FIALE Amélie
JACOB--GUIZON Xavier
RICHEZ Guillaume
DARROU Cédric

Traitement automatique du langage

Rapport de projet

Session 2019-2020

Sommaire

Outils

Nous avons utilisé GitHub car il s'agit d'une plateforme de gestion de versions que nous avions tous eu l'occasion d'utiliser auparavant, annulant la période d'adaptation à la plateforme, ainsi que la création de comptes sur de nouveaux sites.

L'utilisation de la plateforme Trello suit la même logique.

Pour le langage de programmation, nous avons Python 3, avec la librairie *NLTK*.

Par ailleurs afin de pouvoir communiquer à distance nous avons utilisé la plateforme Discord et les réseaux sociaux, tel que messenger.

De plus nous avons aussi utiliser un googledrive pour pouvoir rédiger le rapport du projet tous ensemble.

Objectif du projet

Pour ce projet en traitement automatique du langage, nous devions présenter une liste de 50 assassins, ainsi que les informations sur le meurtre: qui a été tué, quand et où. Les informations doivent être automatiquement extraites de wikipédia et noms des meurtriers doivent commencer par la même lettre.

Étapes du projets

Afin de mener à bien notre projet nous devions faire plusieurs étapes.

Étape 1 : Nous devions récupérer les pages wikipédia en format xml, afin de pouvoir les étudier.

Étape 2 : Il fallait récupérer le texte des fichiers xml.

Étape 3 : nous devions analyser le texte récupérer en fonction de sa forme (tableau, texte) et le trier en bloc.

Étape 4 : Il fallait que l'on trie l'utile et le non utile (exemple: La présentation wikipédia n'est pas utile alors que le meurtrier avec son meurtre sont des information utile)

Étape 5 : Nous devions *tokeniser*, échantillonner les blocs de texte trouvé, les tagger et les diviser en morceaux (*chunks*) afin de les trier avec regex.

Étape 6 : Il fallait associer les différents éléments trouver aux mots d'intérêts et ainsi trouver du sens.

Étape 7 : Il fallait afficher les résultats viables par rapport à notre texte.

Techniques utilisées

Afin de mener à bien notre projet nous avons dû utiliser des techniques en traitement automatique du langage. De ce fait nous avons utilisé la technique décomposer. Cette technique nous a permis de découper le texte en section et ensuite de découper un texte en phrases. Puis grâce à la technique Part-of-Speech tagging nous avons pu assigner une catégorie syntaxique à chaque mot d'une phrase. Nous avons ensuite utilisé le Parsing qui nous a permis de construire un arbre syntaxique d'une phrase. Cette technique permet aussi d'identifier les catégories prédéfinie dans une phrase. Nous avons ensuite utilisé la Désambiguïsation lexicale, qui permet d'identifier le sens exact d'un mot ou d'une entité.

Pour l'utilisation du programme, nous utilisons un fichier prinicipal (« Main.py »), chargé d'appeller les fonctions des scripts annexes. Cela permet d'avoir plus de clarté, et de pouvoir naviguer vers les fonctions, plutôt que de devoir les voir dans le même fichier.

Les difficultés rencontrées

Durant la réalisation de notre projet nous avons rencontrés plusieurs difficultés.

Tout d'abord, nous avons eu du mal au début à télécharger les pages wikipédia en format Xml, cependant nous avons réussit à palier ce problème en ajoutant spécial: export à mettre entre /wiki/ puis ensuite nous avons réussi à enregistrer les pages. De ce fait les pages que nous n'arrivons pas à télécharger au départ on était télécharger au bon format grâce à cela nous avons pu former notre corpus de texte.

De plus nous avons dû utiliser une autre lettre. La lettre qui nous a été assigné au départ était la lettre O cependant nous n'avons pas trouver assez d'assassins commençant par la lettre O. De ce fait, afin de pouvoirs avoir le nombre d'assassin nécessaire, nous avons choisit la lettre E.

Les conditions actuelles ne sont pas très favorables au travail en groupe. Mais nous avons utilisé les réseaux sociaux et les plateformes, tels que Discord, afin de pouvoir nous réunir et ainsi de discuter à propos de notre projet.

Par ailleurs, tout les outils ne fonctionnaient pas sur tous les ordinateurs des membres du groupe. De ce fait nous avons donc décidé de faire le code tous ensemble lors de nos réunion, afin de n'écrire le code que sur un seul ordinateur, celui de Xavier.

Conclusion

Dans la majorité des cas notre projet fonctionne et cela malgré les difficultés rencontrées tout au long de notre projet. De plus ce projet nous a permit d'en apprendre plus sur le traitement automatique des langues, et d'en d'appréhender les outils et techniques en traitement automatique des langues.

Annexes

Annexe I : Extrait du code (« Main.py »)

```
import Outils.OutilsPrincipaux as outils
from pprint import pprint

# Récupération du texte
text = outils.recuperationArbreText("./../XMLSources/List_of_serial_killers_by_country.xml")

for sent in outils.formatage(text):
        # formatage de la phrase sent
        format = outils.formatage(sent)

        # pour les phrases formatées
        for sentFormat in format:
                # preProcess (tokened,tagged)
                preProc = outils.preProcess(sentFormat)
                # création d'un arbre de la phrase tagged
                tree = outils.chunker2Tree(preProc)

                # création d'un arbre taggé avec IOB
                iob = outils.iobTaggSent(tree)

                Relation = outils.creatRelation(tree)

                outils.filtreRelDependance(tree)
```

https://github.com/jacobgui1u/GitTAL