



UNIVERSITÉ
DE LORRAINE



Institut des
sciences du Digital
Management & Cognition

QUI A TUÉ QUI?
QUAND ET OÙ?

DETECTIVE BOT 2020



SOMMAIRE

I. INTRODUCTION

II. PRÉSENTATION PROJET

Les différentes étapes

Les techniques génériques de TAL utilisées

Explications des choix

Points importants

Difficultés

Listes des fonctionnalités

III. TRAVAIL ET ÉQUIPE

Répartition du travail

IV. CONCLUSION

Commentaires individuels

Présentation de l'outil final

Manuel d'instructions

I. INTRODUCTION

MEMBRES DE L'ÉQUIPE

Elise PERROT

Lucille DUMONT

Nicolas CARBONNIER

Antoine BENTINI

Tom WYSOCKI

SUJET

Le but du projet était de programmer un détective qui parcourt Wikipédia afin de déterminer un rapport, une liste constituée des noms de 50 assassins ainsi que leurs victimes, le lieu et la date des meurtres. Les informations doivent être automatiquement extraites de Wikipédia et les noms des meurtriers doivent commencer par la même lettre. Notre groupe a choisi la lettre M.

II. PRÉSENTATION PROJET

Les différentes étapes

Le programme, dénommé Detective Bot 2020, est déployé au moyen du langage Python sur l'éditeur Spyder. Nous avons géré la gestion des versions du code via des commits sur GitHub Desktop.

Dans un premier temps, nous nous sommes réunis afin de déterminer les tâches à accomplir pour le développement du projet. Nous avons mis en place un Trello afin de décomposer les tâches en sous-tâches exécutables indépendamment l'une de l'autre et pour mieux gérer la répartition du travail.

Nous avons tout d'abord choisi un portail thématique afin de récolter le maximum d'information en une seule page. Nous avons élaboré un corpus de plusieurs fichiers XML afin de resserrer davantage notre cercle de recherche de tueurs en "M".

Ensuite nous nous sommes intéressés aux fonctions que devait accomplir notre détective comme l'extraction de données et le traitement de celles-ci. Nous nous sommes ensuite répartis les tâches de programmation et de rédaction afin que chacun puisse commencer à travailler à l'issue de la première réunion.

Au fur-et-à mesure du développement, nous avons rencontré des difficultés, pas forcément attendue, ce qui nous a poussé à poser des temps de réflexion avec toute l'équipe pour ne pas rester coincés.

II. PRÉSENTATION PROJET

LES DIFFÉRENTES ÉTAPES

On a également choisi le thème des serial killer pour restreindre nos recherches et avons donc construit un dossier XML ("Archives_SK") contenant les pages de nombreux serial killer dont le nom de famille commence par "M". L'avantage de ce corpus est que chaque tueur en série possède dans sa balise <text> une section "Infobox" qui rassemble les infos nécessaires à l'enquête: meurtrier + nombres de victimes + temporalité des meurtres + lieu.

Malgré nos efforts, nous n'avons pas pu obtenir des résultats satisfaisants et avons donc choisi de changer de corpus.

LES TECHNIQUES GÉNÉRIQUES DE TAL UTILISÉES

Nous avons utilisé la bibliothèque NLTK afin de procéder à des fonctions de traitements automatiques des langues comme le POS-Tagging

Nous avons également utilisé le tag NNP et NNS pour identifier les tueurs, les victimes, les dates et les lieux car cela semble rapide pour identifier toutes les informations que nous voulions récupérer.

II. PRÉSENTATION PROJET

Explications des choix

POINTS IMPORTANTS

Nous avons pris le parti d'enquêter sur des tueurs dont le prénom OU le nom commencent par la lettre M, cela nous permettant d'avoir plus de données facilement.

Nous avons également choisi, comme décrit dans les techniques de Tal utilisées, de privilégier NNP et NNS car cela facilite le code source.

DIFFICULTÉS

Nous avons choisi de générer une page xml manuellement réunissant les 50 tueurs (appelée Archive_SK) au lieu de faire directement chercher l'enquêteur dans la page des serial killers récupérés par pays pour faciliter la recherche et la manipulation de données dans le fichier.

Être généraliste: La variabilité des fichiers XML; même si ce genre de fichier possède sa structure unique et reconnaissable, des erreurs se glissent parfois dans le code rendant la structure erronée. La recherche d'informations par patterns devient alors compliquée.

II. PRÉSENTATION PROJET

Explications des choix

DIFFICULTÉS

L'angle d'appréhension de données textuelles: Nous avons été tentés de n'utiliser pratiquement que des expressions régulières pour la recherche et l'extraction de données textuelles particulières dans une chaîne de caractères. Cependant, nous nous sommes rendus compte qu'il fallait gérer beaucoup d'expressions régulières contraintes par des mises en forme variables, cela étant difficile à manipuler.

Ne pas trouver les données pertinentes là où on les attendait, et devoir les chercher ailleurs. Cela nécessite une modification des fonctions de recherches et d'extraction et c'est parfois coûteux.

La jonction des informations relatives à un même tueur en série et distinguer les données de chaque tueurs en série trouvé.

II. PRÉSENTATION PROJET

Listes des fonctionnalités

Detective Bot 2020 vous propose d'ouvrir un fichier XML en local. Ensuite, il effectue divers traitements sur le fichier en commençant par lire ce fichier XML. Il y a beaucoup de données et peu d'entre elles sont pertinentes !

Detective Bot est capable de trier les données contenues dans le fichier XML. Pour ce faire, Detective Bot extrait le contenu des balises <text></text> du fichier XML et le nettoie de ses données inutilisables.

Le texte étant extrait, il peut alors être tagué (fonction disponible de la librairie NLTK) afin de trouver les patterns correspondant aux informations dont Detective Bot a besoin pour mener son enquête.

Une fois que notre bot est en possession du nom du tueur, ceux des victimes, du lieu et de la date des meurtres, il peut alors rédiger un rapport, contenant la liste de 50 tueurs et des informations relatives, qu'il affiche dans la console.

```
"""Fonction pour afficher les résultats"""
def show(killerListWithM, allInformation):
    name = ""
    k = -1
    details(allInformation[k])
    for listK in killerListWithM:
        for i in listK:
            name += i + " "
        print(name + ":")
        details(allInformation[k])
        name = ""
        k = k + 1
```


III. TRAVAIL ET ÉQUIPE

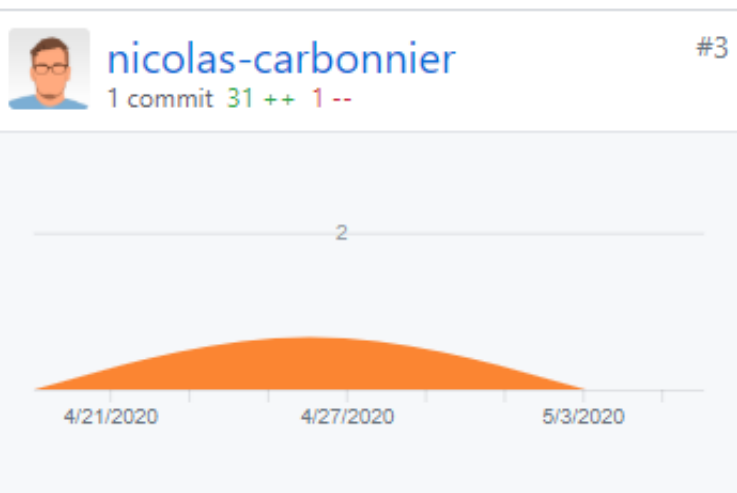
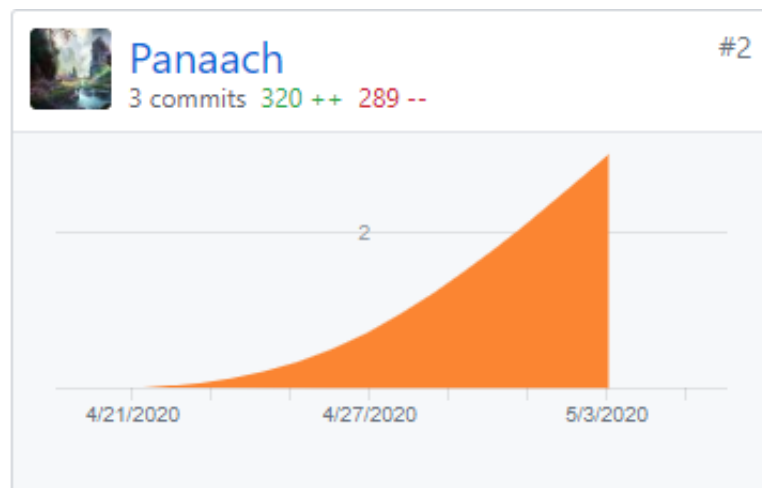
Nous avons pu collaborer tout au long du projet via GitHub Desktop et nos échanges s'effectuent sur Discord. Nous avons également effectué la mise en place d'un Trello dès le début qui nous a permis de bien déterminer les tâches à accomplir. Cela a été d'une grande aide pour l'organisation de l'équipe tout au long du projet ainsi que pour l'avant projet et les différentes parties à inclure dans le rapport.

RÉPARTITION DU TRAVAIL

La répartition initiale du travail a été modifiée au cours du développement pour pallier les difficultés rencontrées par chaque membre de l'équipe afin d'avancer sans perte de temps ce qui nous a permis d'être plus efficace dans l'exécution des tâches à réaliser.

Tout le monde a pu au final participer à toutes les tâches, que ce soit de la programmation ou de la rédaction dans une certaine mesure, selon le niveau de chacun.

III. TRAVAIL ET ÉQUIPE



Tout le monde a participé à l'élaboration du fichier DetectiveBot_2020 ainsi qu'aux pages xml. Lucille a initialisé le projet et Nicolas a rédigé les premières lignes. Tom a créé le corpus et les pages xml tandis que Antoine s'est concentré sur le code (notamment pour afficher la description des tueurs). Elise n'a pas fait de commit dû à un soucis d'outil mais a aidé dans la réalisation du projet en proposant des solutions et dans le rapport.

IV. CONCLUSION

Commentaires individuels

Elise - *"J'ai beaucoup de difficultés en ce qui concerne le code et la programmation. Cependant les membres de l'équipe ont su répondre à mes questions et m'épauler dans ce que je ne comprenais pas, ce qui m'a permis de m'améliorer et d'aider à la progression du projet avec mes propres capacités et plus dans d'autres domaines comme le rapport. Nous avons pu mener ce projet à bien ; la charge des tâches à accomplir a su mettre en avant les points forts et les points faibles de chacun, permettant ainsi une répartition parfois inégale (répartition entre code et rapport pour chaque membre de l'équipe) mais juste du travail. Le terme "travail d'équipe" prend alors tout son sens."*

"Ce projet s'est avéré plus difficile qu'il n'y paraissait et encore plus dans les derniers sprints. Néanmoins, il m'a permis de développer mes compétences en Python et plus globalement en TAL et en programmation. Nous avons su mener à bien ce projet, par l'entraide et l'écoute. Ce n'est pas la première fois que nous collaborons ensemble, voilà pourquoi nous avons des facilités à se joindre et à communiquer. Aussi, le caractère exceptionnel de la situation actuelle a eu de fortes répercussions sur le projet en général. Nous avons fait preuve, plus que jamais, d'adaptabilité, de vivacité et d'esprit d'équipe."

- **Lucille**

IV. CONCLUSION

Commentaires individuels

Tom - *"J'étais heureux de pouvoir enfin participer à un projet en informatique orienté sur le TAL. Souhaitant évoluer professionnellement dans cette voie j'ai vu dans ce projet une opportunité de commencer à acquérir plusieurs compétences, notamment en code. Les événements d'ampleur internationale ont empêchés le déroulement des cours et ne nous ont pas permis d'apprendre à débiter l'apprentissage des outils nécessaires au projet (python, nltk etc.). Néanmoins, comme dans les autres matières, nous avons su mettre en place avec le groupe des stratégies pour mener à bien le projet ; et pour ma part j'ai pris l'habitude de planifier les tâches dans une gestion de projet. Malheureusement, j'ai rencontré des problèmes de santé au cours de cette période et cela a contrarié mon implication dans l'avancement des tâches. De ce fait lorsque fin avril j'ai pu reprendre l'ensemble des cours, j'ai dû combiner entre le retard pris et les difficultés auxquelles nous avons fait face dans le projet. J'ai pris alors toutes mes journées jusqu'à la date du rendu pour faire du mieux que j'ai pu pour comprendre et élaborer des fonctions pour traiter les données. Ce ne fut pas la meilleure expérience à vivre au vu des nombreuses contraintes, mais avec du temps et du recul je le vois comme une chance d'apprendre et ce au-delà des connaissances en la matière. Malgré tout cela permet de se faire la main avec les outils, à travailler en autonomie, innover rechercher et s'adapter pour faire face à tout obstacles quels qu'ils soient."*

IV. CONCLUSION

Commentaires individuels

Nicolas - *"A l'aide de tous les membres et malgré le peu d'expérience en python on a su se débrouiller et collaborer pour obtenir des résultats."*

"Le projet m'a permis d'apprendre de nouvelles compétences en python. L'esprit d'équipe a su se démarquer pour amener ce projet jusqu'au bout." - **Antoine**

IV. CONCLUSION





Présentation de l'outil final

MANUEL D'INSTRUCTIONS

L'utilisateur lance le programme

- L'utilisateur doit ouvrir le fichier **Corpus.xml**
- Après ouverture du fichier, le détective nettoie les informations du fichier pour ne conserver que celles utiles à l'enquête.
- Il enquête si vite, qu'il trouve instantanément le nom de 50 tueurs.
- Son enquête est visible dans la console.

FICHIERS

 Archive_SK.xml	Create Archive_SK.xml	22 hours ago
 Corpus.xml	clean text	9 days ago
 DetectiveBot_2020.py	tueurs et description des tueurs	10 hours ago
 README.md	Initial commit	11 days ago

Fichier contenant le code source:
DetectiveBot_2020.py

ELISE PERROT

LUCILLE DUMONT

NICOLAS CARBONNIER

ANTOINE BENTINI

TOM WYSOCKI