Traitement Automatique des Langues

Rapport de projet

ET5 Informatique

SOW Oumou, BERNIER Chloé et CLAUDE Philippe
08/03/2020

Table des matières

I. Introduction

Dans le cadre de la validation du cours de TAL (Traduction Automatique des Langues), nous avons travaillé pendant un mois sur un projet permettant de comparer trois plateformes open source d'analyse linguistique entre elles : Stanford Core NLP, CEA List LIMA et NLTK. Elles fournissent des logiciels ou des bibliothèques permettant le traitement automatique des langues.

L'objectif principal du projet était de montrer que nous pouvions installer, évaluer et rédiger un rapport décrivant les principaux résultats obtenus mais également les points forts et limitations des plateformes.

Ce projet se divise en deux parties :

- Partie I : évaluation de l'analyse morpho-syntaxique
- Partie II : évaluation de la reconnaissance d'entités nommées

II. Description des plateformes évaluées

CEA List LIMA : est un analyseur multilingue du CEA LIST. Cet outil permet de réaliser des analyses syntaxiques et d'extraire des entités nommées en 9 langues (anglais, français, allemand, espagnol, italien, russe, arabe, chinois et hongrois). Il utilise des règles et des ressources validées par des experts linguistes.

Site utile : https://github.com/aymara/lima/wiki/LIMA-User-Manual

Stanford Core NLP : offre un ensemble d'outils d'analyse du langage naturel. À partir d'un texte simple, il peut fournir les formes de base des mots, leurs parties de discours, normaliser et interpréter des quantités numériques, baliser la structure des phrases en termes de phrases ou de dépendances de mots, et indiquer quels noms de phrases se réfèrent aux mêmes entités. Ces outils utilisent divers éléments d'apprentissage automatique basé sur des règles, d'apprentissage probabiliste et d'apprentissage approfondi. A l'origine, il a été développé pour l'anglais, mais il offre désormais différents niveaux de support pour d'autres langues (français, arabe, chinois, etc.).

Site de référence : https://nlp.stanford.edu/software/

NLTK : est une plateforme pour la création de programmes Python destinés à travailler avec des données en langage humain. Il fournit de nombreux corpus et ressources lexicales, de même qu'une suite de bibliothèques de traitement de texte pour la classification, la tokenisation, le balisage, l'analyse syntaxique et le raisonnement sémantique, etc. C'est donc une boite à outils linguistiques qui utilise des approches hybrides combinant l'apprentissage automatique et des ressources linguistiques.

Site de référence : http://www.nltk.org/

III. Expérimentation

a. Données de test (préparation de données)

Partie I : évaluation de l'analyse morpho-syntaxique

Nous sommes partis d'un fichier « pos_reference.txt.lima », qui contenait le texte avec les tags de LIMA, et nous avons construit le fichier « pos_reference.txt.univ » avec les tags universels et le fichier « pos_test.txt ». Le fichier « pos_test.txt » va nous servir de point de départ pour faire nos tests avec les différentes plateformes, et le fichier « pos_reference.txt.univ » va nous servir de référence dans le script du evaluate.py.

Partie II : évaluation de la reconnaissance d'entités nommées

A partir du fichier « ne_reference.txt.conll », nous avons pu reconstituer le texte « ne_test.txt » sur lequel on a pu ensuite extraire les entités nommées avec les trois différents NE recognizers. Dans ce fichier, la fin d'une phrase est indiquée par une ligne vide. Le fichier « ne_reference.txt.conll » va également nous servir pour évaluer les résultats de l'extraction à la fin à l'aide du script evaluate.py

b. Métriques d'évaluation

Le script evaluate.py, implémenté par un chercheur de l'université de Stanford, est un programme d'évaluation d'un traitement d'un texte candidat en le comparant à une référence. Il nous donne la précision, le rappel et la F-mesure qui est combinaison des deux premières mesures.

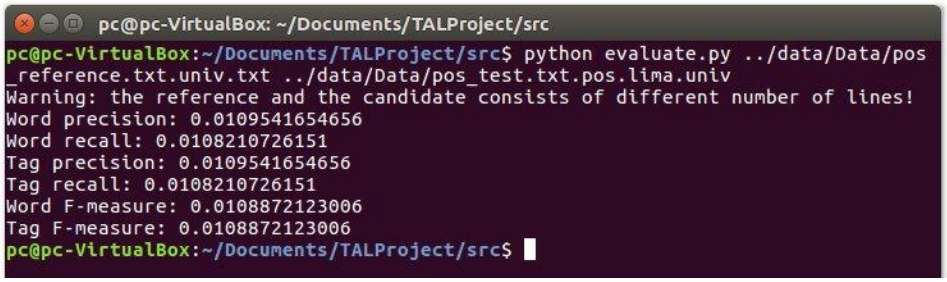
Grâce à ce code, nous pourrons étudier différents taux sur les fichiers en entrée :

- La précision : c'est un taux qui se calcule en fonction du nombre total de mots ou de tags corrects dans le fichier candidat.
- Le rappel : ce taux se calcule en fonction du nombre total de mots ou de tags corrects dans le fichier candidat.
- La F mesure : elle se calcule grâce aux deux points précédents. Sa formule est : *word_fmeasure = (2*word_precision*word_recall)/(word_precision+word_recall)*

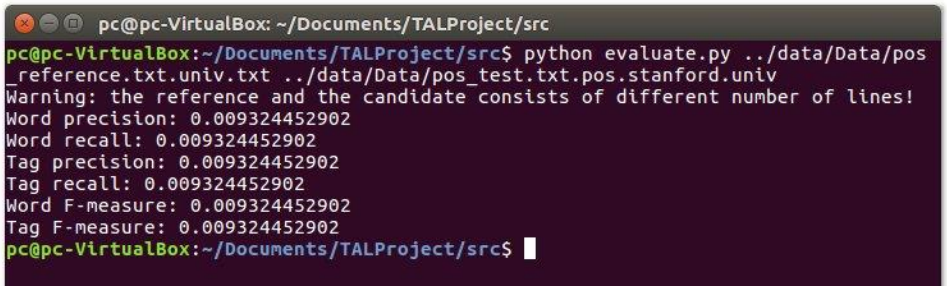
c. Résultats

Partie I : évaluation de l'analyse morpho-syntaxique

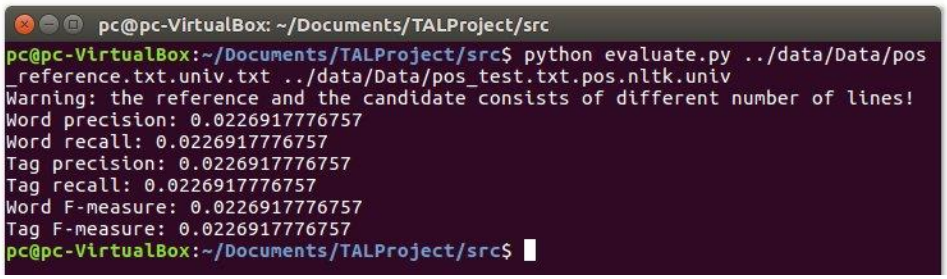
LIMA:



Stanford:

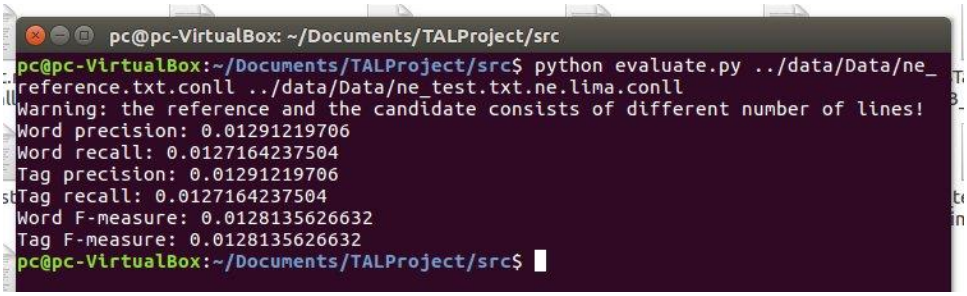


NLTK:

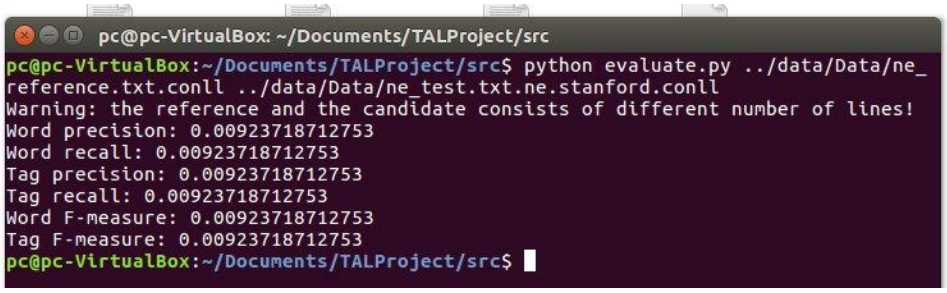


Partie II : évaluation de la reconnaissance d'entités nommées

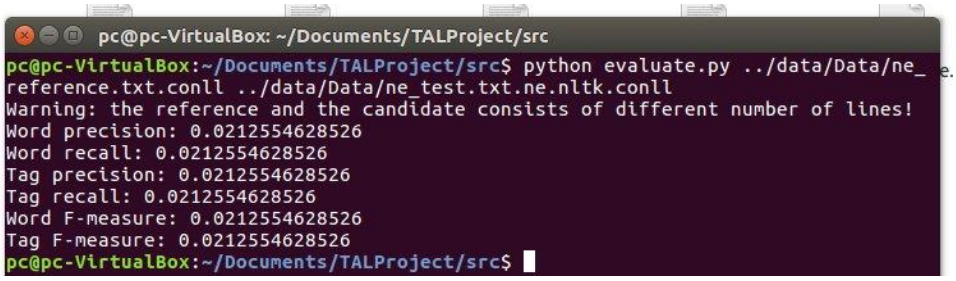
LIMA :



Stanford:



NLTK:



d. Discussion sur les résultats : comparer les plateformes

Les résultats que nous avons obtenus sont très faibles comparer à ce que nous attendions. Ceci peut s'expliquer du fait de l'écart que nous avons entre nos fichiers et les références. En effet, certains groupes de mots se sont séparés lors du traitement dans un fichier et pas dans la référence. Nous ne sommes pas parvenus à trouver la cause de ce dérèglement.

Toutefois, étant donné que nous avons eu ce problème avec les trois plateformes, nous pouvons quand même comparer les résultats entre eux. La précision du NLTK est plus forte que celle de la plateforme LIMA, elle-même plus forte que celle de Stanford que ce soit pour l'évaluation morpho-syntaxique ou bien celle des entités nommées.

Conclusion

a. Résumé du travail fait

Nous avons effectué l'ensemble du travail demander pour les trois plateformes d'analyse linguistique. Cependant, nous avons rencontré de nombreuses difficultés dans la réalisation de ce projet. Le temps imparti était assez court comparer à tout le travail à réaliser. De plus il y'avait beaucoup de travail en amont à faire par exemple compléter les fichiers de correspondance entre les étiquettes LIMA vers PTB et PTB vers Universal.

Pour ce qui est de notre organisation, l'ensemble des membres de l'équipe ont participés à la réalisation de ce projet. Nous disposions d'un unique ordinateur avec les bonnes configurations de la machine virtuelle, cependant nous nous sommes bien organisés pour écrire les scripts et faire les tests par la suite sur la VM.

b. Limitations de chaque plateforme

LIMA est une plateforme plus complète que celle de Stanford car elle fournit en une ligne de commande de nombreuses informations utiles comme les POSTags ou les entités nommées.

NLTK offre également un large choix de fonctionnalités, la documentation est assez riche et facile à trouver et à prendre en main.

c. Proposition pour des améliorations

Stanford Core NLP consomme beaucoup de mémoire, si on souhaite traiter un gros fichier, comme un roman par exemple, on peut imaginer le traiter par morceau afin de réduire l'utilisation de la mémoire.

Sources

http://www.kalisteo.fr/fr/thematic_tx.htm

http://www-list.cea.fr/innover-pour-l-industrie/nos-atouts-pour-les-industriels/produits

https://github.com/stanfordnlp/CoreNLP

https://stanfordnlp.github.io/CoreNLP/memory-time.html