



UNIVERSITÉ
CAEN
NORMANDIE

Université de Caen Normandie
UFR des Sciences
Département Informatique

1^{ère} année de master

COÉCRITURE EN LANGAGE NATUREL

Rapport de projet

https://github.com/Exotikz/natural_language_rpg

Jérôme Legriffon (21307445)
Anne Wartelle (21700164)

Ce projet s'intègre dans l'initiative "la confrérie du code", initiée par Loïs Vanhée. Cette initiative vise à permettre aux apprenants de développer, en plus des compétences techniques classiques, la panoplie des compétences impliquées par la mise en place de projets en pratique (communication, organisation, collaboration, etc) et de garder une vision réflexive sur leur propre développement. Les rapports complémentaires et pédagogique décrivent nos actions complémentaires à l'effort technique et l'évolution de notre développement.

Tuteur : Loïs Vanhée

Année 2019 - 2020

Table des matières

1	Introduction	1
2	État de l'art	2
2.1	Le monde du jeu de rôle sur table	2
2.2	Les jeux vidéo à visée narrative	2
2.3	Le traitement du langage naturel appliqué au jeu de rôle	3
3	Outils existants	4
3.1	TensorFlow	4
3.2	Divers Natural Language Processing	4
3.3	Natural Language ToolKit	4
4	Outils utilisés	5
4.1	Python	5
4.2	NLTK	5
4.3	Git	5
4.4	PyMongo et MongoDB	5
5	Réalisation technique	6
5.1	Architecture du monde	6
5.2	Composition	6
5.3	Création du monde et du personnage	7
5.4	Traitement de la description du personnage	8
6	Difficultés	9
7	Perspectives	10
8	Conclusion	11

Remerciements

Nous tenons à remercier notre tuteur de projet, M. Loïs Vanhée, pour nous avoir formés et accompagnés tout au long de cette expérience avec beaucoup de patience et de pédagogie, ainsi qu'à Christopher Jacquot pour son template \LaTeX .

1 Introduction

Lors de l'année universitaire 2019-2020, nous avons eu comme objectif de mener un projet à bien sous le tutorat d'un professeur/doctorant d'Unicaen. Une pléthore de sujets ont été proposés, mais nous nous sommes focalisés sur "Coécriture en langage naturel", sujet émis par M. Vanhée, que nous avons opté de mettre en premier choix et qui nous a été attribué. Étant en parcours DOP, et ayant reçu comme conseil d'ouvrir nos choix aux autres spécialités afin d'élargir notre champ des possibles, nous avons choisi ce sujet, d'une part car il nous intéressait personnellement, mais également par curiosité.

Notre intérêt pour ce projet se porte dans un premier temps sur la partie "coécriture" que nous avons choisi de placer dans le contexte d'un jeu de rôle. Pouvoir écrire ensemble et créer un monde directement sans avoir à le penser, parfois pendant des mois en amont, est un plus non négligeable par rapport à l'approche classique du jeu de rôle.

Le traitement du langage naturel en informatique est un domaine reliant plusieurs catégories dont la linguistique et l'intelligence artificielle, et il est devenu prédominant dans notre monde actuel. Internet et l'informatique en générale s'étant démocratisés, l'utilisation des machines par des non experts est de plus en plus fréquente. C'est pourquoi il est intéressant de pouvoir communiquer avec une machine par un langage que l'utilisateur comprend, plutôt que celui de la machine. C'est dans cette optique que le traitement du langage naturel en informatique est né. Cependant, de nos jours, le traitement va beaucoup plus loin qu'une simple "traduction" des langages homme/machine : il cherche à faire assimiler aux machines un sens, à comprendre ce qu'on essaie de lui communiquer afin qu'elle puisse répondre "intelligemment" (ou non).

Ce projet s'articule dans cette voie. Nous avons exploré la possibilité d'écrire des histoires dans lesquelles on incarne un personnage, et pour lesquelles plusieurs utilisateurs créent un monde commun, en coopération avec un programme. Seulement, en ce qui concerne les jeux de rôle, le format le plus répandu jusqu'à présent est celui d'un monde pensé en amont dans lequel on joue ensuite, ou alors un monde qui propose des choix limités dans sa construction. Ici, ce que nous cherchons à obtenir, c'est la possibilité de créer un monde directement, de façon intuitive, avec comme seul moyen d'interaction le langage.

2 État de l'art

En recherchant des projets existants en rapport avec celui qui nous avait été assigné, nous avons suivi plusieurs pistes qui sont complémentaires.

2.1 Le monde du jeu de rôle sur table

Premièrement, nous nous sommes penchés sur ce qui existait dans le domaine des jeux de rôle papier. Ils sont généralement beaucoup plus centrés sur la narration que les jeux vidéo de rôle, il était donc logique que nous nous y intéressions en priorité. Nous nous sommes familiarisés avec les jeux de rôle narrativistes, qui mettent en général de côté les systèmes de calcul complexes (conçus en particulier pour simuler des combats) pour laisser un maximum de liberté d'écriture aux joueurs. Ce sont eux qui mènent le jeu, pas le scénariste de la campagne.

Une mécanique utilisée dans certains jeux de rôles nous a semblé pertinente dans le cadre de notre projet. Elle consiste à générer aléatoirement des relations entre personnages joueurs, en lançant des dés : par exemple, il est possible de tirer la relation : « Vous avez entendu des rumeurs à propos du personnage X, et vous ne lui faites pas confiance. Quelles sont ces rumeurs ? » Ce système est notamment utilisé dans les jeux *Monster of the Week* et *Dungeon World*, comme le fait remarquer le développeur Christoffer Petersson dans un article de blog à ce sujet ¹.

Ces jeux sont liés à notre projet, en cela qu'ils permettent aux joueurs d'écrire ensemble leur propre histoire. En effet, dans la plupart des jeux de rôle, le scénario de la campagne prend une place prépondérante et les relations entre les personnages joueurs ont rarement une importance cruciale. Le défi, pour nous, était de transposer les mécaniques de ces jeux de rôle narrativistes, qui sont entièrement mises en œuvre par des êtres humains—capables de comprendre le langage tel qu'on le parle—dans un contexte informatisé. Dans notre cas, le maître de jeu est un programme : toute la question, c'est comment le faire réagir comme le ferait un véritable MJ (maître de jeu) expérimenté.

2.2 Les jeux vidéo à visée narrative

Nous avons également cherché des jeux vidéo qui visaient à simuler un monde complexe peuplé de personnages. Ces jeux systémiques sont souvent conçus pour

1. <https://blog.christoffer.online/2013-07-29-defining-relationships-between-rpg-tabletop-characters/>

un seul joueur : ils permettent bien une coécriture, mais uniquement avec une machine. C'est par exemple le cas de *Dwarf Fortress*, ainsi que de tous les jeux qui en sont inspirés, comme *Rimworld*. Ce dernier mérite une mention pour son système d'*AI narrators* : ces narrateurs virtuels, paramétrables par le joueur, gèrent le flux des évènements du monde. Ce flux peut par exemple être aléatoire, ou proposer un défi de difficulté croissante. On voit bien ici que dans un jeu narratif, il faut essayer autant que possible de garder le joueur engagé en lui proposant des évènements qui ne sont pas anodins. Le jeu *Caves of Qud*, disponible pour le moment en accès anticipé, cherche lui aussi à générer des histoires émergentes pour les joueurs.

Étant donné que notre projet est multi-utilisateurs, on peut s'inspirer de ces jeux existants, mais pas copier leur système de fonctionnement tel quel.

2.3 Le traitement du langage naturel appliqué au jeu de rôle

En décembre 2019, le jeu *AI Dungeon 2* a connu une grande popularité. Il est notable, car il utilise le modèle GPT-2 pour permettre aux joueurs de s'adresser au maître de jeu virtuel comme il le ferait avec un être humain. C'est un projet qui repose entièrement sur l'apprentissage artificiel. Les réponses données par le programme sont souvent impressionnantes de réalisme, mais le contexte de l'histoire ne peut pas être conservé en mémoire dans sa totalité, ce qui entraîne souvent des erreurs de cohérence d'une réponse à l'autre.

Au moment de sa sortie, nous avons déjà effectué notre étude théorique, il aurait donc été difficile de nous en inspirer. De plus, nous souhaitons que le monde de notre jeu conserve une certaine cohérence interne, ce que ne permet pas le système proposé par *AI Dungeon*. C'est également un jeu solo, donc l'aspect multijoueurs de notre projet demandait une approche différente (toutefois, une version multijoueurs d'*AI Dungeon* doit sortir prochainement).

3 Outils existants

3.1 TensorFlow

Pour ce qui est du traitement de langage, il existe de nombreux travaux dans ce domaine. Tout d'abord, il y a *TensorFlow* de Google, dont une partie est dédiée au *Natural Language Processing*. Le travail se fait majoritairement par apprentissage artificiel de textes, documents, etc., et sa grande bibliothèque rend le traitement du langage avec *TensorFlow* très complet, allant de la simple traduction de langage naturel en langage machine jusqu'à la détection automatique de voix et la compréhension d'un message entendu.

3.2 Divers Natural Language Processing

Dans un second temps et pour ne citer que ceux-ci, il y a *Applied Text Analysis with Python* de Benjamin Bengfort, Rebecca Bilbro Tony Ojeda, *Natural Language Processing with PyTorch* de Delip Rao Brian McMahan. Ces travaux ont la particularité de se focaliser entièrement sur le traitement du langage naturel par *machine learning* ou *deep learning*.

3.3 Natural Language ToolKit

Enfin, pour conclure cette liste non exhaustive, il y a *Natural Language Toolkit* de Steven Bird Edward Loper. Destiné exclusivement à Python, il s'agit d'une librairie afin de travailler sur le traitement du langage naturel dans ce même langage.

4 Outils utilisés

4.1 Python

Au début, lors de nos premiers pas, nous avons abordé la question du langage de programmation. En examinant nos besoins, nous sommes tombés d'accord sur Python. Sa grande popularité en fait un excellent choix, que ce soit en terme de choix de bibliothèques disponibles ou bien en terme d'aide accessible en ligne.

4.2 NLTK

Lors de la réflexion qui s'était imposée pour savoir ce avec quoi nous travaillerons pour traiter le langage, nous nous sommes tournés vers *TensorFlow*. Seulement, après avoir été confrontés à cette librairie sur Python, nous nous sommes rendus compte que ce ne serait pas le plus adapté pour faire ce que nous voulions. Après une recherche, nous avons trouvé la bibliothèque python NLTK (Natural Language ToolKit). Elle nous permet de travailler plus à même sur le texte en langage naturel, tout en étant très accessible d'utilisation.

4.3 Git

Git nous sert pour avoir la possibilité de partager notre projet en ligne sans que chacun ait à l'avoir ailleurs. C'est également pratique pour centraliser les changements.

4.4 PyMongo et MongoDB

Pour stocker toutes les informations nécessaire au monde, nous utilisons MongoDB conjointement avec la librairie python PyMongo pour faire le pont entre python et MongoDB.

5 Réalisation technique

5.1 Architecture du monde

Nous avons représenté le monde comme un ensemble d'entités multiples (personnages, lieux, évènements...) reliées entre elles par des relations.

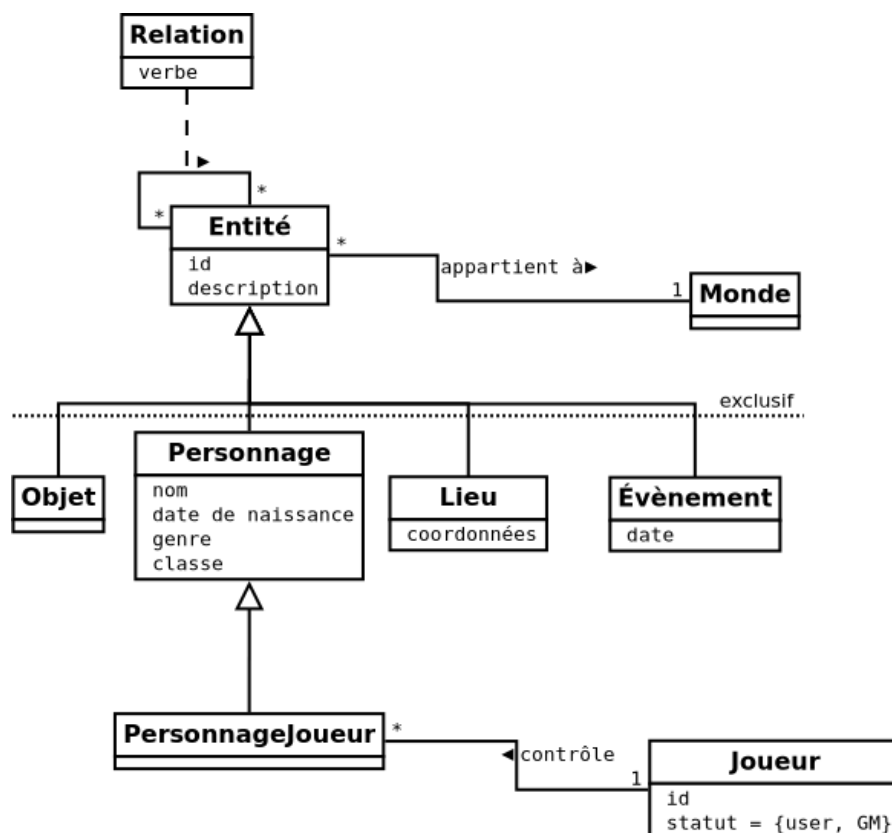


FIGURE 5.1 – Un diagramme UML de l'architecture du monde

5.2 Composition

Nous nous sommes mis d'accord sur une conception du monde par création de personnage. Tout ce qui concerne le monde est ajouté en tant que donnée de celui-ci. Ainsi, les lieux de naissance des personnages deviennent des lieux de ce monde, les races également.

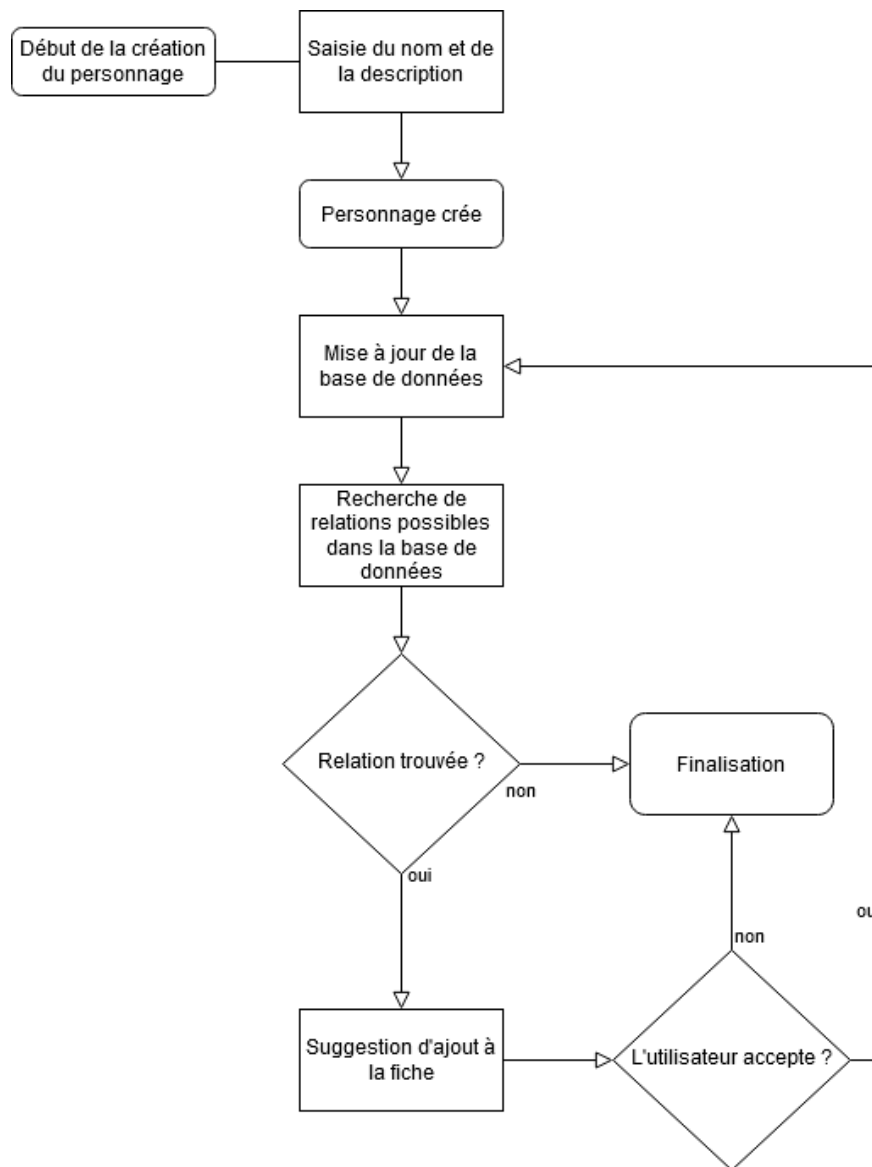


FIGURE 5.2 – Un diagramme décomposant la création d'un personnage

5.3 Création du monde et du personnage

La création du monde est toute simple, il suffit de lui donner un nom et une date actuelle.

La création de personnage quant à elle, nécessite une chose très importante en plus du nom : la description. En effet, c'est avec cette description que le programme tente de trouver toutes les informations nécessaires à remplir, comme le sexe, l'âge ou encore le lieu de naissance.

```
Bienvenue sur ALORS !
Que souhaitez-vous faire ?
  (1) Créer un monde
  (2) Choisir un monde existant
  (3) Quitter
?
Voici l'id (il s'agit de NomDate) des mondes existants :
  Melendra1574
Entrez l'id du monde pour le sélectionner, ou (q) pour retourner en arrière :
Melendra1574
Que souhaitez-vous faire ?
  (1) Créer un personnage
  (2) Voir les personnages
  (3) Quitter
1
Veuillez entrer le nom de votre personnage :
Grom
Saisissez maintenant une description de votre personnage :
Grom est un guerrier fort. Il connaît bien les montagnes du nord car il vient de Haut-Brokk. Du haut de ses 29 ans, il arpente Melendra avec son fidèle compagnon Driss. Il est capable d'endurer le froid pendant tout un hiver. Il est également un combattant hors pair, maîtrisant la hache à deux mains comme personne.
Il semblerait que votre personnage soit de sexe masculin
Est-ce correcte ? : oui (y) ou non (n) : y
Il semblerait que votre personnage soit guerrier
Est-ce correcte ? oui (y) ou non (n) : y
Il semblerait que votre personnage soit âgé(e) de : 29
Est-ce correcte ? oui (y) ou non (n) : y
Il semblerait que votre personnage vient de Haut-Brokk
Est-ce correcte ? oui (y) ou non (n) : y
```

FIGURE 5.3 – Aperçu de la création de personnage dans le terminal

Ces données sont finalement stockées dans une base de donnée MongoDB.

5.4 Traitement de la description du personnage

Afin de déterminer les informations du personnage, le programme se prend de plusieurs manières différentes :

- Par fréquence des mots, par exemple pour déterminer la classe ou le sexe du personnage. Suivant un ensemble de mots donnés en lien avec une classe(hache, fort, combattant, force, etc. pour guerrier), si le nombre d'occurrences de ces mots est grand, on estime que la chance est également grande que ce soit la classe du personnage.
- Par recherche de "sens", comme par exemple pour deviner l'âge ou le lieu de naissance, on cherche une phrase où il y aurait le mot (ou ses dérivés au pluriel, etc) "an", "année", "anniversaire", etc et on décortique le nombre raisonnable (ici de 0 à 150) associé.

Lorsque le lieu de naissance du personnage à été trouvé et approuvé, celui-ci est ajouté au monde s'il ne s'y trouve pas déjà. De même pour la race du personnage.

Si votre personnage vient du même endroit qu'un autre personnage, alors le programme vous montre le nom de ceux-ci et vous demande s'il y a une personne que votre personnage pourrait connaître. Si oui, alors un lien est crée, ici en l'occurrence, un lien de connaissance "à grandi au même endroit".

6 Difficultés

Le premier heurte à été d'envisager de faire travailler une IA sur le traitement langage, impossible sans une base de donnée solide au départ. Nous avons alors opté pour une bibliothèque python, NLTK, qui nous permet de travailler sur le texte en tant que tel plutôt que de laisser un agent le faire. Certes c'est en «dur» mais cela permet déjà une ébauche, avec une possibilité d'amélioration plus tard.

Pris de court par le temps, il s'avère extrêmement difficile de partir de rien et de faire un projet qui doit être fonctionnel. C'est pourquoi nous avons abandonné plusieurs aspects que nous avions considérés au départ, comme une interface utilisateur, afin de nous concentrer sur l'essentiel. Se rendre compte de l'ampleur de la tâche a été très difficile dans un premier temps, mais avec l'aide de notre tuteur, nous avons réévalué le travail à faire et essayé de définir un cadre précis pour obtenir un prototype en temps et en heure.

Lors de la détermination des informations du personnage, si la description parle beaucoup plus de situations mettant en scène le personnage (et donc finalement si la description du personnage est moins détaillée que la description de ce qui existe autour de lui), alors il devient très difficile de différencier avec précision ce qui décrit le personnage de ce qui décrit son environnement.

Dans un souci de moyen, la base de données est faite en locale afin de faire fonctionner le projet. Cependant, une base de donnée en ligne devrait être créée afin de pouvoir réellement utiliser le programme à plusieurs en ligne et non seulement en local.

7 Perspectives

Avec plus de temps, nous aurions pu entreprendre une interface graphique afin de rendre l'expérience utilisateur plus agréable et intuitive. Pouvoir voir le personnage que l'on crée et le monde dans lequel il évolue serait une amélioration majeure du projet.

De même, cela nous aurait également permis d'utiliser plus de contenu de création dynamique. Par exemple, les liens créés actuellement ne sont que des liens de même lieu de naissance. Nous aurions pu ajouter bien plus de lien, comme une possible rencontre dans le monde. Ou bien des liens causés par les actions passées d'un personnage qui se répercutent sur un autre, *etc.*

D'autant que nous aurions pu ajouter plus de choix pour remplir la fiche de personnage, en proposant ce que les autres joueurs ont pu remplir avant. De ce fait, créer une cohérence de plus en plus forte au fur et à mesure que le monde évolue et se complexifie.

Et enfin, nous aurions pu intégrer notre projet à un réseau social existant.

Si le système était réellement déployé, il faudrait se poser la question de la liberté du maître de jeu virtuel. Faudrait-il laisser l'intelligence artificielle tenir les rênes du monde généré, ou alors permettre à certains utilisateurs de passer en revue les décisions qu'elle a prises ? Nommer des administrateurs a des avantages : beaucoup de gens ne sont pas prêts à laisser un algorithme décider entièrement du déroulement de leur jeu de rôle. Mais un inconvénient est que valider tous les nouveaux événements générés deviendrait vite rébarbatif.

Avec une base de données spécialisée dans le jeu de rôle, nous aurions pu utiliser des techniques d'IA pour améliorer l'interaction homme-agent.

Par ailleurs, il serait intéressant de représenter les relations entre les différents éléments du monde sous la forme d'une ontologie. Intuitivement, nous avons d'abord imaginé un système qui y ressemblait ; mais n'ayant pas encore, à ce moment-là, vu en quoi consistait ce concept, nous n'avons pas pu l'utiliser directement.

On pourrait aussi intégrer la notion de conflit à notre modèle de relations : prendre en compte la qualité positive ou négative d'une relation entre deux personnages, et décider que pour qu'une histoire reste intéressante, il faut générer, par exemple, 20% de relations conflictuelles.

8 Conclusion

Notre objectif était de créer une application qui, à partir des textes écrits par différents utilisateurs, crée et maintient un monde et des personnages dans le cadre d'un jeu de rôle. Il fallait également que des relations se créent organiquement entre les personnages existants.

Dans ce but, nous avons utilisé une bibliothèque dédiée au traitement automatique des langues (NLTK) pour extraire du sens des descriptions entrées par les joueurs. Nous avons considéré le contenu de ces descriptions comme des indices d'éléments à enregistrer dans la base de données du monde, et nous avons utilisé du *speech-tagging* pour préciser le sens de ce que nous y avons trouvé. Les relations créées sont, à l'heure actuelle, sommaires et codées "en dur".

Nous avons pu vérifier que le programme avait des résultats satisfaisants pour des cas simples.

Nous aurons appris énormément durant ce projet, et pas seulement sur les points habituellement abordés en informatique. En effet, nous pensions (et avons l'habitude) qu'un projet tel que celui-ci soit un travail de programmation ou de développement dans le but de résoudre un problème, suivi d'un rapport décrivant notre approche de résolution, etc. Or, c'est le cas, mais pas uniquement. Ce projet nous a été formateur dans bien des domaines : d'abord en communication, grâce à des réunions limitées dans lesquelles les bonnes questions devaient être posées—nous avons également eu la possibilité d'organiser des réunions— mais également en terme d'organisation, car finalement, la part de « code » n'a pas été si importante. Il y a eu beaucoup de réflexion autour du sujet, de la recherche sur des exemples similaires... Puis en tant que responsables de projet, nous avons eu des choix à faire, après avoir réalisé que certains objectifs ne pourraient être atteints. Nous avons dû nous focaliser sur certains aspects.

En somme, ce fut une excellente expérience formatrice, sur toute sa durée.

Bibliographie

- [1] Bird, Steven, Edward Loper and Ewan Klein (2009), Natural Language Processing with Python. O'Reilly Media Inc. <https://www.nltk.org/>