Université Claude Bernard Lyon 1

Robin LEMAÎTRE – 2021 – 2022

# Projet d'Orientation en Master

## Développement d'un outil de reconstitution de jardins disparus à partir d'un texte descriptif

–

Cahier des charges

Encadré par Messieurs **Fabien DUCHATEAU** et **Franck FAVETTA**

# I. Contexte

De nombreux jardins, monuments ou autres ouvrages d'arts ayant été détruits dans le cours du temps sont recensés et décrits dans divers témoignages, documents ou encore des reportages. Ces jardins ou monuments disparus peuvent être reconstitués numériquement en combinant différentes sources de données descriptives entre elles, mais cela peut représenter une lourde tâche pour l'historien qui doit dans un second temps trouver les informations pertinentes et les lier entre elles (dans l'espace, ou dans le temps).

Prenons par exemple les jardins suspendus de Babylone, en Irak. Ces jardins n'existent plus aujourd'hui, mais on dispose d'une importante quantité d'informations descriptives[1] sur une des 7 merveilles du monde antique. Le travail de recherche de l'historien se décomposerait ici en deux phases principales, sans compter la représentation : premièrement, il doit relever les différentes entités décrites, puis essayer de trouver des relations spatiales entre elles. Dans la description « […] les piliers qui supportent l'édifice se rejoignent par des arcades voûtées », on relève trois entités : des piliers, un édifice, et des arcades voûtées. Avec cette description, on retrouve également dans un second temps leurs relations dans l'espace.

# II. Présentation du projet

Avec ce contexte, et dans le cadre du Projet d'Orientation pour le Master, Messieurs Fabien DUCHATEAU et Franck FAVETTA, tous deux maîtres de conférences au LIRIS, proposent un sujet exploratoire dont l'objectif est de développer une application qui reconstitue numériquement et visuellement des jardins, disparus ou non (bien que le sujet s'intéresse prioritairement à ceux qui n'existent plus), et qui sont décrits dans divers documents (pages Web, documents anciens…). Cela simplifiera la tâche de reconstitution par l'historien, qui n'aura « plus qu'à » rassembler divers documents descriptifs, le reste du travail étant réalisé par l'application.

# III. Objectifs

Ce projet se divise en quatre étapes, mais l'objectif est d'essayer d'avoir une application fonctionnelle tout au long du développement. On commencera donc par faire un algorithme simple de détection des entités et de leurs relations spatiales, puis on essayera d'avoir une représentation visuelle (ou sous simple forme d'un fichier GeoJSON dans un premier temps), tout en s'informant et en recherchant des informations sur le sujet.

## III.1 Etat de l'art

On s'informera tout d'abord sur les différents algorithmes de geoparsing existants, comme geoparsepy[2], et sur les différentes recherches autour du sujet tout au long du développement. On s'informera également sur le traitement automatique du langage naturel (TALN)[3] et les différents algorithmes qui implémentent ce modèle (e.g. NLKT[4]), ce qui pourrait apporter une

[1] Description des jardins de Babylone : https://fr.wikipedia.org/wiki/Jardins_suspendus_de_Babylone

[2] Le projet geoparsepy : https://pypi.org/project/geoparsepy

[3] Introduction au TALN : https://fr.wikipedia.org/wiki/Traitement_automatique_des_langues

[4] Introduction au NLKT de Python : https://www.nltk.org

aide au geoparsing. On commencera par cette étape avant d'attaquer le développement des algorithmes, pour être certain de bien s'orienter.

### III.2 Développement d'un algorithme de détection d'entités

Cet algorithme détectera diverses entités spatiales (un pont, une allée de châtaigniers, un chemin, un ruisseau…). Il détectera d'abord l'entité principale (une allée) puis ses caractéristiques (une allée longue de 100 mètres bordée de châtaigniers).

Dans une première mesure, les entités que cet algorithme sera en mesure de détecter seront assez simples. On désassemblera une phrase sous sa forme « sujet, verbe, compléments », puis on couplera le résultat à une liste de sujets (« chêne », « lac ») et de compléments (« x mètres », « feuillu »), préalablement fournie à l'algorithme. L'objectif est d'avoir le moins de fausses détections possible (comme par exemple, ne pas confondre un verbe avec un sujet ou son complément, etc.).

Il faudra également gérer le cas où il y a plusieurs mentions pour la même entité. Dans l'exemple « Une allée mène à la fontaine, elle est bordée par des châtaigniers », l'algorithme doit comprendre que le pronom « elle » réfère à l'allée précédemment évoquée.

### III.3 Développement d'un algorithme de détection des relations entre les entités

Cet algorithme détectera les relations spatiales des éléments détectés précédemment. Par exemple, il déterminera qu'un ruisseau entoure un parc, qu'une allée mène à une fontaine située au centre du parc, etc. Il faut anticiper le cas dans lequel une description (ou la détection de relations) serait incomplète. L'algorithme positionnera les éléments par défaut, et le repositionnera en fonction des informations qu'il pourra trouver. En effet, la génération de toutes les possibilités pourrait être longue dans le cas d'un grand parc, par exemple. On pourrait également faire en sorte d'indiquer une ambiguïté à l'utilisateur, et ce dernier n'aurait qu'à compléter la description de quelques mots pour supprimer cette ambiguïté.

Pour cette étape, on pourra utiliser le modèle DE-9IM[5], qui décrit les interactions spatiales entre deux entités, le langage SpatialML[6], ou encore l'algorithme de géocodage Nominatim[7].

### III.4 Représentation numérique du jardin reconstitué

Une fois qu'on aura obtenu toutes les informations sur les entités, on procédera à une reconstitution visuelle du jardin. Cette représentation sera une représentation bi-dimensionnelle d'un jardin vu d'en haut.

Plusieurs interfaces de représentations peuvent être envisagées :

- Une zone de texte descriptif sur un côté, et une représentation 2D instantanée (ou non) sur l'autre côté ;

---

[5] Le modèle DE-9IM : https://en.wikipedia.org/wiki/DE-9IM

[6] Le langage SpatialML : https://cran.r-project.org/web/packages/SpatialML/index.html

[7] L'alorithme de géocodage des données OpenStreetMap : https://nominatim.org

- Une zone de dépôt d'un document ou d'un lien vers une page Web, puis une représentation générée à partir de ce document (ou de cette page) ;

Ces représentations peuvent se faire, dans un premier temps, sous forme d'un fichier quelconque ou en console, dans le but de tester les deux algorithmes.

# IV. Contraintes de développement

## IV.1 Technologies

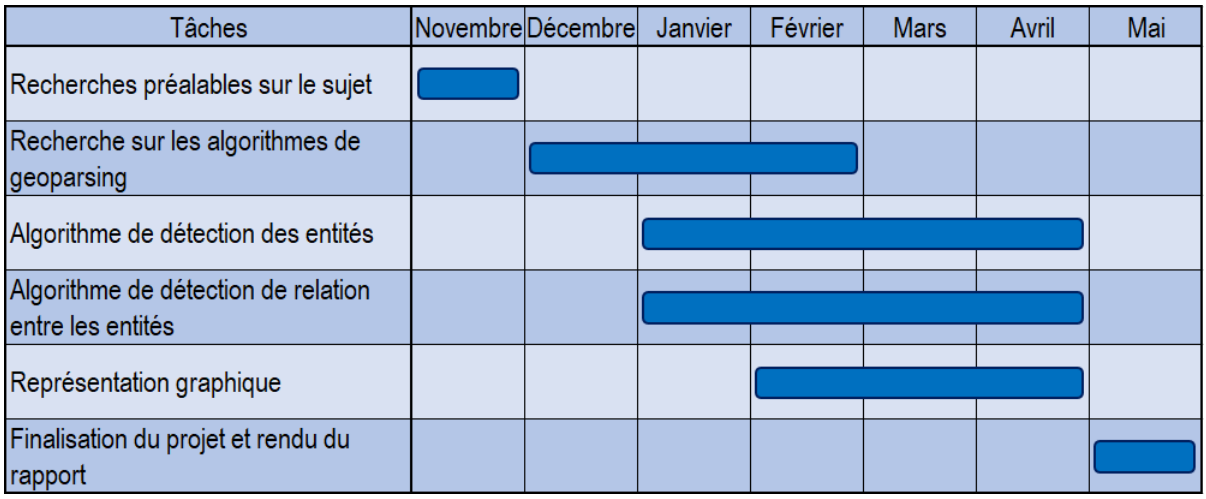
Python sera utilisé pour le développement des algorithmes. Il est adapté au projet, portable, léger et permettra une modularité selon l'implémentation finale de l'application (application Web ou logicielle par exemple). De plus, la plupart des bibliothèques de TALN nous intéressant sont développées en Python. Pour la représentation, on pourra utiliser Tiled[8] ou un Canvas HTML[9], selon le choix final d'implémentation.

## IV.2 Données source et évaluation des algorithmes

On pourra utiliser des données provenant de pages Wikipédia pour les tests. Par exemple, la page du Parc de la Tête d'or de Lyon[10] permettra d'avoir à la fois un texte descriptif assez complet, mais permettra également de comparer la représentation faite par l'application à la réalité. On pourra évaluer nos algorithmes avec ces jardins connus et bien détaillés dont nous connaissons les entités et leurs relations spatiales.

# V. Déroulement prévisionnel du projet

Comme indiqué précédemment dans les objectifs, la recherche, le développement des 2 algorithmes et de la représentation graphique se déroulent plus ou moins simultanément durant le semestre.




[8] Tiled : https://www.mapeditor.org

[9] Les Canvas HTML5 : https://developer.mozilla.org/fr/docs/Web/HTML/Element/canvas

[10] Page Wikipédia du Parc de la Tête d'or : https://fr.wikipedia.org/wiki/Parc_de_la_T%C3%AAte_d%27or