sofiaa FADDI

Encadrant : Fabien DUCHATEAU

Janvier 2017

Introduction

Les services de géolocalisation sont utilisés de plus en plus dans le domaine informatique dans plusieurs types d'applications. Ces dernières utilisent des données cartographiques qui sont définies par différents fournisseurs notamment GeoNames (GN) et OpenStreetMap (OSM), chacun d'eux a ses propres caractéristiques, on les présente comme suit :

- GeoNames est une base de données géographique ouverte décrivant les coordonnées des points de géolocalisation. Elle fournit de nombreuses informations utiles sur différents lieux, par exemple : le nom, la longitude, l'altitude, le code de pays, fuseau horaire etc.
- OpenStreetMap est aussi une base de données cartographique ouverte, c'est une initiative pour créer et fournir des données cartographiques libres sous la forme des points, lignes et polygones.

Chaque fournisseur décrit les liens de manière indépendante, en particulier OSM et GN sont collaboratifs et saisies par des contributeurs. Deux fournisseurs de service peuvent donc avoir des données qui sont incomplètes et/ou contradictoires pour le même point d'intérêt (POI). Ceci engendre un impact défavorable lorsqu'il s'agit de trouver des informations fiables, pertinentes et complètes.
D'où la nécessité des approches d'alignement entre les entités spatiales pour le même POI. Cet alignement permet de faire la correspondance entre deux entités géospatiales en calculant des similarités sur plusieurs critères (e.g. nom, type, adresse, coordonnées spatiales).

Cependant, une méthode d'alignement repose sur une combinaison efficace des similarités (e.g, une moyenne) et d'un mécanisme de décision adapté (e.g, un seuil). Une des problématiques de l'intégration consiste à sélectionner et configurer une fonction de combinaison et un mécanisme de décision.

L'objectif de ce projet de recherche dans un premier temps est de proposer une intégration des données de GN et OSM en faisant la correspondance entre des entités spatiales de chacune de ces bases de données. Ce matching va se focaliser sur la combinaison de plusieurs mesures en se basant sur des critères [1]. Au premier abord, nous allons définir un algorithme qui respecte cette correspondance. Dans un second temps, nous allons présenter une procédure de validation manuelle effectuée par l'utilisateur, et dans un troisième lieu, nous allons proposer une optimisation de l'approche proposée pour faciliter le passage à l'échelle. Pour ce faire, nous avons divisé notre projet en trois grandes parties : matching, prototype permettant la validation et l'algorithme de configuration automatique de la fonction de combinaison.

1. Matching

Dans cette phase nous allons définir la correspondance entre les entités dans OSM et celle de GN. On choisira d'abord une source principale (schéma primaire). C'est à partir d'elle que nous allons commencer à traiter les comparaisons avec l'autre source. Pour réaliser le matching nous allons utiliser différentes méthodes d'alignement d'entités entre les sources de données, et c'est ainsi que nous allons comparer un élément du schéma primaire avec un ou plus de la seconde source.
Nous allons donc construire un outil qui fait la correspondance de deux entités géospatiales entre les sources, autrement dit, créer un algorithme qui permet d'évaluer la pertinence des données ainsi que la possibilité de dire qu'une entité X dans OSM correspond à l'entité Y dans GN. Pour cela, trois algorithmes sont nécessaires :

- Le blocking, un principe qui repose sur la comparaison de sous-groupes, certains appartiennent au schéma primaire et d'autres à la source secondaire. Par exemple nous allons prendre un sous-groupe de données du schéma primaire et le comparer avec un autre sous-groupe de la deuxième source afin d'éviter le produit cartésien (Figure 1.1). Le regroupement sera effectué en fonction de plusieurs critères tels que : les coordonnées géographiques les plus proches dans un rayon donné, la similarité des chaînes de caractères de certains attributs, la région, etc.
- Le matching représente l'algorithme qui calculera le score de pertinence entre deux entités géospatiales données. Pour l'attribut des coordonnées de géolocalisation (longitude et altitude), on calculera la distance (euclidienne) entre les entités [4] en prenant en considération certaines contraintes de précision [3] qui pourront avoir un impact sur le résultat de l'algorithme. Pour les autres attributs on choisira une méthode de calcul de distance entre chaînes de caractères telle que la distance Levenshtein, similarité phonétique, distance Jaccard, distance de JaroWinkler ou n-gram [2] .
- À la fin du matching, chaque paire d'entité d'un même sous-groupe possède un score de pertinence. Ce score sera basé sur une moyenne pondérée des différentes valeurs de similarités. Les poids de pondération seront initialement égaux et raffinés dans la partie 3 (algorithme de configuration automatique de la fonction de combinaison).
- Un algorithme de décision. Dans notre contexte, nous utilisons un seuil que l'utilisateur pourra varier.

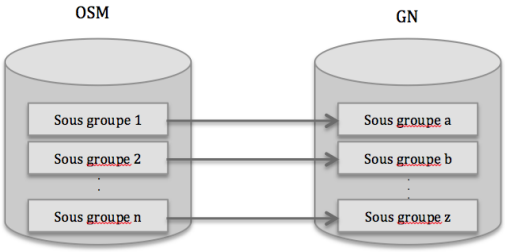

Figure 1.1 : Principe de blocking

En utilisant ces différents algorithmes, nous pouvons donc stocker des correspondances.
Pour mieux illustrer ces techniques, nous allons prendre un exemple de la Tour Eiffel, nous allons d'abord définir les attributs de ce POI dans les sources comme l'indique le tableau suivant :

Attributs	**GeoNames**	**OpenStreetMap**
Nom	Tour Eiffel	Eiffel Tour
Point	48.858084, 2.294359	48.858737, 2.293612
Type	Monument historique	Tourisme
Forme	Point	Polygone
Region	Paris, France	Paris, France

Tableau 1.1

Les attributs sélectionnés sont : le nom, les coordonnées géographiques (latitude, longitude) les plus proches et la région, on appliquera l'algorithme de matching qui retournera les indices de similarité de chaque attribut (Tableau 1.2), et à partir de là, on calculera le score de pertinence entre les deux entités, en prenant chaque attribut avec le même poids (33,3%) sur le score final.

Attributs	**GeoNames**	**OpenStreetMap**	**Indice de similarité**
Nom	Tour Eiffel	Eiffel Tour	99%
Point	48.858084, 2.294359	48.858737, 2.293612	98%
Region	Paris, France	Paris, France	100%

Tableau 1.2

De cette manière, le score de pertinence entre les deux entités géospatiales sera 98.66%, et selon la valeur du seuil, nous allons déduire que les deux entités sont alignées ou pas. Ainsi la correspondance, les entités et le score de pertinence, seront stockés dans la base de données.

2. Prototype

Afin de montrer concrètement le résultat des algorithmes proposés et de permettre à l'utilisateur de valider les résultats, nous avons pensé à créer deux interfaces :

(a) **Affichage sous forme de tableau**
C'est une page de validation qui permet à un utilisateur de confirmer ou invalider les correspondances (Figure 1.2). Elle comprend des filtres pour différencier les correspondances validées, rejetées et en attente et elle inclut aussi un tableau contenant une entité spatiale et ses correspondances en fonction du score de pertinence, et c'est à l'utilisateur de valider le matching le plus approprié.

(b) **Affichage cartographique**
C'est une autre manière de visualiser les correspondances afin de les valider par l'utilisateur, au lieu d'avoir un tableau, nous allons utiliser des cartes géographiques (Figure 1.3).
Ainsi, l'expérience utilisateur décrite précédemment pourra donner l'opportunité à l'utilisateur d'indiquer une erreur de matching ou une correspondance possible non existante. De ce fait, une correction récurrente par l'utilisateur permettra de l'ajouter ou de le supprimer à la liste des correspondances.

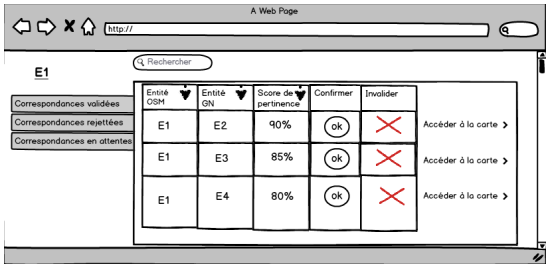

Figure 1.2 : Matching sous forme de tableau

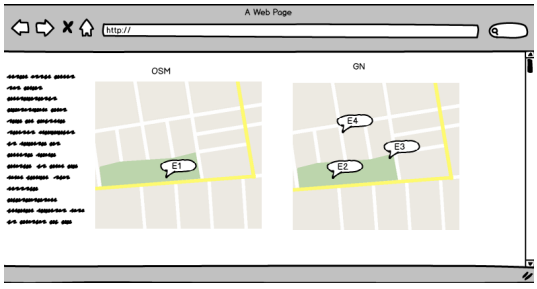

Figure 1.3 : Matching cartographique

3. Algorithme de configuration automatique de la fonction de combinaison

Pour optimiser le résultat on affecte pour chaque attribut un poids en fonction du degré d'importance afin de calculer la moyenne pondérée par conséquent, le score de pertinence, comme on a déjà vu dans la section de matching. Ces poids sont modifiables selon les validations de l'utilisateur, par exemple, nous pouvons modifier le poids de chaque attribut d'une entité et avec plusieurs tests nous pouvons obtenir un meilleur score de pertinence. Cette heuristique nous permettra d'optimiser notre algorithme sur des petites collections. En optimisant cette fonction de combinaison, nous pouvons faire le passage à l'échelle et valider un grand nombre de correspondances en une seule fois.

Livrables

— Un algorithme d'alignement avec le principe de blocking.
— Un algorithme de configuration automatique de la fonction de combinaison.
— Un prototype avec GUI.
— Une base de données contenant les correspondances et les entités importées.
— Rapport final.

Technologies

Nous divisons les technologies en deux catégories, celles qui seront utilisées côté client et d'autres côté serveur.

Front-End :
— AngularJS 2
— Ajax et JQuery
— CSS 3 et Bootstrap

Back-End :

— Python 3
— SGBD MySQL

Calendrier gantt

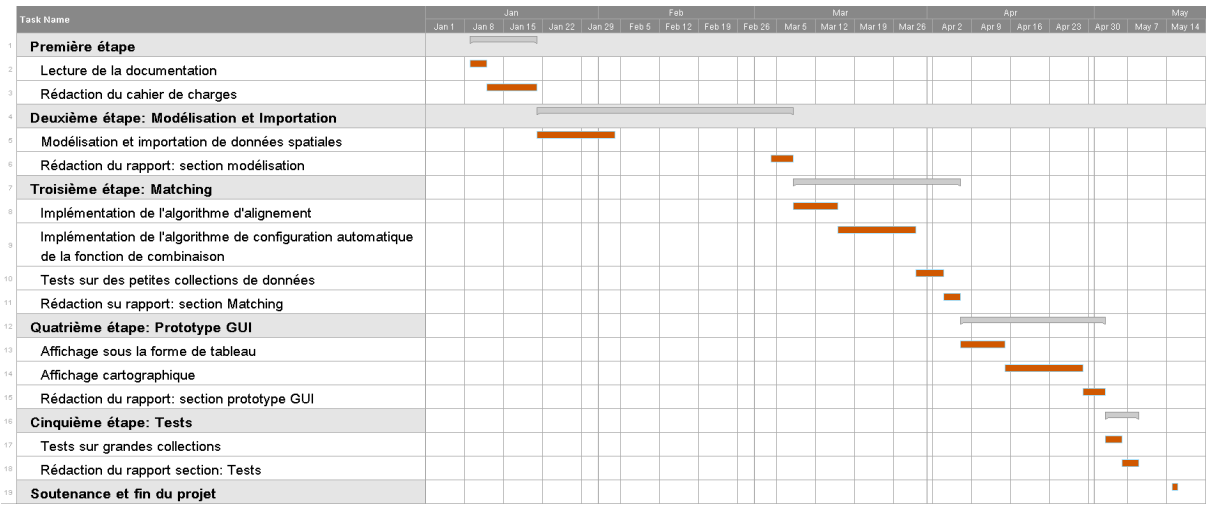

Figure 1.4 : Planification

Références

[1] Philip A Bernstein, Jayant Madhavan et Erhard Rahm. "Generic schema matching, ten years later". In : *Proceedings of the VLDB Endowment* 4.11 (2011), p. 695–701.

[2] William Cohen, Pradeep Ravikumar et Stephen Fienberg. "A comparison of string metrics for matching names and records". In : *KDD workshop on data cleaning and object consolidation*. T. 3. 2003, p. 73–78.

[3] Grant McKenzie, Krzysztof Janowicz et Benjamin Adams. "A weighted multi-attribute method for matching user-generated points of interest". In : *Cartography and Geographic Information Science* 41.2 (2014), p. 125–137.

[4] Anthony MORANA et Thomas MOREL. "GeoBench : un outil d'alignement entre entités spatiales pour la construction d'un benchmark cartographique". In : *ACM GIS* (2014).