Alignement spatial entre Geonames et OpenStreetMap

Amaia NAZABAL et Sofiaa FADDI

Mai 2017

Résumé. De nombreux fournisseurs cartographiques exposent des points d'intérêt (musées, parcs, etc.) utilisés dans des domaines d'application variés (tourisme, "smart city", etc.). D'un fournisseur à l'autre, les données sur un même point d'intérêt peuvent être incohérentes et/ou incomplètes, et l'utilisateur doit donc interroger plusieurs fournisseurs pour obtenir des informations de qualité. Pour répondre à ce problème, ce projet s'intéresse à l'alignement automatique de deux fournisseurs (Geonames et OpenStreetMap) afin de détecter leurs points d'intérêt équivalents.

Mots-clés : intégration spatiale, géolocalisation, OpenStreetMap, GeoNames, matching, alignement spatial.

1 Introduction

Ce rapport présente notre travail de projet d'orientation en Master, réalisé dans l'équipe BD du laboratoire LIRIS. et encadré par Fabien Duchateau. Ce travail s'inscrit dans la continuité du projet UNIMAP, dont l'objectif consistait à produire à la volée une carte unifiée à partir des informations issues de plusieurs fournisseurs cartographiques [15].

Les services basés sur la localisation (LBS) sont souvent utilisés dans le monde numérique à travers différentes plateformes mobiles et web. Les fournisseurs cartographiques sont des LBS qui fournissent diverses informations sur des points d'intérêts (POI), par exemple les musées, les restaurants, les montagnes. Un point d'intérêt est représenté sur la carte d'un fournisseur par une ou plusieurs entités spatiales. Chaque fournisseur décrit et représente les POI d'une manière différente et indépendante. Cette hétérogénéité engendre des données incomplètes ou contradictoires. Mais elle aussi peut influencer la fiabilité, l'exactitude et la pertinence des informations lorsqu'il s'agit de trouver un lieu.

Afin de résoudre ce problème, un processus d'alignement permet de faire la correspondance entre des entités spatiales équivalentes de différents fournisseurs. Comme il n'existe pas d'identifiant commun entre les entités des différents fournisseurs, l'alignement se base sur une comparaison approximative (calcul de similarités) entre les autres attributs (par exemple, le nom, le type). L'une des difficultés concerne la prise en compte de la forme du POI (un fournisseur peut représenter un fleuve avec un point à sa source ou à son embouchure tandis qu'un autre utilise une ligne composée de milliers de points). Les différentes similarités calculées doivent être combinées pour décider si la paire d'entités comparées est une correspondance ou non. Il est donc crucial de combiner intelligemment ces similarités pour obtenir une qualité d'alignement acceptable. Enfin l'intervention de l'utilisateur (validation des correspondances détectées) peut servir à améliorer la façon de combiner les similarités.

Pour traiter ce problématique nous proposons ASMA (Alignement Spatial Multiformes et Adaptatif) un outil qui sera exposé tout au long de ce rapport avec les différentes visions et solutions apportées à ce sujet. Nous avons étudié l'alignement entre les fournisseurs Geonames et OpenStreetMap, en nous focalisant sur la prise en compte des entités multi-formes et sur l'amélioration de la combinaison des similarités. Après une étude de l'existant, ce rapport propose des algorithmes permettant de suggérer des correspondances aux utilisateurs et de configurer automatiquement la fonction de combinaison des similarités (Section 3) . Ensuite, il expose les expérimentations pour évaluer notre approche ainsi qu'une optimisation celle-ci pour faciliter le passage à l'échelle (Section 4). Des perspectives sont présentées dans la dernière section.

2 Pré-requis et travaux existants

2.1 Pré-requis

Une entité spatiale représente un POI et se compose d'attributs primaires, à savoir un identifiant relatif au fournisseur, des coordonnées géographiques (également appelées attributs géographiques), un type (catégorie de POI) et au moins un nom. Elle peut également inclure des attributs secondaires, qui sont en général optionnels, par exemple : une adresse, un numéro de téléphone, un site web [6].

Chaque fournisseur représente ces entités avec sa propre structure de données (schéma) qui comporte plus ou moins de niveau de précision par rapport à la description de ces POI. Quant aux données, chaque fournisseur utilise différentes techniques pour alimenter sa base de données spatiales. Par exemple, les fournisseurs Geonames (GN) et OpenStreetMap (OSM) possèdent des données issues d'un travail collectif (communautés de bénévoles [3]). Par conséquence, ces données sont soumises à un certain nombre de variations et/ou d'imperfections (par exemple, incohérences, degré de précision, erreurs).

Cette hétérogénéité des représentations entraîne un impact négatif sur l'alignement spatial, i.e., quand il faut comparer deux entités (de fournisseurs différents) afin de déterminer si elles représentent le même POI. Quatre types de différences caractéristiques dans ce domaine ont été identifiées [7, 8] :

- différence sémantique, quand des entités qui représentent le même POI ont des différences dans les valeurs de ses attributs. (par exemple, la tour Eiffel dans OSM est classifiée comme une "*attraction touristique*" alors que dans GN, elle est catégorisée comme un "*monument commémoratif*");
- différence spatiale, au niveau des coordonnées (par exemple, dans OSM la tour Eiffel est un ensemble de coordonnées géographiques qui délimitent le contour du POI, cependant dans GN l'attribut spatial s'exprime comme un point, c'est à dire une latitude-longitude);
- différence de schémas : les fournisseurs ne possèdent pas les mêmes schéma et format pour stocker leurs données. Dans GN les entités sont exprimés à travers des noeuds, même si un POI dans la réalité correspond à une surface (e.g. La Seine). Du côté de OSM, les entités sont représentées avec différentes structures, ainsi un POI peut être représenté comme un noeud, un chemin (ligne) ou un polygone (e.g. la Seine est un chemin);
- disponibilité : une entité d'un fournisseur peut correspondre à zéro ou plusieurs entités dans un autre fournisseur. Par exemple, *La Fontaine des Innocents* est représentée dans OSM comme un ensemble des coordonnées ("way"), mais elle n'existe pas comme une entité dans GN. Un autre exemple avec le *centre sportif Emile Anthoine*, qui est une entité ponctuelle dans GN mais qui apparaît sous plusieurs entités dans OSM (terrain de football, salle de sport et les bâtiments qui le composent).

ASMA gère cette hétérogénéité avec les techniques de blocking et alignement qui seront détaillés dans les sections suivantes.

2.2 Travaux similaires

De nombreux travaux portent sur l'alignement de données en général [14], mais ceux sur l'alignement spatial sont encore rares. L'aspect géographique est primordial et nécessite des solutions adaptées.

Karma est un outil d'intégration de données qui propose une approche semi-automatique [13]. Il offre une interface graphique qui permet aux utilisateurs d'interagir et d'affiner le modèle proposé en utilisant des graphes RDF à partir d'une ontologie donnée. Il utilise cette ontologie pour réaliser l'intégration de collections de données. L'ontologie peut être modifiée pour fournir à l'utilisateur une plus grande flexibilité.

L'approche de Vivek et Seghal consiste à combiner les différents attributs (géographiques et non géographiques) pour améliorer la précision de résultats [10]. Ils proposent l'utilisation de différentes mesures de similarité pour trouver les correspondances entre entités afin d'intégrer des collections de lieux. La notion des poids pour calculer une mesure de similarité globale et l'utilisation des algorithmes d'apprentissage pour trouver la combinaison idéale fait partie aussi de leur démarche. Les expérimentations montrent de bons résultats, mais le jeu de données est très peu hétérogène (plus de 90% des correspondances ont des noms identiques) et peu dense (peu d'entités proches sur une même zone).

GeoBench est un outil qui fait l'alignement puis l'intégration de données entre trois fournisseurs cartographiques (Google Maps, Geonames et Here), en utilisant leurs services web [7, 8]. Cet outil prend en compte uniquement les points géographiques (i.e. les chemins, polygones et multipolygones sont exclus de l'échantillon) et n'aligne que quelques dizaines d'entités par run (mode online et restrictions des API).

À la différence de ces outils, notre approche met en oeuvre l'alignement avec des POI de toutes formes (points, lignes, polygones et multipolygones) et en recourant aux dumps des fournisseurs OSM et GN (ce qui implique des algorithmes performants). Cependant le principe du blocking et l'algorithme de matching de ces approches ont servi de sources d'inspiration pour notre processus d'alignement.

3 Notre approche ASMA

ASMA est un outil qui permet de trouver des correspondances entre les fournisseurs cartographiques OSM et GN. Il inclut quatre phases (Figure 1.1). La première phase est l'importation. Elle consiste en la récupération des données à partir de sources externes (dumps) ainsi que leur transformation selon le schéma relationnel de notre BD locale. Les détails de ce processus sont expliqués dans l'annexe (Section 7.1 Processus d'importation). La deuxième phase est l'alignement, nécessaires à cause des conflits entre les entités des fournisseurs (section 3.1). La phase d'alignement comprend deux étapes : le blocking pour regrouper les entités qui seront comparées (section 3.2) et l'alignement proprement dit qui compare plus finement chaque paire d'entités d'un même groupe et produit des correspondances (section 3.3). La troisième phase concerne la validation des correspondances par l'utilisateur (section 3.4). Enfin, la dernière phase qui correspond à l'apprentissage, qui cherche une combinaison optimale des mesures de similarité pour améliorer les prochains processus d'alignement (section 3.5).

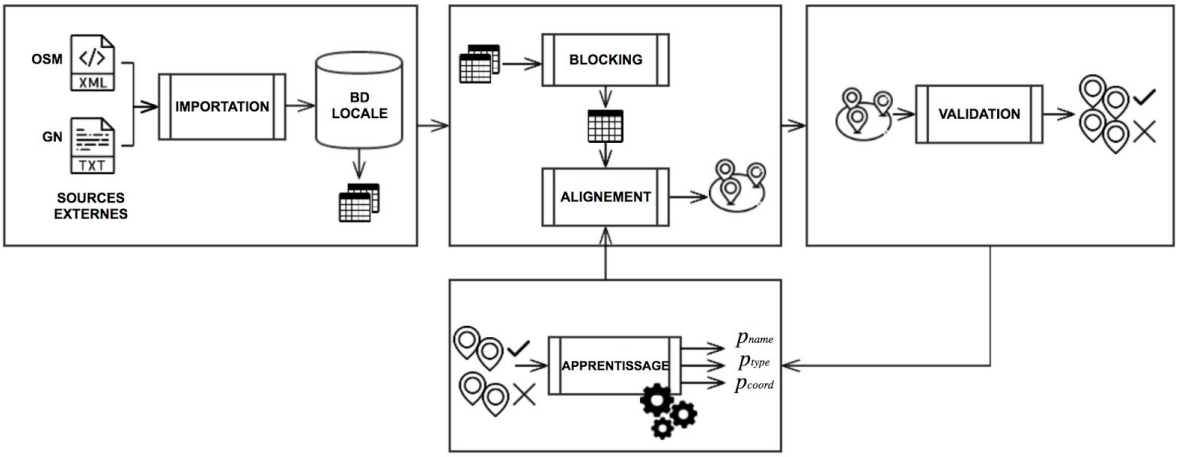


Figure 1.1 Enchaînement de processus

3.1 Conflits lors de l'alignement:

Pour les différences au niveau des coordonnées géographiques, trois catégories ont été identifiées [6] :

- Erreurs dans les attributs géographiques qui ne correspondent pas à la réalité, (e.g. une grande marge d'erreur à cause de bruit dans l'appareil récepteur qui empêche d'avoir les bonnes coordonnées)
- Imprécision de l'information spatiale (e.g. PDOP[1] inférieure à 4 chiffres après la virgule)
- Ambiguïté dans l'information spatiale (e.g., les limites géographiques entre régions).

Les causes de ces conflits sont détaillées dans l'annexe (section 7.2 Conflits lors de l'alignement).

Au niveau des autres attributs (non géographiques), les conflits sont syntaxiques (différences au niveau des chaînes de caractères), sémantiques (différences au niveau de la signification) et structurels (différences au niveau du type et du nombre d'attributs).

3.2 Blocking

L'intérêt de l'algorithme de blocking est de réduire l'espace de recherche (des entités à comparer), dans notre cas entre une entité GN et plusieurs entités OSM. Ce principe a été adopté parce que les données fournies par OSM sont beaucoup plus détaillés et complètes par rapport à GN.

Dans ce sens, on a utilisé les attributs géographiques pour limiter la recherche. Ainsi, l'algorithme de blocking prend en entrée une entité de GN et à partir d'un rayon défini, il cherche dans la BD locale toutes les entités OSM qui sont incluses dans de ce rayon.

La Figure 1.2 illustre le principe du blocking sur une carte : le marqueur vert correspond aux coordonnées de l'entité de GN, le cercle rouge est le rayon de recherche et les points violets sont les entités OSM incluses

[1] PDOP : (Position dilution of precision) C'est une valeur qui spécifie l'effet multiplicateur dans la navigation satellitaire géométrique dans la mesure de position précis. Le PDOP est liée à la déviation verticale et horizontale de la position précise.

dans la surface du rayon donné, et donc les entités qui seront comparées avec celle de GN.

Figure 1.2: Rayon de recherche pour l'algorithme de blocking

La quantité des entités qui seront retournées par le blocking peut être limitée par un rayon de recherche qui est flexible (modifiable) ou par un seuil prédéfini. Ainsi, si le nombre d'entités dans le rayon de recherche est supérieur à une limite K définie, alors, l'algorithme retournera uniquement la quantité des entités définie par ordre de distance euclidienne, autrement dit, il retournera les K entités les plus proches de l'entité GN .

3.3 Alignement

Le but de l'alignement est de calculer des similarités pour chaque attribut primaire (type, nom et coordonnées) et de produire un score de pertinence global qui sera assigné à chaque correspondance candidate.

Notre algorithme d'alignement prend comme entrée la liste d'entités résultante de l'algorithme de blocking. Selon la distance euclidienne entre chaque entité OSM et l'entité GN, l'algorithme assigne un indice de similarité aux coordonnées (Sim_{coord}), en normalisant cette distance. Ainsi, le Sim_{coord} d'une correspondance candidate entre deux entités GN et OSM a une valeur plus élevée si leur distance est faible, et inversement (voir exemples dans le Tableau 1.1). Le Sim_{coord} est normalisée en fonction du rayon de recherche.

Coordonnées géographiques OSM	Coordonnées géographiques GN	Distance euclidienne	Sim_{coord}
45.7149820 4.8345973	*45.7152100 4.8353100*	61m	*93.9%*
45.7483223 4.8253267	*45.7480600 4.8258300*	48m	*95.2%*

Tableau 1.1: Exemples de similarités entre coordonnées géographiques (rayon de 1km). La première ligne corresponde au Port Édouard Herriot et la deuxième à la Gare de Lyon Perrache.

Pour la mesure de similarité entre noms, l'algorithme applique une mesure très utilisée, la distance de Levenshtein[16], et calcule l'indice de similarité du nom (Sim_{name}) entre toutes les paires d'entités. La formule ci-dessous donne le score obtenu en appliquant Levenshtein entre les chaînes x_1 et x_2.

$$f(x_1, x_2) = \frac{max(taille(x_1), taille(x_2)) - \sigma(x_1, x_2)}{max(taille(x_1), taille(x_2))}$$

La distance de Levenshtein est au plus la taille du mot plus grand. Ainsi, pour calculer le score de similarité, l'algorithme fait une règle de trois comme dans la formule précédente, où σ est la fonction de Levenshtein (c.f. Tableau 1.2).

Noms des entités OSM	Noms des entités GN	σ	Sim_{name}
PORT ÉDOUARD HERRIOT	*PORT ÉDOUARD-HERRIOT*	1	*95%*
LYON-PERRACHE	*GARE DE LYON-PERRACHE*	8	*61.9%*

Tableau 1.2: Exemples de similarités entre noms.

Au niveau des types, ce n'est pas possible de calculer directement l'indice de similarité, étant donné les

conflits sémantiques et structurels importants (voir l'annexe section 7.2 Conflits lors de l'alignement). Nous avons combiné plusieurs stratégies pour augmenter les chances d'avoir un indice juste.

La similarité de types (Sim_{type}) applique successivement ces trois stratégies et le score est arbitrairement fixé selon la stratégie gagnante, comme le montrent les exemples du tableau 1.3. En cas d'échec des trois stratégies, le score de Sim_{type} vaut 0.

Type d'entité OSM	**Type d'entité GN**	**Stratégie utilisée pour trouver la correspondance de type**	Sim_{name}
tourisme=hotel	*S.HTL (hotel)*	Correspondance manuelle pour types valides.	*100%*
water=lake	*H.LK (lake)*	Recherche du nom du type dans les valeurs des tags	*70%*
highway=secondary	*R.RD (road)*	Utilisation du service externe de synonymes pour trouver la correspondance	*50%*
amenity=pharmacy	*S.HTL(hotel)*	Correspondance manuelle pour types invalides.	*0%*

Tableau 1.3: Exemples de similarités entre types.

Pour faire évoluer cette approche, l'interface utilisateur offre une option pour indiquer une correspondance entre types (voir Section 3.3 Interface utilisateur).

Enfin, le score de pertinence global pour une correspondance candidate est une moyenne pondérée des scores de similarité entre coordonnées, entre noms et entre types. Il représente le degré de similarité globale entre deux entités données. Plus grand est le score de pertinence, plus grande est la probabilité que les deux entités représentent le même POI.

$$\phi(Sim_{name}, Sim_{type}, Sim_{coord}) = Sim_{name} \times p_{name} + Sim_{type} \times p_{type} + Sim_{coord} \times p_{coord}$$

La formule précédente exprime comment calculer le score de pertinence. Chaque indice Sim a un poids p qui est utilisé lors de calcul pour déterminer le score global ϕ.

Correspondances	Sim_{name}	Sim_{type}	Sim_{coord}	ϕ
PORT ÉDOUARD-HERRIOT	*95%*	*0%*	*93.9%*	*56.78%*
GARE DE LYON-PERRACHE	*61.9%*	*100%*	*95.2%*	*83.8%*

Tableau 1.4: Assignation du score de pertinence avec p_{name} = 0.4, p_{type} = 0.4 et p_{coord} = 0.2.

La recherche des valeurs optimales pour p_{name} , p_{type} et p_{coord} est un problème qui sera abordé plus tard dans la section 3.5 (apprentissage).

3.4 Interfaces utilisateur pour la validation

Dans ASMA l'utilisateur peut effectuer actions de validation et invalidations. Ces actions peuvent s'appliquer aux correspondances suggérées, mais aussi entre les types de OSM et GN.

La validation de types est utile afin d'alimenter la BD avec des correspondances (in)valides entre types. Par exemple (Figure 1.3), si la classe "*L.PRT*" (port) de GN et la clé-valeur "*landuse=industrial*" de OSM n'est pas une correspondance de type définie dans la BD, l'utilisateur peut l'ajouter et elle sera prise en compte pour les prochains alignements (augmentation de la valeur de Sim_{type}). Par contre, si l'utilisateur décide de signaler les types comme non-équivalents, lors du prochain alignement, la valeur de Sim_{type} entre ces types sera de 0.

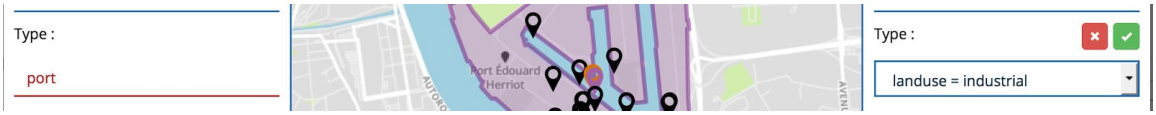


Figure 1.3: Validation de types de GN et OSM

Par rapport à l'alignement des entités, plusieurs correspondances peuvent être proposées (résultat de l'algorithme du blocking). Ainsi, l'utilisateur peut choisir une, plusieurs ou aucune correspondance par une entité donnée. GN est une base de données plus petite que OSM, en cette raison c'est possible de trouver les trois cas cités précédemment. Les correspondantes candidates (pour une même entité GN) sont triées par score de pertinence global décroissant.

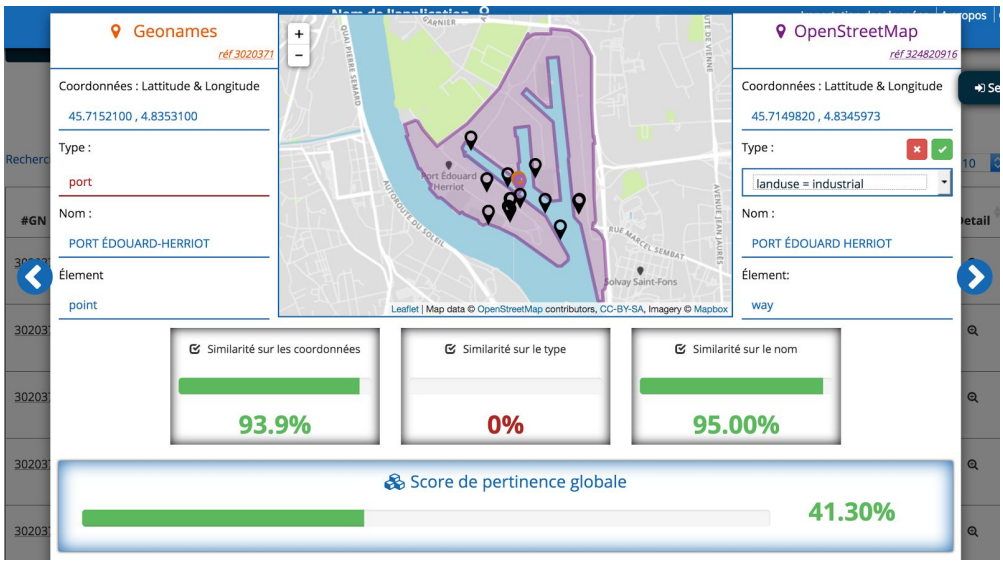


Figure 1.4: Alignement du Port Édouard-Herriot entre GN et OSM

La Figure 1.4 montre la correspondance avec le score de pertinence plus élevé pour le POI "Port Édouard-Herriot". Le score de pertinence n'est pas très élevé étant donné que l'algorithme n'a pas trouvé de correspondance entre types dans la BD. Cependant, la similarité du nom et les coordonnées jouent également un rôle important dans le calcul qui permet de trouver (en première position) la correspondance valide. Plus de détail dans l'annexe dans la section 7.3 Interfaces de l'utilisateur.

3.5 Apprentissage

Des valeurs par défaut ont été données au poids de chaque mesure de similarité dans la combinaison. L'intérêt de l'apprentissage est de trouver la combinaison des poids optimale de façon à maximiser le score de pertinence pour les correspondances valides et de le minimiser dans le cas de correspondances invalides. Les mesures de similarité retournent des valeurs selon une distribution uniforme, bien qu'avec des tendances (i.e., probabilité plus élevée d'avoir une correspondance valide si une similarité est supérieure à 90%). L'idée est d'apprendre les valeurs ou intervalles qui offrent la plus grande probabilité d'avoir une correspondance pour une mesure donnée, puis d'en déduire l'influence de cette mesure.

Le processus d'apprentissage reçoit comme entrée les correspondances validées et invalidées par l'utilisateur lors de la phase précédente (données d'entraînement) et les trois mesures de similarité (Sim_{name}, Sim_{type} et Sim_{coord}). Dans notre contexte, nous avons opté pour un algorithme d'apprentissage supervisé de classification: l'arbre de décision CART (similaire à C4.5 mais il supporte des résultats numériques et ne calcule pas des ensembles de règles). CART construit des arbres binaires en utilisant la caractéristique et le seuil qui produisent le meilleur gain d'information à chaque noeud. Il offre de bonnes performances et reste facile à interpréter. L'algorithme a été implémenté en utilisant la librairie scikit-learn[2].

L'arbre de décision crée un modèle prédictif sur les attributs (règles simples de décision de type "if-else") à partir des données d'entraînement. L'étape de validation s'effectue par cross-validation. Un arbre est construit

[2] http://scikit-learn.org/stable/modules/tree.html

pour chaque mesure de similarité, et permet d'identifier l'influence de cette mesure (ou de cet attribut) sur l'ensemble de résultats. Dans la figure suivante les ronds gris représentent les ramifications de l'arbre où la valeur de l'indice gini[3] est utilisée pour calculer la division et les règles qui seront définies afin de déterminer si certains rangs de valeur amènent ou pas vers une correspondance valide.

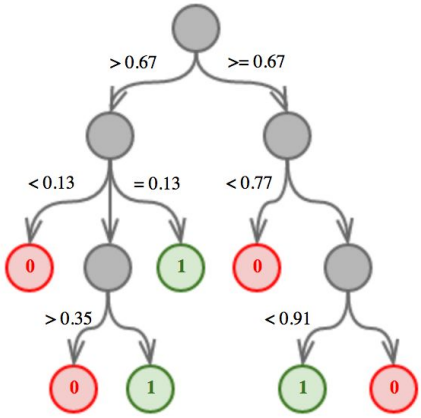


Figure 1.5: Un arbre de décision pour la similarité de nom. Les feuilles verts avec la valeur 1 représentent les rangs où l'arbre amène vers une correspondance positive, et les feuilles rouges avec la valeur 0 les correspondances négatives.

Dans l'arbre de la Figure 1.5, les rangs qui amènent vers une correspondance valide sont $[0.35, 0.67[$, $[0.77, 0.91[$, autrement dit dans 48% de cas la mesure de similarité sur le nom conduit à une correspondance valide. De la même façon, l'algorithme calcule l'influence des autres mesures (sur le type et les coordonnées) et il normalise les valeurs pour trouver leur poids. En plus d'affiner les valeurs des poids affectées initialement par défaut, l'une des découvertes suite à l'implémentation de l'algorithme d'apprentissage est que les coordonnées géographiques ont une infime influence dans la validité d'une correspondance, notamment à cause de l'effet du blocking. À long terme, quand plus de données d'apprentissage seront disponibles, l'intuition est que les valeurs apprises se stabilisent.

4. Validation expérimentale

Les algorithmes décrits ci-dessus ont été testés pour vérifier la qualité de l'alignement. Un jeu de données a été spécialement généré, puis notre prototype a été testé pour aligner ce jeu de données d'abord avec les poids par défaut, puis avec des poids appris.

4.1 Protocole

Le jeu de données a été généré selon plusieurs critères qui sont décrits. Pour cette évaluation nous avons expertisé 300 entités GN, pour lesquelles nous avons manuellement associé l'entité correspondante chez OSM. Le choix de commencer par une entité GN se justifie par le fait que sa BD contient moins d'entités que celle d'OSM. La Figure 1.6 montre la répartition de ces 300 entités GN (20% aléatoires, 20% de type région ou commune, etc.). Toutes les correspondantes sont 1:1 (une entité GN alignée avec une seule entité OSM) et toutes ont une correspondance. Enfin, étant donné que le prototype retourne plusieurs correspondances classées par ordre décroissant de score de pertinence, nous évaluons d'abord le premier résultat (*top-1*) puis les deux premiers (top-2). La qualité est mesurée selon trois mesures très connues, à savoir la précision, le rappel et la F-mesure. Elles se calculent à partir du classement des correspondances selon quatre catégories :

- True Positive (TP): l'ensemble des correspondances validées correctes et retournées par l'outil.
- False Positive (FP): l'ensemble des correspondances validées incorrectes et retournées par l'outil.
- True Negative (TN) l'ensemble des correspondances validées incorrectes et non retournées par l'outil.
- False Negative (FN): l'ensemble des correspondances validées correctes et non retournées par l'outil.

Un résumé de ces catégories est montré en Figure 1.7, et les trois mesures sont détaillées dans l'annexe

[3] Le coefficient Gini est une mesure d'inégalité utilisée dans les arbres de décision pour déterminer les ramifications en fonction des valeurs des observations et des objectifs.

7.5.

Figure 1.6: Les échantillons qui ont été pris pour l'évaluation.

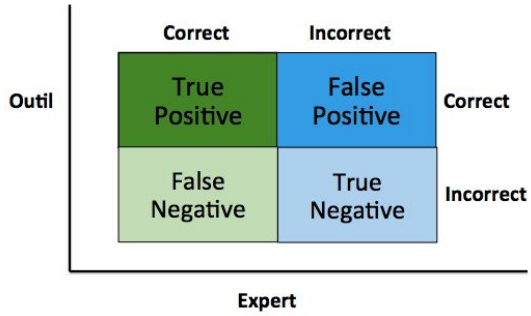
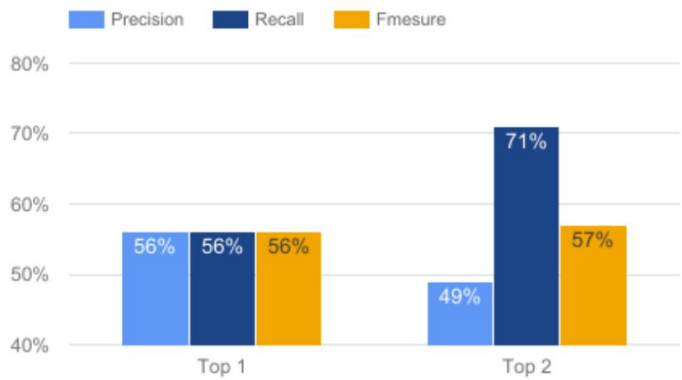


Figure 1.7: Division de résultats dans quatre quadrants.

4.2 Évaluation de la qualité sans apprentissage

Cette première évaluation a été réalisée avec les poids initiaux pour calculer le score de pertinence, c'est à dire sans apprentissage. Ces poids considérés sont les suivants: $p_{name} = 0.4$, $p_{type} = 0.4$ et $p_{coord} = 0.2$.

La figure 1.8 expose la qualité obtenue pour l'alignement des 300 entités GN dans le *top-1* et le *top-2*, et ce pour chaque mesure (précision, rappel, F-mesure). La précision est assez faible à *top-1*, seulement une correspondance sur deux est correcte. Cela s'explique aussi par le fait que notre BD n'ait pas toutes les correspondances entre types, et donc certaines correspondances entre entités sont pénalisées par une similarité nulle sur le type. Le rappel est également autour de 50%, une entité sur deux n'est pas classée en première position. Cependant, nous remarquons qu'à *top-2*, le rappel augmente de manière significative (71%). De nombreuses correspondances correctes sont donc également bien classées (en seconde place). La précision diminue par contre légèrement. À noter également que certaines correspondances ont été manquées pour des raisons de performance : le rayon de blocking, initialement de 1 km, a été baissé à 500 mètres car il contenait énormément d'entités dans la zone de blocking, ce qui ralentissait fortement le temps d'exécution de l'algorithme. Des correspondances impliquant des POI larges (parcs, forêts, rivières) ont pu être manqué à cause de la réduction du rayon de blocking (voir aussi annexe 7.1.3 limitations de l'importation).



	True Positive	False Positive	False Negative
Top 1	168	132	132
Top 2	213	219	87

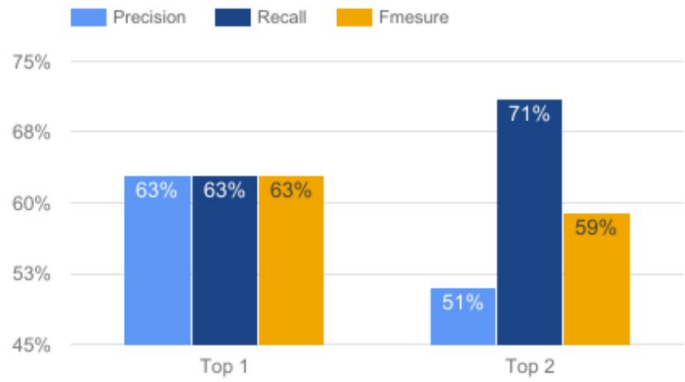
Figure 1.8: Evaluation de la qualité de l'alignement sans apprentissage (courbe précision rappel, F-measure et tableau de répartition des correspondances).

4.3 Évaluation de la qualité avec l'apprentissage

La deuxième évaluation a été faite avec les poids résultants de l'algorithme d'apprentissage sur le jeu de données. Pour apprendre ces poids, nous avons choisi pour une cross-validation (K=10 runs) avec un ensemble d'entraînement de 90% (soit 270 correspondances) et un ensemble de données test de 10% (soit 30

correspondances). Après d'une dizaine d'exécution (afin de lisser les résultats dûs au hasard), les poids optimaux sont le résultat de la moyenne de toutes les valeurs obtenues normalisées entre elles. Ce qui donne : $p_{name}=0.58$, $p_{type}=0.33$ et $p_{coord}=0.09$. Les mêmes principes utilisés dans la section précédente ont été pris en compte pour ces calculs.

La figure 1.9 illustre la qualité de l'alignement avec les nouveaux poids. Une légère amélioration (+7%) est constatée en précision et en rappel, ce qui signifie que l'algorithme classe une peu plus de correspondances correctes en première place (63%). Les résultats pour le *top-2* ne s'améliorent pas par contre. Cela signifie que 7% de correspondances moins bien classées (à partir de la troisième place et plus bas) ont été remontées en première place dans le classement. Il faudrait plus tard approfondir l'analyse les résultats afin de vérifier les autres raisons qui font que la moitié des correspondances proposées (51%) dans le *top-2* sont incorrectes.



	True Positive	False Positive	False Negative
Top 1	189	111	111
Top 2	213	189	87

Figure 1.9: Evaluation de la qualité de l'alignement avec apprentissage (courbe précision rappel, F-measure et tableau de répartition des correspondances).

5 Conclusion et perspectives

Dans ce rapport, nous avons résumé l'approche utilisée pour le processus de l'alignement spatial des entités de différents sources (OpenStreetMap et et Geonames). Le défi le plus grand a été de gérer l'hétérogénéité des représentations de chaque fournisseur cartographique. Notre approche ASMA est un adaptative parce qu'elle s'alimente de l'information validée par l'utilisateur pour améliorer la combinaison de ses mesures de similarité. Elle permet aussi de gérer l'hétérogénéité des entités sous toutes ses formes (point, chemin et polygones) dans la limite du rayon de blocking, ce qui n'avait pas été traité auparavant.

Comme perspectives à long terme, on peut citer trois alternatives. D'abord l'importation de données dans un processus distribué, la limitation décrite dans la section 7.1.3 (Limitations de l'importation de l'annexe) explique que le temps d'exécution de l'importation est très élevé à cause du volume de données (chez OSM), une approche distribuée serait idéale pour améliorer ce temps d'importation. Deuxièmement, le calcul des poids optimaux par type d'entité, actuellement ce calcul est fait en prenant toutes les entités en général, cependant classifier les entités par type (des hôtels, des parcs, etc.) pourrait être une démarche intéressante lors de l'alignement, puisque les entités de même types partagent d'autres informations (ex, les horaires d'ouverture, le type de cuisine) qui pourraient être exploitées. Une amélioration de la qualité à *top-1* faciliterait une validation de masse (ensemble de correspondances partageant des caractéristiques et des scores de similarité proches). Troisième point, l'intégration de données, c'est à dire fusionner les entités correspondantes en une seule.

Ce projet de recherche nous a permis de nous familiariser avec des approches de recherche qui en général diffèrent par rapport aux autres projets que nous avons l'habitude de mettre en place dans les autres UEs. Par ailleurs, nous avons pris conscience que des outils/APIs proposant un même service ne rendaient pas forcément une information identique voire complète et que selon le contexte il est parfois nécessaire d'en conjuguer plus d'une afin d'obtenir une information approfondie et optimale. Enfin, la construction des expérimentations est un procédé qui demande de réflexion.

6 Références

[1] Description des fichiers DUMP de Geonames.
http://download.geonames.org/export/dump/readme.txt
[2] Description du système clé-valeur (tag) et des types chez OpenStreetMap
http://wiki.openstreetmap.org/wiki/Map_Features
http://wiki.openstreetmap.org/wiki/Tags
[3] Description de la communauté de OpenStreetMap
http://wiki.openstreetmap.org/wiki/About_OpenStreetMap
[4] LANGLEY, Richard B., et al. Dilution of precision. GPS world, 1999, vol. 10, no 5, p. 52-59.
http://gauss.gge.unb.ca/papers.pdf/gpsworld.may99.pdf
[5] Description des recommandations de OpenStreetMap par rapport à la précision des appareils GPS.
http://wiki.openstreetmap.org/wiki/Accuracy_of_GPS_data
[6] BERJAWI, Bilal, et al. Uncertainty visualization of multi-providers cartographic integration. Journal of Visual Languages & Computing, 2014, vol. 25, no 6, p. 995-1002.
http://liris.cnrs.fr/Documents/Liris-7016.pdf
[7] : Anthony MORANA et Thomas MOREL. “GeoBench : un outil d’alignement entre entités spatiales pour la construction d’un benchmark cartographique”. Rapport de recherche (2014).
[8] Anthony MORANA et Thomas MOREL. "GeoBench: a geospatial integration tool for building a spatial entity matching benchmark." *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2014.
[9] Description de classification de types dans Geonames.
http://www.geonames.org/export/codes.html
[10] SEHGAL, Vivek; GETOOR, Lise; VIECHNICKI, Peter D. Entity resolution in geospatial data integration. En Proceedings of the 14th annual ACM international symposium on Advances in geographic information systems. ACM, 2006. p. 83-90.
[11] Description du stockage de données dans OpenStreetMap
http://wiki.openstreetmap.org/wiki/Planet.osm
[12] Fawcett, Tom. "An introduction to ROC analysis." *Pattern recognition letters* 27.8 (2006): 861-874.
[13] Szekely, Pedro, et al. "Connecting the smithsonian american art museum to the linked data cloud." *Extended*
Semantic Web Conference. Springer Berlin Heidelberg, 2013.
[14] Köpcke, Hanna, Andreas Thor, and Erhard Rahm. "Evaluation of entity resolution approaches on real-world match problems." *Proceedings of the VLDB Endowment* 3.1-2 (2010): 484-493.
[15] Projet UNIMAP http://liris.cnrs.fr/unimap/
[16] Levenshtein, V. I. (1966, February). Binary codes capable of correcting deletions, insertions, and reversals. In
Soviet physics doklady (Vol. 10, No. 8, pp. 707-710

7 Annexes :

7.1 Processus d'importation

Le processus d'importation permet de stocker localement les données des deux fournisseurs GN et OSM. L'intérêt de ce processus est de ne pas dépendre des services externes pour atteindre l'objectif du projet, mais aussi d'améliorer les performances lors de l'intégration. Chaque fournisseur met à disposition ses données dans des fichiers (DUMP), plus spécifiquement au format TXT pour GN et au format XML dans OSM.

L'importation des donnés de chaque fournisseur est faite de manière indépendante.

7.1.1 Importation GeoNames :

L'importation de données consiste à copier deux fichiers. Le premier (cf Figure 1.10) contenant toutes les entités géographiques sous forme de points, avec certains attributs primaires notamment les coordonnées, le nom de l'entité et d'autres attributs secondaires [1], et le deuxième avec le nom et la description qui correspondent à la classification de type définie par GN (cf Figure 1.11).

$$\Pi \text{ geonameid}\rightarrow\text{id, name, asciiname}\rightarrow\text{ascii\_name, alternatenames}\rightarrow\text{alternative\_name, latitude, longitude, feature class}\rightarrow\text{fclass, feature code}\rightarrow\text{fcode, cc2, admin1 code}\rightarrow\text{admin1, admin2 code}\rightarrow\text{admin2, admin3 code}\rightarrow\text{admin3, admin4 code}\rightarrow\text{admin4, population, elevation, dem}\rightarrow\text{gtopo30, timezone, modification date}\rightarrow\text{moddate}\,(\text{PAYS})$$

Figure 1.10 : Projection des attributs correspondant aux points géographiques de GN.

$$\Pi_{\text{code, name, description}}(\text{featureCodes})$$

Figure 1.11: Projection des attributs correspondants au type de GN.

7.1.2 Importation OpenStreetMap:

L'importation d'entités est faite à partir du fichier XML contenant les entités géographiques sous forme des nœuds (entité sous forme de point, comme un musée), des chemins (entités sous forme de ligne, comme un rivière ou sous la forme de polygone, comme un parc) et des relations (entité sous forme de multipolygone, comme une région entière). Le fichier est donc, divisé en trois grandes parties qui correspondent à chaque forme.

À différence de GN, OSM n'a pas une structure rigide des données (modèle Relationnel), mais plutôt un modèle clé-valeur, free tagging system [2], pour stocker ses attributs. Ce système de stockage lui garantit une grande flexibilité pour offrir un haut niveau de granularité dans les données, mais il entraîne certains défis dans l'alignement des entités qui sont détaillés dans la section 7.2 Conflits lors de l'alignement (Conflits de type).

```
<node id="26863008" lat="42.6985978" lon="0.3944255" version="13" timestamp="2017-01-24T02:36:43Z" changeset="45418729"
uid="90780" user="Verdy_p">
    <tag k="ele" v="3174" />
    <tag k="name" v="Pic Schrader" />
    <tag k="name:es" v="Gran Bachimala" />
    <tag k="name:fr" v="Pic Schrader" />
    <tag k="natural" v="peak" />
    <tag k="alt_name:fr" v="Grand Batchimale" />
</node>
<way id="183036874" version="2" timestamp="2012-09-26T18:54:40Z" changeset="13263022" uid="849013" user="thibautRe">
    <nd ref="1933977604" />
    <nd ref="1933977622" />
    <nd ref="1933977640" />
    <nd ref="1933977607" />
    <nd ref="1933977629" />
    <nd ref="1933977627" />
    <nd ref="1933977621" />
    <tag k="name" v="Carrer dels Avellaners" />
    <tag k="highway" v="unclassified" />
</way>
<relation id="5761660" version="3" timestamp="2017-01-04T02:49:03Z" changeset="44888000" uid="339581" user="nyuriks">
    <member type="way" ref="387134290" role="outer" />
    <member type="way" ref="136391986" role="outer" />
    <member type="way" ref="387134289" role="outer" />
    <tag k="name" v="Parc Natural de la Vall de Sorteny" />
    <tag k="type" v="multipolygon" />
    <tag k="leisure" v="nature_reserve" />
    <tag k="website" v="http://www.sorteny.ad/" />
    <tag k="boundary" v="national_park" />
    <tag k="wikidata" v="Q3364586" />
    <tag k="wikipedia" v="ca:Parc natural de la vall de Sorteny" />
</relation>
```

Figure 1.12 : Structure du fichier XML OSM

La Figure 1.12 illustre une partie du fichier XML de OSM utilisé pour ce processus. Chaque entité est porteuse des tags, soit un nœud, un chemin ou une relation. Les nœuds sont les uniques formes qui

contiennent les attributs géographiques, étant donné les chemins et les relations sont composés d'eux, et sous le même principe, les relations sont composées, à leur tour, de chemins ou d'autres relations.

```
for $node in doc("osm.xml")//node
    let $id := $node/@id
    let $latitude := $node/@lat
    let $longitude := $node/@lon

    let $relation_reference := $node/../relation/member[@type='node' and @ref=$id]/../@id
    let $way_reference := $node/../way/nd[@ref=$id]/../@id

return concat($id, $latitude, $longitude, $relation_reference, $way_reference).
```

Figure 1.13: Partie de la logique d'importations de nœuds à partir du fichier XML

La portion de code exposée dans la Figure 1.13 décrit une partie du processus d'importation. Tous les nœuds sont parcourus pour extraire les attributs primaires, et pour récupérer les références vers d'autres entités. Ainsi, l'identifiant d'une relation ou d'un chemin sont des références du nœud qui le compose.

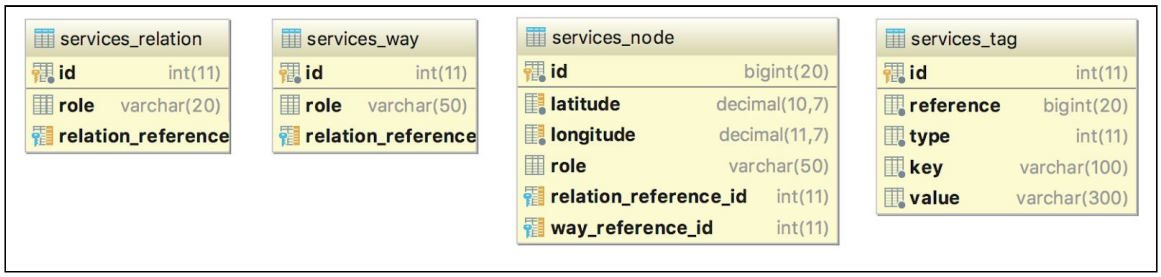


Figure 1.14: Schéma relationnel de OSM dans BD locale

La Figure 1.14 décrit le schéma relationnel adopté dans notre approche. Les tables *services_relation*, *services_way* et *services_node* stockent les entités selon la forme du POI. De plus, la table *services_tag* stocke les tags associés à chaque entité indépendamment de sa forme. Dans cette table, les attributs *reference* et *type* permettent de retrouver l'entité à laquelle le tag est associé. Le système de tag chez OSM étant par définition ouvert (*free tagging system*), les données représentant le nom du tag et sa valeur (key et value) ne sont pas normalisées.

7.1.3 Limitations de l'importation:

OSM est une grande base de données volumineuse, car elle stocke non seulement des informations sur les points d'intérêt mais également toutes les informations sur le fond de carte (différentes couches, contours, altitudes, etc.). À ce jour, OSM contient 784 Go de données non compressées [11], et l'organisation compte sur des serveurs très puissants pour la gestion de ces données géolocalisées.

Dans notre approche, le processus d'importation pour OSM a présenté plusieurs inconvénients. Les fichiers dumps de OSM sont très lourds (e.g., 84 Go non compressé pour la France, 9 Go pour la région Rhône-Alpes). Sur les machines de travail, l'importation de dumps OSM nécessitait donc plusieurs heures voire plusieurs jours. Le code a donc été optimisé (requêtes avec insertions multiples dans le SGBD, report des traitements spécifiques, optimisation de la gestion de la mémoire, etc.). L'organisation des données, notamment pour les POI multi-points (ligne ou polygone) implique que le nom du POI n'est rattaché à qu'à un seul point ou relation. Il est donc nécessaire de rattacher chaque point aux bonnes informations du POI, mais des recherches à ce niveau ralentissent fortement le processus d'import. Notre solution a consisté à lire le fichier XML ligne par ligne afin d'éviter de faire des recherches sur le fichier, puis de réaliser ces traitement ultérieurement.

7.2 Conflits lors de l'alignement

7.2.1 Conflits dans les attributs géographiques:

L'imprécision GPS correspond à la catégorie des erreurs des attributs géographiques et imprécision de l'information spatiale. Quand on parle des attributs géographiques, on parle aussi de bruit qui est inhérent aux

LBS. Ces bruits peuvent se produire à cause de multiples facteurs comme : le bruit dans l'appareil récepteur, l'horloge du satellite, influences atmosphériques, SA[4] (Selectivity Availability) et phénomènes de propagation par réflexe[5] (multipath propagation) [4].

Les fournisseurs cartographiques recommandent de prendre certaines précautions pendant le mappage de traces géographiques, cette précaution est la valeur de PDOP (Position Dilution of Precision) qui doit être à partir de quatre chiffres après la virgule pour éviter une dilution de précision trop grande [5]. Une autre pratique recommandée est de désactiver le SA de l'appareil récepteur, pour réduire la déviation standard de la pseudo-distance[6]. Cette dernière représente la distance entre le récepteur et chaque satellite en tenant en compte le délai de propagation ionosphérique et troposphérique et le bruit occasionné pour des facteurs déjà mentionnés avant [4].

Mais, même avec ces précautions les données géographiques varient entre fournisseurs. D'autres composants peuvent impacter le calcul de distance : le degré de précision de l'appareil récepteur utilisé, qui peut être plus ou moins sensible aux bruits, la fréquence de l'appareil (simple ou double, la seconde option étant capable de supprimer le délai ionosphérique), et le moment et le lieu dans lequel la trace a été effectué. Ainsi, compte tenu de toutes ces variations auxquelles les coordonnées géographiques sont soumises, les conflits géographiques sont souvent un problème dans l'intégration des entités spatiales.[4].

Des autres différences au niveau des coordonnées géographiques peuvent être encore subdivisées en trois nouvelles catégories : (i) Différence de localisation : des entités qui ont différents attributs géographiques mais, qui correspondent au même POI. (ii) Position équivalente : des entités qui ont la même position mais qui correspondent à deux POI différents et (iii) Superposition : des entités ont la même position mais représentent des POI différents où un d'eux est contenu dans l'autre.[7][8]

7.2.2 Conflits dans les attributs primaires:

Une partie du problème de l'intégration est la correspondance entre les noms des entités provenant des différentes sources de données. Certaines des conflits dans l'intégration de nom: les permutaciones et abréviations. (Tableau 1.9)

Fournisseur	**Conflits de nom**	
	Permutation	**Abbreviations**
Le nom de l'entité dans OSM	*Husa Imperial Hotel*	*Ctra. del Plans de Ransol*
Le nom de l'entité dans GN	*Imperial Atiram Hotel*	*Carretera de Ransol*

Tableau 1.9: Exemples de conflits de nom entre les deux fournisseurs.

Dans la littérature scientifique, on trouve plusieurs techniques de mesure de similarité entre chaînes de caractères. Entre elles on peut citer: la distance de levenshtein, distance jaro winkler ou comparaison pour NGram. La distance de levenshtein est définie par la quantité des opérations (suppression, édition ou ajout de caractères) requises pour transformer une chaîne de caractères dans une autre. Où toutes les opérations ont le même coût[10]. Dans notre approche on a choisi utiliser la distance de levenshtein pour trouver la correspondance entre noms.

Le type de chaque entité est un attribut indispensable, lors de l'intégration, qui permet de restreindre la quantité des entités candidates à comparer, afin d'avoir un alignement plus précis. De ce fait, nous l'avons utilisé comme un élément clé dans l'alignement.

Chaque fournisseur propose sa propre structure pour les types de POI. La base OSM les présente sous

[4] SA (Service Availability) est une caractéristique de certain appareils GPS qui ajoutent intentionnellement le temps.

[5] Multipath : (Multipath propagation) C'est le phénomène de propagation qui résulte dans signaux de radio qui atteindront le récepteur à travers plusieurs chemins.

[6] Peudo-distance (Pseudorange) est la distance entre le satellite et le récepteur, en incluant les délais de propagation et des erreurs comme le Multipath.

forme de clé-valeur, avec une liste non standardisée des valeurs pour les types les plus courants [2]. Quant à Geonames [9], il les décrit par des classes et des mots clés comme illustré dans le tableau 1.10. Ces éléments sont classés selon des catégories qui diffèrent d'un fournisseur à un autre.

La différence entre la présentation des structures des données des fournisseurs a un grand impact sur l'alignement des données, plusieurs types de conflits ont été déterminés lors de l'intégration :

- Structurel: L'hétérogénéité dans les structures d'attributs de différentes sources. GN a deux attributs (classe et code) qui définit les types de chaque entité. Par contre, dans OSM on trouve des tags qui sont utilisés pour décrire les types mais aussi des autres attributs. (e.g. un tag peut être utilisé pour indiquer l'adresse d'une entité: addr:country=*, addr:city=*, addr:postcode=*, etc.).
- Sémantique (schéma): Les relations à niveau des classes et des sous-classes qui varient entre sources. (e.g. la classe S dans GN est utilisé pour représenter monuments, restaurants, etc. alors que dans OSM est utilisé la clé historic pour les monuments et la clé amenity pour les restaurants)
- Sémantique (instance) : Des différences dans les noms du type qui représentent le même concept. (e.g. la clé-valeur amenity:hospital dans OSM est équivalente à la classe et sous-classe S,HSP). Subsomption de concepts (e.g. la classe et sous-classe S,CTRM qui s'utilise dans GN pour représenter des ensembles de bâtiments de services médicaux comme: hôpital, faculté de médecine, clinique, pharmacies, cabinets médicaux, etc mais que dans OSM sont représentés par clés indépendantes).

Fournisseur	Structure	Utilisation
OSM	Clé	Dans OSM, chaque tag est une catégorie. Par exemple *amenity* utilisé pour cartographier des équipements pour les visiteurs et les résidents (e.g. banque, pharmacie)
	Valeur	C'est la valeur d'un tag par exemple *restaurant*.
GN	Classe	Chaque classe est une catégorie. Par exemple la classe *S* qui définit des bâtiments, des boutiques et des fermes.
	Code	Chaque classe contient des codes définissant le type de l'entité, par exemple *REST* (restaurant) dans la classe *S*.

Tableau 1.10: Description des différences de structure de type dans OSM et GN.

Avec tous les conflits, il est difficile de faire la correspondance entre les entités d'une manière automatique ainsi, il faut passer par une phase d'analyse pour examiner des différents conflits.

Dans ce sens, nous avons réalisé une première phase des correspondances manuelles entre les types afin de faciliter le matching. Pour cet effet, nous avons sélectionné les types les plus utilisés dans les deux fournisseurs. Les informations générées ont été testées avec des entités réelles afin de les valider, ensuite elles ont été stockées dans une table de la base de donnée locale.

Pour l'alignement automatique, l'approche est différente. Le même est détaillé dans la section 3.2.3. (Algorithme d'alignement).

7.3 Interfaces de l'utilisateur

L'application a été réalisé en HTML5/CSS3 et utilise en grande majorité le JAVASCRIPT. Nous avons utilisés plusieurs librairies (Bootstrap, jQuery et Leaflet) et API (API-GEONAME et OVERPASS) qui nous ont facilité l'implémentation visuelle et la réalisation d'une application évolutive qui propose plusieurs fonctionnalités à savoir l'importation, la visualisation des correspondances, la recherche des POI et le changement des paramètres et poids des attributs choisis pour calculer le score de pertinences.

À continuation la présentation des interfaces:

Accueil: Cette page permet à l'utilisateur de chercher un POI et de l'afficher sur une carte géographique. Nous utilisons l'API de GN pour récupérer son identifiant. ce dernier sera par la suite envoyé au serveur afin de visualiser la liste des correspondances dans la page qui suit. (c.f Figure 1.15)

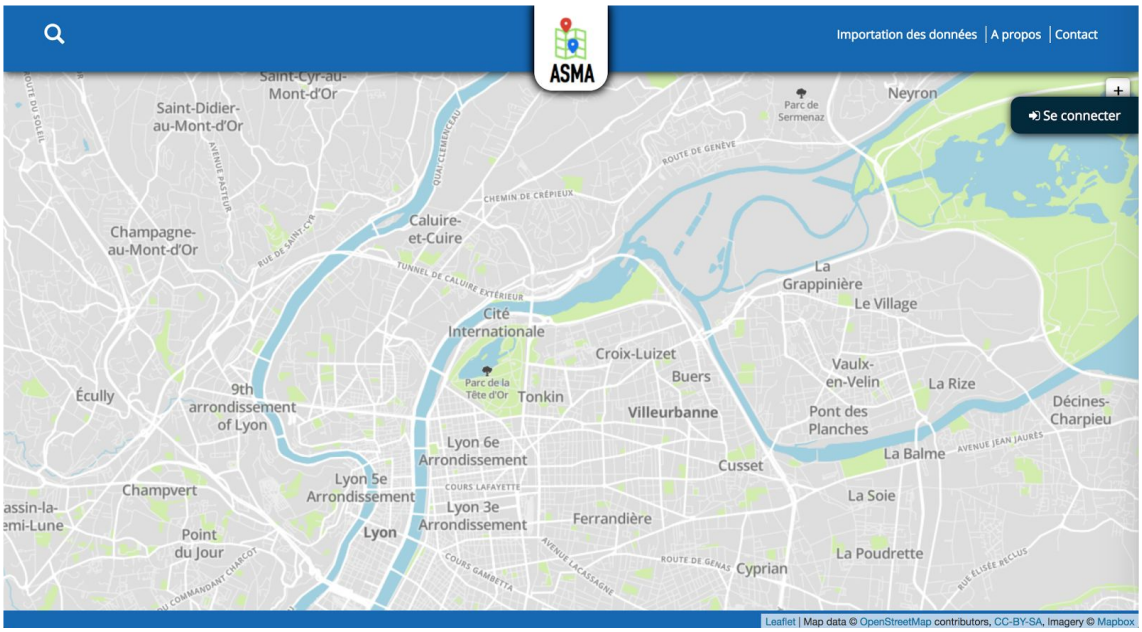


Figure 1.15 : Interface Accueil

Correspondance (Figure 1.16) : Cette interface affiche à l'utilisateur la liste des correspondances trouvées automatiquement. Ainsi, il pourra les valider ou les invalider en se basant sur différentes informations communes comme le nom , le type et les coordonnées géographiques. Un utilisateur pourra aussi faire la correspondances entre les types ce qui contribuera à un meilleur affinage des résultats de l'algorithme du matching. En effet, les correspondances des types validées ou invalidées sont stockées afin d'être réutilisé. En ce qui concerne les entités qui n'ont pas de type, un utilisateur pourra choisir un tag. Cependant, pour que ces règles soient prises en compte, plusieurs utilisateurs doivent faire la même validation.

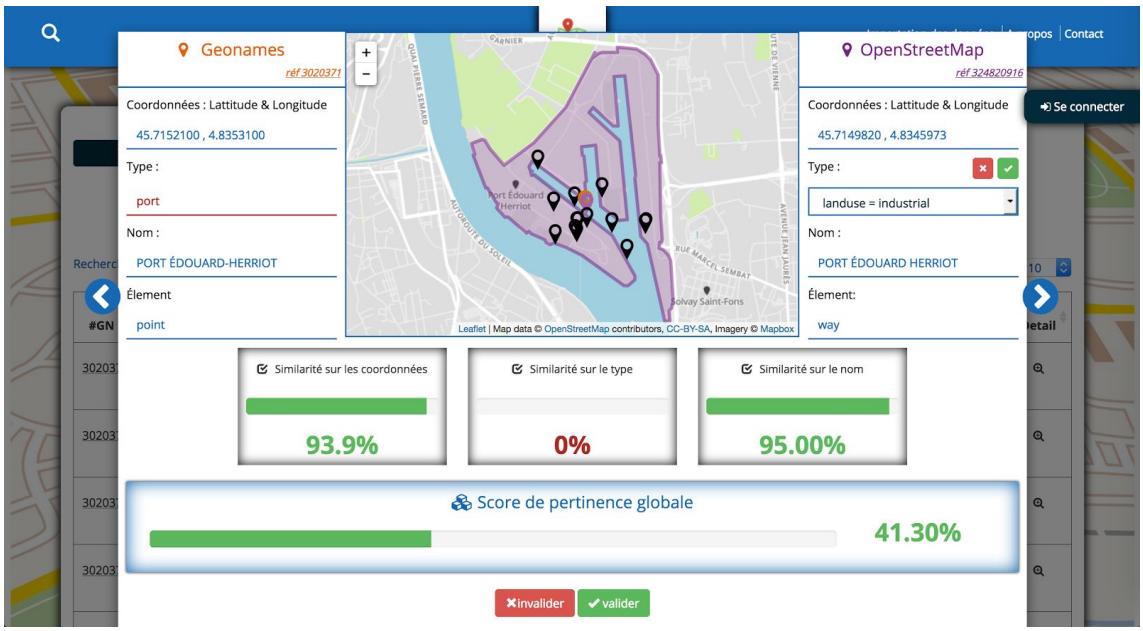


Figure 1.16: Interface correspondances

Configuration: Cette page est privée, elle est destinée aux utilisatrices connectées . Elle permet de changer les poids des indices de similarité afin de recalculer le score de pertinence. L'utilisateur pourra aussi visualiser l'historique des précédents poids (c.f. Figure 1.16).

Importation: Vue que nous n'avons pas importer les dumps de tous les pays, l'utilisateur pourra les importer et démarrer les algorithmes afin de visualiser les correspondances (c.f. Figure 1.17).

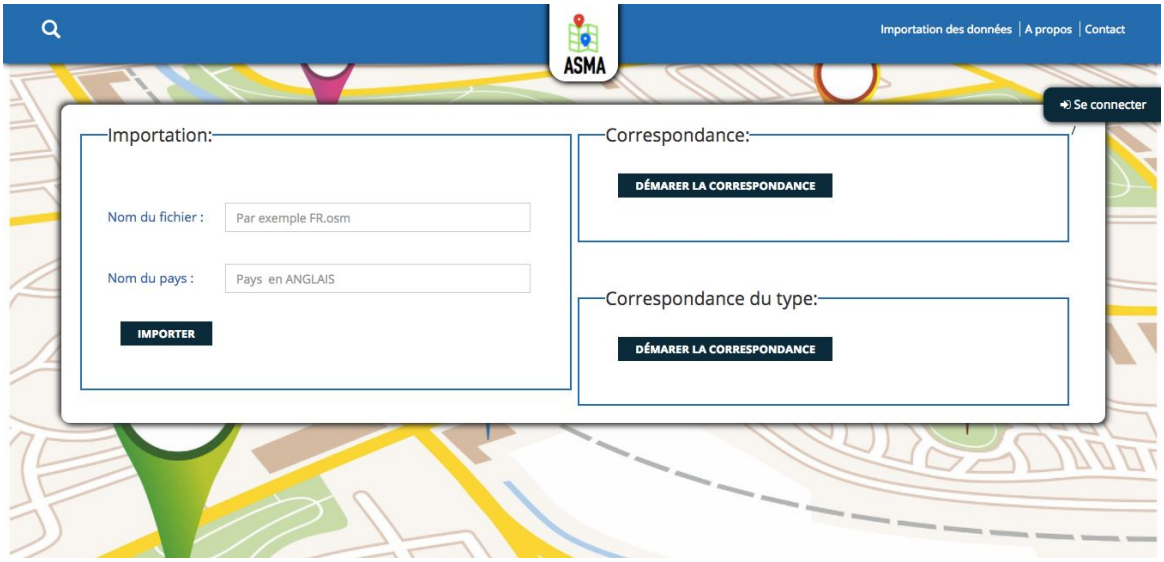


Figure 1.17: Interface correspondances

Authentification: Cette page permet à un utilisateur de se connecter à l'application pour accéder aux pages privées (configuration et importation) (cf Figure 1.18)

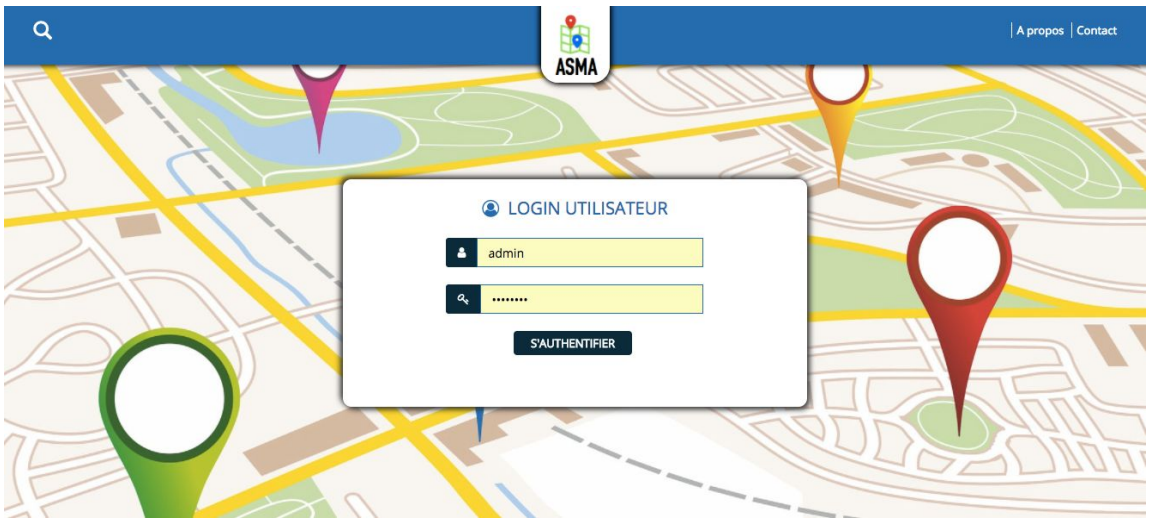


Figure 1.18: Interface correspondances

7.4 Caracteristiques des ordinateurs où nous avons a exécuté les processus précédemment décrits

- Mac PRO, processeur 2.7 GHz Intel Core i5, mémoire 8 GB 1867 MHz DDR3. 120GB SSD.
- Mac Air , processeur 1.6 GHz Intel Core i5, mémoire 4 Go 1600 MHz DDR3. .120GB SSD.

7.5 Précision, Rappel et F-mesure:

La précision est la proportion de correspondances trouvées qui sont correctes. Elle mesure la capacité de l'outil à refuser les correspondances incorrectes, ainsi, elle est définie par le rapport entre les correspondances validées correctes et l'ensemble des correspondances découvertes : [12]

$$P = \frac{TP}{TP + FP}$$

Le rappel est la proportion de correspondances trouvées et correctes parmi toutes celles attendues (correctes d'après l'expertise). Il mesure la capacité de l'outil à donner toutes les correspondances correctes,

ainsi, il est défini par le rapport entre les correspondances trouvées correctes et l'ensemble des correspondances de l'expertise :[12]

$$R = \frac{TP}{TP + FN}$$

La F-measure (F-score) est la moyenne harmonique de la précision et du rappel , elle mesure la capacité de l'outil à donner toutes les correspondances correctes et à refuser les autres:[12]

$$Fmeasure = 2 \times \frac{P \times R}{P + R}$$