FEURTE Morgan
GONCALVES Jeffrey
VOLLMER Florent
ET5, Informatique

Rapport - Projet EIT

Evaluation d'outils de TAL

Année : 2018 - 2019

Enseignant : Nasredine SEMMAR

Polytech Paris Sud
Maison de l'Ingénieur - Bât 620 - Centre Scientifique d'Orsay
91405 Orsay - France

Table des matières

Introduction

L'objectif de ce projet est d'étudier deux outils d'analyse linguistique. Ces deux outils en question sont CEA List LIMA et Stanford Core NLP. Ces outils sont basés sur des procédés différents : CEA List LIMA est une plateforme d'analyse linguistique multilingue utilisant uniquement des règles formelles et des ressources validées par des experts linguistes, tandis que Stanford Core NLP peut aussi se baser sur l'apprentissage statistique à partir de corpus annotés.

Malgré leurs différences dans le procédé, ces deux outils suivent les mêmes étapes pour donner leurs résultats. En effet, les modules suivants sont présents chez les deux outils :

- Une étape de tokenisation pour isoler chaque mot présent dans les chaînes de caractères
- Une analyse morphologique pour vérifier si chaque mot analysé (token) appartient bien à la langue et ainsi lui associer des propriétés syntaxiques et un lexeme (ou champs lexical) qui vont servir dans la suite du traitement.
- Une analyse morpho-syntaxique pour réduire le nombre d'ambiguïtés concernant certains mots d'un point de vue grammatical, qui restent présentes à la suite de l'analyse morphologique
- Une analyse syntaxique afin d'identifier les relations entre les différents mots
- POS (Part-Of-Speech) tagger : Il s'agit de l'analyse grammaticale de texte, qui se base sur les étapes précédentes pour associer un POS à chaque token en fonction de la nature du mot et du contexte.
- Une reconnaissance d'entités nommées pour identifier différents entités reconnaissables (dates, lieux, expressions numériques, organisations, etc) sur un ou plusieurs tokens et ainsi les remplacer par un unique token

Nous rappelons évidemment que ces modules, bien que présents chez les deux outils, n'utilisent pas les mêmes données : règles formelles pour LIMA, statistiques pour Stenford. Aussi, pour chacune des plateformes, il faudra d'abord lancer l'analyse linguistique sur les phrases afin d'en retirer l'analyse fournie par chaque outil sous forme de matrices.

Le but de notre étude sera de d'évaluer et de comparer LIMA et Stanford. Pour cela, nous avons codé un programme permettant de ressortir pour chaque phrase son POS tag (ou son entité nommée) associé, après avoir récupéré l'output brut de nos plateformes. Cependant, nous avons besoin d'un point de comparaison universel pour que celle-ci soit pertinente : c'est ici que les tags universels interviennent. Nous pouvons alors remplacer les tags propres à chaque plateforme par ces tags universels et ainsi réaliser une comparaison pertinente. Enfin, nous évaluerons aussi la performance de chaque plateforme grâce à des fichiers contenant des mots associés à des tags universels comme expliqué précédemment.

État de l’art

Les différentes approches d’analyse linguistique

Il y a plusieurs approches d’analyse linguistique. La première d’entre elles est basée sur les règles. Une approche par une grammaire basée sur des règles est basiquement un système de règles “fait à la main” basé sur des structures linguistiques qui imitent la manière humaine de créer des structures grammaticales. Cela implique une intervention humaine dans le développement et l’amélioration d’un système progressif.
Les plus grands avantages de cette approche sont qu’il y a toujours un moyen de vérifier si le système peut répondre à une requête d’un utilisateur ou non (et comment il peut y répondre), que les règles de grammaire sont développées de manière vraiment flexible et qu’une telle approche ne nécessite pas un important corpus d'entraînement comme pour l’approche par machine learning.
À l’inverse, cette comporte certains point négatifs. En effet, une telle approche nécessite un expert linguistique pour définir les règles, ces règles doivent être créées et maintenues manuellement et peuvent parfois se contredire.

La seconde approche est basée sur le machine learning. Elle est basée sur des algorithmes qui “apprennent à comprendre” un langage sans que ce langage soit explicitement programmé. Cela est possible par l’utilisation des statistiques (comme pour Stanford), où le système commence par analyser un corpus d'entraînement (corpus annoté) pour construire son propre “savoir” et produire ses propres règles et ses propres classifications.
Les principaux avantages d’une telle approche se trouvent dans son automatisation et sa performance. En effet, une approche par machine learning ne nécessite pas d’intervention humaine important comme pour l’approche avec des règles grammaticales et sa performance est généralement meilleure que les autres types d’approches.
Cependant, le développement d’algorithmes de machine learning est assez complexe dans la pratique et un manque d’apprentissage du système peut engendrer une grande différence dans la précision des résultats.

Enfin, il existe une approche hybride qui combine les deux approches décrites précédemment. Toutefois, cette approche est peu présente donc il est encore assez difficile de l’évaluer précisément.

Choix des plateformes

Il existe plusieurs motivations pour lesquelles nous avons choisi LIMA et Stanford. La première motivation est le fait que LIMA et Stanford sont des plateformes gratuites et open-sources, contrairement à certaines plateformes de Microsoft, Google, Amazon ou IBM. La seconde motivation est le fait qu'il est intéressant de comparer ces deux plateformes, puisqu'elles suivent des méthodologies d'analyse linguistique différents (règles pour LIMA et apprentissage statistique pour Stanford). Enfin, certaines plateformes plus anciennes comme NLTK ont des limites importantes de performances, ce qui nous a poussé à ne pas les considérer.

Le fait que LIMA soit encore en développement et que nous soyons relativement proches de l'équipe s'en occupant est naturellement aussi une des raisons pour lesquelles son étude est intéressante, puisque nous pouvons avoir plus facilement avoir des explications supplémentaires et donner un feed-back de notre expérience. Par opposition, CoreNLP est un outil se présentant comme plus abouti, et peut servir de référence.

Descriptions des analyseurs linguistiques

Bien qu'étant similaires dans les grandes lignes, CoreNLP et LIMA ont chacunes leurs particularités, dont cette partie a vocation à présenter les plus intéressants.

La pipeline de LIMA a quelques particularités :

- La **tokenization** de LIMA est "absolue", ce qui veut dire qu'elle ne traite pas l'ambiguïté des tokens générés. Ceux-ci peuvent en revanche être fusionnés ou éclatés lors des étapes suivantes.
- La **POS** est divisée en quatre étapes :
    - Le **dictionary check**, qui cherche dans un dictionnaire des mots correspondant aux tokens traités pour déterminer leur lexeme.
    - Le **hyphenated words**, qui va s'occuper des mots qui n'ont pas été trouvés tels quels dans le dictionnaire en y cherchant plutôt leurs sous-éléments (radicaux, mot composés…). Cela est particulièrement utile pour le traitement des langues comme l'allemand et le hongrois, qui comportent de nombreux mots composés à partir d'autres.
    - La **reconnaissance d'entités spécifiques** : c'est ce qui correspond à la reconnaissance d'entités nommées décrite en introduction
    - Le **POS-tagging** à proprement parler : il utilise l'algorithme "Viterbi".

Celle de Core NLP quand-à-elle, présente les caractéristiques suivantes :

- L'étape **cleanxml** retire les tags d'un document, permettant de traiter efficacement des documents au format xml.
- **ssplit** sépare la suite de tokens en phrases.
- **truecase** rétablit la casse (estimée) correcte d'un mot.
- La **POS** utilise un algorithme d'entropie maximale.
- L'analyse morphologique est séparée en deux sous-étape : **lemma** et **gender**, qui reconnaissent, comme leurs noms l'indique, le lemme et le genre des tokens.
- La reconnaissance d'entités nommées se fait d'abord par **ner**, qui se base sur l'ensemble des entités issues des corpus d'apprentissage, et **regexner**, qui est basée sur des règles pour trouver les entités nommées restantes.
- **sentiment** peut analyser les sentiments présents dans une séquence.
- **dcoref** permet d'associer des mots, comme les pronoms, aux tokens auxquels ils réfèrent, générant ainsi un "graphe de coréférence".

Description des expérimentations

Dans cette partie, tous les résultats proviennent de l'implémentation de plusieurs scripts en Python trouvables à l'adresse suivante.

CEA List LIMA

Afin de pouvoir analyser les résultats de LIMA en rapport avec l'extraction d'entités nommées et avec l'analyse morpho-syntaxique de manière pertinente, nous devons tout d'abord implémenter quelques scripts.

Dans le cas de l'analyse morpho-syntaxique, on souhaite modifier l'affichage des étiquettes du fichier de sortie de LIMA en un texte de la forme : "When_WRB it_PRP 's_VBZ ...". Les étiquettes produites par LIMA respectant la syntaxe Penn TreeBank (PTB), nous pouvons lancer une première analyse de la plateforme en comparant ces étiquettes à celles du fichier de référence correspondant aux étiquettes PTB. Nous obtenons les résultats suivants :

Metrics	Number	Readable number
Word precision	0,8771929825	87,72%
Word recall	0,9090909091	90,91%
Tag precision	0,6403508772	64,04%
Tag recall	0,6636363636	66,36%
Word F-measure	0,8928571429	89,29%
Tag F-measure	0,6517857143	65,18%

Pour effectuer la même analyse sur les étiquettes universelles, il faut convertir les étiquettes produites directement par LIMA afin de réaliser l'évaluation. Afin de transformer les étiquettes PTB en étiquettes universelles, il faut dans un premier temps remplacer les étiquettes LIMA par leurs équivalents PTB suivants :

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

De plus, LIMA interprète les noms propres composés comme un seul et unique token. Afin que la conversion se déroule correctement et que la lecture du fichier soit pertinente, nous ajoutons de manière temporaire un mot-clé “Espace” à la place des espaces d’un nom propre composé. Par exemple, le nom propre “Boca Raton” deviendra “BocaEspaceRaton”. Une fois que cela a été réalisé, nous pouvons lancer l’évaluation. Nous obtenons les résultats suivants :

Metrics	Number	Readable number
Word precision	0,8771929825	87,72%
Word recall	0,9090909091	90,91%
Tag precision	0,7105263158	71,05%
Tag recall	0,7363636364	73,64%
Word F-measure	0,8928571429	89,29%
Tag F-measure	0,7232142857	72,32%

Nous remarquons dans un premier temps que les statistiques concernant Word (Word precision et Word recall) restent identiques dans les deux cas. Cela s’explique par le fait que la segmentation est réalisée de la même manière dans les deux cas. De plus, nous observons que les valeurs concernant le Tag sont plus hautes dans le cas des étiquettes universelles. En observant la table des correspondances, on peut observer que plusieurs étiquettes PTB sont réunies sous une seule étiquette universelle. Les tags universels étant donc moins complexes, il est donc logique d’observer plus de correspondances dans ce cas précis.

Stanford Core NLP

Cette partie va être réalisée en 3 étapes : tout d’abord l’évaluation de l’outil de désambiguïsation morpho-syntaxique, puis celle de l’outil de reconnaissance des entités nommées, et enfin celles des outils de reconnaissances d’entités nommées des deux outils, que sont Stanford et LIMA.

Évaluation de l’outil de désambiguïsation

Dans un premier temps, 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 :

Metrics	Number	Readable number
Word precision	0,746478873239	74,65%
Word recall	0,481818181818	48,18%
Tag precision	0,732394366197	73,24%
Tag recall	0,472727272727	47,27%
Word F-measure	0,585635359116	58,56%
Tag F-measure	0,574585635359	57,46%

Nous pouvons remarquer que le recall (soit le pourcentage d'éléments pertinents sélectionnés) est considérablement inférieur à la précision (le nombre d'éléments correspondants 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.

Ensuite, pour le passage des étiquettes PTB en étiquettes universelles, nous obtenons les résultats suivants en lançant l'évaluation :

Metrics	Number	Readable number
Word precision	0,990909090909	99,09%
Word recall	0,990909090909	99,09%
Tag precision	0,972727272727	97,27%
Tag recall	0,972727272727	97,27%
Word F-measure	0,990909090909	99,09%
Tag F-measure	0,972727272727	97,27%

Cette différence flagrante dans nos résultats peut s'expliquer par le fait que les étiquettes universelles étant moins diverses, elles sont donc moins complexes que les étiquettes PTB. On peut donc en déduire que Stenford est beaucoup plus performant lorsqu'il utilise des étiquettes universelles plutôt que des étiquettes PTB.

Évaluation de l'outil de reconnaissance d'entités nommées

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.

Évaluation de l'outil de reconnaissance d'entités nommées des deux plateformes

Dans cette partie, nous allons donc évaluer la capacité de reconnaissance d'entités nommées des deux plateformes que sont Stanford et LIMA.

Pour cela, il faut dans un premier temps réaliser un table de correspondance des étiquettes d'entités nommées entre les deux plateformes (cf tableau ci-dessous).

Lima	**Stanford**
Organization.ORGANIZATION	ORGANIZATION
Location.LOCATION	LOCATION
Person.PERSON	PERSON
_	/O
DateTime.TIME	/O
DateTime.DATE	/O
Numex.NUMEX	/O
Numex.NUMBER	/O

Grâce à cette table, nous pouvons désormais transformer les outputs de LIMA pour qu'ils suivent la même syntaxe que ceux de Stanford.

Ensuite, il faut transformer les étiquettes des entités nommées de Stanford en syntaxe universelle (notamment transformer /O en _O). Dans ce cas, nous n'avons pas besoin de rajouter de mot-clé "Espace" puisque Stanford prend déjà compte de cas : par exemple, "Boca Raton" devient "Boca_LOCATION Raton_LOCATION".

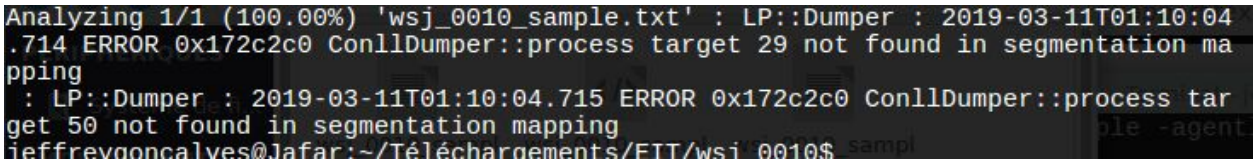
Etant donné que notre script evaluate.py ne peut prendre en compte que des fichiers ayant des étiquettes universelles ou PTB, il faut transformer le fichier du corpus de référence, sous standard d'étiquetage ENAMEX.

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.

Conclusion

Difficultés rencontrées

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. Cependant, comme nous n'écartons aucune hypothèse, on peut se demander si ces erreurs n'ont pas eu une influence sur nos résultats.



Exemple d'erreurs rencontrées

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 ces transformations : malheureusement, certaines lacunes subsistent. Ces scripts ne sont pas parfait et pourraient poser problème sur un corpus plus grand.

Répartition des tâches

Les répartitions des tâches dans notre groupe a été la suivante :

- **Morgan Feurte** : partie LIMA du projet, évaluation de la reconnaissance d'entités nommées des deux plateformes et rédaction du rapport
- **Jeffrey Goncalves** : partie Stenford du projet, évaluation de la reconnaissance d'entités nommées des deux plateformes et rédaction du rapport
- **Florent Vollmer** : partie Stenford du projet, évaluation de la reconnaissance d'entités nommées des deux plateformes et rédaction du rapport

Conclusion

A travers ce projet, nous avons pu en apprendre plus sur le fonctionnement des plateformes d'analyses linguistiques et nous pu réfléchir à leurs points forts et leurs points faibles des différentes approches utilisées.

Malgré leur objectif commun, nous remarquons que LIMA et Stanford présente des différences, essentiellement dans leur standards d'étiquetage et dans leur approches. 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. Cependant, nous remarquons que dans un contexte plus simplifié et généralisé, l'apprentissage statistique de Stanford s'en sort avec une précision proche de 100% là où LIMA est un peu à la traîne.