Master 2 Informatique

spécialité Données, Connaissances et Langues

SENTERRITOIRE

À la découverte des acteurs importants du territoire Montpellier Agglomération

Travail d'Étude et de Recherche

Février 2013

Auteurs :

Audry Barimbane,
Dimitri Dupuis,
Nidhal Gaddour,
Mathilde Salthun-Lassalle

Encadrant :

Eric Kergosien

Remerciements

Nous remercions notre encadrant, Eric Kergosien, de nous avoir guidé et accompagné dans ce projet.

Sommaire

Table des figures

Chapitre 1
Présentation Générale

1 – Contexte de projet

Dans le cadre du TER de Master 2, notre choix s'est porté sur le projet Senterritoire. Ce projet met en collaboration des acteurs du monde scientifique (informaticiens, géographes,...), des laboratoires, TETIS et LIRMM, et est soutenu par la Maison des Sciences de l'Homme de Montpellier (MSH-M). Il a pour but de fournir un environnement décisionnel basé sur une analyse de textes liés à l'aménagement du territoire. Le projet se focalise dans un premier temps sur l'extraction automatique des descripteurs géo-spatiaux dans la région du bassin de Thau. Ces derniers seront par la suite enrichis par des informations caractérisant plus largement le concept du territoire, notamment les opinions des acteurs, afin d'analyser automatiquement l'utilisation de ces concepts dans les textes et les perceptions qu'ils véhiculent. L'enjeu final est la découverte de réponses à des questions telles que « les journalistes ont-ils une perception positive de l'aménagement du quartier de la gare ? ».

Dans le cadre d'un stage de recherche mené au LIRMM, Tahrat Sabiha a réalisé la première étape de cette analyse automatique de textes : une chaîne de traitements linguistiques (Text2Geo) pour l'extraction d'entités nommées spatiales et de type organisation dans un corpus de textes. Le fond exploité est un ensemble de documents d'actualités exprimant des opinions sur le territoire et indexé manuellement. Ce travail a été conçu avec la plate-forme pour le TALN Linguastream qui a permis l'organisation de la chaîne en modules.

2 – Besoins

L'opération d'extraction peut être lancée via Linguastream, en ligne de commande ou au travers d'une application tierce. Notre contribution au projet est motivée tout d'abord par le besoin d'intégration de la chaîne Text2Geo à l'application et celui de la récupération de ses résultats à des fins de stockage. Cela permettra à terme une visualisation des entités nommées trouvées et donnera la possibilité à un utilisateur distant d'un accès en ligne.

Le second apport demandé est la validation des entités nommées spatiales à l'aide de ressources externes, qu'il nous faudra définir.

3 – Objectifs

L'application TerridocViewer1, développée dans le cadre de travaux de recherches, offre la possibilité de naviguer dans un graphe de concepts construit à partir d'un thésaurus, représentant un territoire implicitement, décrit par un fond documentaire indexé manuellement. L'incorporation de la chaîne de traitements spécifiques au projet Senterritoire suppose l'extension des fonctionnalités de l'application et de ses bases de données.

La chaîne de traitement au sein de l'application sera elle-aussi prolongée, par une étape de validation des entités nommées extraites et, suite à notre proposition et après discussion avec M. Eric Kergosien sur les besoins et le cahier des charges, par un module facilitant l'analyse statistique des résultats de cette étape. L'objectif de l'ajout de ces deux derniers modules est le contrôle de la qualité des étapes précédentes, à savoir l'extraction des entités puis la validation des entités, par l'utilisateur.

En raison de la courte durée de notre TER, nous sommes amenés à ne réaliser que quelques briques d'un projet plus ambitieux. L'utilisateur aura à terme la possibilité de visualiser les entités nommées extraites. Dans un second temps, ce projet s'enrichira d' une étape d'extraction d'opinion. Nous détaillons les perspectives de notre travail dans la partie quatre.

En résumé, notre rôle premier dans ce projet peut être organisé en quatre points essentiels :

- La prise en main de l'application Web j2ee N-tiers existante TERRIDOCViewer[1], de ses technologies (ajax, dojo, struts2..) et de son architecture (bases de données...). En d'autres termes, il s'agit d'une analyse de l'existant préalable aux phases de conception et de développement
- Le développement d'un module permettant de lancer la chaîne Text2Geo, pour le stockage des documents résultants de la chaîne et l'extraction des entités nommées spatiales identifiées
- Le développement d'un module de validation des entités (spatiales) extraites à l'aide d'une ou plusieurs ressources à déterminer
- Le développement d'un module de visualisation de statistiques de l'étape de validation des entités nommées spatiales

4 – Organisation du rapport

Le premier chapitre de ce rapport ci-dessus a introduit le périmètre de notre travail dans ce projet. Nous y présentons le contexte de ce projet et les objectifs à atteindre. Nous présentons dans le chapitre deux l'état de l'art de l'application TerrodocViewer1 existante, son architecture MVC et le schéma de ses bases de données. Cette analyse est suivie des spécifications de notre apport à ce projet. Dans le chapitre trois, nous détaillons notre contribution. Nous exposons les différents modules développés, les modifications apportées aux schémas de bases de données. Le chapitre quatre conclut notre travail et expose les perspectives de l'application.

[1] TerridocViewer : http://t2i.univ-pau.fr/Terridoc/

Chapitre 2
Étude préalable

1 – Extraction des entités nommées

1.1. Définition :

Avec la démultiplication des textes numériques et le besoin d'extraction d'information, notamment en temps réel, l'un des enjeux majeurs aujourd'hui pour le TAL, est de capter l'information dans le texte et d'accéder à son sens. Les unités linguistiques porteuses d'informations sont diverses : mots, segments, phrases, paragraphes… et les entités nommées (EN) sont au nombre de celles-ci. Selon (Chinchor, 1998), la notion d'EN est utilisée pour regrouper tous les éléments du langage qui font référence à une entité unique et concrète, appartenant à un domaine spécifique (ie. Humain, économique, géographique, etc.). Dans la définition traditionnelle, ce terme désigne l'ensemble des noms de personne, de lieux et d'organisations. Cet ensemble a été élargi par l'ajout d'autres expressions comme les dates, les unités monétaires les pourcentages et autres (Eherman, 2008).

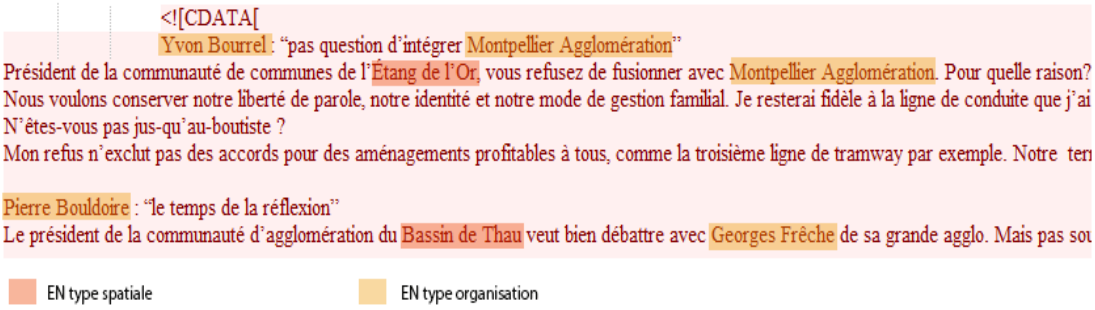


Figure 1: Aperçu sur les EN dans un article

Dans notre corpus les entités nommées sont très présentes. Les types auxquels nous nous intéresserons sont les entités nommés spatiales (ex : Bassin de Thau) ainsi que les organisations (ex: Montpellier Agglomération) (cf. figure1).

1.2. Détection des entités nommées :

La détection des entités nommées représente un intérêt majeur pour le domaine de l'extraction d'information et les applications comme l'indexation, l'aide à la décision, la veille, les systèmes de question/réponse et autres. Ce besoin a suscité de nombreux travaux sur le domaine dès le milieu des années 90. Depuis, plusieurs campagnes d'évaluations ont été organisées autour de ce sujet (i.e. cycle MUC[2], IREX[3], CoNNL[4], ACE[5], ESTER[6], HAREM[7]) (Eherman, 2008) Le traitement des entités nommées s'organise autour de trois processus. Le premier, détection ou identification, vise le repérage de l'entité dans le texte en se basant sur un ensemble des indices

[2] Message Understanding Conferences
[3] Information Retrieval and Extravtion Exercise
[4] Conference on Natural Language Learning
[5] Automatic Content Extraction
[6] Evaluation des Systèmes de Transcription Enrichie d'Emission Radiophonique.
[7] Evalição de sistemas de Reconhecimento de Entitades Mencionadas.

internes et externes (McDonald, 1996). Ces indices présentent des éléments importants pour former des méthodes de détections automatiques des entités nommées.

Le deuxième processus, la catégorisation, consiste à déterminer le type de l'entité à partir de catégories sémantiques prédéfinies. La catégorisation ou l'annotation des entités nommées est une étape importante pour le reste du traitement, elle vient dans le but de produire une analyse linguistique selon un ou plusieurs niveaux (morphologique, syntaxique, sémantique). On note la présence de différentes approches pour réaliser cette tâche (approches Statistiques, approches Symboliques et approches Mixtes).

Le dernier processus est la normalisation d'entités détectées. C'est une étape qui tente de fournir une représentation standard pour les différents types de variations pour chaque entité. (Eherman, 2008).

2 – Analyse de l'existant

2.1. Architecture de l'application et technologies

L'application Senterritoire présente une architecture 3-tiers (client, applicatif, données) sous une infrastructure Java ee(cf. Figure).La technologie Java ee correspond à plusieurs spécifications de logiciels, et de frameworks constituant ensemble un système pour développer et déployer des applications. Aussi, l'application TerridocViewer, s'appuie sur les Framework Struts2, Hibernate, Dojo et utilise le serveur d'applications Tomcat.

Le Framework Struts2 implémente une architecture dit **MVC2 (modèle, vue, contrôleur)**. Le contrôleur sert pour la gestion de requêtes clientes par un servlet, la partie modèle se charge de la gestion et de la récupération des données à envoyer et la vue qui a pour rôle l'affichage de l'interface utilise des JSP et Ajax.

Le Framework Hibernate intégrant de nombreuses fonctionnalités permettant l'accès aux bases de données est utilisé pour simplifier le développement et gérer l'accès aux bases de données MySQL. Le Framework Dojo est utilisé pour permettre le développement d'interface interactive.

- **Tiers client (navigateur) :**

Cette partie prend en charge l'affichage de l'interface utilisateur. Elle est constituée du navigateur. L'affichage de l'interface est développé avec le Framework Dojo.

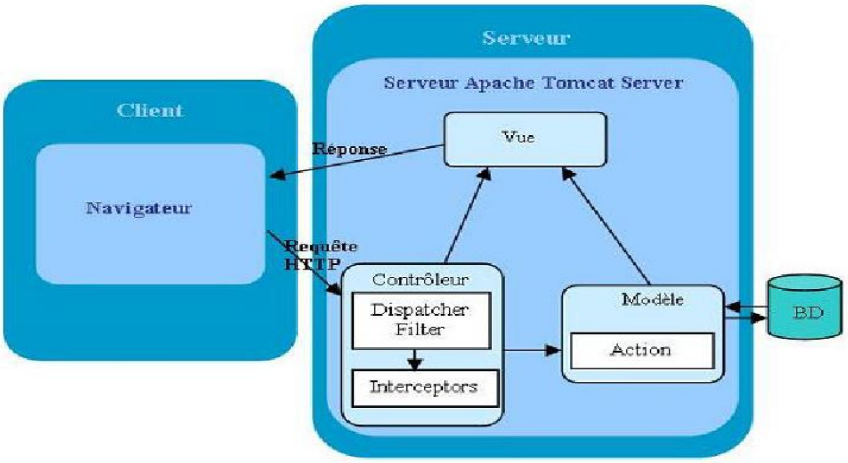


Figure 2: Architecture de l'affichage

- **Tiers applicatif (Serveur Tomcat) :**

Cette partie permet la gestion des requêtes et des réponses clientes. Elle intègre les modules vue, contrôleur et modèle selon l'architecture Struts2. La figure2 illustre les rôles de ces modules suivant cette architecture.

- **Tiers données :**

Le tiers données représente l'ensemble des données manipulées par l'application. Les bases de données thésaurusr, terridoc, paramétrage et senterritoire nous ont été fournies par M. Eric Kergosien au début du projet.

2.2. Technologies et concepts

Les concepts clés de l'application sont Struts2, Hibernate et Dojo :

a. Struts 2

Struts est un Framework pour applications Web Java développé par la fondation Apache dans le cadre du projet Jakarta. Il met en œuvre le modèle MVC2 basé sur une seule servlet faisant office de contrôleur et des JSP pour l'IHM. L'application de ce modèle permet une séparation en trois couches de l'interface, des traitements et des données. Struts se concentre sur la vue et le contrôleur. L'implémentation du modèle est laissé libre aux développeurs : ils ont le choix d'utiliser un outil de mapping objet/relationnel, des EJB ou toute autre solution. La figure suivante illustre l'architecture MVC de Struts2 :

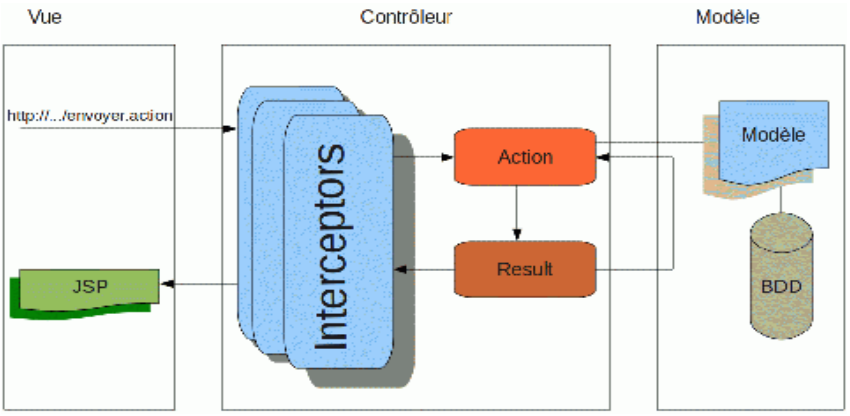


Figure 3: Architecture de Struts MVC

b. Dojo ToolKit

Dojo est un Framework JavaScript facilitant le développement d'applications Web 2.0.l est à la fois une bibliothèque JavaScript et une bibliothèque de Widgets. Il permet le développement d'applications RIA. Il est conçu pour les navigateurs, les mobiles et les serveurs.

c. Hibernate

Hibernate est un Framework permettant la gestion de la persistance des objets en base de données relationnelle. Il permet ainsi de représenter une base de données en objets. Il rend facile la recherche des données dans une base de données par la création des objets et les traitements de remplissage de ces objets par accès à la base de données.

Une architecture illustrative est présentée ci-après :

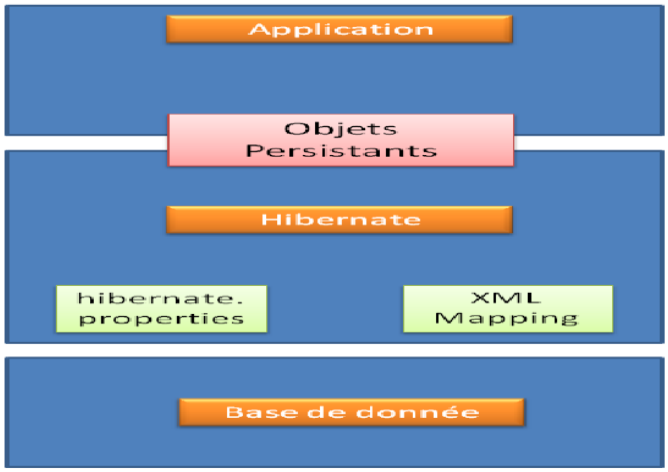


Figure 4: Architecture Hibernate

d. Design pattern DAO (data access object)

L'utilisation d'Hibernate dans l'application s'accompagne de la mise en place du design pattern DAO. Il fournit une abstraction pour les méthodes de requêtes à la base de données lors d'action comme la suppression, la sélection ou l'insertion. Une classe DAO est implémenté par objet mappé d'une base de données. Il réalise la liaison entre la base de données et la couche métier.

2.3. Les bases de données de l'application

a. Bases de données thesaurusr

Cette base de données est utilisée pour le stockage de thésaurus utile pour l'indexation des documents. Nous ne la détaillons pas ici car elle n'est pas en lien avec notre projet.

b. Base de données terridoc

Elle permet de stocker les données concernant des documents liés à un fond documentaire décrit par une notice et a également vocation à accueillir des entités nommées de type spatial et temporel qui en sont extraites. Elle concerne le projet Terridoc pour lequel l'application a été initialement prévue. Malgré une finalité allant dans le même sens que celle du projet qui nous concerne (stockage d'entités nommées spatiales extraites de documents), elle ne peut être réutilisée en l'état car elle ne prend pas en compte certaines de ses spécificités et besoins futurs comme par exemple, le stockage d'entités nommées "acteurs" ou encore des opinions. Cette base a néanmoins servi de modèle de conception pour la base Senterritoire dont nous avons usé et que nous présentons dans les spécifications de notre contribution.

c. Base de données paramétrage

Cette base contient les données en rapport avec les préférences de paramétrages de l'application par l'utilisateur. Voici la description de quelques tables de sa base :

Table utilisateur : pour l'identification de l'utilisateur

Table configuration : pour le stockage des données concernant les sources de données à utiliser (thésaurus,…).

Nous serons amenés à y avoir recours en ajoutant des tables spécifiques à la configuration de la nouvelle fonctionnalité de lancement de la chaîne non encore prise en compte.

2 – Spécification de notre contribution

2.1. Spécification des besoins

La chaîne de la figure ci-dessous résume les traitements qui vont être réalisés au sein de l'application web. Tout comme la chaîne de traitements Text2Geo, notre travail s'articule autour de modules à la fois dépendants et indépendants les uns des autres qui correspondent à des étapes de traitement des entités nommées extraites.

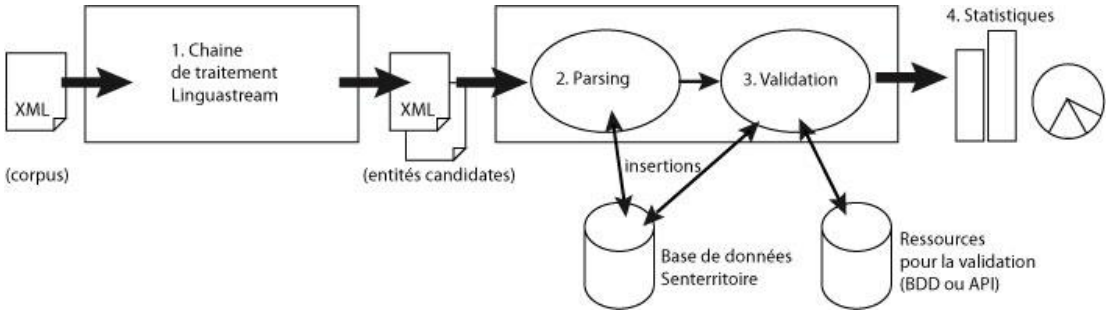


Figure 5: description de l'ensemble du processus et des modules à créer au sein de l'application.

Les tâches associées à chaque brique du schéma sont décrites ci-dessous. Vous les trouverez développées dans le chapitre concernant notre contribution.

a. Module d'intégration de la chaîne de traitement Text2Geo :

Afin de préparer à l'intégration de la chaîne, nous serons amenés à réaliser les points suivants :

- Analyse de la chaîne : Cette analyse succincte doit nous amener à comprendre le rôle de la chaîne, la configuration nécessaire à la faire fonctionner et les résultats qu'elle produit.
- Création de fichiers au format XML pour l'entrée de chaîne : Il est nécessaire de définir une normalisation des noms de balises des fichiers XML parsés en entrée de la chaîne. Pour nous y aider, des documents classés par opinions nous ont été fournis au format doc en guise d'exemple de corpus.
- Création d'un formulaire de lancement de la chaîne : Le choix des paramètres est laissé à l'utilisateur qui lance l'exécution via un formulaire. (cf figure maquette du formulaire)

b. Parsing et stockage dans la base :

Les tâches suivantes réaliseront l'objectif de stockage des entités extraites du corpus dans la base :

- Création d'une base de données aux tables spécifiques au projet Senterritoire (cf. spécifications de la base Senterritoire)
- Développement d'un module de parsing des fichiers produits en sortie de la chaîne : Les entités nommées que contiennent ces fichiers, candidates à la validation, sont insérées dans la base à l'issue de ce traitement.

c. Module de validation :

La réalisation de ce module nécessite 4 étapes :

- Étude comparative des ressources existantes pour la validation d'entités nommées : Le recours à une ressource de type service-web ou base de données permet d'automatiser la tâche de validation. Il nous appartient de vérifier la pertinence d'une ressource dans cette tâche par cette étude.
- Mise en place de l'interrogation à/aux ressources choisies : Plusieurs ressources parmi les meilleures de l'étude pourront être interrogées pour la validation d'une même entité spatiale candidate. En effet, les informations la concernant peuvent différer d'une ressource à une autre.
- Insertion des entités validées dans la base Senterritoire (en les distinguant des entités non validées)
- Affichage des résultats de la validation : une interface doit rendre compte à l'utilisateur du déroulement du processus de validation.

d. Module de visualisation de statistiques :

Il comprend 2 phases :

- Analyse du besoin statistique et analyse des solutions : Les problématiques soulevées par cette analyse sont le choix des informations pertinentes de la base Senterritoire pour une visualisation, le choix de la présentation de ces données et les moyens techniques envisageables pour la réaliser.
- Développement du module Statistiques au sein de l'application : l'affichage des graphiques doit être généré côté client à partir de données récupérées côté serveur dans la base.

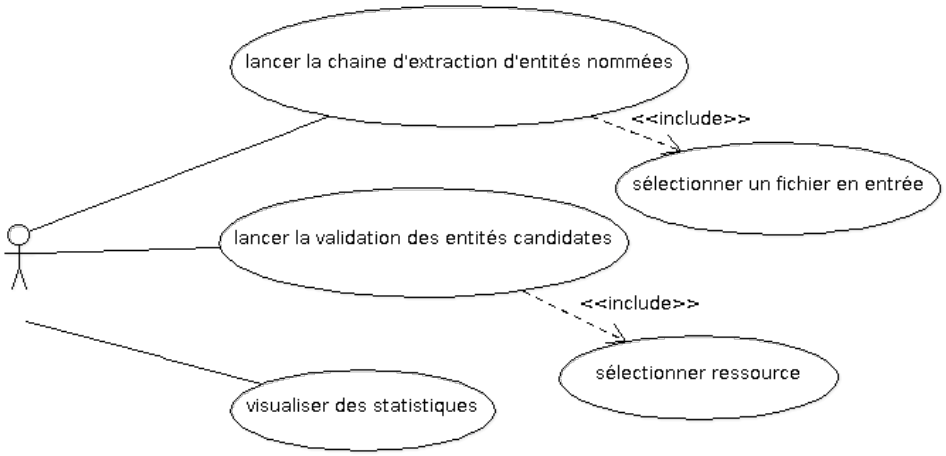


Figure 6: Diagramme de cas d'utilisation.

L'utilisateur des fonctionnalités présentées ci-dessus est un expert technique. Le diagramme de cas d'utilisations correspond aux besoins définis initiaux : lancement de la chaine d'extraction des entités nommées à travers l'application web et le module de validation des entités. La partie de visualisation des statistiques a été ajouté par la suite.

2.2. Spécification technique

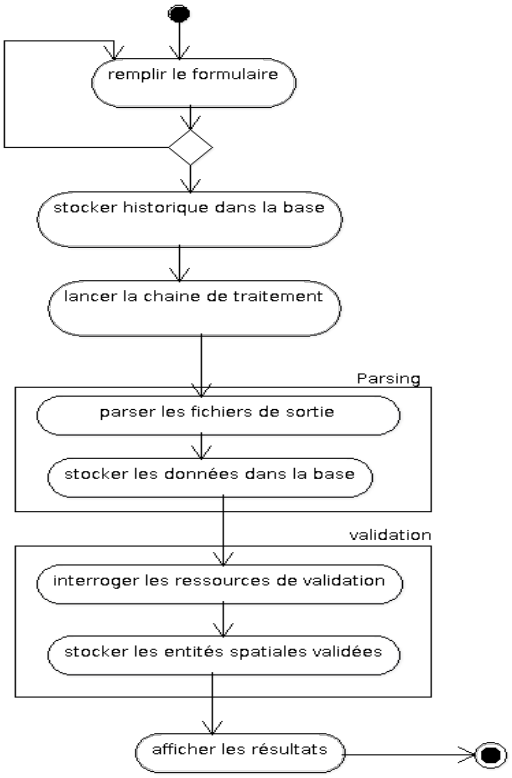


Figure 7: Diagramme d'activité qui résume les actions des modules à implémenter.

Grâce au diagramme d'activité on peut différencier les trois principaux modules de notre contribution et les liens entre eux : Tout d'abord nous avons le lancement de la chaine de traitement, puis ensuite le parsing de la sortie et le stockage dans la base et enfin l'étape de validation.

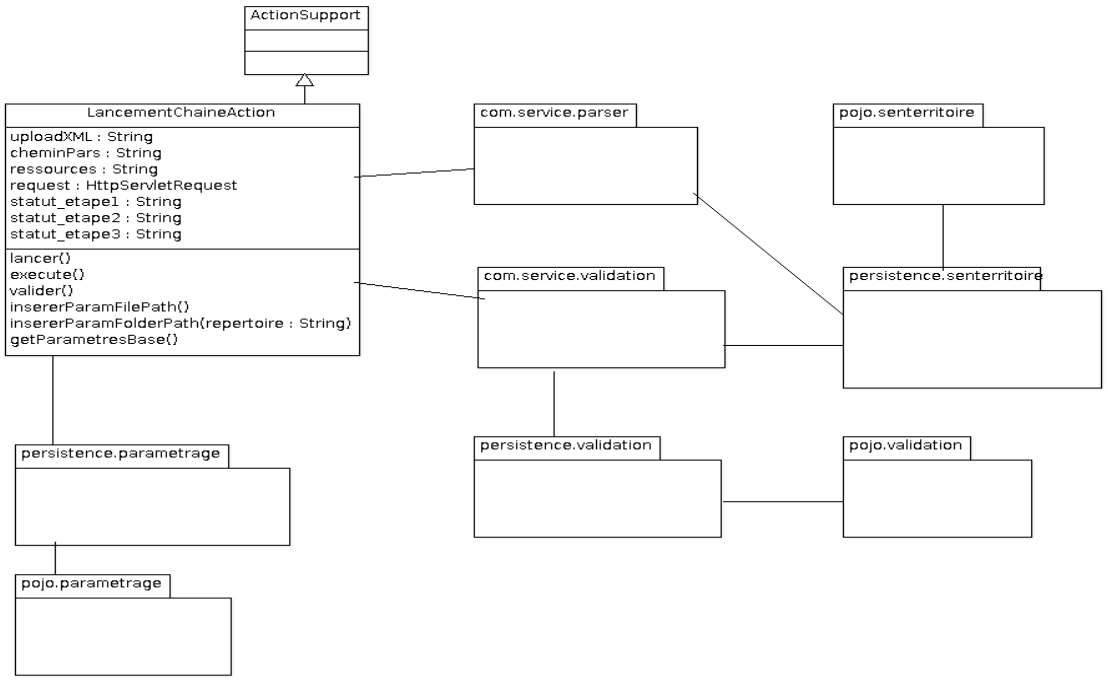


Figure 8: Diagramme de classes général pour le lancement de la chaine.

Le diagramme de classes nous montre toutes les classes crées lors de notre travail. Les packages pojo.parametrage, pojo.validation, pojo.senterritoire vont contenir nos classes correspondantes aux objets crées pour le mapping relationnel Hibernate.

Les packages persistence.validation, persistence.parametrage et persistence.senterritoire vont contenir nos classes DAO contenant les méthodes nécessaire aux requêtes sur la base.

Le package com.service.parser inclut les classes nécessaires au parsing de la sortie de la chaine de traitement et au stockage dans la base.

Le package com.service.validation inclut les classes pour la validation et également le stockage dans la base.

2.3. Schéma des bases de données

a. base de données Senterritoire

La base de données senterritoire est prévue pour stocker les informations relatives aux entités nommées, issues des fichiers XML produits en sortie de la chaine d'extraction. Elle est vouée à conserver la description des textes en entrée, le contexte d'extraction d'une entité, son type et son statut de validation. Elle a été construite sur le modèle de la base terridoc. Nous décrivons ici l'ensemble des tables (Fig.1). La base est prévue pour prendre en compte les entités de type spatial, organisation (acteur) et temporel ainsi que les informations géographiques, elles-mêmes composées

d'entités spatiales, temporelles et thématiques. Dans un premier temps, nous nous limitons au stockage et à la validation des entités nommées spatiales. Certaines tables ne seront par conséquent pas utilisées.

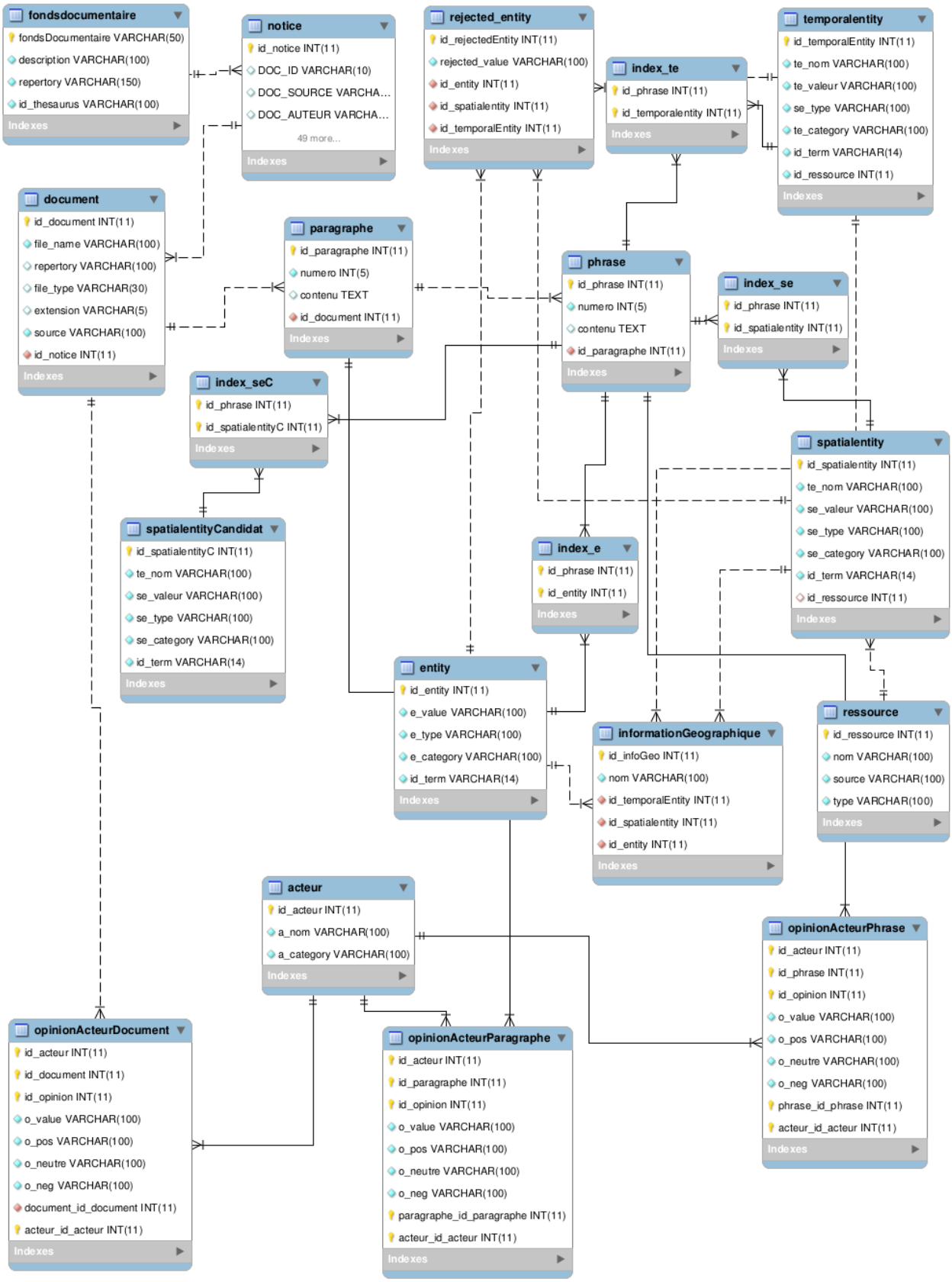


Figure 9: Ensemble des tables de la base 'Senterritoire'.

Tables relatives aux documents :

Dans les tables relatives aux documents seront stockés toutes les informations sur un document dans le détail jusqu'à la phrase. Ces tables seront principalement remplies par l'étape de parsing de la sortie de la chaine de traitement.

notice : description détaillée du document (origine, auteur, type...)

document : description des fichiers parsés avant la validation

paragraphe : ensemble des paragraphes par phrases

phrase : ensemble des phrases d'un document

fondsdocumentaire : description du fond documentaire.

Tables relatives aux entités:

Dans ces tables seront stocké toute information relative a une entité nommée. Ces tables seront remplies en partie par le module de parsing et aussi par le module de validation une fois les entités validées.

acteur : stockage des acteurs, des entités nommées de type organisation.

entity : stockage de tous les types d'entités non (encore) validées

informationGeographique : stockage d'information géographique

opinionActeurDocument, opinionActeurParagraphe, opinionActeurPhrase :

regroupent les opinions des acteurs

rejected_entity : stockage d'entités équivalentes à d'autres entités.

spatialentity : stockage des entités spatiales validées par une ressource.

spatialentityCandidat : stockage des entités marquées spatiales dans le fichier xml parsé, en attente de validation.

temporalentity : stockage des entités temporelles validées.

Table relatives à la validation des entités :

ressource : description des ressources disponibles pour la validation, nécessaire au module de validation.

b. base de données Parametrage

La volonté de conserver une trace des lancements successifs de la chaine de traitement nous impose le stockage des informations relatives aux fichiers en entrée et sortie de cette même chaine. Nous avons créé une table à cet effet dans la base Parametrage.

Table indexation : conserve les paramètres de configuration pour le lancement de la chaine et le lancement de la validation, c'est-à-dire le nom du fichier en entrée de la chaine et du répertoire de sortie obtenu.

Chapitre 3
Contribution

1 – Chaine de traitement Text2Geo

Le but de Text2Geo est l'identification des entités nommées de type spatial et organisation. Ce traitement s'effectue sur un ensemble d'articles journalistiques en entrée, et fournit en sortie l'ensemble des entités nommées spatiales et organisations détectés. Nous préciserons, dans cette partie, l'organisation des données en entrée et en sortie de la chaîne de traitement et nous détaillerons par la suite les différentes étapes de ce processus de traitement linguistique défini.

2.1. Les entrées / sorties

a. Les données en sortie

La chaine de traitement, défini sous Linguastream[8], est destinée à traiter un seul fichier fourni comme entrée. Ce fichier regroupe les articles journalistiques, collectés d'avance, sous un format XML prédéfint (cf. figure10).

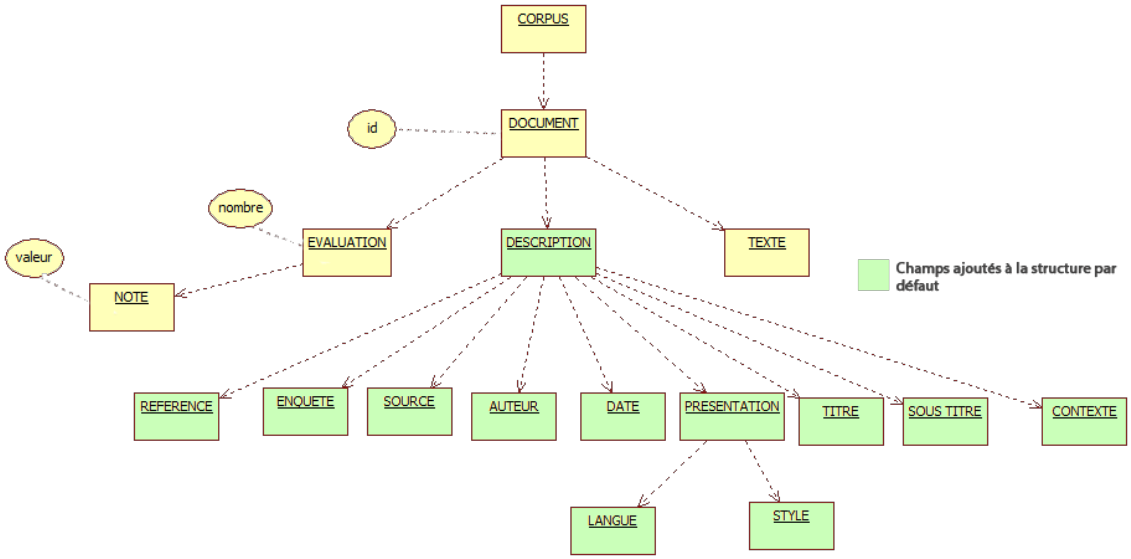


Figure 10: La structure du fichier XML.

La figure précédente présente la structure XML. C'est une structure que nous avons défini, à partir du modèle de DEFT[9]. Nous avons modifié et ajouté certains champs pour adapter la structure de départ à nos données, la figure10 montre les champs ajoutés à la structure. Le fichier en entrée présente un corpus composé par plusieurs documents. Chaque document présente un article journalistique décrit, dans le fichier, par l'ensemble de la structure définie par le schéma.

Nous avons procédé, par la suite, à la conversion des données au format de départ .doc sous le format XML défini. Les articles, sur lesquels nous avons travaillés, présentent des enquêtes et des interviews avec des personnes et des responsables politiques de l'agglomération. Le sujet de ces articles porte, en général, sur les opinions qu'ont les acteurs sur des projets d'aménagement et des projets politiques dans l'agglomération. Les opinions se répartissent sur trois catégories (Pour, Contre et Neutre), et les articles étaient classifiés selon ce critère. Nous avons gardé alors cette

[8] http://www.linguastream.org/whitepaper.html
[9] Défi fouille de textes (DEFT) cf. http://deft.limsi.fr/

information (EVALUATION à NOTE), au cours de la conversion, pour des finalités plus larges du projet.

b. Les données en sortie

Figure 11: Aperçu sur les fichiers de sortie d'un article traité

La sortie de traitement par la suite est un dossier. Ce dossier contient, à son tour, un ensemble de dossiers dont chacun correspond à un document traité (article journalistique). Pour chaque article, la chaîne de traitement linguistique génère un fichier XML (document1 dans la figure11) qui contient la structure par défaut (<DOCUMENT></DOCUMENT>) et deux fichiers textes. Les fichiers textes listent les deux types d'entités nommées détectés pour chaque article. Dans la figure11 le document 2 correspond à la liste des entités spatiales dans le texte et le document 3 correspond à la liste des entités organisation.

2.2. Le traitement linguistique

La chaîne de traitement linguistique à intégrer dans cette application est déjà fournie au début. Programmer cette chaîne n'est pas une tâche de notre travail. Par contre, nous avons été appelés à étudier Text2Geo pour comprendre son fonctionnement. Nous présenterons dans cette partie un résumé sur les différentes étapes de traitement en se basant sur le travail de recherche de (Tahrat, 2012) et la version implémentée sous linguastream (cf. figure12). Cette description peut porter quelques petites différences avec celle qui va exister réellement sous l'application.

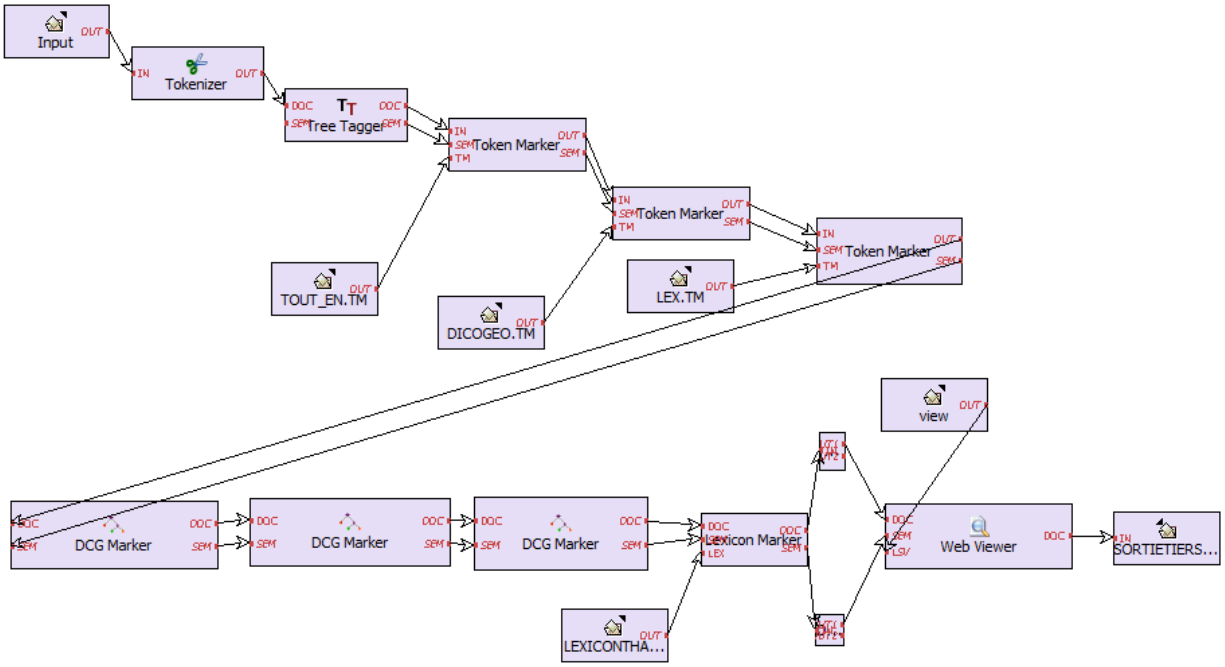


Figure 12: la chaine de traitement linguistique sous Linguastream.

La chaîne de traitement linguistique se compose de 3 principales phases de traitement :

a. L'étiquetage des données (analyse morphosyntaxique)

Les données en entrée sont en format XML. Le processus suivant, consiste à passer les données au *Tokenizer* pour la tâche de segmentation qui vise à découper le texte en unités lexicales. La tâche finale de cette phase consiste à effectuer l'analyse morphosyntaxique avec *TreeTagger* afin de désigner la morphologie et la fonction syntaxique (lemme + catégorie) pour chaque mot dans le texte.

b. Le marquage des Tokens

Cette deuxième phase, divisée en trois sous étapes, est basée sur l'utilisation de patrons syntaxiques. La première a pour but de créer un squelette vide pour tous les tokens marqués. Le squelette prend la forme suivante :

```
<intro>

<type>non</type>

<stype>non</stype>

</intro>

<egn>non</egn>

<val>non</val>
```

Ce squelette permettra de stocker les informations identifiées lors des prochaines étapes du traitement. Ces informations vont être décisives pour identifier les EN des différents types.

Dans la deuxième étape de marquage, l'attribut <val>non</val> subit un changement pour les mots (Tokens) qui figurent dans la base des termes utilisée *DICOGEO.TM* et qui correspondent à des noms de lieux. L'attribut <val> prend, par la suite, la valeur oui dans ce cas.

<val>oui</val>

La troisième étape de marquage (TokenMarker3) se charge de repérer les indicateurs spatiaux présents dans le texte et de préciser le type et le sous-type de chacun. Un lexique (LEX.TM) est utilisé dans ce but. Pour les tokens identifiés comme indicateur spatial à l'aide de ce lexique, la balise <intro> </intro> est mise à jour en précisant le type <type>geo|distance|route|…</type> et le sous-type <stype>KM|yard|vielle|…</stype>.

c. La détection des instances

La troisième phase de traitement est répartie sur trois modules appelés *DCG Marker*. Ces modules implémentent des règles logiques prédéfinies dans le but d'identifier des instances dans le texte qui valident ces règles. Dans un premier temps, le premier module procède à l'identification des entités nommés (EN) qui se présentent sous des types primitifs non complexes. Par la suite, le deuxième module applique ses règles dans le but d'identifier des formes plus complexes. Finalement, le dernier module implémenter procède un filtrage de résultat pour séparer les EN spatiales et les EN organisation.

2 – Parsing et stockage de données

L'objectif défini pour cette étape est de parser les résultats obtenus après le traitement linguistique pour remplir notre base de données. Garder ces informations dans une base de données, de façon permanente, est important et plus pratique pour les prochaines étapes de traitement. La sortie de la chaine de traitements linguistiques se présente sous forme de fichiers structurés. Nous avons défini un modèle adapté pour pouvoir interroger ces données et les stocker dans la base « senterritoire ».

Figure 13: Le parsing et le stockage dans la base.

Le lancement du parsing se fait via l'onglet indexation en faisant appel à une instance de la classe *Parsing.java*. Cette classe s'occupe de parcourir le contenu du répertoire racine fournit en paramètre et de remplir la base de données avec les éléments en relation avec chaque article. Ces éléments sont extraits en faisant appel à une instance de la classe *FoldParser* pour chaque document traité. Les méthodes définies dans la classe *FoldParser* permettent de récupérer ces éléments à partir des fichiers relatifs au document sélectionné.

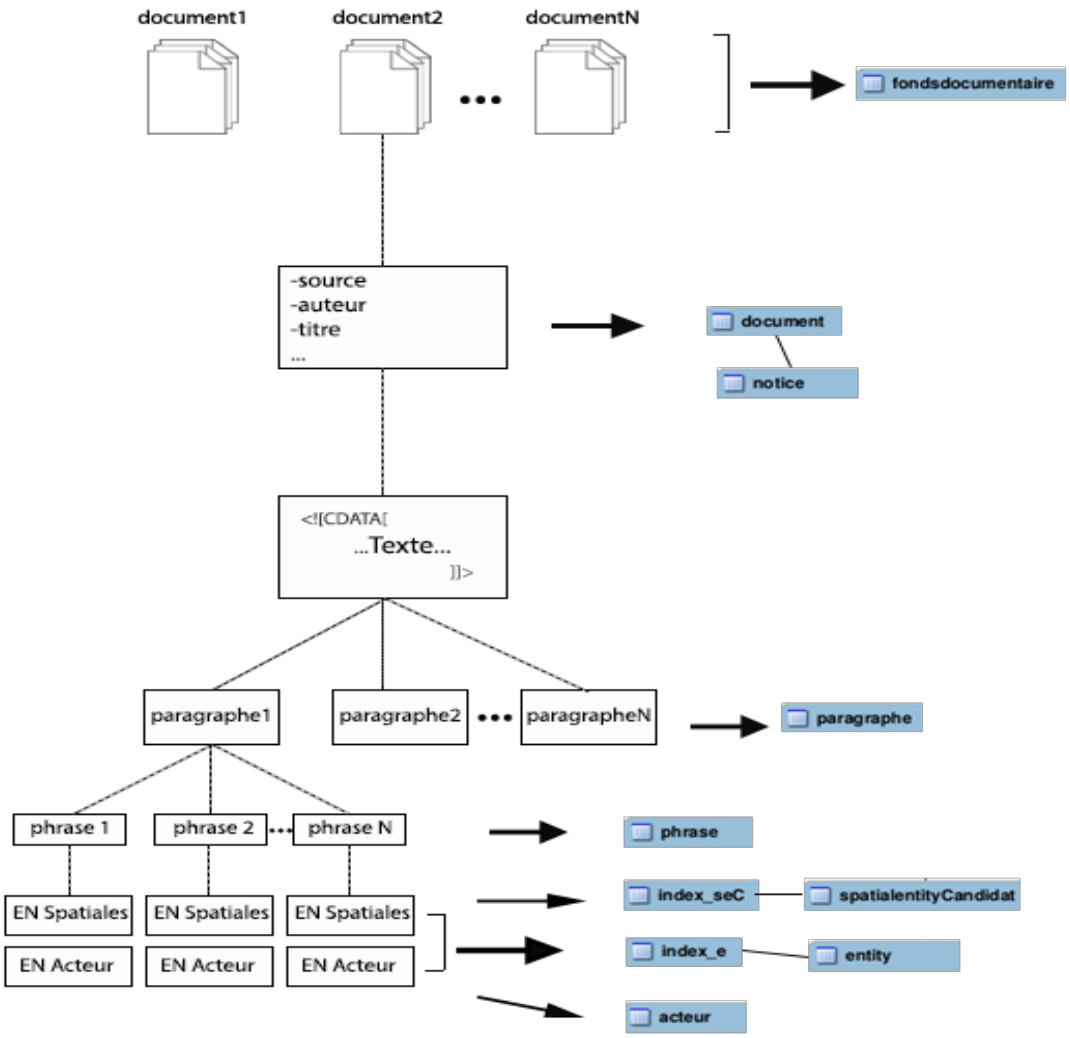


Figure 14: Schéma détaillé du parsing.

Pour chaque article traité, la chaine de traitements renvoi trois fichiers en sortie : un fichier XML et deux fichier texte. Nous avons implémenté un module de parsing à l'aide de l'API JDOM ce qui permettra une manipulation simple et optimisée pour les fichiers de type 'XML'.

La première étape de parsing consiste à interroger la table indexation de la base de données « parametrage » pour récupérer le chemin d'accès au résultat du traitement linguistique. Le contenu du dossier racine est organisé sous forme de sous-dossiers où chaque sous-dossier correspond à un document (article) traité. Un sous-dossier contient donc les trois fichiers générés par l'étape de traitements linguistiques. Pour chaque sous-dossier, le module commence par traiter le fichier XML afin de récupérer en mémoire les informations liées à l'article en cours de traitement. Par la suite, il récupère la liste des EN Spatiales et celle des Acteur à partir des deux fichiers textes. Les données récupérées seront insérés dans la base « senterritoire » à l'aide des classe de mapping Hibernate. La figure précédente précise les tables mappées au cours de cette tâche.

Au moment de l'insertion dans la base, le module segmente les textes en paragraphes, les paragraphes en phrase et identifie la présence des EN dans les phrases. Cette opération a pour but de repérer les EN et les situer par rapport à la structure générale du texte dans le texte. Les relations sont conservées dans la base et gérés par les classes de mapping au cours de l'insertion.

3 – Validation des entités

3.1. Veille et comparatif de ressources pour la validation d'entités nommées

a. Objectifs et démarche de la veille

Notre objectif est le choix d'une ressource permettant la validation d'entités nommées de type spatial ou organisations. Nous avons recueilli à des fins de comparaison pour chaque ressource trouvée les données suivantes: son nom, son descriptif, des lien(s) vers des informations, le type de ressource, les auteurs, la source, le domaine couvert, le mode d'accès et la licence, les limites, les avantages. Nous recherchons avant tout une base libre et gratuite, téléchargeable ou accessible aisément via un service-web, en langue française et couvrant le plus exhaustivement possible la zone du Bassin de Thau qui nous intéresse. Ce dernier critère est de loin le plus difficile à remplir. Afin de vérifier la capacité d'une ressource à y répondre, nous avons extrait dans les textes analysés par la chaîne un certain nombre d'entités représentatives afin d'interroger (manuellement) chacune des ressources étudiées à leur propos (quand cela était possible). Les résultats obtenus figurent dans la dernière colonne de notre tableau comparatif où nous listons les mots trouvés/non trouvés.

Voici la liste des mots extraits des textes qui nous ont servis pour des tests :

- Entités de type organisation :

Midi Libre, Montpellier Agglomération, Georges Frêche, Philippe Sans, Sétois

- Entités de type géographique :

Montpellier, (bassin de) Thau, Poussan ou Frontignan, Hérault, Languedoc-Roussillon, Port-Marianne, (aéroport de) Fréjorgues, Sétois.

Nous établissons ensuite un classement sur la base de ces critères.

b. Bilan de notre étude et description des ressources trouvées

- **Domaine :**

Nous avons comparé 29 ressources pour la validation d'entités spatiales et 8 pour la validation d'entités organisation. Lorsque le domaine de la ressource était uniquement les noms propres comme pour le projet Prolex du Laboratoire d'informatique (LI) de l'université François-Rabelais de Tours, les tests ont révélés une inefficacité à reconnaître les entités nommées de type organisation. Bien que nous ayons dans notre comparatif des ressources dont le domaine ne concernent que des organisations (unesco), les ressources les plus adaptées pour la validation des entités nommées organisation se sont révélées être celles dont le domaine était général. Les données des ressources géographiques se focalisent sur des noms géographiques, des informations démographiques, des adresses, des cartes, des régions administratives ou de la végétation et des territoires agricoles.

- **Accès :**

L'interrogation des ressources que nous avons récoltées se fait par deux biais principaux : par les moyens d'une API ou le requêtage d'une base téléchargée qui peut être relationnelle ou xml (dump rdf). Le langage d'interrogation peut donc différer (sql, sparql, etc.) ainsi que le format de retour. Les données de type sémantique peuvent s'appuyer sur un thésaurus ou une ontologie. En ce qui concerne ces données disponibles en ligne, les sparql end point constituent un point d'accès supplémentaire. L'accès est parfois soumis à des conditions plus ou moins restrictives : licence payante (getty thesaurus), demande à faire auprès d'un organisme (la poste), nombre de requête

limité (geonames)...Le critère de l'accès a ainsi joué un rôle essentiel dans le filtrage des ressources possibles.

- **Type de ressources**

Il existe des standards de protocole de communication pour les services-web de type géographique maintenus par l'Open GeoSpatial Consortium. Par exemple, les web feature services (les plus intéressants) permettent de manipuler et de mettre à jour des données de type géographiques. Un service-web peut également donner accès à un lexique géographique. Quant aux bases de données à thématique géographique rencontrées, elles peuvent appartenir à l'une de ces catégories : Systèmes d'informations géographiques (SIG) qui permettent de créer, d'organiser et de présenter des données alphanumériques spatialement géoréférencées (pouvant servir à la production de plans et de cartes), et les lexiques comme par exemple les gazetteers, des dictionnaires géographiques donnant des informations sur des pays, régions ou continents ainsi que leurs statistiques sociales et leurs caractéristiques physiques. Dans la problématique qui est la nôtre, il importe davantage d'avoir accès à des données de type lexique que des données géoréférencées.

- **Langue**

Nous privilégions, du fait de la zone géographique voulue, les ressources en langue française ou multilingue, autorisant la recherche en langue vernaculaire.

c. Tests sur la zone géographique

Une de nos préoccupations essentielles est la recherche d'une base efficace à couvrir la zone du bassin de Thau, ses éléments spatiaux (comme par exemple ses communes), ses organisations et ses acteurs locaux. Or, bon nombre des ressources que nous avons collectées, bien que très précises, sont génériques et adaptées à l'échelle mondiale. Les résultats de nos tests pour les organisations ne sont pas très bons : dans le pire des cas rien n'a été trouvé ; dans le meilleur, nous échouons à trouver des personnalités uniquement célèbres localement comme le conseiller municipal Philippe Sans tandis que Georges Frêche, plus connu, est parfois trouvé. Ce constat n'est pas surprenant en raison du grand nombre de personnalités existantes et de la grande zone géographique couverte par ces ressources. Il existe un site web renseignant l'ensemble des élus de France mais il n'est pas possible d'utiliser ses données autrement que via un formulaire en ligne. Les résultats des tests pour les entités spatiales montrent leurs limites également : les noms de commune ou de région donne une réponse quasi systématiquement y compris pour les petites communes dans les très grandes bases, cependant, il est plus rare de trouver le terme de "Thau" (nom d'un étang) et rarissime, de rencontrer des termes se rapportant à des quartiers (Port-Marianne à Montpellier) ou des composants d'entités spatiales très petites (aéroport de Fréjorgues).

d. Les meilleures ressources

Nous avons estimé comme meilleures, les ressources DBPedia et freebase pour les organisations et les ressources NGA GEOnet Names Server, Fuzzy Gazetteer, wikimapia et freebase, pour les entités spatiales.

- **Organisations :**

Les deux bases retenues sont accessibles via des services-web, couvrent un domaine général et une zone géographique large. Les tests réalisés sont les plus concluants de tous. Ces ressources sont efficaces également pour les entités spatiales. Dbpedia est une encyclopédie collaborative dont il existe une version française. DBpedia a la particularité d'être au format rdf, ce qui suppose l'utilisation des standards du web de données comme le sparql ou le xml. Pour éviter la complexité de l'interrogation en sparql, il est possible d'utiliser le lien

http://fr.dbpedia.org/page/{nom-de-l'entité}. Quant à Freebase, il s'agit d'un corpus entièrement composé d'entités, ce qui correspond tout à fait à nos attentes. Le format de réponse est le Json assez aisément manipulable. Cette seconde solution est probablement plus facile à mettre en place et à utiliser.

Tableau 1: Comparatif des meilleurs résultats.

Nom	DBpedia	FreeBase
Descriptifs	Données de wikipedia disponibles sous une forme Web de données	Corpus sur les films,les noms des lieux, de livres d'émission télé,des entreprises
Type	Encyclopédie,service-web	Service-web,corpus,graphe d'entités
Auteurs	collaboratif	google
Source	wikipedia	
Domaine	général	général
Mode accès/licence	Sparql en point,tléchargement rdf	Freebase API,creative commons Attribution License
Limites	Utilisation de la version française-interrogation SPARQL-les élus peu connu n'ont pas forcément une page personnelle	Pas de célébrité peu connues
Avantages	Permet de retrouver des infos comme le nom du maire de Montpellier-efficace avec les entités spatiales-aide à la désambiguïsation	Format de réponses Json-requêtage simple-ne contient que des entités-aide à la Désambiguïsation-efficace avec les entités spatiales
Test	**Trouvés :**Midi Libre,Montpellier Agglomération,Frêche,Sétois ,Port-Marianne,Fréjorgues,Poussan **Non trouvé** :Phillipe Sans	**Trouvé** :Georges Frêche,Midi Libre,Montpellier Agglomération **Non trouvé** :Phillipe Sans, Sétois

- **Spatiales :**

Nous avons retenu en majorité des services-web, ainsi qu'une base téléchargeable au format .txt, NGA Geonet Names Server, pour insertion dans une base de données relationnelle. Wikimapia est très performant sur les entités spatiales locales même petites et localisées. Freebase est performant à la fois pour les entités spatiales et organisations. Fuzzy Gazetteer, enfin, a l'avantage de permettre des recherches floues pour contourner les erreurs typographiques comme les inversions de lettres, mais son interrogation est difficile.

Tableau 2 : Comparatif des meilleurs résultats.

Nom	NGA Geonet Name Server	Wikimapia	Freebase	FuzzyGazetter
Descriptif	Base de données de la NGA (National Geospatial-Intelligence Agency) et le U.S. Board on Geographic Names's (BGN).	base de donnée géographique,inspirée de google maps et wikipédia ,visant à cartographier et décrire la terre vue par satellite	corpus sur les films,les noms des lieux,de livres d'émission télé,des entreprises,	FuzzyG (or Fuzzy Gazetteer) propose des noms de lieux mondiaux et gère les variations d'écriture et la recherche floue.
Type	base de données + services web map service + web feature service	site web, service web	service-web, corpus, graphe d'entités	gazetteer
Auteurs	national geospatial intellignecy agency	Alexandre Koriakine,Evgeniy Saveliev	google	isodp project, Christian Kohlschütter
Source				
Domaine	coordonnées, noms géographiques, mondial, en dehors des États-Unis	planète terre	général	administratif, hydrographique, localités, lieux peuplés, végétation
Mode accès/licence	téléchargement par pays .txt + services-web	url	freebase API, Creative Commons Attribution License	url ?http://isodp.hof-university.de/fuzzyg/query/ ?q=bassin+de+Thau
Limites	Pas de suppression d'une entrée sauf duplication - Web services plutôt orienté cartes	orienté cartes	pas de célébrités peu connues	Beaucoup d'homonymes-mode d'accès difficiles-format de réponse difficile
Avantages	Mise à jour mensuelle.	le plus précis sur les zones locales même très petites, formats de retour possibles (xml (default), kml, json, jsonp, binary)	format de réponse Json - requêtage simple - ne contient que des entités - aide à la désambiguïsation - efficace avec les entités spatiales	On peut régler la précision de la recherche (permet de traiter les erreurs de frappe,…)
Test	**Trouvés** :Thau (étang de Thau), Montpellier, La Paillade, Poussan, Languedoc-Roussillon **No n trouvé** :Port-Marianne, Fréjorgues	montpellier,languedoc-roussillon,bassin de thau,hérault, aéroport de Fréjorgues, port marianne **Non-trouvé : /**	**Trouvés** :Georges Frêche,Midi Libre,Montpellier Agglomération **Non trouvés :** Philippe Sans, Sétois	**Trouvés** :Montpellier, Bassin de Thau, Poussan, Hérault, Languedoc, Langudeoc-Roussillon, Roussillon, La Paillade **Non trouvés** :Port-Marianne, Fréjorgues

Dans notre version de l'application Senterritoire, seulement deux ressources (uniquement pour la validation d'entités spatiales) pourront être utilisées. Nous avons conservé NGA et WIKIMAPIA. Contrairement à FuzzyGazetteer, l'accès à ces bases est facile à mettre en œuvre. Toutes deux proposent des informations de localisation qui faciliteront la visualisation géolocalisée future des entités. Nous avons écarté Freebase pour le moment car son domaine est général.

3.2. Module de validation

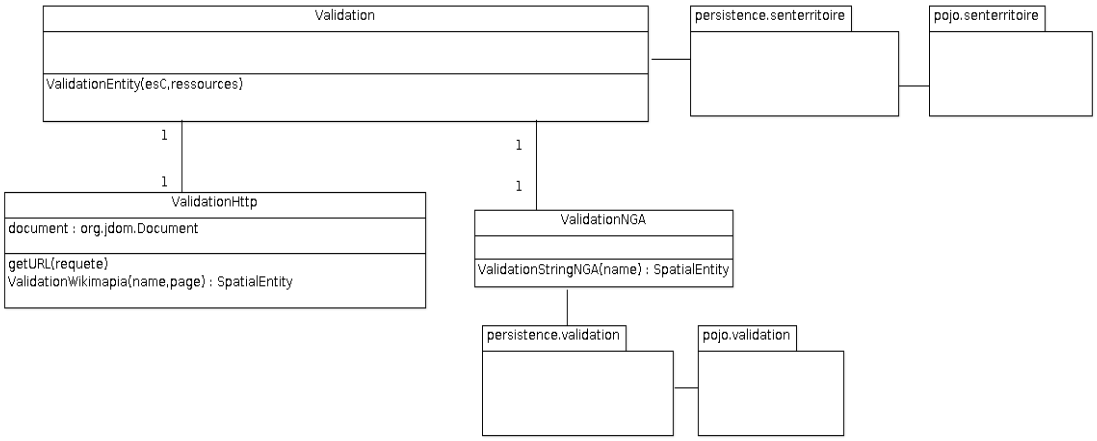


Figure 15: Diagramme de classes pour la validation.

Comme nous pouvons le voir dans le diagramme de classes de la validation (cf. figure13) Validation utilise les classes Hibernate Senterritoire et on utilise deux autres classes "ValidationHttp" et "ValidationNGA" en fonction de la ressource choisi. On remarque également que la partie avec la base de la National Geospatial-Intelligence Agency nécessite également des ressources Hibernate mais lié à cette base.

Dans la partie suivante nous verrons le fonctionnement général du module de validation.

a. Module d'interrogation des données

L'étape de validation peut être décomposée en plusieurs parties (cf. figure13), La première étape consiste à récupérer les entités spatiales candidates, c'est à dire les entités spatiales trouvés par la chaîne de traitement linguistique et ensuite stockés dans la base par le module de Parsing que l'on trouvera dans la table "SpatialEntityCandidat".

Les éléments récupéré de la table ainsi que les ressources à utiliser pour la validation saisies dans le formulaire vont être passés en argument a la méthode ValidationEntity(List<SpatialEntityCandidat>,List<String>). Ces entités vont être ensuite validées pour chaque ressource et finalement, chaque entité validée sera inséré dans la base Senterritoire. Ensuite, les candidats vont être validés de différente façon suivant la ressource, plus précisément, les requêtes ne vont pas être les mêmes pour chaque.

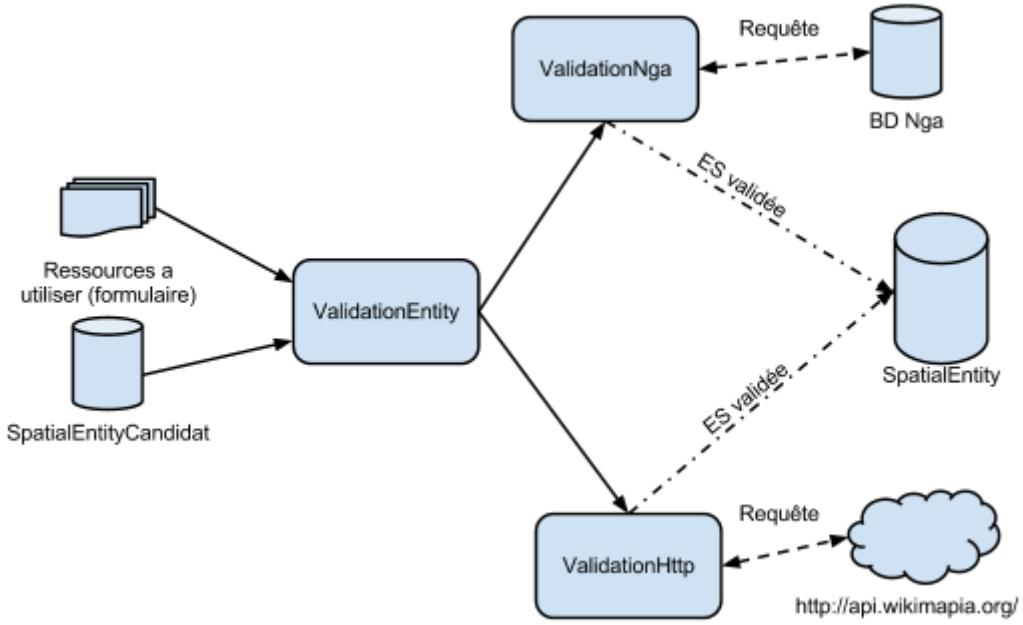


Figure 16: Schéma de principe de la validation

b. Validation National Geospatial-Intelligence Agency (NGA)

La base de données de la National Geospatial-Intelligence Agency est une ressource, interrogeable en ligne, qui référence plusieurs lieux et données Géospatiales dans le monde.

Dans le cadre de notre projet, nous avions la possibilité de télécharger la base en local, ce que nous avons choisi car cela permet d'avoir une ressource pour la validation hors-ligne consultable sans internet mais également car les bases téléchargeables sont localisés par pays.

Le fichier récupéré étant en texte simple (.txt) il y a eu une étape de conversion en .csv puis une importation en MySql pour pouvoir utiliser cette ressource sur un système de gestion de base de données.

Une fois cette base prête, nous allons envoyer des requêtes a une base de données relationnelle, pour cela nous utiliserons donc Hibernate comme pour le reste du projet. Un mapping sera donc réalisé ainsi que des requêtes de sélection.

Les éléments récupérés de la base par la requête seront le nom de l'entité spatiale ainsi que ses coordonnées (cf. figure15). Cette base nous proposant plusieurs noms pour le même lieu, nous avons la possibilité de tester sur ces différents noms.

Enfin la validation d'une entité sera réalisé par la comparaison d'une chaîne de caractères entre le nom récupéré de la requête et celui de l'entité spatial candidate que l'on veut valider, cette comparaison se fera grâce au calcul de la *distance de Levenshtein*[10] entre ces deux derniers.

[10] http://fr.wikipedia.org/wiki/Distance_de_Levenshtein

#	RC	UFI	LATITUDE	LONGITUDE	FULL_NAME_RO	FULL_NAME_RG
1	1	-1968141	49.85	4.93333	Ruisseau de Rebaix	Rebaix, Ruisseau de
2	1	-1969595	49.8833	4.75	Semoy	Semoy
3	1	-1963481	51.05	3.71667	Lys	Lys
4	1	-1961335	51.15	2.71667	Yser	Yser
5	1	-1852216	49.25	6.83333	Rosselle	Rosselle
6	1	-1827782	50.3667	7.6	Moselle	Moselle

Figure 17: Aperçu de la base National Geospatial-Intelligence Agency (NGA).

c. Validation WikiMapia

Pour la validation à partir de Wikimapia une requête http va être générée de la forme : “http://api.wikimapia.org/?function=search&q=Montpellier"&format=xml&count=1000[...]” avec function= qui précise le mode : ici recherche, q= qui va correspondre a la requête voulu : ici Montpellier, format=xml précise qu’on veut récupérer les données renvoyées au format xml.

La structure de données xml renvoyées va contenir tous les résultats de la requête dans une balise <place> et diverses informations dont les plus importantes sont le nom (<name>) et les données de localisation (<location>).

L’étape suivante sera de parser le fichier xml renvoyé pour récupérer les informations voulues. Comme pour la base “Nga” nous allons récupérer le nom de l’information géographique ainsi que ses coordonnées.

Pour valider une entité spatiale nous allons de la même façon que précédemment, comparer deux chaînes de caractères basés sur la *distance de Levenshtein*.

```
−<place id="23567340">
    <name>Ville de Montpellier</name>
    <url/>
  −<location>
      <lon>-35.2206284</lon>
      <lat>-5.8202792</lat>
      <north>-5.8201057</north>
      <south>-5.8204526</south>
      <east>-35.2204192</east>
      <west>-35.2208376</west>
    </location>
  −<polygon>
      <point x="-35.2207491" y="-5.8201057"/>
      <point x="-35.2208376" y="-5.8203245"/>
      <point x="-35.220505" y="-5.8204526"/>
      <point x="-35.2204192" y="-5.8202284"/>
    </polygon>
</place>
```

Figure 18: Aperçu de la structure du fichier xml renvoyé.

d. L’insertion d’une entité validée

Après avoir été validée, une entité spatiale est ensuite insérée dans la base. Dans un premier temps les informations de cette dernière vont être complétées en précisant avec quelle(s) ressource(s) elle a été validée, ce paramètre va être utilisé pour l’étape d’affichage des statistiques plus tard.

Il y a ensuite deux cas pour l'insertion dans SpatialEntity, en effet pour ne pas mettre de doublons dans la table, nous allons vérifier pour chaque élément inséré si il n'existe pas déjà dans la base : Une entité spatiale apparait comme déjà présente si il existe un autre élément avec le même nom et validé avec la même ressource.

Si l'entité spatiale n'existe pas, nous faisons une simple insertion dans la table "SpatialEntity" et la table "Index_se" (cf. figure17) pour garder la liaison entre la phrase et l'entité.

Dans le cas contraire, pour éviter les doublons, nous allons seulement insérer dans la table "index_se" pour ne pas perdre l'information liant une entité spatiale à une phrase. Pour cela nous allons utiliser comme identifiant d'une entité dans la table index celui de son équivalent dans la table 'SpatialEntity'. Pour résumer, si l'on trouve "Montpellier" dans deux phrases différentes, on pointera sur la même entité validée dans "SpatialEntity".

#	id_phrase	id_spatialentity
1	1	7
2	2	1
3	2	7
4	8	1
5	8	7
6	9	1

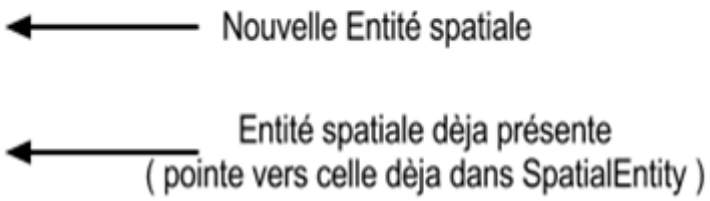


Figure 19: Aperçu de la table 'index_se'.

4 – Interface

4.1. Onglet indexation

a. Analyse, spécification

Ce module consiste en l'ajout d'un onglet et d'un formulaire permettant le lancement de tout le processus de traitement des entités d'un corpus de l'extraction à l'affichage en passant par la validation. Ses spécifications ont été faites dans la partie spécification du besoin chapitre 2.2.

b. Présentation du module crée

Dans ce module, la classe LancementChaineAction.java joue le rôle du contrôleur, treenavig.jsp et indexation.jsp sont respectivement la vue principale et celle qui affiche les résultats du traitement de la chaine, enfin le modèle est matérialisé par les classes des packages persistence.senterritoire (design pattern dao) et pojo.senterritoire (mapping hibernate).

La génération de l'onglet créé est réalisée par le biais de méthodes écrites en JavaScript et l'emploi de composants dojo. Son contenu est le résultat d'un appel à la classe action avec pour paramètres ceux entrés via le formulaire par l'utilisateur. Cet appel renvoie la page indexation.jsp avec ses variables instanciées.

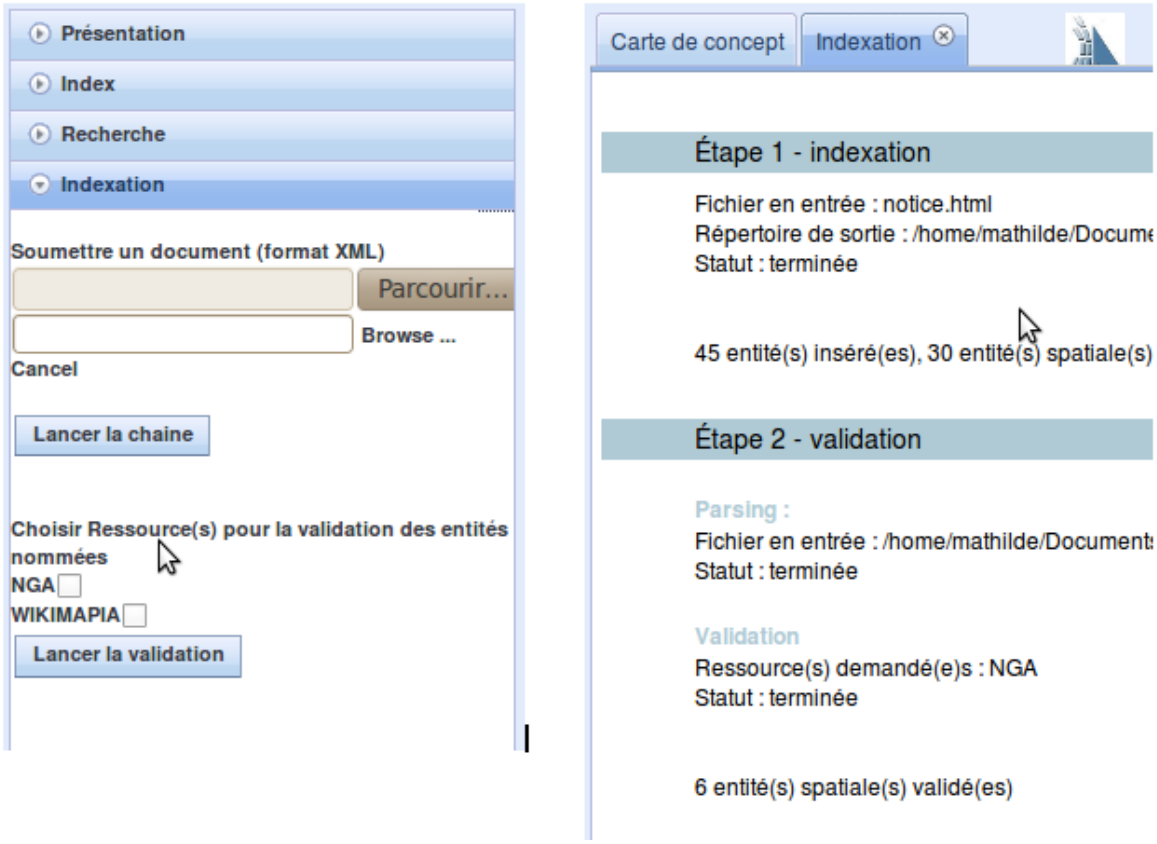


Figure 20: vue du formulaire, vue des résultats du lancement de la chaîne dans l'onglet indexation.

4.2. Onglet statistiques

a. Analyse, spécification

Une fois la validation effectuée, l'utilisateur peut souhaiter connaître le détail des résultats tel que le taux de réussite de la validation par une ressource donnée. L'onglet 'Statistiques' que nous proposons a cette vocation. Par ce biais, des actions comme la comparaison de résultats de validation ou la détection d'entités nommées non validées deviennent possibles.

Dans un premier temps, nous avons défini les informations que cet outil pourra délivrer en nous basant sur les données stockées dans la base Senterritoire lors de l'étape de validation. Nous en avons déduit des représentations visuelles explicites.

- taux total d'entités spatiales validées
- taux total d'entités spatiales non-validées
- taux d'entités spatiales validées par ressource
- liste des entités validées
- liste des entités non-validées
- fréquence d'une entité

Nous ne traitons ici que les entités de type spatial. Lorsque plus d'un type d'entités sera pris en compte par l'application, les statistiques pourront être enrichies de comparaisons basées sur ce critère.

Pour l'affichage des statistiques sous forme visuel, nous avons choisi d'utiliser la bibliothèque Highcharts. Cette bibliothèque permet de créer plusieurs types de graphiques

interactifs en JavaScript. Notre choix est tombé sur Highcharts puisque elle offre une documentation complète en ligne qui nous a permis d'appréhender rapidement avec ses solutions proposés.

b. Présentation du module crée

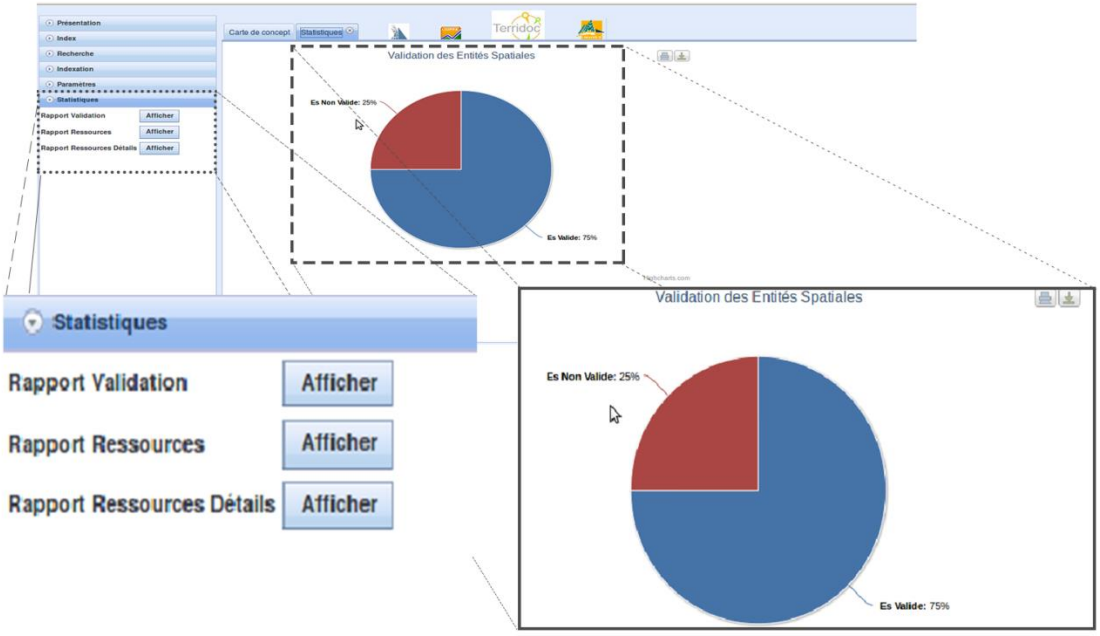


Figure 21: Aperçu de l'onglet 'Statistiques'

Sous l'onglet 'statistiques' nous avons mis en place trois boutons qui correspondent aux différents types de résultats que nous souhaitons afficher. Le premier permet de lancer un graphique de type camembert pour afficher le pourcentage des EN spatiales validés et non-validés par les ressources choisies. Le deuxième bouton lance un graphique de type Barre (BarChart) pour afficher les pourcentages correspondant au EN spatiales validées par chaque ressource. Le dernier bouton est défini pour afficher la liste des entités nommées détectés. Ce dernier affichage est organisé sous la forme d'un tableau qui rassemble les deux types d'EN : les EN validés sont affichés en vert et les non-validés en rouge. Le tableau contient la colonne ressource qui indique la/les ressource(s) de validation pour les EN de première type.

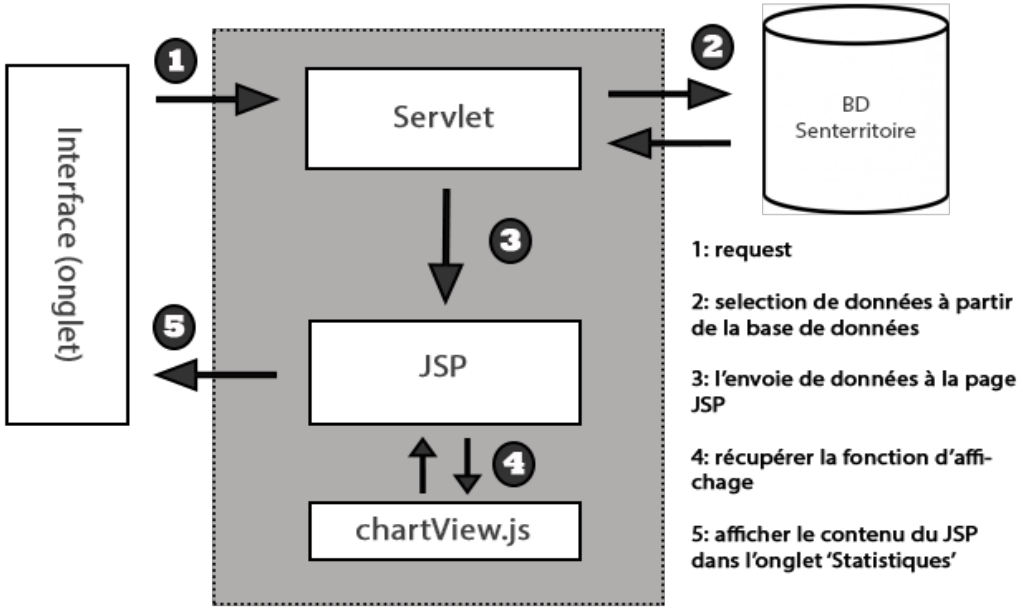


Figure 22: vue du formulaire, vue des résultats du lancement de la chaîne dans l'onglet indexation.

En cliquant sur l'un des boutons un onglet sera créé. La génération de cet onglet est réalisée par le biais de méthodes écrites en JavaScript et l'emploi de composants dojo. Son contenu est le résultat d'un appel à une servlet définit pour le bouton. Cet appel renvoie la page JSP correspondante au type d'affichage lancé. Pour chacun des deux types de graphiques, nous avons défini une fonction JavaScripts dans le fichier ChartView.js, et une page JSP qui inclue l'appel de la fonction correspondante.

Chapitre 4
Perspectives et Conclusion

1 – Perspectives

Nous avons mis en place en un mois les bases nécessaires à l'intégration d'une chaine de traitement Linguastream. Néanmoins, celle-ci ne nous ayant pas été fournie, elle reste encore à ajouter à notre application.

L'étape suivante vers la finalisation de notre travail sera d'étendre le processus de validation et de stockage aux entités de type organisation et temporel. Nous avons d'ores et déjà sélectionné des ressources intéressantes pour la validation d'entités organisation. Nous avons également implémenté les différentes étapes de ce processus de telle manière que l'ajout ultérieur de ressources soit possible. Enfin, la base Senterritoire possède les tables adéquates pour le stockage.

Lors de la saisie dans le formulaire de lancement de la chaine, les informations sur les fichiers en entrée et sortie de la chaine sont conservées avec la date du lancement. Ces informations qui constituent un historique, pourront permettre à l'utilisateur de relancer la chaine dans des configurations déjà entrées précédemment. Cette possibilité n'est pas encore implémentée. Le même besoin d'un historique se fait sentir pour la fonctionnalité de statistiques obtenues après chaque validation d'entités à des fins de comparaison entre les différentes configurations ou les différentes ressources.

Une des visées de l'opération d'intégration de la chaine, est la visualisation des entités nommées extraites. Il s'agira de proposer d'une part un moyen attractif de visualiser des entités spatiales sur une carte géographique et probablement d'autre part un moyen de signifier des opinions exprimées sur les territoires localisés par ces entités. Afin d'anticiper sur les besoins de localisation des entités spatiales, nous stockons dès l'étape de validation les coordonnées et le tracé des zones géographiques rencontrées grâce aux mêmes ressources qui réalisent la validation.

Enfin, afin d'augmenter les sources de données en entrée de la chaine et de profiter de sa mise en ligne au travers de l'application web, le contenu de sites web ou de blogs répondant à notre thématique pourrait être exploité de manière automatique.

2 – Difficultés rencontrées

La première difficulté rencontrée a été le fait de travailler sur une application J2EE. En effet, même si nous avons tous des connaissances en Java, travailler en J2EE était nouveau pour nous et il nous a fallu un certain temps pour faire fonctionner l'application au départ. Malgré cela le fait d'avoir cette expérience en Java et de connaitre le patron MVC nous a permis de nous adapter facilement au J2EE. Nous avons acquis grâce au projet une certaine expérience des technologies J2EE.

Une grosse partie du projet nécessitait une connaissance du framework Hibernate que nous avions déjà utilisé en cours, cela nous a permis de consolider nos connaissances à ce sujet.

Le fait de reprendre une application déjà existante a été également une difficulté dans le sens où il faut un temps d'adaptation pour comprendre son fonctionnement.

Nous avons également rencontré des problèmes au niveau de l'adaptation de la base de données à l'application. Des changements du schéma relationnel ont dû être faits fréquemment pour l'adapter à nos besoins et adapter l'application en conséquence.

En ce qui concerne l'organisation du TER, nous avions une durée assez limitée pour le projet (un mois) et une des personnes du groupe a eu des problèmes de matériel l'empêchant de coder. Ces deux faits nous obligeaient à limiter l'étendue de notre contribution et à une gestion du temps et de la répartition des tâches efficaces.

3 – Conclusion

Dans la première partie du rapport nous avions fixé les objectifs pour le travail d'étude et de recherche qui étaient définis en quatre principaux points:

- Prise en main de l'application J2EE existante TERRIDOCViewer
- Le développement d'un module permettant de lancer la chaîne de traitement linguistique.
- Le développement d'un module de validation des entités.
- Le développement d'un module de visualisation des statistiques.

Au terme de notre projet nous pouvons faire le constat que ces objectifs ont été atteints, l'application déjà existante a été enrichie par les différents modules développés : Nous avons intégré un module pour l'indexation et le lancement de la chaîne de traitement ainsi que le traitement des sorties de cette dernière puis enfin la validation et l'affichage de statistique.

Ce TER nous a permis de nous familiariser avec de nouvelles technologies dont J2EE. Des technologies et frameworks vus en cours ont aussi été utilisés, cela nous a permis de renforcer nos connaissances sur ces dernières.

Finalement, ce TER a été l'occasion pour nous de travailler en équipe de manière organisée avec des objectifs définis, des rapports hebdomadaires, un travail collaboratif grâce au gestionnaire de versions.

Bibliographie

Ehrmann, M. (2008). Les Entités Nommées, de la Linguistique au TAL : Statut théorique et méthodes de désambiguïsation. *Thèse réalisée dans le cadre d'une convention CIFRE (Association National de la Recherche Technique) au sein du Centre de recherche Xerox (XRCE) à Grenoble (sous la direction de* VICTORRI, B*)*.

Kergosien, E. (2011). Point de vue ontologique de fonds documentaires territorialisés indexés. *Thèse pour l'obtention du 'Doctorat de l'Université de Pau et des Pays de l'Adour' (sous la direction de* GAIO, M*.)*

McDonald, D. (1996). Internal and external evidence in the identification and semantic categorization of proper names. *In Boguraev, B. & Pustejovsky, J., éditeurs: Corpus processing for lexical acquisition,* (p. 21-39), MIT Press, Cambridge, MA, USA.

Tahrat, S. (2012). Extraction d'Information Géospatiale dans les documents. *Mémoire de recherche Master2 IFPRU, (sous la direction de Teisseire, M. & Roche, M.)*. Laboratoire d'Informatique Robotique Microélectronique de Montpellier (LIRM) , Université Montpellier II_Sciences et Techniques de Languedoc, France.

Table des annexes

Annexe 1
Gestion du projet

Planification

Notre TER s'est déroulé durant le mois de février. Tout au long de son avancement, nous avons rencontré régulièrement notre encadrant à raison d'une fois par semaine. La demande initiale en développement sur ce projet a évoluée au fil des semaines. Nous avons donc adapté notre planification en conséquence pour rester réactifs face aux changements du besoin. Notre stratégie a été une planification plus détaillée sur une période d'une semaine que sur une période d'un mois. Chaque réunion hebdomadaire donnait lieu à un découpage des tâches à réaliser et une répartition entre les membres. Chaque fin de semaine, nous faisions un bilan de notre avancée de la semaine en termes de tâches effectuées, à faire et en cours dont nous rendions compte à notre encadrant.

Cf. les différents comptes rendus de semaine

Récapitulatif des tâches

semaine	date du CR	Objectifs	dead line
1 et 2	25/01 01/02	● Formater les fichiers sous format doc (contre, pour, et neutre) au format XML en suivant un modèle XML fournit à enrichir. ● Tester la chaine de traitement linguistique fournie et proposition d'amélioration. ● Effectuer l'étude comparative de solutions existantes sur le net pour la validation des entités nommées, des types « Organisation », « Personnes » et « Spatiales », retournées par la chaine de traitement Linguastream. ● Etudier les technologies pour ajouter un module graphique pour le lancement de la chaine de traitements à partir de l'application web existante.	04/02
3	08/02	● Finir l'étude des bases de données géographiques et des organisations pour la validation ● JDOM : Coder le module de parsing XML du fichier de sortie.	14/02

		• Hibernate : Préparation de requêtes pour l'insertion validés dans la base • Hibernate / ? : Préparation de requêtes d'interrogation pour l'étape de validation • Création du module d'indexation / module visuel	
4	15/02	• Parsing XML • Etapes validations : - Recherche par motif flou - Interrogation par Web Services - Insertion dans la Base de données • Formulaire (lancement+visualisation) • Affichage entités valides • Rapport projet	19/02
5	22/02	• Rapport rendu TER. • Finalisation module Parsing Xml • Finalisation module Validation • Module d'indexation • Module Visualisation • début diapos	04/03

Outils de travail collaboratif

Notre effectif est de 4 personnes. Nous nous sommes retrouvés quotidiennement pour travailler néanmoins des outils de collaboration ont été nécessaires. Afin de collaborer sereinement à l'amélioration des fonctionnalités de Senterritoire, nous avons choisi de déposer l'application Senterritoire sur un espace de travail privé sur Assembla (https://www.assembla.com/), dont l'accès a été partagé et les versions gérées via svn.

Les documents de travail comme la documentation, les compte-rendu de travail ou les ordres de création ou d'insertion de la base étaient échangés via le système de partage de dossier dropbox entre les membres du groupe et Eric Kergosien.

Annexe 2
Vues de l'application

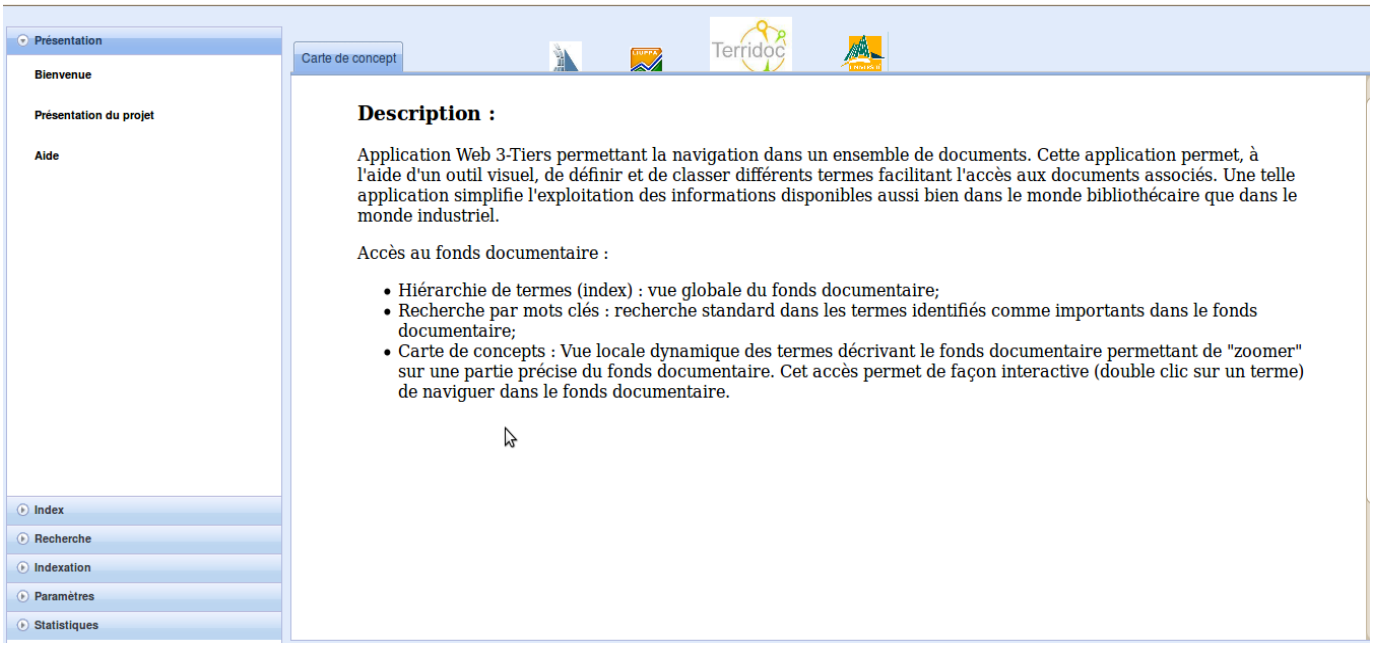


Figure ci-dessus : La page d'accueil de l'application. A gauche, le menu, à droite des onglets.

Figure ci-dessous : Le formulaire de lancement et de la validation de la chaine Text2Geo dans le menu Indexation.

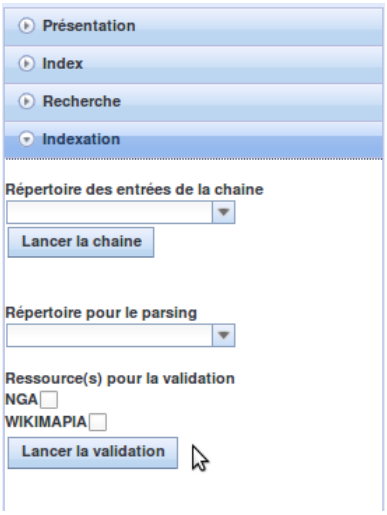


Figure ci-dessous : Le menu indexation à gauche commande le lancement de modules dont les résultats sont affichés dans l'onglet indexation à droite.

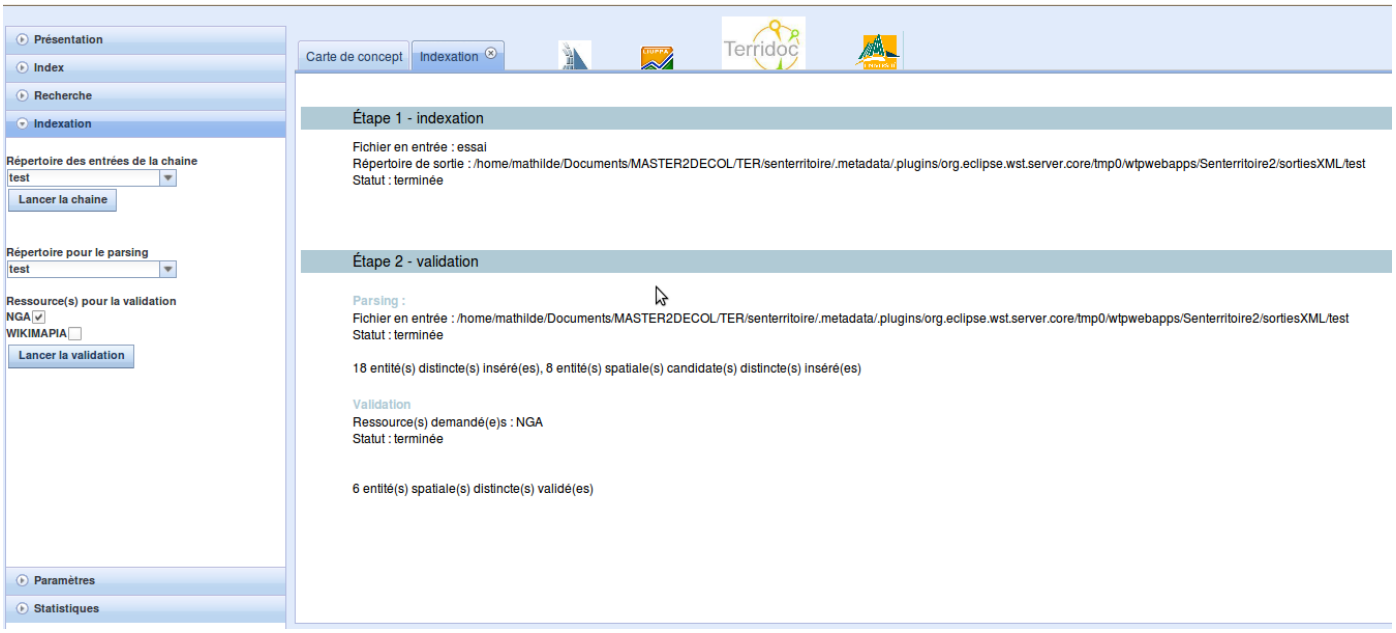
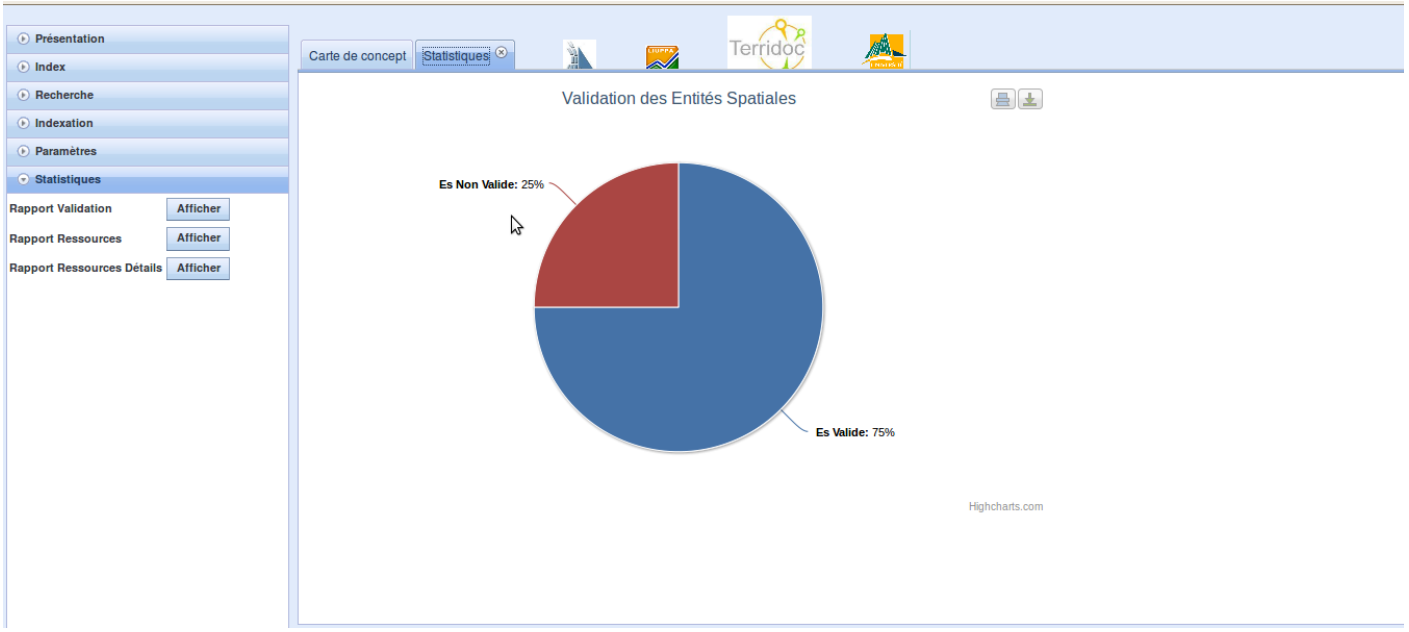


Figure ci-dessous : Le menu Statistiques à gauche permet de choisir les statistiques à afficher dans l'onglet Statistiques.



Glossaire

XML :	Extensible Markup Language.
DTD :	Document Type Definition.
XPS :	XML Paper Specification.
SGML :	Standard Generalized Markup Language.
SQL :	Structured Query Language
SPARQL :	SPARQL Protocol and RDF Query Language
J2EE :	Java Enterprise Edition
JDOM :	Java Document Object Model
MVC :	Model–View–Controller
EN :	Entité Nommée
JSP :	JavaServer Pages
DAO :	Data Access Object
POJO :	Plain Old Java Object
API :	Application Programming Interface
JSON :	JavaScript Object Notation