

Rapport Ingénierie Linguistique - Lettre D

BOUMOKONIA Olivia BROGGI Thomas
CHOLEZ Maud DA COSTA Sophie

24 avril 2017

Table des matières

1	Introduction	2
2	Notre corpus	4
3	Déroulement de la réalisation	5
3.1	Planning	5
3.2	Retour sur les tâches réalisées	6
3.3	Github	7
4	Nos champs lexicaux	9
5	Techniques utilisées et choix d'implémentation	12
6	Problèmes lors de la réalisation	14
7	Rétrospective et conclusion	16
A	Activité Github	17

Chapitre 1

Introduction

Notre équipe pour ce projet est composée de quatre membres : BOUMO-KONIA Olivia, BROGGI Thomas, CHOLEZ Maud et DA COSTA Sophie. Nous sommes tous les quatres étudiants issus de la licence en mathématiques informatique appliquées aux sciences humaines et sociales, parcours Sciences Cognitives. Ce projet a été conçu dans le cadre d'un projet en ingénierie linguistique enseigné par M.Amblard.

Le but de ce projet étant de créer un enquêteur qui doit traverser des pages Wikipédia afin de déterminer un meurtrier, une victime, quand et où le meurtre de la victime a eu lieu. Une des conditions de ce projet était que les noms de ces meurtriers doivent tous commencer par la lettre "D" Pour cela, nous devons donc identifier automatiquement, à l'aide des outils donnés dans le sujet du projet, des noms propres, des verbes de meurtre, des noms communs relatifs à la mort, etc... Nous avons ainsi déterminé des champs lexicaux en rapport avec notre sujet.

Nous avons alors fait le choix d'un corpus afin de limiter nos recherches, tout en ayant une quantité suffisante de pages et d'informations disponibles. En effet, nous avons choisi de traiter notre programme sur le thème des zombies dans les films. Nous nous sommes dit que ce corpus nous permettrait d'obtenir un certain nombre de meurtriers, car rappelons-le, notre enquêteur doit pouvoir repérer entre 50 et 100 meurtriers. De plus, notre corpus doit être en anglais, ce qui engendre le fait que les outils qui vont constituer notre enquêteur doivent être adaptés à cette langue.

Auparavant nous avons acquis quelques notions en traitement automatique des langues, notamment grâce au cours de TAL suivi en L2 MIASHS, enseigné par M.COUCEIRO. Certaines commandes utilisées pour le projet nous ont été fournies durant les TD du cours d'ingénierie linguis-

tique, dispensé par M.AMBLARD. Le reste des commandes - notamment les commandes Python - nous ont été mises à disposition grâce à nos recherches sur diverses sources.

Chapitre 2

Notre corpus

Comme nous avons pu le dire dans l'introduction, nous avons choisi le corpus qui a pour sujet « les zombies dans les films ». Cependant, ce choix n'était pas notre première suggestion de corpus. En effet, nous avons tout d'abord pensé aux criminels, aux séries criminelles, et à d'autres exemples de ce type.

Néanmoins, nous avons estimé que nos premières idées étaient trop « banales » et trop « communes ». Nous avons alors décidé d'avoir un corpus original. C'est pour cela que nous avons voté à l'unanimité pour un corpus sur les zombies et plus particulièrement les zombies dans les films afin que le corpus soit moins conséquent. De plus, c'était un choix unanime au sein du groupe car tous les membres de l'équipe ont aimé ce thème. Le thème des zombies est assez conséquent car ils sont traditionnellement dans les films en tant que tueurs ; on peut par exemple citer « The Walking Dead », comme une série où les zombies ont le rôle principal avec des morts de caractères principaux ou pas assez régulières (no spoil).

Pour pouvoir extraire le corpus, nous avons suivi la partie « constitution d'un corpus » donnée dans le sujet du corpus, en suivant le lien suivant :

<https://fr.wikipedia.org/wiki/Sp/unhbox/voidb@x/bgrouplet/unhbox/voidb@x/setbox/@tempboxa/hbox{e/global/mathchardef/accent@spacefactor/spacefactor-}/accent19e/egroup/spacefactor/accent@spacefactorcial:Exporter>

128 pages Wikipedia étaient liées au thème "Zombie", nous avons extrait 500 pages Wikipedia contenant un lien redirigeant vers la page "Zombie (film)". Ainsi nous avons pu constituer notre corpus sous forme d'un fichier texte que nous avons pu utiliser dans notre programmation Python.

Chapitre 3

Déroulement de la réalisation

3.1 Planning

N°	Tâche :	Description :	A faire pour le :	Fait par :	Terminé le :
1	Choix du groupe	Faire des groupes de 3 ou 4 étudiants	10.02.2017	Tout le monde	27.01.2017
2	Choix de la lettre	Les noms des meurtriers doivent commencer par la même lettre à définir	10.02.2017	M le professeur	27.01.2017
3	CoreNLP 3.7.0	Installation du logiciel qui nous permettra d'effectuer certaines commandes sur notre corpus	27.01.2017	Tout le monde	27.01.2017
4	GitHub	Mise en place d'un répertoire sur GitHub	10.02.2017	Sophie	11.02.2017
5	Thème corpus	Choix d'un thème pertinent	10.02.2017	Tout le monde	20.02.2017
6	Générer corpus	Création du corpus à partir de Wikipédia	24.02.2017	Sophie, Maud	23.02.2017
7	Segmenter corpus	Tokenisation à l'aide de CoreNLP	03.03.2017	Olivia, Thomas	01.03.2017
8	Trouver les entités nommées	Identification et classements des noms présents dans le corpus	17.03.2017	Sophie, Maud	problèmes rencontrés
9	Python ou Java ?	Choix du langage à utiliser pour trouver les informations souhaitées	17.03.2017	Tout le monde	01.03.2017

10	Champs lexicaux	Mise en place de différents champs lexicaux afin de pouvoir retrouver les données dans le corpus ensuite	07.04.2017	Sophie, Olivia, Maud	12.04.2017
11	Segmenter le corpus en phrases	Découper le corpus en phrases pour faciliter l'analyse	15.04.2017	Thomas	20.04.2017
12	Tri dans le corpus	Faire un tri dans le corpus pour ne garder que les phrases qui contiennent des mots inclus dans nos champs lexicaux	20.04.2017	Thomas	22.04.2017
13	Entités nommées avec Python	Identifier les entités nommées du corpus allégé sous Python	22.04.2017	Thomas	22.04.2017 (en cours)
14	Trouver les meurtriers et leurs victimes	A partir du corpus épuré, mettre en avant les noms propres représentant des meurtriers en ne gardant que ceux qui commencent par la lettre D et le nom de leur victime	22.04.2017	Thomas	22.04.2017 (en cours)
15	Mettre en avant les contextes	Trouver les éléments liés au meurtre : comment a-t-il été tué, et pourquoi ?	23.03.2017	XXX	XXX
16	Rédaction du rapport	Le rapport doit reprendre le but du projet et les explications de ce qui nous a permis de nous rapprocher du but final	24.04.2017	Sophie, Olivia, Maud	23.04.2017
17	Support de soutenance	Diapositives pour animer la soutenance orale du projet	24.04.2017	Sophie, Olivia, Maud	23.04.2017

3.2 Retour sur les tâches réalisées

Comme pour tout projet, avant de passer à l'étape de la réalisation, il faut mettre en place une période dite d'avant-projet. Ainsi, durant ce laps de temps, nous avons notamment choisi les membres de notre groupe. Ce choix n'a pas été difficile, car nous avons l'habitude de travailler ensemble et donc nous savions que nous travaillerions dans de bonnes conditions. Ensuite, M. AMBLARD, avec qui nous avons cours de travaux dirigés, nous a attribué la lettre D.

Pour la conception de notre enquête, nous avons eu besoin du logiciel CoreNLP, qui nous a permis d'utiliser certaines commandes, fournies par notre professeur. Le cœur de notre projet reposait sur l'élaboration d'un corpus autour d'un thème, qui se devait pertinent pour que nous arrivions à trouver les données nécessaires pour atteindre notre but - pour rappel : trouver 50 à 100 noms de meurtriers commençant par la lettre D et établir le contexte de chaque drame. La génération de ce corpus s'est faite à partir d'un ensemble de pages de Wikipédia, autour du thème des zombies.

Les premières étapes que nous avons réalisées ont été de segmenter le corpus à l'aide de la commande fournie dans le TD1 et du logiciel CoreNLP :

```
java -mx1g -cp ''' edu.stanford.nlp.pipeline.StanfordCoreNLP
```

```
-props StanfordCoreNLP-french.properties -annotators  
tokenize,ssplit,pos -le corpus.txt
```

Ensuite, nous avons tenté de définir les entités nommées avec ce même logiciel et une autre commande, mais nous n'avons pas réussi à le faire, nous avons donc été obligés de trouver une autre solution via Python. Nous en parlerons dans la partie concernant les problèmes que nous avons rencontrés.

La suite des événements a été de mettre en place des champs lexicaux nous permettant de faire un tri dans les données présentes dans le corpus. En effet, tout n'était pas utile au sein du corpus, il fallait donc réussir à reconnaître ce qui avait un lien avec notre enquêteur et ce qui n'en avait pas. Les champs lexicaux étaient donc là pour pouvoir mettre en avant les phrases dont le contexte était intéressant. Par exemple, une phrase telle que "The cat is eating an apple." n'avait aucun intérêt, contrairement à une phrase contenant les mots : "kill", "gun", "murder", ...

Une fois que les champs lexicaux étaient créés, nous pouvions les utiliser pour créer un corpus ne contenant que les informations pertinentes. Ce nouveau corpus ne contient plus que les phrases pour lesquelles le programme avait trouvé des termes appartenant à nos champs lexicaux. Cette étape devrait nous permettre de trouver les meurtriers et le contexte dans lequel s'est déroulé son acte. Mais avant de nous lancer dans cette recherche, nous devons identifier les entités nommées présentes dans le nouveau corpus allégé. Etant donné que nous n'arrivions pas à utiliser la commande présente dans le logiciel CoreNLP, nous avons dû effectuer des recherches assez longues afin de trouver une solution avec Python.

Une fois que nous avons trouvé notre méthode, nous l'avons appliqué à notre corpus. Mais malheureusement à cet instant, la reconnaissance des entités nommées dans ce nouveau fichier prit beaucoup de temps, si bien que bloqués par les délais, nous n'avons pas eu le résultat final de cette étape. Cependant, Thomas a tout de même décidé de commencer la fonction qui permet de trouver « qui a tué qui ? ». Pour se faire, il a utilisé le résultat qui devrait être fourni par la reconnaissance des entités nommées, et il a cherché les phrases pour lesquelles il y avait une EN représentant une personne ou deux personnes, commençant par la lettre D. Suite à cela, il fallait regarder si autour de ces termes, il y avait un terme représentant un meurtre, en s'appuyant sur notre corpus.

3.3 Github

Au sein de notre cursus, nous avons l'habitude de réaliser des projets en groupe. De cette façon, nous avons découvert l'outil qu'est GitHub. L'utilisation de ce dernier était obligatoire pour l'élaboration de notre enquêteur. Ainsi, vous pourrez trouver ci-dessous une capture d'écran de notre activité sur cet outil. De plus, si vous souhaitez visiter notre répertoire, vous pouvez vous y rendre en cliquant sur le lien suivant : <https://github.com/socosta/TAL-Lettre-D>. Une capture d'écran

des commits détaillées de chaque membre du groupe est disponible en annexe, sur la figure A.1, bien que le travail ait été collaboratif tout le long du projet, cette activité détaillée était l'un des documents demandés.

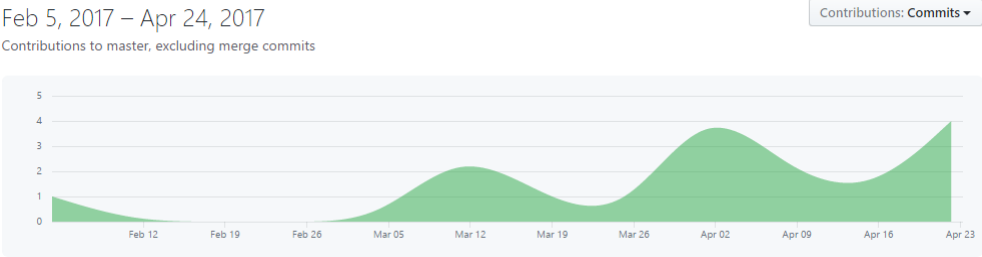


FIGURE 3.1 – Activité Github globale du projet

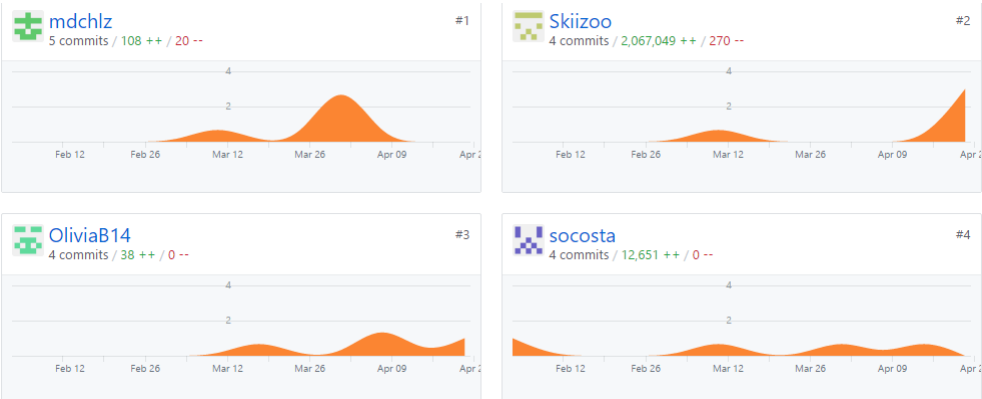
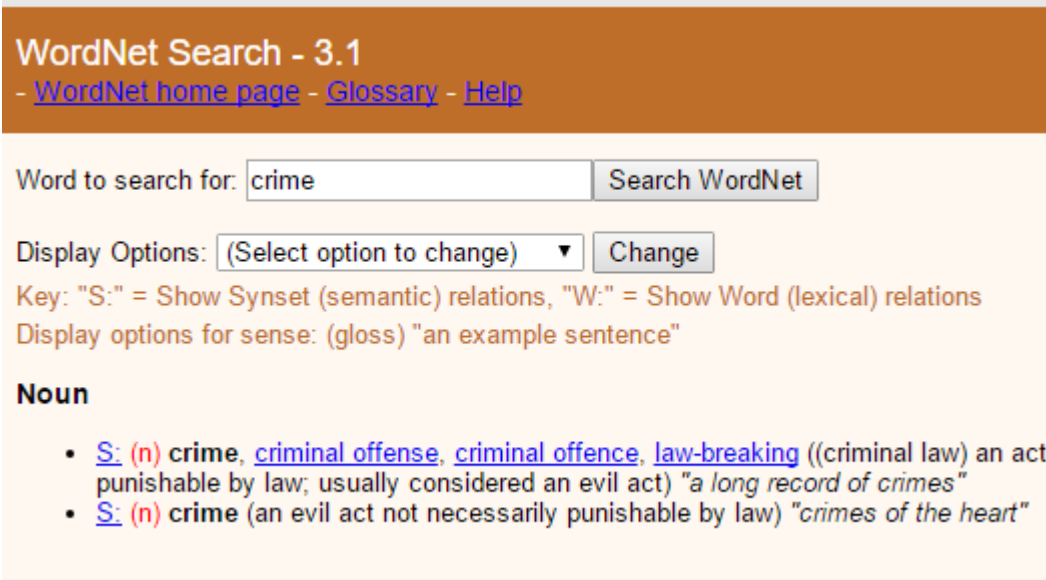


FIGURE 3.2 – Activité Github du projet par membre

Chapitre 4

Nos champs lexicaux

Nous avons choisi quatre champs lexicaux à exploiter pour trouver les meurtriers et les victimes dans notre corpus : "arme", "murder", "dead victim", et "crime". Nous avons remarqué plusieurs noms et verbes se trouvant dans plusieurs de nos champs lexicaux, c'est pourquoi à l'aide du site nuagedemots.fr, nous avons réalisé un nuage des mots de ces champs, afin de visualiser leur occurrence. Les champs lexicaux contiennent différentes catégories de mots : on y trouve des verbes, des noms communs, des adjectifs et des adverbes. A l'aide de l'outil Wordnet (<http://wordnetweb.princeton.edu/perl/webwn>), nous avons recherché les différents mots associés aux titres de nos champs lexicaux. Par exemple :



WordNet Search - 3.1
- [WordNet home page](#) - [Glossary](#) - [Help](#)

Word to search for:

Display Options:

Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations
Display options for sense: (gloss) "an example sentence"

Noun

- [S:](#) (n) **crime**, [criminal offense](#), [criminal offence](#), [law-breaking](#) ((criminal law) an act punishable by law; usually considered an evil act) "a long record of crimes"
- [S:](#) (n) **crime** (an evil act not necessarily punishable by law) "crimes of the heart"

FIGURE 4.1 – Champ lexical du mot "crime" - Wordnet

WordNet Search - 3.1

- [WordNet home page](#) - [Glossary](#) - [Help](#)

Word to search for:

Display Options:

Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations

Display options for sense: (gloss) "an example sentence"

Noun

- [S:](#) (n) **dead** (people who are no longer living) *"they buried the dead"*
- [S:](#) (n) **dead** (a time when coldness (or some other quality associated with death) is intense) *"the dead of winter"*

Adjective

- [S:](#) (adj) **dead** (no longer having or seeming to have or expecting to have life) *"the nerve is dead"; "a dead pallor"; "he was marked as a dead man by the assassin"*
- [S:](#) (adj) **dead** (not showing characteristics of life especially the capacity to sustain life; no longer exerting force or having energy or heat) *"Mars is a dead planet"; "dead soil"; "dead coals"; "the fire is dead"*
- [S:](#) (adj) **all in**, [beat](#), [bushed](#), **dead** (very tired) *"was all in at the end of the day"; "so beat I could flop down and go to sleep anywhere"; "bushed after all that exercise"; "I'm dead after that long trip"*
- [S:](#) (adj) **dead** (unerringly accurate) *"a dead shot"; "took dead aim"*
- [S:](#) (adj) **dead** (physically inactive) *"Crater Lake is in the crater of a dead volcano of the Cascade Range"*
- [S:](#) (adj) **dead**, [numb](#) ((followed by 'to') not showing human feeling or sensitivity; unresponsive) *"passersby were dead to our plea for help"; "numb to the cries for mercy"*
- [S:](#) (adj) **dead**, [deadened](#) (devoid of physical sensation; numb) *"his gums were dead from the novocain"; "she felt no discomfort as the dentist drilled her deadened tooth"; "a public desensitized by continuous television coverage of atrocities"*

FIGURE 4.2 – Champ lexical du mot "dead" - Wordnet

- [WordNet home page](#) - [Glossary](#) - [Help](#)

Search WordNet

Change

Display options for sense: (gloss) "an example sentence"

- S: (n) **crime**, **criminal offense**, **criminal offence**, **law-breaking** ((criminal law) an act punishable by law; usually considered an evil act) "*a long record of crimes*"
- S: (n) **crime** (an evil act not necessarily punishable by law) "*crimes of the heart*"

FIGURE 4.3 – Champ lexical du mot "crime" page 2 - Wordnet



FIGURE 4.4 – Nuage de mots contenus dans nos différents champs lexicaux avec une taille proportionnelle aux occurrences. - Nuagedemots.fr

Chapitre 5

Techniques utilisées et choix d'implémentation

Comme vous l'avez peut être deviné, nous avons fait le choix d'utiliser le langage de programmation Python. En effet, nous n'avons jamais utilisé le langage Java dans le domaine linguistique, ce n'était donc pas une bonne idée d'en faire usage ici. Au cours de l'enseignement que nous avons suivi, nous avons découvert, ou parfois redécouvert, des notions en lien avec le traitement automatique des langues, que nous avons bien évidemment utilisées dans notre conception. Nous avons traité des points tels que la tokenisation, la reconnaissance des entités nommées, le pos-tagging, ... Nous allons revoir ce à quoi cela correspond et la raison de leur utilisation.

Premièrement, la **tokenisation** est l'opération de segmenter un acte langagier en sous-unités appelées tokens. Les tokens les plus courants sont le découpage en mots ou en phrases. Nous avons effectué une tokenisation en phrases - dans le but de ne garder que les phrases intéressantes contenues dans notre corpus. La tokenisation en mots n'était pas vraiment appropriée étant donné que lorsque nous trouvions un mot qui avait de l'intérêt pour l'enquêteur, nous devions aussi connaître le reste du contexte de la phrase pour répondre à l'ensemble des questions.

Deuxièmement, la **reconnaissance des entités nommées**. Une entité nommée est représentée par tous les éléments du langage qui font référence à une entité unique et concrète, appartenant à un domaine spécifique. Ainsi, on les regroupe en quatre catégories : personne, organisation, localisation et date. Pour nous, l'intérêt de ces termes est de pouvoir mettre en évidence les noms de nos meurtriers et aussi trouver le lieu sur lequel ils ont commis leur crime. Afin de trouver comment effectuer cette action avec Python, nous avons dû nous plonger au cœur de la documentation, cela n'a pas été simple, mais Thomas a fini par la trouver.

Enfin, le **pos-tagging**, ou *étiquetage morpho-syntaxique*, est l'action d'asso-

cier aux mots d'un texte les informations grammaticales leur correspondant. Cette étape a été faite lorsque nous avons appliqué la première commande fournie pour l'utilisation de CoreNLP. Mais nous n'en avons pas fait usage au moment où nous rédigeons ce rapport.

D'un point de vue général, l'implémentation de notre code, réalisé en Python, permet les étapes suivantes :

- * L'ouverture de notre corpus, en mode lecture,
- * La tokenisation de ce dernier en phrases,
- * La mise en place d'une fonction permettant de chercher les phrases contenant des termes présents dans nos champs lexicaux,
- * Ces phrases sont ensuite mises de côté pour élaborer un corpus plus léger,
- * Ensuite vient la reconnaissance des entités nommées,
- * Puis la recherche de la réponse à la question « qui a tué qui ? » , en s'appuyant sur le résultat fourni par l'étape précédente.

Chapitre 6

Problèmes lors de la réalisation

Etant donné que nous ne sommes pas vraiment habitués à écrire des programmes pour le traitement automatique des langues, la réalisation de ce projet nous a posé quelques problèmes.

Cela a commencé lorsque nous avons dû utiliser le logiciel CoreNLP. Nous n'avions aucune connaissance à sujet. Alors quand nous avons essayé de faire fonctionner les commandes fournies par M.AMBLARD, nous avons été face à des difficultés. En ce qui concerne la commande qui permet d'effectuer le pos-tagging, nous n'arrivions pas à la faire fonctionner sur notre corpus. Nous étions face à une erreur mettant en cause la place disponible dans notre machine virtuelle. A force de persévérance, avec l'aide de certains de nos camarades d'autres groupes, nous avons réussi à obtenir le résultat souhaité. Cependant, lorsque nous avons voulu utiliser la commande permettant l'identification des entités nommées, cela ne fonctionnait pas, malgré plusieurs essais, le résultat était toujours le même : le terminal n'arrivait pas au bout de sa tentative. Ainsi, nous nous sommes dit que la taille importante de notre corpus était peut-être en cause, mais même sur un extrait, cela n'allait pas. Alors, nous avons tenté de trouver une solution. Et Olivia trouva ce site : <http://www.lattice.cnrs.fr/sites/itellier/SEM.html>. A partir d'un texte, il est possible de mettre en avant différentes choses : *pos*, *chunk* et *entités nommées*. Ci-dessous, une capture d'écran afin de se rendre compte de ce que cela donne lorsque nous voulons trouver les entités nommées d'un texte :

Part-Of-Speech	Chunking	Named Entity
<p>In Louisiana's Seven Doors Hotel in 1927, a lynch mob murders an artist named Schweick, whom they believe to be a warlock. This opens one of the Seven Doors of Death, allowing the dead to cross into the world of the living. Several decades later, Liza, a young woman from New York, inherits the hotel and plans to re-open it. Her renovation work activates the hell portal, and she contends with increasingly strange incidents. A plumber named Joe investigates flooding in the cellar and a demonic hand gouges out his eye. His body and another are later discovered by a hotel maid, Martha. Liza encounters a blind woman named Emily, who warns that reopening the hotel would be a mistake. Joe's wife Mary-Anne and their daughter Jill arrive at the hospital morgue to claim Joe's corpse. Jill finds her mother lying on the floor unconscious, her face burned by acid. Liza meets with Dr. John McCabe, and receives a phone call informing her of Mary-Anne's death. After the funerals, Liza encounters Emily at the hotel. Emily tells Liza the story of Schweick, and warns her to not enter room 36. When Emily examines Schweick's painting, she begins to bleed and flees the hotel. Liza ignores Emily's advice, and investigates room 36. She discovers an ancient book titled "Eibon". She sees Schweick's corpse nailed to the bathroom wall. She flees the room in terror, but is stopped by John. She takes him to room 36 but both the corpse and the book are gone. Liza describes her fearful encounters with Emily, but John insists that Emily is not real. While in town, Liza spots a copy of "Eibon" in the window of a book store, but when she rushes in to grab it, a different book is in its place. The shop owner says the book has been there for years, prompting Liza to remark to John that perhaps it is all in her head. At the hotel, a worker named Arthur attempts to repair the same leak as Joe, but is killed off-screen by ghouls. Liza's friend Martin visits the public library to find the hotel's blueprints. He is struck by a sudden force and falls from a ladder, resulting in paralysis. Spiders ravage his face and kill him. Martha is cleaning the bathroom in Room 36 when Joe's animated corpse emerges from the bathtub. Joe pushes her head into an exposed nail, killing her and destroying one of her eyes. The walking corpses of Schweick, Joe, Mary-Anne, Martin and Arthur invade Emily's house. She pleads with them to leave her alone, and insists she will not return with Schweick. She commands her guide dog to attack the corpses, but the dog turns on Emily, tearing out her throat. At the hotel, spirits terrorize Liza. John breaks into Emily's house, which appears to have been abandoned for years, and finds "Eibon". He returns to the hotel and tells Liza that it is a gateway to Hell. They flee to the hospital, but it has been overrun by zombies. Liza is attacked, but John gets a gun out of his desk and shoots the shambling corpses. Only Harris and Jill are found still alive, but Harris is killed by flying shards of glass. Jill, having shown signs of possession since the funeral, finally attacks Liza. John is forced to kill Jill. Escaping the zombies, John and Liza rush down a set of stairs but find themselves back in the basement of the hotel. They move forward through the flooded labyrinth and stumble into a supernatural wasteland of dust and corpses. No matter which direction they travel, they find themselves back at their starting point. They are ultimately blinded just like Emily, succumb to the darkness, and disappear.</p>		

Une légende est associé au surlignage :

POS	Chunks	Entités nommées
ADJ, ADJWH	UNKNOWN	Company
ADV, ADVWH	AP	FictionalCharacter
CC	AdP	Location
CL, CLO, CLR, CLS	CONJ	Organization
CS	NP	Person
DET, DETWH	PP	Product
ET	VN	
I		
NC, NPP		
P, P+D, P+PRO		
PONCT		
PREF		
PRO, PROREL, PROWH		
V, VIMP, VINP		
VPP, VPR, VS		

Ainsi, nous nous sommes dit que nous avions une solution de secours si nous n'arrivions pas à établir les entités nommées à partir de Python et de la librairie NLTK. Sachant qu'il est possible d'extraire le résultat du site sous les formes suivantes : HTML, coll ou text. Heureusement, nous avons fini par trouver la solution pour établir la reconnaissance des entités nommées avec Python. Mais un nouveau problème s'est présenté à nous, cela a pris beaucoup plus de temps que nous ne l'imaginions.

Nous avons principalement connu des embûches à cause de nos connaissances : auparavant, nous n'avions utilisé NLTK que partiellement et nous sommes loin de maîtriser l'ensemble des possibilités présentes dans cette librairie. Aussi, c'est la première fois que nous devons utiliser le logiciel LATEX, et c'est aussi un point qui nous a posé problème. Nous sommes loin de connaître le fonctionnement de ce dernier, ainsi la mise en page de notre rapport est restée très basique ne sachant pas comment faire autrement - si cela était possible.

Chapitre 7

Rétrospective et conclusion

Malgré beaucoup de recherches, notamment sur wordnet, nous estimons que nos champs lexicaux n'étaient peut être pas assez complets. En effet, compte-tenu de la taille de notre corpus, nous sommes probablement passés à côté de beaucoup d'autres mots qui auraient pu nous indiquer un meurtrier ou tout autre indication qui aurait pu nous aider dans notre étude. Donc nous pensons que nous ne pouvions pas mettre la main sur toutes les informations disponibles dans le corpus.

Néanmoins, nous avons pu tout de même obtenir un corpus, tokenizer le corpus, établir des champs lexicaux et encore repérer les entités nommées. Grâce à toutes ces étapes, nous sommes capables de déterminer les tueurs et le lieu de leur meurtre (ou bien le film dans certains cas), ainsi que les victimes et la façon dont cela s'est passé.

Nous aurions pu, éventuellement, créer des champs lexicaux directement via Python. En effet, grâce à cette technique nous aurions pu avoir des champs lexicaux plus vastes. De plus, faire les champs lexicaux grâce à Python nous aurait permis un travail moins fastidieux. Mais nous avons déjà mis en place nos champs lexicaux lorsque nous avons pris connaissance de cette technique via Python.

Enfin, notre plus gros défaut a été de ne pas savoir gérer notre temps avec les autres projets en cours. Durant quelque temps, nous sommes restés axés sur nos problèmes et nous les avons laissés de côté, à tort. Ce qui explique que notre enquêteur nest pas abouti aujourd'hui.

Annexe A

Activité Github

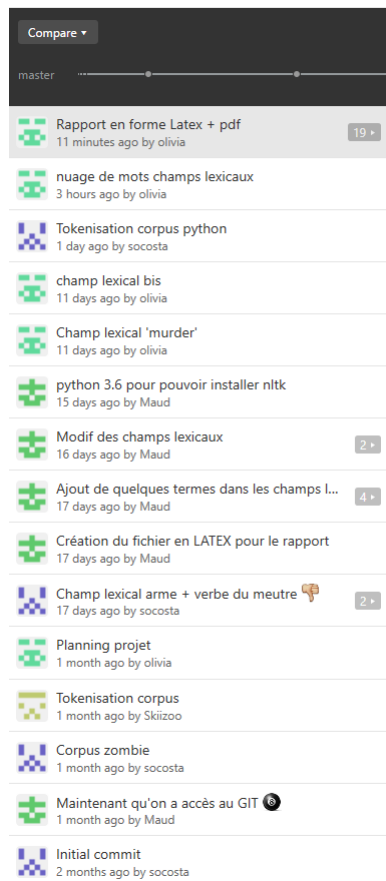


FIGURE A.1 – Activité détaillée Github des membres du groupe

Table des figures

3.1	Activité Github globale du projet	8
3.2	Activité Github du projet par membre	8
4.1	Champ lexical du mot "crime" - Wordnet	9
4.2	Champ lexical du mot "dead" - Wordnet	10
4.3	Champ lexical du mot "crime" page 2 - Wordnet	11
4.4	Nuage de mots contenus dans nos différents champs lexicaux avec une taille proportionnelle aux occurrences. - Nuagedemots.fr	11
A.1	Activité détaillée Github des membres du groupe	17