COLLIN
Khévin
ET5 - IIM

Mars 2024

Rapport de Traitement Automatique des Langues

Objectif du projet :

On souhaite faire une analyse et évaluer les performances du POS tagging et détection d'entités nommées de NLTK, avec ceux du CoreNLP Java de Stanford. On va ainsi regarder la précision et le rappel de l'étiquetage des tags que donnent NLTK et Stanford, en comparant à un fichier de référence contenant des tags étiquetés par un être humain (on part du postulat que ce fichier de référence a été correctement étiqueté).

Résultats du projet :

```
python evaluate.py pos_test.txt.pos.nltk.univ pos_reference.txt.univ
Warning: the reference and the candidate consists of different number of lines!
```
Word precision: 0.009033154655548938
Word recall: 0.008360128617363344
Tag precision: 0.009033154655548938
Tag recall: 0.008360128617363344
Word F-measure: 0.008683620401736723
Tag F-measure: 0.008683620401736723

```
python evaluate.py pos_test.txt.pos.stanford.univ pos_reference.txt.univ
Warning: the reference and the candidate consists of different number of lines!
```
Word precision: 0.009231685527099465
Word recall: 0.008543867707854846
Tag precision: 0.009231685527099465
Tag recall: 0.008543867707854846
Word F-measure: 0.008874469201774893
Tag F-measure: 0.008874469201774893

```
python evaluate.py ne_reference.txt.conll ne_test.txt.ne.nltk
```
Tag precision: 0.8813071116408423
Tag recall: 0.8813071116408423
Tag F-measure: 0.8813071116408423

```
python evaluate.py ne_reference.txt.conll ne_test.txt.ne.stanford
```
Tag precision: 0.894020659515296
Tag recall: 0.894020659515296
Tag F-measure: 0.894020659515296

COLLIN
Khévin
ET5 - IIM

Mars 2024

Points forts et limitation des deux plateformes

POS Tagging :

Il y a un problème de groupement de mots dans pos_reference, ces mots ne sont pas groupés par nltk et stanford, ce qui décale les mots suivants et vient donner un score de précision médiocre.
On pourrait essayer de grouper les mots, les entités nommées (comme Pierre Vinken) ou groupes de mots (comme 61 years old) avant de faire le POS Tagging.
On ne peut malheureusement pas faire un groupement de mots correct, en effet pour prendre l'exemple des deux premières phrases, "Pierre Vinken" est identifié par le fichier référence comme un groupe de mots, alors que "Mr. Vinken" est reconnu comme deux mots distincts (voir image du fichier de référence ci-dessous).

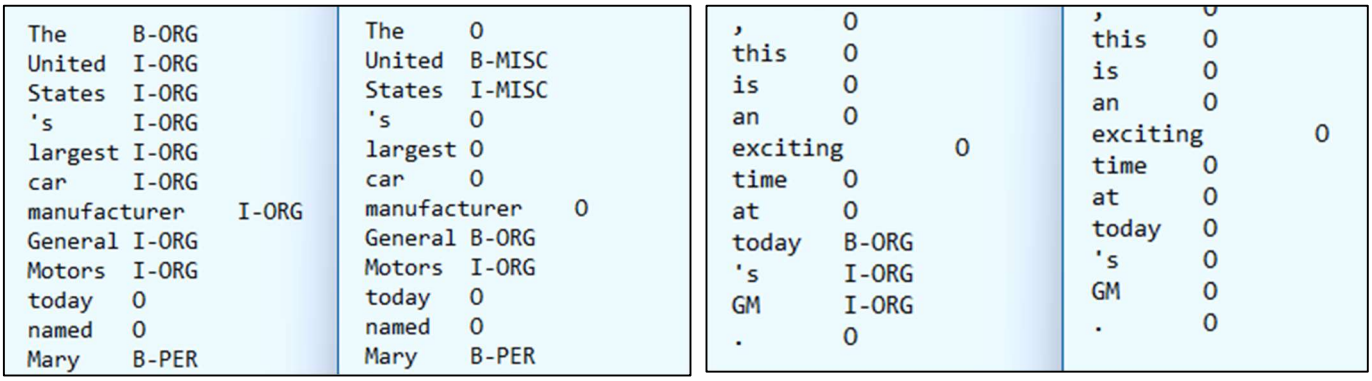
Pourtant, "Pierre Vinken" ainsi que "Mr. Vinken" sont tous deux reconnus par NLTK et Stanford, comme étant <NNP><NNP>, ce qui nous empêche d'écrire une grammaire faisant une distinction pour ces deux cas. L'évaluation nous donne alors une précision et un rappel de moins de 1% pour les deux systèmes de POS Tagging.
On pourrait essayer dans un futur projet de changer le fichier evaluate.py de sorte à ce qu'il ignore les groupes de mots en les comptant comme tags incorrects et en avançant dans l'autre fichier du nombre de mots de l'entité avec plusieurs mots.
NLTK et Stanford ont donc tous les deux la même limite quant au POS Tagging. La seule différence marquante concernant les deux plateformes est le temps nécessaire à l'étiquetage des mots : NLTK le fait en 4 secondes, tandis que Stanford prend 17 minutes, soit une performance 250 fois moins rapide.
On rajoute par ailleurs un affichage console de l'avancée de la tokenization de Stanford pour que l'utilisateur voie où en est le programme et qu'il soit rassuré que ça n'a pas planté en cours d'éxecution.

COLLIN
Khévin
ET5 - IIM

Mars 2024

Reconnaissance d'entités nommées

La reconnaissance des entités nommées va souffrir du même problème que le POS Tagging qui est la séparation d'un même tokens en plusieurs sous-tokens. On va néanmoins être capables de lister les sources de séparation de tokens et régler manuellement ces erreurs au moment de l'écriture des tags. On parvient tout de même à obtenir le même nombre de lignes, avec presque les mêmes mots au début (certains guillemets sont remplacés par d'autre, on saute également les sous-tokens pour ne laisser que le premier afin de ne pas décaler les lignes) et l'évaluation est alors possible pour NLTK et Stanford.
On remarque alors plusieurs soucis avec NLTK : certaines entités qui englobent des entités, ne sont pas reconnues et seules les entités de plus bas niveau sont reconnues. Parfois ce sont des entités qui sont indiquées comme "Others" alors qu'il s'agit d'entités nommées (General Motors et Akerson) et NLTK va même séparer General Motors en 2 entités différentes...



The	B-ORG		The	O
United	I-ORG		United	B-MISC
States	I-ORG		States	I-MISC
's	I-ORG		's	O
largest	I-ORG		largest	O
car	I-ORG		car	O
manufacturer	I-ORG		manufacturer	O
General	I-ORG		General	B-ORG
Motors	I-ORG		Motors	I-ORG
today	O		today	O
named	O		named	O
Mary	B-PER		Mary	B-PER

,	O		,	O
this	O		this	O
is	O		is	O
an	O		an	O
exciting	O		exciting	O
time	O		time	O
at	O		at	O
today	B-ORG		today	O
's	I-ORG		's	O
GM	I-ORG		GM	O
.	O		.	O

General	O		General	B-ORG
Motors	O		Motors	B-PER
recovered	O		recovered	O
from	O		from	O
their	O		their	O
bankruptcy	O		bankruptcy	O
a	O		a	O
year	O		year	O
after	O		after	O
the	O		the	O
appointment	O		appointment	O
of	O		of	O
Akerson	O		Akerson	B-PER

Sur ces 3 images, le fichier de référence ne_reference.txt.conll est situé à gauche, et le fichier de sortie de NLTK est à droite

En ce qui concerne Stanford, on observe le même type d'erreurs mais pas forcément au même endroit. Stanford en revanche ne sépare pas en 2 les entités (par exemple General Motors) comme l'a fait NLTK. Stanford a malheureusement comme inconvénient d'être ici aussi 200 fois moins rapide que NLTK.

On regarde également le rapprochement entre les résultats de NLTK et Stanford avec evaluate.py et on trouve une précision et un rappel de 93%, ce qui nous montre que les deux plateformes vont avoir

raison souvent aux mêmes endroits et se tromper souvent sur les mêmes entités nommées.
En rassemblant tous ces critères, on conclut que Stanford apporte un résultat dans l'ensemble un petit peu plus correct, de l'ordre de 1% de plus, mais que sa lenteur est un handicap considérable dans le monde de l'informatique d'aujourd'hui. De plus, en considérant que les deux plateformes font nécessairement des erreurs, on ne peut pas dire que la peu d'erreur supplémentaire de NLTK le rend inutilisable car Stanford fait quasiment le même nombre d'erreur.
NLTK est donc la plateforme qui sera la plus utile dans la majorité des cas.

Contributeur : Khévin Collin