

# Reconstitution d'un jardin à partir d'un texte

Projet d'orientation pour le Master

2021 – 2022

Robin Lemaître – Université Lyon 1

Encadré par Messieurs Fabien Duchateau et  
Franck Favetta

**Résumé.** Ce document présente un projet d'initiation à la recherche dont le but est de proposer une solution à la problématique des détections d'entités et de relations spatiales à partir d'un texte afin de pouvoir faciliter le travail des historiens qui souhaitent reconstituer graphiquement un parc ou un jardin à partir d'une description textuelle. Le prototype est développé en Python à l'aide de la librairie NLTK afin de mettre en œuvre des techniques du traitement automatique du langage naturel.

**Mots-clés :** traitement du langage, grammaire, relations spatiales, extraction d'entités, jardin.

## I. Introduction

De nombreux jardins ou parcs n'existent plus de nos jours, et le seul souvenir qu'il nous reste est au mieux des photographies, ou encore des descriptions dans divers ouvrages. L'historien(ne) qui souhaiterait se faire une idée de la représentation d'un de ces jardins devra d'abord réunir diverses sources, descriptions, extraits de textes, pour ensuite faire des liens entre eux et en tirer une description physique (un dessin, une représentation assistée par ordinateur, etc.).

Le projet d'orientation pour le Master introduit le domaine de la recherche technique et scientifique aux étudiants du M1 Informatique, ces derniers devant choisir parmi un large panel de sujets. Le sujet sur lequel j'ai choisi d'effectuer mes recherches a pour but de proposer des solutions visant à délester les historiens de la partie de conception, grâce à un algorithme qui analyse une description d'un jardin et en retourne une représentation graphique. Ce projet est proposé par Messieurs Fabien Duchateau et Franck Favetta, tous deux maîtres de conférences à l'Université Claude Bernard Lyon 1 pour l'enseignement et au LIRIS pour la recherche, dans l'équipe de *Base de Données*.

Quotidiennement, nous utilisons des outils avec lesquels on peut dialoguer, comme les assistants vocaux sur nos téléphones ou nos enceintes connectées, les messageries automatiques sur des sites internet (« chatbot »), ou encore grâce aux traducteurs automatiques. Ces outils ont un point commun : ils utilisent des techniques du traitement automatique du langage (ou traitement automatique du langage naturel, TALN) qui permettent une analyse lexicale d'une requête (une question par exemple) afin de comprendre cette dernière, et ainsi pouvoir retourner une réponse adaptée. C'est en mettant en œuvre des techniques du TALN comme la détection d'entités que nous chercherons une solution au problème de reconstitution d'un jardin à partir d'une description textuelle. Nous commencerons par décrire l'avancée des recherches sur le TALN et les modèles de relations topologiques (grammaire, modèle DE-9IM), puis nous détaillerons en section III notre proposition pour détecter et extraire des entités ainsi que leurs relations. La section IV décrira notre approche pour représenter graphiquement ces entités. L'implémentation d'un prototype sera détaillée en section V, puis une conclusion et des perspectives seront amenées.

## II. Problématique et état de l'art

Le domaine du traitement du langage a fait l'objet de nombreuses recherches depuis la fin des années 1950<sup>i</sup> <sup>ii</sup>. Plus récemment, parmi les utilisations connues du grand public de ces techniques, on retrouve par exemple celle de l'outil Siri d'Apple<sup>iii</sup>, où l'on discute avec un « chatbot ». L'algorithme traite notre requête, en extrait les informations importantes et pertinentes, et retourne un résultat. Les techniques mises en œuvre par l'algorithme de la marque à la pomme utilisent les avancées des recherches sur le Traitement Automatique du Langage Naturel, un domaine regroupant des méthodes qui permettent notamment l'extraction d'informations

pertinentes depuis un texte. On se limitera au traitement de la langue anglaise pour ce projet, car d'un point de vue technique la formulation des phrases est parfois plus simple qu'en français, mais on utilisera aussi des exemples en français dans ce rapport.

### A. La détection d'entités

Parmi les techniques du TALN, certaines permettent de détecter des entités dans un texte. Les recherches ont été principalement orientées sur la détection d'entités nommées (NER pour Named Entity Recognition), comme « Lyon », « Wall Street » ou encore « Michael Jackson »<sup>iv</sup>. En plus de la détection d'entités non nommées, telles que « un chien », « le département » ou « ce lac », il y a également la problématique qui est de trouver des relations ou des liens entre ces entités. En effet, « un lac » ne désigne pas une entité précise et connue, contrairement à Lyon (on sait qu'il s'agit d'une ville française de plus de 500 000 habitants qui se situe dans le Rhône, mais on ne se sait pas de quel lac on parle). Dans le cas de la détection des entités nommées, une technique très répandue, le *geoparsing*, que l'on pourrait traduire par « analyse syntaxique géographique », permet de situer le contexte et/ou le sujet d'une phrase ou d'un groupe de phrases, en lui associant des coordonnées géographiques (latitude et longitude)<sup>v</sup>. Dans ce domaine du TALN, les techniques de *POS-Tagging*<sup>vi</sup> et de *NP-Chunking*<sup>vii</sup> cherchent à résoudre la problématique d'étiquetage des mots et de groupes de mots.

### B. La détection des relations spatiales

Si une partie primordiale de la problématique est la détection des entités dans un texte, une autre partie tout aussi importante est la détection des relations spatiales entre les entités. Si la détection syntaxique d'une relation peut se faire comme pour les entités, il faut pouvoir la décrire. En effet, que dire de la phrase « des arbres bordent le chemin » ou encore « cette allée mène au lac » ? Comment définir la relation « border » ou « mener à » ? Des travaux de recherche et de développement ont permis de mettre au point différents modèles de description de relations spatiales, comme le modèle DE-9IM<sup>viii</sup>. Ce dernier décrit au total 6561 relations entre deux régions, que ces dernières soient représentées par des points, des lignes ou des polygones. Ces 6561 relations peuvent être ramenées à 10 types de relations. Il existe plusieurs types de relations spatiales : les relations topologiques (e.g. « Le lac est à l'intérieur du parc »), des relations cardinales (e.g. « Le lac est au nord du parc »), ou encore des relations de positionnement précis (e.g. « Le lac est à 3 mètres du chemin »)<sup>ix</sup>.

### C. Positionnement

Notre objectif est de proposer une méthode permettant de détecter des entités et des relations spatiales à partir d'un texte, et plus spécifiquement pour la reconstitution de jardins historiques. Ces propositions sont donc basées sur les recherches actuelles dans le domaine du TALN, notamment la détection d'entités (nommées ou non), ainsi que la définition des relations spatiales du modèle DE-9IM pour lier les entités. Pour détecter les types de mots ou trouver des sous-ensembles de mots dans une phrase, nous nous servirons également des méthodes de *POS-Tagging* et de *NP-Chunking*.

Notre proposition de solution se divise en deux parties, comme le montre la Figure 1. La première permet de détecter des entités avec des relations spatiales depuis un texte, en appliquant un étiquetage de mots pour chaque phrase (*POS-Tagging*), la syntaxe grammaticale (*NP-Chunking*) et la détection de relations. La seconde partie concerne la résolution des ambiguïtés de positionnement des entités puis la visualisation des données. La Figure 1 montre aussi des exemples pour les entrées/sorties (cadres gris en pointillés) pour illustrer les transformations apportées par chaque étape.

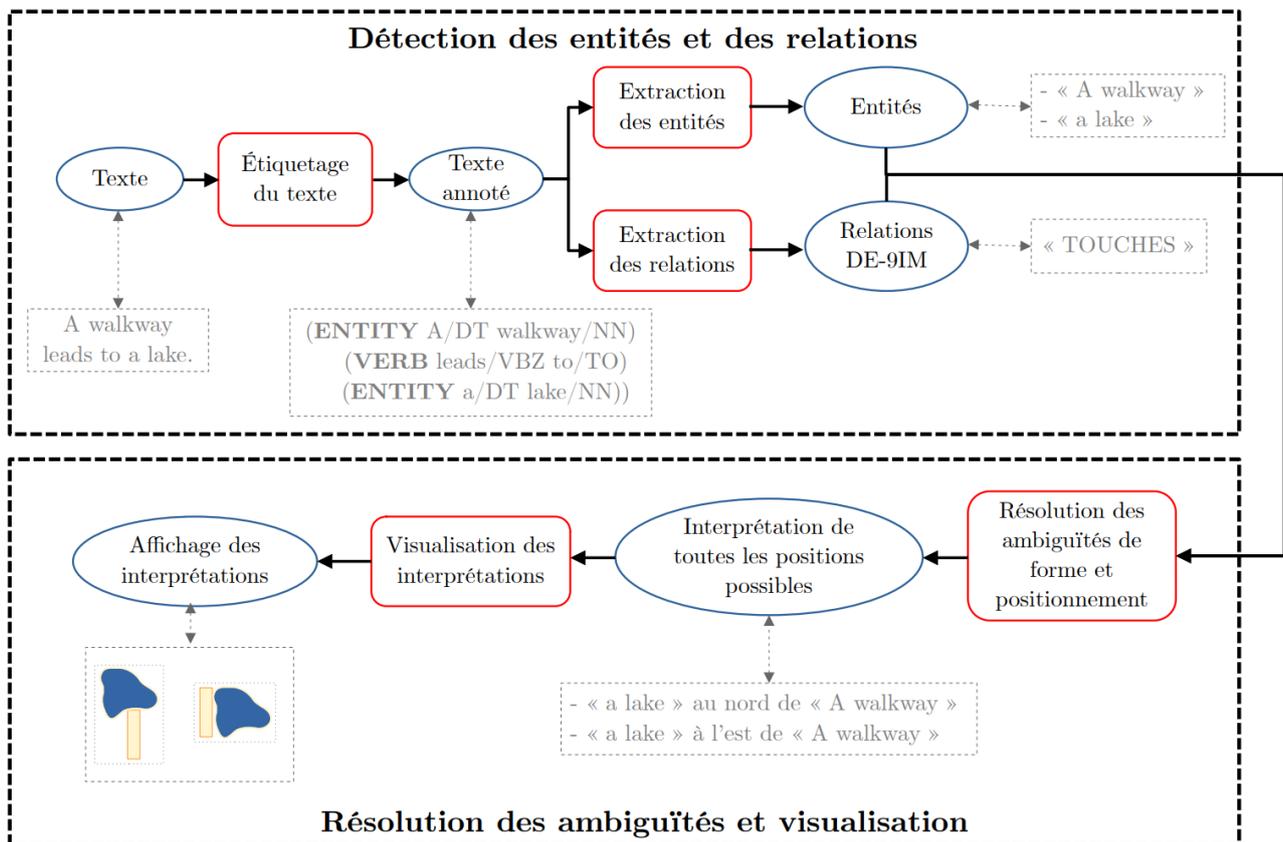


Figure 1 - Les différentes étapes de notre proposition

### III. Détection des entités et leurs relations

En mettant en œuvre les différentes techniques du TALN décrites auparavant, on peut proposer une approche afin de détecter des entités et leurs relations dans une phrase. Bien sûr, nous restons dans le contexte des jardins et des parcs, avec du vocabulaire et certaines tournures de phrases particulières à ce domaine.

Dans un premier temps, les étapes consistent à détecter le type de formulation de la phrase. En effet, on peut formuler une même description de plusieurs manières, comme on peut le voir avec les deux phrases suivantes : « Il y a un lac au nord du parc » et « Au nord du parc se situe un lac ». Les entités n'apparaissent pas dans le même ordre, et la relation n'est syntaxiquement pas la même (« il y a » et « se situe »), mais elles ont la même signification. Quand on trouve le type de formulation, la tâche est extrêmement simplifiée pour détecter les relations. En effet, peu importe la formulation d'une phrase, la description d'une entité atomique (« un lac ») restera a priori la même si on souhaite que la phrase ait un sens.

#### A. Étiquetage du texte

La première étape pour pouvoir étudier une phrase est l'étiquetage morpho-syntaxique (*POS-Tagging* en anglais), qui permet d'associer à chaque mot son type grammatical (un déterminant, un nom, un adjectif qualificatif, un verbe, etc.). Cette technique est la base dans un travail de traitement de langage, car elle permet simplement de donner un sens grammatical à la phrase. Il existe divers algorithmes de *POS-Tagging* mais tous fonctionnent sur le même principe : la liste des mots et du type grammatical correspondant sont déjà connus. Nous utilisons l'algorithme Perceptron Tagger<sup>x</sup>, car il a beaucoup été entraîné avec des corpus préétiquetés et donne donc des résultats très pertinents. Prenons comme exemple la phrase suivante, qui illustrera toutes les étapes de notre proposition : « There is a lake to the north of the park Tête d'Or. A large island emerges from the lake. ». Cette phrase sera étiquetée de la sorte : « There/EX is/VBZ a/DT lake/NN to/TO the/DT north/NN of/IN the/DT park/NN Tête/NNP d/NN 'Or/NNP./ A/DT large/JJ island/NN emerges/VBZ from/IN the/DT lake/NN./ », avec *VBZ* signifiant un verbe conjugué à la troisième personne au présent de l'indicatif, *NN* représentant un nom, etc.

En complément du *POS-Tagging*, chaque phrase est découpée en fonction d'une grammaire (e.g., identifier le groupe nominal, le groupe verbal). Une grammaire est un ensemble de règles permettant de construire des phrases, ou encore de vérifier si elles sont correctement formulées. Si on part du principe qu'une phrase à analyser est bien formulée, alors on peut déconstruire la phrase en groupes de mots qu'on prendra soin de labelliser. Cette étape utilise la technique du *NP-Chunking*, qui permet de découper une phrase à l'aide d'une grammaire que l'on aura rédigée auparavant. Par exemple, on peut décider dans notre grammaire qu'un déterminant suivi immédiatement d'un nom devra être labellisé « Entité » (exemple complet de grammaire dans la section V). Pour la deuxième phrase de notre exemple, le découpage s'effectue de la sorte : « (SPREL (ENTITY A large island) (VERB emerges from) (ENTITY the lake)) », avec, par exemple, SPREL qui est défini par une étape supplémentaire du *NP-Chunking* (i.e. nouvelle exécution de l'algorithme de *NP-Chunking* sur son propre résultat), qui indique une composition de deux « ENTITY » séparés par un « VERB ».

## B. Extraction des entités

Chaque phrase du texte a été annotée, et il faut maintenant en extraire les entités. Dans notre grammaire, le cas le plus simple concerne les groupes nominaux (déterminant suivi d'un nom commun), que nous avons étiqueté comme des entités. La grammaire s'inspire des expressions régulières pour définir les règles de découpage d'une phrase, et des structures plus complexes (e.g., déterminant + adjectif(s) + noms composés) peuvent donc être détectées (e.g., « un grand lac »). Les noms propres (généralement des entités nommées) comme le Parc Tête d'Or sont également détectés et extraits.

Pour une même phrase, plusieurs personnes peuvent chacune avoir leur propre interprétation d'une entité, comme sa taille, ou sa couleur. De ce constat, on en déduit qu'une grammaire, aussi bien écrite qu'elle puisse l'être, ne pourra jamais satisfaire tout le monde. Entre deux phrases, on peut parler d'une même entité mais avec un pronom différent, comme dans la phrase « Le lac est grand. Il est situé au nord » où la même entité qui représente le lac est référencée deux fois, par « le lac » et par « il ». On peut également citer la phrase « Un lac est situé au nord. Une île émerge du lac », dans laquelle on retrouve « Un lac » et « du lac », qui représentent une entité techniquement différente d'un point de vue syntaxique. Comment faire le lien entre deux entités qui, finalement, sont les mêmes ? Pour avoir les résultats les plus pertinents possibles, il est nécessaire de s'inspirer de l'interprétation humaine, afin de définir certaines règles à partir de différents constats :

- « Un lac » signifie à priori que l'on se réfère à une entité qui n'a pas déjà été citée avant, donc on considère une nouvelle entité ayant pour nom « Un lac » ;
- « Il », ou encore « Ce dernier » : ces pronoms se réfèrent-ils au principal sujet cité précédemment, ou au dernier objet de la phrase précédente ? Dans le cadre de nos recherches, nous ne considérerons que le second cas ;
- « Le lac » signifie à priori que l'on se réfère à une entité qui a déjà été citée dans le passé et qui contient « lac » dans son nom ;
- Une phrase peut également être composée de deux parties séparées par un pronom tels que « qui », comme dans la phrase « Un chemin mène au lac, qui est entouré par des arbres ». Dans ce cas, comme pour le pronom « Il », nous avons décidé que « qui » est une entité qui se réfère à la dernière entité citée dans la phrase précédente (ici, « lac »).

Tous ces cas spécifiques mais fréquents sont ajoutés dans la grammaire afin de détecter une majorité d'entités ainsi que leur réutilisation à d'autres endroits du texte. Dans notre texte d'exemple, les entités détectées sont les suivantes : « a lake », « the park Tête d'Or », « A large island », « the lake » (cette dernière se référant à la première entité détectée selon les règles énoncées auparavant).

## C. Détection des relations

Dans le cadre de cette recherche, nous nous limiterons au traitement des relations spatiales les plus fréquemment utilisées à savoir les relations topologiques (non orientées) et les relations cardinales<sup>x</sup>. Il existe plusieurs façons de formuler une phrase afin de décrire une relation spatiale entre deux entités. De la même manière que pour détecter des entités, nous allons d'abord devoir détecter la construction de la phrase avant de trouver des relations. En effet, bien que l'on puisse détecter des entités relativement facilement dans tous

les cas, les relations spatiales, elles, dépendent syntaxiquement de la formulation de la phrase (cf. l'introduction de la partie III). Une fois que l'on a compris comment la phrase est construite, on peut savoir où chercher le groupe verbal décrivant la relation dans la phrase. Pour la phrase « Au nord du parc se situe un lac », on comprendra que le type de la phrase est construit avec l'énonciation d'une région (le nord du parc), d'un groupe verbal décrivant une relation spatiale (se situe), et enfin l'énonciation d'une entité (un lac).

Une fois la détection des entités et des relations réalisée (i.e. notre phrase est découpée en morceaux qui représentent des entités et une relation par exemple), on se rend compte que l'on peut résumer le résultat avec un tuple du type < « Entité A », « Entité B », « Relation spatiale » >, ou encore < « Entité A », « Relation spatiale », « Entité B » >. Nous avons décidé de choisir la seconde représentation que nous trouvons plus naturelle et compréhensible à la lecture : une entité A est liée d'une certaine manière à une entité B. Il existe des spécificités de langage dans ce contexte de parcs et de jardins. Nous sommes dans un contexte de description de jardins, donc on se concentre ici sur des phrases qui permettent de situer deux entités l'une par rapport à l'autre. Par ailleurs, si on essayait d'analyser sur une phrase d'un contexte différent (« Il courait pour ne pas louper son train »), aucune entité ni relation ne seraient détectées car la grammaire de l'algorithme de *NP-Chunking* ne prévoit pas ce cas. On se concentre donc uniquement sur des verbes relationnels et descriptifs. En effet, une fois le verbe de la phrase obtenu, l'idée est de l'associer à une relation issue du modèle DE-9IM, afin de simplifier par la suite le nombre de cas à traiter. Par exemple, les descriptions « A touche B », « A mène à B », « A conduit à B » représentent toutes la même relation entre A et B, c'est-à-dire qu'entre ces 3 phrases, on placera A par rapport à B (ou inversement) de la même manière.

Pour une plus grande simplicité d'implémentation, on considère, parmi les relations issues du modèle DE-9IM (cf. Figure 8 en annexe) et dans le cadre de ce POM, que les relations « CONTAINS » et « COVERS », ainsi que « WITHIN » et « COVERED-BY » sont, respectivement deux à deux, symétriques. Nous retiendrons donc uniquement les deux premières relations. L'implémentation de ce modèle dans l'application permettra ainsi de simplifier la mise en place des relations entre les entités, car le nombre de relations sera réduit à 5 : « touches », « CONTAINS », « WITHIN », « CROSSES » et « DISJOINTS ». Pour résumer, le groupe verbal d'un texte décrivant une relation entre deux entités sera converti en une relation DE-9IM. De plus, nous détectons aussi les relations cardinales (« au nord de », « à l'ouest »). En effet, bien que ces relations soient représentées par la relation « CONTAINS » de DE-9IM (ou « WITHIN » selon la tournure de la phrase), cette dernière ne les décrit pas assez. Il faut donc créer des relations cardinales spéciales dans ces cas-là. Cependant, pour la phrase « Il y a une roseraie au nord du parc » doit-on comprendre que la roseraie se situe au nord à l'intérieur ou au nord à l'extérieur du parc ? Nous nous limiterons au cas où l'entité se situe au nord à l'intérieur de l'autre.

En prenant en compte les synonymes, il existe des centaines de mots anglais permettant de décrire des relations (par exemple, « take », « conduct », « direct » sont des synonymes décrivant la relation « lead »). Grâce à l'outil WordNet<sup>xi</sup>, on peut détecter tous ces synonymes et les placer dans une catégorie globale décrite par le modèle DE-9IM. Après application des règles de ce modèle sur notre liste de tuples, les groupes verbaux sont remplacés par une des relations décrites par le modèle. Le tuple <a lake, surrounded by, trees> deviendra alors <a lake, WITHIN, trees>. L'ordre est important : si la première entité du tuple était trees, la relation aurait été « CONTAINS » pour être équivalente.

À cette étape, nous avons extrait les entités et les relations (remplacées par celles du modèle DE-9IM) d'une phrase ou d'un groupe de phrases. Il existe également des ambiguïtés dans les relations spatiales, mais qui ne concernent pas la détection grammaticale. Nous les détaillerons dans la partie suivante. Pour notre phrase d'exemple, les tuples extraits ressembleraient à ces deux-là :

- < « a lake », « NORTH », « the park Tête d'Or » >
- < « A large island », « WITHIN », « the lake » >

#### IV. Ambiguïtés et visualisation

Jusqu'à présent, nous avons vu comment on peut détecter et extraire des entités et leurs relations spatiales au sein d'un groupe de phrases. Cependant, dans une optique de visualisation des résultats, de nouvelles réflexions

sont nécessaires, comme la détection préalable de leur forme et le positionnement des entités entre elles, et donc l'extraction de toutes les interprétations possibles de notre algorithme.

### A. Ambiguïtés sur la forme

Comment choisir une forme qui n'a pas été fournie par la description textuelle ? Il y a plusieurs façons de représenter une entité : une image, une représentation 3D, une forme polygonale colorée, ou encore simplement un nom que l'on affiche. L'affichage d'une image peut se révéler peu pratique. En effet, ce format n'est pas du tout adaptable. Si on utilise une image de lac pour représenter cette entité, elle sera plus ou moins déformée en fonction de sa taille et de sa position finale, et la superposition d'images (e.g., une île sur un lac) ne donne pas forcément un résultat satisfaisant. L'affichage sous forme d'un nom est la solution la plus simple, mais elle n'est pas vraiment réaliste ni très parlante (que penser du mot « lac » écrit juste au-dessus du mot « chemin » ?). Dans le cadre de cette recherche, nous nous limiterons donc à choisir une forme vectorielle arbitraire pour chaque type d'entité. On n'oubliera pas de fournir également une forme par défaut pour les toutes les autres entités non décrites, ou les entités peu courantes.

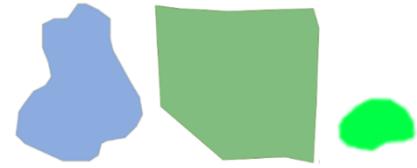


Figure 2 - Exemple de formes par défaut pour un lac, un parc et une île

### B. Ambiguïté de positionnement

#### 1. Positionnement des entités

L'interprétation d'une relation entre deux régions dépend des personnes. Par exemple, pour la phrase « A mène à B », on peut comprendre que A et B sont proches et éventuellement orientés l'un vers l'autre. Pour chaque relation, on va estimer que A sera placé par rapport à B ou inversement. Parfois, une relation issue d'un groupe verbal ne peut pas être retrouvée parmi les relations DE-9IM. Dans ce cas, la relation est considérée comme inconnue. Il existe également la situation où aucune relation n'est exprimée dans la description textuelle. À noter que cette dernière ambiguïté peut être transitive, comme dans le cas où A est en relation avec B, B en relation avec C, mais où rien ne nous indique directement la relation entre A et C.

Pour résoudre ce problème d'ambiguïtés, nous proposons une méthode qui consiste à ajouter incrémentalement les relations et qui résout les conflits générés par la dernière relation ajoutée avec les relations ajoutées précédemment. Chaque entité est positionnée dans un sens l'une par rapport à l'autre (par exemple, dans le cas du lac qui se situe au nord du parc, on considère que le lac est situé par rapport au parc). Étudions le cas où nous avons 2 relations pour 3 entités, comme dans « Un chemin mène à la fontaine, la fontaine est au nord de la prairie ». Le chemin et la fontaine seront positionnés lors du travail sur la relation « mène à », et lors du travail sur « au nord de », on se rendra compte que la fontaine a déjà été citée auparavant et a donc déjà été traitée. Ici, nous faisons face à une relation cardinale (le nord), et nous considérons arbitrairement que pour une telle relation, la fontaine se situe par rapport à la prairie (et donc on modifie le sens des dépendances avec le chemin qui se situe désormais par rapport à la fontaine, si ce n'était pas déjà le cas avant).

Nous avons donc proposé de positionner les entités les unes par rapport aux autres. Notons que le parc est une entité que l'on considère comme différente des autres, car c'est par rapport à cette dernière que nous positionnons toutes les autres. En effet, nous travaillons dans un contexte de reconstitution de parcs et de jardins, nous avons décidé de partir du principe que même si le mot « parc » n'est pas présent dans une description, toutes les entités extraites se situent à l'intérieur. Dans un cas général, l'idée est donc de positionner deux entités l'une par rapport à l'autre. Le choix de l'entité référente et référée se fait en fonction de la relation, de l'existence préalable d'une des deux entités, ou encore du degré de précision dans le positionnement par rapport au parc apporté. Notre but est d'obtenir des chaînages entre toutes les entités extraites du texte. Prenons l'exemple suivant (Figure 3) pour illustrer nos propos.

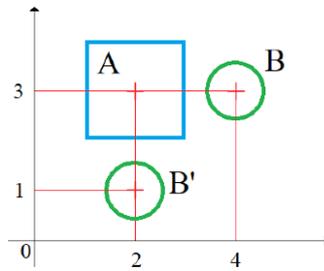


Figure 3 - Représentation des positions relative d'une entité B à une entité A

Comme représentée sur cette figure, l'entité A possède des coordonnées physiques (2,3), et l'entité B possède deux coordonnées relatives (2,0) et (0,-2). Au total, il y aura 2 interprétations possibles, qui ont l'entité A (2,3) en commun, puis B (2,0) ou B (0,-2) pour chacune d'entre elles.

Pour notre exemple principal, nous aurions donc :

1. Nom : « a lake », coordonnées : [ [0, -1] ], entité référente : « the park » ;
2. Nom : « A large island », coordonnées : [ [0, 0], [0, 1] ], entité référente : « the lake » ;

Pour l'entité « a lake » (1), l'idée ici est donc d'indiquer que cette entité se trouve au nord (-1 sur l'axe des ordonnées pour illustrer) du centre de son entité référente (ici, le parc). « A large island » (2), elle, se réfère au lac précédemment cité, et se situe soit relativement en son centre (ici, [0,0], qui pourrait signifier « les mêmes coordonnées que le centre de l'entité référente »), soit au sud (1 sur l'axe des ordonnées) du lac.

## 2. Extraction des interprétations

Une fois les positions relatives définies, il s'agit d'extraire toutes les interprétations possibles du texte initial.

Une liste de chaînages peut ressembler à ceci :  $A \rightarrow B \rightarrow C, D \rightarrow B, E \rightarrow B$ . Nous devons utiliser un algorithme qui permet d'additionner chaque paire de coordonnées d'une entité à toutes les paires de coordonnées de son entité référente, avec comme point d'arrêt de l'algorithme le fait de tomber sur une entité de type « parc » (on ne peut pas remonter au-dessus, car on considère que le parc est l'entité principale qui contient toutes les autres), ou alors tomber sur une entité qui a déjà été traitée (dans l'exemple de la liste de chaînages citée en début de paragraphe, pour la chaîne  $D \rightarrow B$ , on se contenterait de simplement renvoyer les coordonnées de B, qui ont déjà été traitées. En additionnant les coordonnées entre elles, on finit par donner des coordonnées physiques aux entités. Comme nous ne traitons pas une entité sur laquelle nous sommes déjà tombés, cette solution peut mener à des incohérences avec un grand nombre d'entités.

Pour chaque coordonnée ajoutée, on crée de nouvelles interprétations. Cela peut s'apparenter à un produit cartésien de toutes les coordonnées. Quand toutes les entités ont été traitées par notre algorithme, nous avons toutes les interprétations possibles. Notons que nous avons fait le choix de toutes les extraire. Nous pourrions mettre en place des règles supplémentaires à cette étape pour ne filtrer que les interprétations valides ou pertinentes, en excluant celles qui ont moins de sens, par exemple qui ont un lac par-dessus une fontaine ou un chemin en dehors du parc.

À cette étape de raisonnement, pour notre exemple principal, nous aurions une liste d'interprétations qui pourraient s'apparenter à la suivante (liste limitée ici à 2 interprétations possibles) :

1. Interprétation 1
  - a. « a lake », position décrivant le nord du parc « Tête d'Or » ;
  - b. « A large island », position décrivant le centre du lac, qui est en fait le nord du parc « Tête d'Or ».
2. Interprétation 2
  - a. « a lake », position décrivant le nord du parc « Tête d'Or » ;
  - b. « A large island », position décrivant le sud du lac, qui est en fait le nord du parc « Tête d'Or ».

### 3. Visualisation

Il est difficile de visualiser clairement un grand nombre d'interprétations. Diverses solutions s'offrent à nous. Par exemple, une interaction directe par l'utilisateur avec le résultat, qui permet de filtrer les interprétations à afficher. Dans le cadre de cette recherche, nous nous limitons à afficher toutes les interprétations possibles.

## V. Implémentation

Nous allons implémenter un prototype permettant de mettre en œuvre les différentes solutions proposées auparavant. L'idée est de concevoir une interface utilisable par un(e) historien(ne), qui indique clairement les ambiguïtés relevées par l'algorithme, et qui propose différentes représentations graphiques des interprétations. Nous avons réalisé une interface client Web (HTML/CSS) qui interagit avec un serveur Python tournant sous Flask. Il existe une puissante librairie Python, NLTK<sup>xii</sup> (*Natural Language ToolKit*), dont on se servira pour analyser le texte.

### A. Les grandes étapes de l'algorithme

Le traitement d'un texte se fait donc en trois grandes étapes, qui s'appuient sur la solution proposée précédemment :

1. La détection des entités dans le texte ;
2. La détection des relations spatiales entre les entités ;
3. Le placement des entités entre elles en fonction de leur relation spatiale.

Finalement, on crée une réponse au format JSON que l'on envoie en retour au client, où les résultats seront affichés.

### B. Métier

L'algorithme reçoit le texte « brut » du client web. Grâce à la bibliothèque NLTK, chaque mot de la phrase est étiqueté avec son type grammatical. Ensuite, on peut grouper les mots d'une phrase en fonction de ce qu'ils représentent (une entité, un verbe...). À chaque « groupe verbal » est associé une relation issue du modèle DE-9IM, qui permet ensuite de placer relativement deux entités entre elles. Finalement, il reste à régler les contraintes de placement et à associer des coordonnées finales physiques aux entités, qui peuvent ensuite être directement placées sur un canvas HTML avec un algorithme JavaScript.

#### 1. Un peu de grammaire

La bibliothèque NLTK possède une fonction qui permet de faire du *POS-Tagging*. Une fois que l'on a étiqueté les mots, on va pouvoir les regrouper. Par exemple, une composition d'un déterminant et d'un nom commun pourrait former une entité. NLTK permet de découper de la sorte une phrase avec une grammaire que l'on aura préalablement définie. La grammaire permettant de trouver des « ENTITY » dans un cas général peut ressembler à la règle décrite dans la Figure 4.

```
1 grammar = r"""
2     ENTITY:
3     {<DT|PP\$\$>?<JJ>*<NN.*>+}
4     VERB:
5     {<VB.*>+<TO|RP|IN>*}
6     SPREL:
7     {<ENTITY><VERB><ENTITY>}
8     """
```

Figure 4 - Des règles de grammaire simples qui permettent de trouver des entités

Nous définirons d'autres règles grammaticales que les 3 précédentes. En effet, il y a beaucoup de façons de formuler une même description. Nous avons un total de 12 règles dans notre grammaire.

#### 2. Analyse de l'arbre produit par NLTK

Nous avons développé un algorithme permettant d'analyser le texte étiqueté par le *POS-Tagging* et le *NP-Chunking*. L'algorithme se contente de lire chaque mot de l'arbre produit précédemment (sous forme de

texte), et d'appliquer des règles en fonction du mot. Si le mot est « SPREL », l'algorithme s'attend à lire le mot « ENTITY », suivi d'un groupe de mots définissant une entité, puis le mot « VERB », suivi d'un autre groupe de mots qui indique la relation, puis à nouveau « ENTITY » pour la dernière entité. Ainsi sont formées différentes relations. De notre algorithme sortira finalement un tableau contenant une ou plusieurs entités composées (« SPREL » est une entité composée de deux entités).

### 3. Forme d'une entité

Initialement, on plaçait uniquement le nom des entités les uns par rapport aux autres, sans se préoccuper de la forme, qui n'était pas indispensable au bon fonctionnement de l'algorithme de détection des entités / relations.

Cependant, le résultat ne ressemblait pas à grand-chose (cf. Figure 5).

L'hypothèse des images étant écartée comme décrit dans la partie IV.A, nous ne pouvions cependant pas dessiner manuellement chaque entité possible.

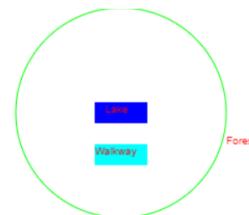


Figure 5 - Premier résultat, sans formes



```
entity.shape = [
  [105,0], [120,0], [140,10], [160,25],
  [160,60], [165,80], [195,135], [210,160],
  [215,190], [205,210], [185,230], [155,235],
  [145,240], [140,260], [125,270], [80,270],
  [15,240], [0,225], [5,185], [15,170],
  [30,150], [50,140], [60,125], [55,100],
  [45,75], [45,35], [65,25], [80,25],
  [88,24], [91,20]]
```

Figure 6 - Un lac et sa liste de points prédéfinis

Nous avons donc dessiné et enregistré localement la forme de quelques entités de base (un chemin, une allée, un lac, ou encore une fontaine), comme décrit sur la Figure 6. À l'étape de placement des entités, ces formes pourront éventuellement être ajustées. Les couleurs sont également personnalisables. On peut choisir la couleur principale (couleur de remplissage

de l'entité), et celle du contour.

### 4. Position des entités

Pour chacune des 5 relations DE-9IM ainsi que les positions de placement Nord/Sud/Est/Ouest, on définit des règles qui indiquent que par exemple, pour une relation « touches », on place B par rapport à A, et que B peut se situer soit dessus, soit en dessous, soit à droite ou à gauche de A.

### 5. Extraction des interprétations

Les positions relatives ne permettent pas directement un affichage des entités sur un canvas HTML. L'idée est de transformer les coordonnées relatives d'une entité avec les coordonnées relatives ou physiques de son entité référente. Au passage, cela permet de créer toutes les interprétations possibles du texte. Nous créons un algorithme récursif du type décrit en section 7. La Figure 7 décrit ces interprétations au formatage JSON.

```
1 "versions": [
2   [ {"name": " a lake", "pos": [100.0, 90.0], "shape": []},
3     {"name": " a fountain", "pos": [130.0, 175.0]}],
4   [ {"name": " a lake", "pos": [100.0, 90.0], "shape": []},
5     {"name": " a fountain", "pos": [180.0, 125.0]}],
6 ]
```

Figure 7 - Représentation des différentes interprétations dans la réponse JSON

## C. Client

Le client est une page Web HTML simple, qui permet d'entrer du texte, et d'afficher le résultat sur un canvas HTML, grâce à un script JavaScript. L'historien(ne) entre une description de son parc dans une zone de texte dédiée, il peut également pré remplir cette dernière avec des phrases déjà toutes faites, pour essayer rapidement l'application. Il envoie la description en cliquant sur un bouton. Une fonction permet alors de récupérer le texte de la textarea, puis de requêter le serveur Flask avec ce texte dans les données. La réponse est un fichier JSON décrivant des listes de listes d'entités (chaque liste étant en fait une interprétation possible du texte).

Une entité est constituée notamment d'une forme (une liste de points que l'on va relier entre eux), une couleur, une position et un nom. On peut facilement parser du JSON avec JavaScript, autant d'objets Entity que

nécessaires sont créés. Ensuite, on itère sur ces entités, et comme elles ont toutes le même « fonctionnement » (on relie les points entre eux, on colorie, on rajoute une étiquette avec le nom de l'entité à côté pour plus de compréhension), on les dessine de la même manière sur le canvas. La réponse est composée de plusieurs versions (plusieurs représentations possibles dues aux ambiguïtés de positionnement des entités). On va donc créer un canvas par interprétation. Entre-temps, on affiche sur une fenêtre intermédiaire la liste des entités relevées dans la description, la liste des relations, et la liste des ambiguïtés de relations des entités.

## D. Résultats

Les captures d'écran de l'application figurent en annexe.

Sur la Figure 9 est représentée l'interface globale de l'application. On aperçoit une liste de canvas sur la droite, car il y a 4 interprétations du texte entré (« There is the lake to the north of the park. A large path leads to the lake. An island emerges from the lake »). La Figure 10 représente la liste des entités / relations / ambiguïtés relevées pour cette même phrase.

## VI. Conclusion et perspectives

Nous avons proposé une solution permettant d'aider des historiens à représenter graphiquement un parc ou un jardin à partir d'une description textuelle. Bien que la détection d'entités nommées (NER) a fait l'objet de nombreuses recherches et est de plus en plus aboutie, en plus d'être à l'origine de nombreux algorithmes très efficaces, on trouve beaucoup moins d'articles scientifiques concernant la détection d'entités non nommées et leurs relations spatiales. Des modèles existent (comme DE-9IM) et nous ont permis de proposer une solution à la problématique de ce sujet. Nous avons pu proposer une solution mettant en œuvre des techniques du TALN, de la saisie du texte à l'obtention d'une liste précise des entités et des relations spatiales extraites. À la date de fin de rédaction de ce document de recherche, le département Recherche de Google a présenté un projet (Imagen)<sup>xiii</sup> qui répond plus ou moins à l'aspect principal de cette problématique (à savoir la génération d'une image par rapport à un texte descriptif), ce qui montre bien un intérêt dans la recherche du domaine de reconnaissance des entités non nommées. Ce projet nous a fait réaliser à quel point une interprétation pouvait varier d'une personne à une autre, et du nombre très élevé d'ambiguïtés à résoudre, que ce soit dans la détection d'entités ou de relations. Nos recherches pourraient être approfondies pour certains points, à commencer par une meilleure façon de détecter les références d'entités entre les pronoms au sein de plusieurs phrases (e.g. « Il », « Ce dernier », différence entre « Un lac » et « Ce lac », etc.). Il faudrait approfondir les recherches dans la détection de relations métriques. Concernant les formes, nous avons proposé d'en utiliser des polygonales par défaut, mais comment les détecter et en affecter aux entités dans tous les cas ? Une extension à cette question serait la question de la détection des chevauchements entre entités. Finalement, il reste la problématique de la représentation des données. Nous avons proposé d'afficher toutes les données, mais il vaudrait mieux mettre en évidence les résultats et les différentes interprétations.

La bibliothèque NLTK est très complète et régulièrement mise à jour, et n'est pas compliquée à mettre en œuvre. Notre version de démonstration fonctionne bien avec des descriptions simples et montre bien toutes les fonctionnalités implémentées. L'implémentation dans notre algorithme de ces différentes interprétations a représenté une bonne partie du travail. Ce projet est extensible et est prévu pour être beaucoup plus complet, et nous avons commencé avec la liste des synonymes pour décrire des relations. La limitation principale se trouve du côté de la grammaire : grâce à l'implémentation d'un algorithme d'apprentissage machine, nous ne serions plus limitées à quelques tournures de phrases (bien que nous ayons implémenté les principales), à moins d'en écrire des dizaines à la main dans le code. Il faudrait exploiter davantage le texte pour choisir la forme idéale d'une entité, soit en détectant mieux les adjectifs d'un nom, soit en récupérant la liste de points sur OpenStreetMap. Ce sera nécessaire à terme. Il faudrait également prendre en compte la place physique occupée par une entité (une « hitbox ») dans une optique de résolution des chevauchements entre les entités. Pour évaluer la pertinence de notre approche, il faudrait comparer un parc déjà existant, comme le parc de la Tête-d'Or, au résultat généré par notre proposition afin de peaufiner le code du prototype en conséquence.

## VII. Annexes

Tableau décrivant les 5 relations DE-9IM utilisées dans le cadre de nos recherches.

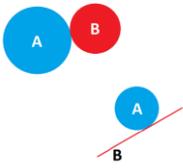
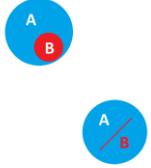
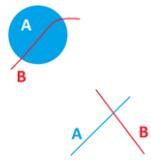
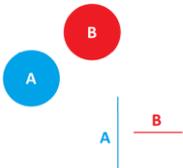
Nom de la relation	Schéma	Description
<i>touches</i>		« A touches B » ou « B touches A ». Les deux entités ont au moins un pixel extérieur en commun.
<i>CONTAINS</i>		« A CONTAINS B ». Une entité se trouve entièrement à l'intérieur d'une autre. Dans notre cas, « A coveredBy B » est également décrit ici.
<i>WITHIN</i>	Même schéma que le précédent	« B WITHIN A » dans le cas du schéma précédent. Dans notre cas, « B covers A » est également décrit ici.
<i>CROSSES</i>		« A CROSSES B » ou « B CROSSES A ». Si tous les pixels au niveau de A étaient supprimés, B serait coupé en 2 parties distinctes.
<i>DISJOINT</i>		« A DISJOINT B » ou « B DISJOINT A ». A et B n'ont aucun pixels en commun.

Figure 8 - 5 relations parmi les 10 du modèle DE-9IM

Sur la Figure 9, l'interface générale est représentée. Sur la droite, on aperçoit une liste de différentes interprétations.

## study of spatial relationships between entities

Robin Lemaître - Université Lyon 1

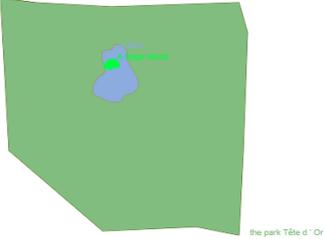
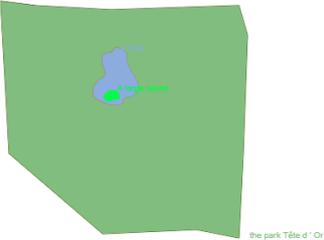
The text	Result (text)	Result (draw)
<p>Write a description here, or select a pre-defined text below.</p> <p>There <b>is a lake</b> to the <b>north</b> of the park Tête d'Or. A <b>large island</b> <b>emerges from</b> a lake.</p> <p><input type="button" value="Render!"/></p> <p><input type="button" value="A walkway leads to a lake, where there is an island."/></p> <p><input type="button" value="A tree-lined path leads to a fountain, which is surrounded by flowers."/></p> <p><input type="button" value="An island emerges from the lake."/></p> <p><input type="button" value="A path leads to the lake. It is surrounded by trees. An island is located in the lake."/></p> <p><input type="button" value="There are flowers to the north of the park."/></p> <p><input type="button" value="There is a lake and a fountain."/></p> <p><input type="button" value="a lake is surrounded by trees."/></p> <p><input type="button" value="There is the lake to the north of the park. A large path leads to the lake. an island emerges from the lake."/></p>	<p>The following entities / relations have been detected :</p> <p><b>Entities</b></p> <ul style="list-style-type: none"> <li>• a lake (id: 2)</li> <li>• the park Tête d'Or (id: 0)</li> <li>• A large island (id: 6)</li> </ul> <p><b>DE-9IM relations</b></p> <ul style="list-style-type: none"> <li>• &lt; a lake (id: 2), the north, the park Tête d'Or (id: 0)&gt;</li> <li>• &lt; A large island (id: 6), within, a lake (id: 2)&gt;</li> </ul> <p><b>Spatial relations ambiguities</b></p> <ul style="list-style-type: none"> <li>• within : A large island can be located in a lake from different location and size ( a lake can be partially or nearly-totally covered by A large island, or A large island can be located to the center or South of a lake).</li> </ul> <p><b>Different interpretations</b></p> <ul style="list-style-type: none"> <li>• Version 0</li> <li>• Version 1</li> </ul>	
		

Figure 9 - Interface globale de l'application

Sur la Figure 10, sont représentées les entités détectées, les relations détectées (représentation des tuples), et les détails des ambiguïtés relevées.

## Result (text)

The following entities / relations have been detected :

### Entities

- a lake (id: 2)
- the park Tête d'Or (id: 0)
- A large island (id: 6)

### DE-9IM relations

- < a lake (id: 2), the north, the park Tête d'Or (id: 0)>
- < A large island (id: 6), within, a lake (id: 2)>

### Spatial relations ambiguities

- within : A large island can be located in a lake from different location and size ( a lake can be partially or nearly-totally covered by A large island, or A large island can be located to the center or South of a lake).

### Different interpretations

- Version 0
- Version 1

Figure 10 - Résultats partiels sous forme textuel indiquant la liste des éléments relevés

## VIII. Références

---

- i Alan Turing, *Computing Machinery and Intelligence*, 1950
- ii Erwin Reifler, *The solution of MT linguistic problems through lexicography*, in *Proceedings of the National Symposium on Machine Translation*, 1960
- iii Apple Siri ([https://fr.wikipedia.org/wiki/Siri\\_\(logiciel\)](https://fr.wikipedia.org/wiki/Siri_(logiciel))), 2011
- iv Goyal, A., Gupta, V., & Kumar, M. (2018). *Recent named entity recognition and classification techniques: a systematic review*. *Computer Science Review*, 29, 21-43
- v Pouliquen, B., Kimler, M., Steinberger, R., Ignat, C., Oellinger, T., Blackler, K., ... & Best, C. (2006). *Geocoding multilingual texts: Recognition, disambiguation and visualisation*. arXiv preprint cs/0609065
- vi *Étiquetage grammatical des corpus de parole : problèmes et perspectives*. *Revue française de linguistique appliquée*, Vol. 4, No. 2. (1999), pp. 113-133, Valli A., Véronis J.
- vii NP-Chunking tests and references : <https://ifarm.nl/erikt/research/np-chunking.html>
- viii Christian Strobl, *Dimensionally Extended Nine-Intersection Model (DE-9IM)*, 2008
- ix « *Information Systems Urban Planning* », Robert Laurini, <https://www.taylorfrancis.com/books/mono/10.1201/9781315274713/information-systems-urban-planning-robert-laurini>
- x Algorithme Perceptron Tagger de NLTK : [https://www.nltk.org/\\_modules/nltk/tag/perceptron.html#PerceptronTagger](https://www.nltk.org/_modules/nltk/tag/perceptron.html#PerceptronTagger)
- xi WordNet, <https://wordnet.princeton.edu/>
- xii NLTK website : <https://www.nltk.org/>
- xiii Google Imagen website : <https://imagen.research.google/>