Célestin COLLIN
Elian RAYNAUD

Traitement Automatique des Langues

-

Projet ET5

Introduction

Ce projet de TAL a pour but de comparer différents outils d'analyse morpho-syntaxique. En l'occurrence, nous allons étudier l'outil de Stanford, et l'outil NLTK. Cette comparaison s'effectue sur deux composants de ces outils : le *postagger* et le *NER* (named entities recognizer).

Le *postagger* : il va permettre d'analyser chaque token d'un texte (en utilisant le contexte) et ainsi déterminer si ce token est un verbe, un nom, etc.

Le *NER* : un token (ou ensemble de token) peut représenter une entité nommée (c'est-à-dire un lieu, une personne, etc.). Dans ce cas, le *NER* va permettre d'identifier cette entité, et d'analyser son contenu.

Pour évaluer ces outils, on va comparer leurs résultats à un résultat de référence via le script *evaluate.py* fournit dans le projet. Ce script nous fournit plusieurs facteurs d'évaluations :

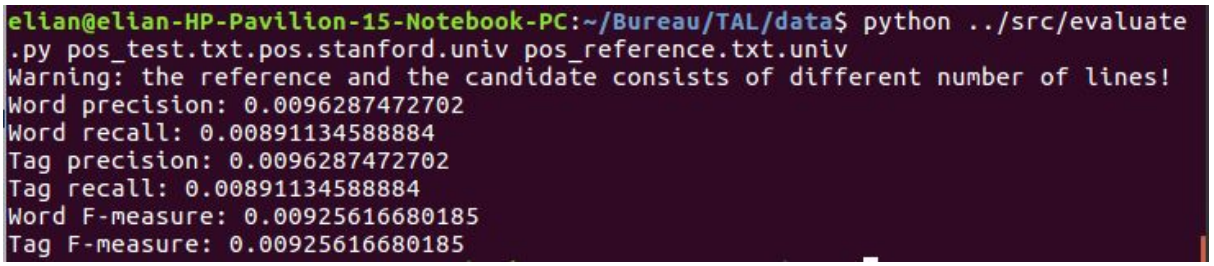
- la *word precision* : le ratio de mots ayant le bon *tag* par rapport à la référence
- le *word recall* : le quotient entre le nombre d'éléments classés dans le bon type et le nombre d'éléments appartenant à ce type

Les autres facteurs ne nous seront pas utiles : *tag precision* et *tag recall* sont équivalents, et les F-measure fournissent une moyenne harmonique qui ne nous est pas utile.

Résultats

Outil : Stanford

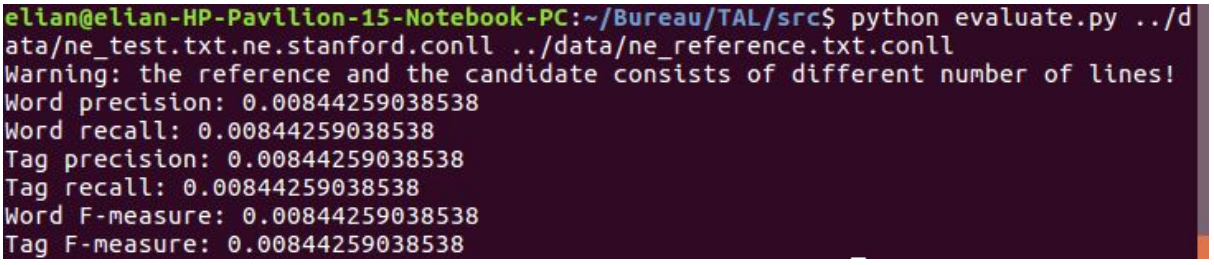
Résultats pour l'évaluation du postagger de Stanford :



Comme on peut le voir, la précision est très faible. Il y a ici 2 explications possibles :

- cela peut être dû aux multiples conversions entre les étiquettes PTB, lima, et univ, qui engendrent une grande perte d'information
- cela peut-être dû à une erreur de désynchronisation entre le fichier à évaluer et le fichier référence, ce qui expliquerait une si faible précision

Résultats pour l'évaluation du NER de Stanford :

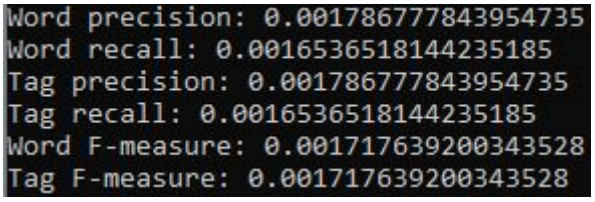


Encore une fois, les résultats ne sont pas brillants. C'est d'autant plus étonnant que si les O (entités qui ne sont pas des entités nommées) sont comptabilisés dans le script evaluate.py, cela devrait assurer une précision assez élevée car ils représentent la majorité du texte à traiter.

En revanche, si le script d'évaluation ne considère que les entités nommées, alors le résultat paraît plus cohérent. En effet, la reconnaissance des entités nommées de CONLL est beaucoup plus précise que celle de Stanford. Prenons un exemple dans le texte à traiter : "The United States's largest car manufacturer General Motors [...]". Ici, CONLL identifie tout ce bloc comme une organisation, et reconnaît également United States comme un emplacement, alors que Stanford lui ne reconnaît que United States comme un emplacement. De plus, Stanford ne fait pas de distinction entre un B - … et un I - … contrairement à CONLL, ce qui engendre encore plus d'imprécision.

Outil : NLTK

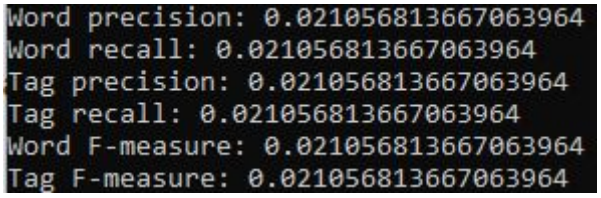
Résultat du POS tagger nltk :



On obtient donc une précision de 0.1%, ce qui est très mauvais… Cependant, la faible précision s'explique non pas par la mauvaise performance du modèle, mais plutôt par le fait que le format des fichiers de référence n'est pas toujours en accord avec le résultat donné par le framework : en effet certain mot, comme l' sont chez l'un découpé en deux (l et ') alors qu'il reste ensemble chez l'autre. Cela crée donc des erreurs importantes.

Cela peut aussi s'expliquer par la grande perte d'information lors de la conversion d'un système de tag au système universel ; celui-ci possède en effet un nombre très restreint de tags, ce qui oblige donc à perdre des informations lors de la conversion.

Résultat de la reconnaissance d'entités nommées :



On arrive ici à un bien meilleur résultat, puisque l'on atteint les 2% de précision. Les raisons pour cette faible précision sont les mêmes que pour le tagger. Cependant en implémentant la détection des débuts et fins de mots, on arrive à un résultat plus proche de la référence.

Il est néanmoins important de préciser que pour les deux résultats, la précision est bien meilleure lorsque l'on compare le fichier résultat et le fichier de référence à la main. On s'aperçoit alors que les résultats affichés ne sont dû qu'à de petits décalages de lignes, mais le texte en soi est en grande majorité correctement interprété par le système.

En regardant ainsi le fichier on trouve une précision plus proche de 70-80%, ce qui est beaucoup plus correct.

Conclusion

Les résultats semblent faussés par des erreurs de synchronisation des formats. Outre ce problème, nous avons pu observer, en lisant les fichiers de sortie des différents analyseurs morpho-syntaxiques, des grosses différences de précisions entre les différents outils. L'outil de Stanford est le plus rudimentaire : ses étiquettes sont assez restreintes (notamment pour les entités nommées) et il ne fait pas de distinctions entre le début des entités et ce qui est à l'intérieur (ainsi il n'identifie pas les blocs correspondant à une unique entité). L'outil NLTK lui est plus précis. En effet, il identifie les entités en bloc, même si elles sont composées de plusieurs mots différents.

Outre les différences de précisions des étiquettes, l'outil CONLL semble beaucoup plus puissant que les deux autres. Comme on l'a montré précédemment avec l'exemple sur General Motors, il a une meilleure compréhension du texte et prend en compte un contexte bien plus large que les deux autres.