PROJET TAL - ÉVALUATION D'OUTILS DE TAL

-Traitement Automatique des langues-

My-Linh HO, Jérémy TOUBOUL

08/03/2020

ET5 - Polytech Paris-Saclay

INTRODUCTION

Le langage caractéristique propre de l'homme est la capacité de communiquer au moyen d'un système de signe doté d'une sémantique et d'une syntaxe. De tout temps, les hommes ont tenté de communiquer avec leurs semblables. L'apprentissage d'une langue est un processus long et épuisant. L'apparition de l'informatique ainsi que des outils de traitement automatique du langage dans les années 50 offrent une nouvelle perspective d'avenir pour les langues.

Le traitement automatique de la langue naturelle est une science multidisciplinaire s'appuyant sur la linguistique, l'informatique ainsi que l'intelligence artificielle. Aujourd'hui, on compte environ 7 000 langues parlées dans le monde (ce chiffre ne prend pas en compte les dialectes), et ces langues suivent des structures très précises. Un des objectifs du traitement des langages naturels est de repérer le rôle de chacuns des mots dans les phrases que l'on écrit. Cela peut par exemple revenir à identifier les noms, verbes, etc. (Part of Speech) mais également les entités nommées et à comprendre l'information qu'elles contiennent (nom de lieu, de personne, etc.). Des méthodes existent aujourd'hui pour identifier de telles informations dans un corpus.

L'objectif de ce projet était de comparer des méthodes de tokenisation de POS (Part Of Speech) et de reconnaissance d'entités nommées, et de comparer les résultats obtenus avec des corpus déjà annotés pour observer la précision des méthodes.

Pour réaliser cette étude, nous avons utilisé les deux méthodes suivantes :

- L'analyseur multilingue LIMA qui fournit des tokens dans un format qui lui est propre.
- L'analyseur du Stanford NLP Group qui fournit des tokens au format PTB.

PRÉSENTATION RAPIDE DES ANALYSEURS

CEA List LIMA est une plateforme d'analyse linguistique multilingue utilisant des règles et des ressources validées par des experts linguistes. Lima a été designé pour offrir une large flexibilité de configuration sans sacrifier la rapidité d'exécution et la qualité des résultats. La flexibilité et l'efficacité de LIMA viennent de deux besoins: la prise en compte de la problématique du multilinguisme et couvrant un large gamme d'applications, et la nécessité de traiter de très grands corpus avec les niveaux d'analyse de plus en plus approfondis requis par les outils de recherche avancée tels que les systèmes de questions-réponses.

Stanford Core NLP est une boîte à outils linguistiques utilisant l'apprentissage statistique à partir de corpus annotés. Ces outils peuvent être utilisés dans un pipeline, pour convertir une chaîne contenant du texte en langage humain en listes de phrases et de mots, pour générer des formes de base de ces mots, leurs caractéristiques morphologiques, et pour donner une analyse syntaxique de dépendance de structure, qui est conçue pour être parallèle parmi plus de 70 langues, en utilisant le formalisme des Dépendances Universelles.

NLTK est une boîte à outils linguistiques utilisant des approches hybrides combinant l'apprentissage automatique et des ressources linguistiques. Il fournit des interfaces faciles à utiliser pour plus de 50 corpus et ressources lexicales tels que WordNet, ainsi qu'une suite de bibliothèques de traitement de texte pour la classification, la tokenisation, le tagging, parsing, l'analyse syntaxique et le raisonnement sémantique. NLTK convient aux linguistes, aux ingénieurs, aux étudiants, aux éducateurs, aux chercheurs et aux utilisateurs industriels. NLTK est disponible pour Windows, Mac OS X et Linux. Mieux encore, NLTK est un projet libre, open source et communautaire. Pour ce projet, nous n'avons pas utilisé cette technologie, mais l'avons tout de même mise en pratique lors de nos Travaux Pratiques.

DESCRIPTION DU PROGRAMME

Le script principal de ce projet repose sur plusieurs méthodes principales :

La première est la fonction ***build_vocab*** qui prend une table d'équivalences de tokens et qui en constitue un dictionnaire en python. Il est donc facile de trouver l'équivalent d'une étiquette à travers ce dictionnaire.

La méthode ***extract_sentences*** a pour but de reconstituer les phrases d'origine à partir d'un corpus annoté. Elle parcourt les lignes en ne gardant que le token, et joint tous les tokens en les séparant par des espaces. On fait toutefois attention à supprimer les espaces avant les différentes ponctuations (points, virgules, etc.) et également à conserver les sauts de lignes entre les phrases différentes.

Une autre méthode importante est la méthode ***translate*** qui prend un corpus annoté, dans un format en 2 colonnes, et qui transcrit le tag de chaque mot par son équivalent dans le vocabulaire fourni en paramètre.

Un groupe de fonctions importantes sont des fonctions qui récupèrent le résultat d'une analyse POS/NE et qui les met au format 2 colonnes dont nous avons besoin pour nos analyses. Pour Lima par exemple, le résultat obtenu est un tableau à plusieurs colonnes. Quand nous procédons à une analyse POS nous gardons la colonne correspondante, et quand nous faisons une analyse NE nous récupérons le type d'entité nommée dans une autre colonne par le biais d'une expression régulière.

Enfin la dernière fonction est celle qui traduit les entités nommées au format CoNLL. Ce format permet de garder une information supplémentaire sur les entités nommées en plusieurs mots. En effet le premier mot de l'entité nommée se voit attribué un tag commençant par "B-" (Beginning) et les mots suivants dans le groupe ont un tag commençant par "I-" (inside). Pour ce faire, on garde en mémoire le dernier tag rencontré, si le suivant est différent alors on fait commencer le tag par B, sinon par I. De la même façon, si on rencontre une fin de phrase alors on sort de l'entité nommée.

RESULTATS

Dans le cadre de ce projet nous sommes donc parti des corpus de référence fournis, nous en avons extrait les phrases d'origines et ré-appliqué des analyseurs syntaxiques. A partir de ces fichiers reformattés nous avons pu appeler le script evaluate qui donne une précision des analyses en regard des fichiers de référence.

Tag precision	**POS**	**NE**
LIMA	0.0092	0.0143
Stanford	0.0099	0.0092

Les scores obtenus lors de ces analyses sont relativement faibles.

Il est possible que ces scores faibles ne soient pas uniquement dûs à la qualité des méthodes syntaxiques utilisées mais également au format dans lesquels les résultats ont été mis. En effet si un décalage a lieu au sein des phrases du corpus alors il peut causer des incohérences dans la phase de comparaison. De plus nous avons repéré par exemple pour le fichier de référence des entités nommées que certains tokens étaient munis de 2 tags, ce que notre convertisseur CoNLL ne génère pas.

Nous avons donc de faibles taux de précision à cause des problèmes dans les fichiers de références, qui auraient pris trop de temps à corriger au vu du temps que nous avions pour ce projet et de la charge de travail que nous avions en parallèle.

Une amélioration à apporter pourrait être une vérification intermédiaire sur la concordance des fichiers test et référence pour être sûr de la similarité de leurs structures.

RÉPARTITION DU TRAVAIL

La majorité de ce projet a été effectuée en groupe car les fonctions écrites sont souvent ré-utilisées au sein du script.

Concernant les TP, nous avons travaillé dessus en binôme lors des séances prévues à cet effet dans le planning. Cependant nous avons rencontrés plusieurs problèmes nous empêchant de bien les réaliser. Notamment le TP2, avec les même problèmes que les autres binômes, et l'installation de la machine virtuelle avec la bonne version d'Ubuntu était aussi un peu compliquée …