



Projet EIT

Etude des différences entre CEA List LIMA et Stanford CoreNLP

-

Enseignant : Nasredine SEMMAR

Membres du groupe : Adrien LAVILLONNIERE
Corentin MANSCOUR
Hien Minh NGUYEN

Rédigé le : 09/03/2019

Nombre de pages : 11

TABLE DES MATIÈRES

| | | |
|-------|---------------------------------------------------------------------------------------------------|----|
| 1 | INTRODUCTION | 2 |
| 2 | ETAT DE L'ART | 3 |
| 2.1 | CEA LIST LIMA | 3 |
| 2.2 | Stanford CoreNLP | 3 |
| 2.3 | NLTK | 4 |
| 2.4 | spaCy | 4 |
| 2.5 | Pourquoi LIMA et CoreNLP? | 4 |
| 3 | RÉSULTATS DE LA COMPARAISON | 5 |
| 3.1 | Parties I et II : LIMA | 5 |
| 3.2 | Partie III : Evaluation de l'outil de désambiguïsation morpho-syntaxique de Stanford | 6 |
| 3.3 | Partie IV : Evaluation de l'outil de reconnaissance d'entités nommées de Stanford | 7 |
| 3.4 | Partie V : Evaluation de l'outil de reconnaissance d'entités nommées de Stanford | 7 |
| 3.4.1 | Le pré-traitement des entités | 7 |
| 3.4.2 | Le corpus de référence | 8 |
| 4 | CONCLUSION | 9 |
| 4.1 | Problèmes rencontrés | 9 |
| 4.2 | Conclusion | 9 |
| 4.3 | Répartition des tâches | 10 |
| | BIBLIOGRAPHIE | 11 |

1

INTRODUCTION

CEA List LIMA et Stanford CoreNLP sont deux plateformes d'analyse linguistiques utilisées pour étudier des textes écrits par des humains. Grâce à leurs nombreux outils, il est par exemple possible d'apposer un tag à tous les mots d'un texte afin de savoir leur nature grammaticale ou encore déterminer la structure d'une phrase.

Le but de ce projet est d'étudier ces plateformes d'analyse linguistique par l'expérimentation. Afin de comparer les plateformes, nous devons utiliser des scripts Python convertissant les étiquettes propres aux deux plateformes en étiquettes universelles. Pour cela, on dispose d'une table de correspondance entre les étiquettes Penn TreeBank et les étiquettes universelles (nous devons créer notre propre table de correspondance pour la plateforme de Stanford).

Enfin, nous disposons également d'un script d'évaluation, `evaluate.py`, qui nous permet de comparer les fichiers générés par les plateformes à des fichiers de référence. L'évaluation se présente sous la forme de 6 résultats : word precision, word recall, tag precision, tag recall, word F-measure et tag F-measure. Cela nous permet d'obtenir des informations sur la segmentation (les résultats pour word) ou la désambiguïsation morpho-syntaxique (les résultats pour tag) des plateformes.

En lançant les analyseurs sur les mêmes étiquettes, nous pourrions comparer leur performance, leur précision et leur taux de réussite.

A travers plusieurs analyses de résultats, nous évaluerons et comparerons leur approches, afin de mieux comprendre ce qu'ils sont capables de faire et leur performance respective. Nous pourrions éventuellement proposer des solutions pour les améliorer.

Le code source du projet ainsi que certains résultats sont disponibles ici : <https://github.com/minh-n/EIT>.

2 | ETAT DE L'ART

Il existe plusieurs bibliothèques permettant l'analyse linguistique. Voici quelques-unes d'entre elles.

2.1 CEA LIST LIMA

Article de présentation de LIMA : [Besançon et al., 2010](#)¹

CEA Lima (List Multilingual Analyzer) est une plateforme d'analyse conçue par le CEA List. Lima a été développé avec 4 objectifs :

- supporter un nombre élevé de langues,
- supporter une grande diversité d'applications,
- avoir suffisamment d'extensibilité pour pouvoir ajouter de nouvelles fonctionnalités,
- pouvoir être utilisé dans des contextes requérant de l'efficacité et de la performance.

Les outils du CEA List utilisent un système de règles élaborées par des linguistes.

Cette approche a pour avantage de ne pas avoir à utiliser un corpus annoté de textes, ce qui peut éviter à l'équipe de développement une période fastidieuse d'annotation à la main. De plus, une grammaire fixée peut également éviter d'obtenir des cas imprévus qui, dans le cas du machine learning, pourraient être causés par des ambiguïtés du corpus ou des problèmes au niveau de l'algorithme d'apprentissage.

Malgré tout, cette méthode reste coûteuse en effort car il faut faire appel à des spécialistes qualifiés pour établir ces règles.

2.2 STANFORD CORENLP

Article de présentation de CoreNLP : [Manning et al., 2014](#)²

Stanford CoreNLP est une bibliothèque Java développée par l'université de Stanford, regroupant un ensemble d'outils destinés à l'analyse linguistique. Elle est également utilisable sous Python et est réputée pour sa vitesse.

Cette plateforme présente de nombreuses fonctionnalités comme l'analyse du sentiment d'un texte, la reconnaissance des entités nommées et le part-of-speech tagging.

1. Lien direct : http://www.lrec-conf.org/proceedings/lrec2010/pdf/537_Paper.pdf

2. Lien direct : <http://aclweb.org/anthology/P14-5010>

Elle supporte plus de 6 langues dont le français. Comme les autres librairies de NLP, il est simple à utiliser et nécessite un nombre minimal de lignes de code pour obtenir des résultats.

La librairie comprend un grand nombre d'outils différents utilisant des règles, des probabilités ou du deep learning pour leur analyses. L'approche par machine learning de Stanford est intéressante : au lieu d'avoir à écrire une multitude de règles, le système peut s'entraîner lui-même sur un corpus de texte annoté. Il peut ainsi établir des règles statistiques par lui-même avec relativement moins d'intervention humaine. Cela permet un gain de temps considérable par rapport à l'écriture de règles précises (l'annotation de texte est un processus relativement facile, qui pourrait être effectué par des non-spécialistes).

Il existe bien sûr des inconvénients à cette méthode : la quantité de texte du corpus annoté a une conséquence directe : le modèle produit peut se retrouver en situation de sous ou sur-apprentissage et donner des résultats incohérents.

2.3 NLTK

C'est une librairie en Python, très répandue dans le monde de la recherche et de l'éducation. Elle sert essentiellement au preprocessing d'un texte, à sa tokenisation et à son étiquetage morpho-syntaxique. Dû à son âge, elle présente des performances inférieures aux outils industriels. Il est basé sur une méthode de reconnaissance statistique.

2.4 SPACY

Un outil industriel compatible avec les librairies de machine learning comme PyTorch ou scikit-learn. L'un de ses atouts principaux est la vitesse d'exécution. Il est basé sur une méthode de reconnaissance statistique.

2.5 POURQUOI LIMA ET CORENLP ?

Le choix de comparer ces deux plateformes a été fait pour différentes raisons. Tout d'abord, ces deux plateformes sont gratuites et disponibles en open source sur internet. De plus, LIMA est développé au CEA par des chercheurs nous faisant cours. Il est donc intéressant de s'y pencher. La documentation qui entoure les deux plateformes est également développée et nous permettra de résoudre d'éventuels problèmes techniques que nous rencontrerons.

Ensuite, les deux plateformes possèdent des approches différentes : les outils de Stanford utilisent principalement de l'apprentissage automatique, alors que les outils du CEA List sont basés sur des règles. Cette différence sera intéressante à comparer lorsque nous analyserons les résultats.

3 | RÉSULTATS DE LA COMPARAISON

3.1 PARTIES I ET II : LIMA

Afin d'interpréter les résultats produits par LIMA concernant l'extractions d'entités nommées et l'analyse morpho-syntaxique, nous devons élaborer plusieurs scripts.

Pour l'analyse morpho-syntaxique, il est nécessaire de modifier l'affichage des étiquettes du fichier de sortie donné par LIMA en un texte de la forme : "When_WRB it_PRP 's_VBZ ..." A la base, les étiquettes produites par LIMA respectent la syntaxe Penn TreeBank (PTB). On peut donc se lancer dans une première évaluation de l'outil en utilisant le fichier de référence correspondant aux étiquettes PTB. On obtient les résultats suivants :

- Word precision : 0.877192982456
- Word recall : 0.909090909091
- Tag precision : 0.640350877193
- Tag recall : 0.663636363636
- Word F-measure : 0.892857142857
- Tag F-measure : 0.651785714286

Pour évaluer l'analyse morpho-syntaxique sur les étiquettes universelles, il faut convertir les étiquettes produites nativement par LIMA afin d'utiliser l'outil d'évaluation fourni. Afin de transformer les étiquettes PTB en étiquettes universelles, il faut tout d'abord remplacer les étiquettes LIMA par leurs équivalents PTB suivants :

- SCONJ -> CC
- SENT -> .
- COMMA -> ,
- COLON -> :

LIMA considère les noms propres de plusieurs mots comme un seul et unique token. Afin que la conversion se fasse bien et que le parsing du fichier à convertir soit cohérent, nous pouvons avoir recours à l'ajout temporaire d'un mot-clé, "Espace", entre toutes les sous-parties de ce nom propre. Par exemple, "Boca Raton" deviendra "BocaEspaceRaton". Ensuite, nous pouvons lancer le script `evaluate.py`. Nous obtenons les résultats suivants :

- Word precision : 0.877192982456
- Word recall : 0.909090909091
- Tag precision : 0.745614035088
- Tag recall : 0.772727272727
- Word F-measure : 0.892857142857
- Tag F-measure : 0.758928571429

Nous remarquons que toutes les valeurs liées à Word restent identiques. Cela est logique, puisque la segmentation est réalisée de manière identique dans les deux cas. Nous remarquons également que les valeurs liées à Tag sont plus hautes dans le cas des étiquettes universelles. En observant la table des correspondances, beaucoup de tags PTB différents sont réunis sous un même tag universel. Les tags universels étant de ce fait moins complexes, il est normal d'obtenir plus de correspondance d'une méthode à l'autre.

D'autre part, certains de nos scripts n'étant pas fonctionnels, soit à cause de nos compétences en Python, soit à cause d'inconsistences dans les fichiers générés, des modifications ont dû être faites à la main et non de manière automatisées.

3.2 PARTIE III : EVALUATION DE L'OUTIL DE DÉSAMBIGÜISATION MORPHO-SYNTAXIQUE DE STANFORD

Afin d'évaluer les résultats de l'outil de désambiguïsation morpho-syntaxique, nous pouvons passer par 3 étapes.

Tout d'abord, il faut lancer l'outil POS tagger sur le fichier sur lequel on veut effectuer les tests : `wsj_0010_sample.txt`. Le POS tagger produit alors un fichier contenant le texte originel taggé avec des étiquettes Penn Treebank (PTB). Nous obtenons les résultats suivants pour ce fichier PTB lorsque comparé avec le fichier de référence.

- Word precision : 0.746478873239
- Word recall : 0.481818181818
- Tag precision : 0.732394366197
- Tag recall : 0.472727272727
- Word F-measure : 0.585635359116
- Tag F-measure : 0.574585635359

Nous pouvons remarquer que le rappel (soit le pourcentage d'éléments pertinents sélectionnés) est considérablement inférieur à la précision (le nombre d'éléments pertinents sur le total des candidats sélectionné). Cela pourrait être expliqué par le fait que les étiquettes PTB sont plus précises et entraîneraient l'élimination de certains mots pertinents.

Nous pouvons ensuite faire passer les étiquettes PTB en étiquettes Universelles. En lançant l'évaluation, nous obtenons les résultats suivants :

- Word precision : 0.990909090909
- Word recall : 0.990909090909
- Tag precision : 0.972727272727
- Tag recall : 0.972727272727
- Word F-measure : 0.990909090909
- Tag F-measure : 0.972727272727

Cette disparité peut s'expliquer par le fait que les étiquettes PTB sont plus complexes que les étiquettes universelles. Nous pouvons donc dire que Stanford est plus performant en utilisant des étiquettes universelles qu'en utilisant des étiquettes PTB.

3.3 PARTIE IV : EVALUATION DE L'OUTIL DE RECONNAISSANCE D'ENTITÉS NOMMÉES DE STANFORD

Pour cette partie, nous avons créé un script permettant de représenter les données sous le format demandé (Entité nommée - Type - Nombre d'occurrences - Proportion dans le texte). Il n'y a pas grand chose à remarquer à part que la reconnaissance d'entités nommées de Stanford est fonctionnelle et qu'elle est similaire à celle de LIMA, avec moins de complexité.

3.4 PARTIE V : EVALUATION DE L'OUTIL DE RECONNAISSANCE D'ENTITÉS NOMMÉES DE STANFORD

Dans cette partie, nous devons évaluer la reconnaissance d'entités nommées des deux plateformes.

3.4.1 Le pré-traitement des entités

Pour cela, il faut d'abord établir une correspondance entre les étiquettes des entités nommées des deux outils (table 1).

| Lima | Stanford |
|---------------------------|-----------------|
| Organization.ORGANIZATION | ORGANIZATION |
| Location.LOCATION | LOCATION |
| Person.PERSON | PERSON |
| – | /O |
| DateTime.TIME | /O |
| DateTime.DATE | /O |
| Numex.NUMEX | /O |
| Numex.NUMBER | /O |

TABLE 1 – Tableau de correspondance entre les étiquettes des entités nommées LIMA et celles de Stanford

Nous devons ensuite transformer les outputs de LIMA pour qu'ils suivent la même syntaxe que ceux de Stanford, grâce à la table de correspondance ci-dessus.

Ensuite, il faudra convertir les étiquettes des entités nommées type Stanford en syntaxe universelle (transformation des /O en _O...), afin que le script `evaluate.py` puisse les lire. Cette fois, il n'y a pas besoin d'ajouter un mot-clé Espace pour les noms propres composés de plusieurs mots, comme pour le cas de LIMA : Stanford prend en effet en compte ce cas, "Boca Raton" devient donc "Boca_LOCATION Raton_LOCATION". Dans notre cas, nous utilisons le même script que pour transformer les étiquettes LIMA, avec la différence que nous enlevons cette fois le mot-clé Espace.

3.4.2 Le corpus de référence

Etant donné que notre script `evaluate.py` ne peut prendre en compte que des fichiers ayant des étiquettes universelles ou PTB, il faut transformer le fichier du corpus de référence, sous standard d'étiquetage ENAMEX.

Malheureusement, nous n'avons pas réussi à transformer les étiquettes ENAMEX pour obtenir un fichier exploitable. Notre script de conversion ne permet en effet d'obtenir que les informations contenues dans les balises. Le reste est perdu. Le fichier en sortie est donc incomplet et ne nous permet pas de faire la comparaison.

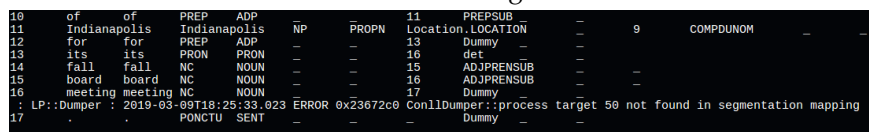
4 | CONCLUSION

4.1 PROBLÈMES RENCONTRÉS

Lors de ce projet, nous avons remarqués que les fichiers n'étaient pas parfaits : présence de] et \$ dans certains fichiers .pos.ref

LIMA étant encore une plateforme en développement, des erreurs subsistent : par exemple, lors de la création des fichiers .conll, des erreurs ConllDumper peuvent apparaître. Nous n'avons malheureusement pas trouvé l'origine de cette erreur.

FIGURE 1 – Erreur rencontrée lors de la génération d'un fichier .conll



| | | | | | | | | | | | |
|-------------------------------------------------------------------------------------------------------------------------|--------------|--------------|--------|------|-------|--|----|-------------------|--|---|-----------|
| 10 | of | of | PREP | ADP | | | 11 | PREPSUB | | | |
| 11 | Indianapolis | Indianapolis | | NP | PROPN | | 12 | Location.LOCATION | | 9 | COMPDUNOM |
| 12 | for | for | PREP | ADP | | | 13 | Dummy | | | |
| 13 | its | its | PRON | PRON | | | 14 | det | | | |
| 14 | fall | fall | NC | NOUN | | | 15 | ADJPRESUB | | | |
| 15 | board | board | NC | NOUN | | | 16 | ADJPRESUB | | | |
| 16 | meeting | meeting | NC | NOUN | | | 17 | Dummy | | | |
| ; LP::Dumper : 2019-03-09T18:25:33.023 ERROR 0x23672c0 ConllDumper::process target 50 not found in segmentation mapping | | | | | | | | | | | |
| 17 | . | . | PONCTU | SENT | | | | Dummy | | | |

Afin de comparer les deux outils, il a fallu passer par plusieurs étapes de transformations des étiquettes. Nous n'avons pas eu le temps de compléter tous les scripts nécessaires à cette transformation : malheureusement, certaines lacunes subsistent. Ces scripts ne sont pas parfaits et pourraient poser problème sur un corpus plus grand.

4.2 CONCLUSION

Ce projet nous a permis d'en apprendre plus sur le fonctionnement de plateformes d'analyse linguistique, et plus particulièrement sur l'évaluation de la segmentation (les résultats liés à word) et la désambiguïsation morpho-syntaxique (les résultats liés à tag).

Malgré leur objectif commun, nous remarquons que LIMA et Stanford présente des différences, essentiellement dans leur standards d'étiquetage. Lorsque nous passons leurs étiquettes en étiquettes universelles, la précision et le rappel pour la segmentation ou la désambiguïsation morpho-syntaxique se rapproche. Stanford présente de faibles résultats lorsque les étiquettes universelles ne sont pas utilisées. LIMA possède donc un clair avantage.

Le point principal qui différencie les deux plateformes est la gestion des entités dans le cas des noms propres. Prenons par exemple "Boca Raton" : LIMA le reconnaît comme une seule entité (donc un seul token) comme un humain normal pourrait le faire. Stanford, au contraire, le reconnaît comme deux entités. LIMA est donc objectivement un meilleur outil en ce qui concerne la tokenization. Cela pourrait

faire la différence lors de la désambiguïsation morpho-syntaxique par rapport aux résultats de Stanford.

La différence est sans doute causée par l’approche des deux plateformes : Stanford utilise en effet une approche statistique et LIMA une approche par règles. Avec une étude approfondie, nous pourrions comparer plus de points et révéler plus de différences.

4.3 RÉPARTITION DES TÂCHES

Dans ce projet, nous avons tous essayé de toucher aux différentes parties. Nous avons en effet beaucoup réfléchi ensembles pour l’interprétation des résultats. Les rôles principaux sont les suivants :

- Adrien LAVILLONNIERE : reconnaissance d’entités nommées, rédaction des scripts permettant la transformation des étiquettes des entités nommées,
- Corentin MANSCOUR : désambiguïsation de Stanford, rédaction de divers scripts LIMA, rédaction du rapport,
- Minh NGUYEN : rédaction des scripts d’analyse morpho-syntaxique de LIMA, rédactions d’autres scripts de conversion d’étiquetage, rédaction du rapport.

BIBLIOGRAPHIE

Besançon, Romaric, Gaël de Chalendar, Olivier Ferret, Faiza Gara, Olivier Mesnard, Meriama Laïb et Nasredine Semmar

- 2010 « LIMA : A Multilingual Framework for Linguistic Analysis and Linguistic Resources Development and Evaluation », anglais, in *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, sous la dir. de Nicoletta Calzolari (Conference Chair), Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner et Daniel Tapias, European Language Resources Association (ELRA), Valletta, Malta, ISBN : 2-9517408-6-7.

Manning, Christopher, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard et David McClosky

- 2014 « The Stanford CoreNLP Natural Language Processing Toolkit », in *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics : System Demonstrations*, Association for Computational Linguistics, Baltimore, Maryland, p. 55-60, DOI : [10.3115/v1/P14-5010](https://doi.org/10.3115/v1/P14-5010), <http://aclweb.org/anthology/P14-5010>.