# Rapport - Traitement Automatique des Langues

## 1. Introduction

Aujourd'hui, on recense 141 langues officielles dans le monde. Dans cette optique, il paraît évident que le domaine du Traitement Automatique des Langues ait émergé dans un monde où l'on est amené à rencontrer de plus en plus de personnes parlant une autre langue.
À l'heure de la mondialisation, la traduction semble être devenue une nécessité.
Le Traitement Automatique des Langues (TAL) se développe de plus en plus grâce aux nouvelles technologies de traitement statistique de l'information (Machine Learning et Deep Learning).

Ainsi, la problématique de ce projet est d'effectuer une comparaison des performances de plateformes d'analyse linguistique disponibles.
Nous allons comparer 2 plateformes :

- **Stanford Core NLP :** une boîte à outils linguistiques utilisant l'apprentissage statistique à partir de corpus annotés.

- **NLTK :** une boîte à outils linguistiques utilisant des approches hybrides combinant l'apprentissage automatique et des ressources linguistiques.

Avant de commencer, il est important de rappeler que l'analyse linguistique standard se compose de ces 5 modules:

- **Découpage (ou Tokenization) :** Ce module consiste à découper les chaînes de caractères du texte en mots (tokens), en prenant en compte le contexte ainsi que les règles de découpage. Ce module utilise généralement des règles de segmentation ainsi que des automates d'états finis.

- **Analyse morphologique :** Ce module a pour but de vérifier si le mot (token) appartient à la langue et d'associer à chaque mot des propriétés syntaxiques qui vont servir dans la suite des traitements. Ces propriétés syntaxiques sont décrites en classes appelées catégories grammaticales. La consultation de dictionnaires de formes ou de lemmes permet de récupérer les propriétés syntaxiques concernant les mots à reconnaître.

- **Analyse morpho-syntaxique :** Après l'analyse morphologique, une partie des mots restent ambigus d'un point de vue grammatical. L'analyse morphosyntaxique réduit le nombre des ambiguïtés en utilisant soit des règles ou des matrices de désambiguïsation. Les règles sont généralement construites manuellement et les matrices de bi-grams et tri-grams sont obtenues à partir d'un corpus étiqueté et désambiguïser manuellement.

- **Analyse syntaxique (Syntactic analysis ou Parsing) :** Ce module consiste à identifier les principaux constituants de la phrase et les relations qu'ils entretiennent entre eux. Le résultat de l'analyse syntaxique peut être une ou plusieurs structures

syntaxiques représentant la phrase en entrées. Ces structures dépendent du formalisme de représentation utilisé : un arbre syntagmatique, un arbre de dépendance ou une structure de traits. L'analyse en dépendance syntaxique consiste à créer un arbre de relations entre les mots de la phrase. Le module d'analyse syntaxique utilise des règles pour l'identification des relations de dépendance ou des corpus annotés en étiquettes morpho-syntaxiques et en relations de dépendance.

- **Reconnaissance d'entités nommées (Named Entity Recognition) :** Ce module consiste à identifier les dates, lieux, heures, expressions numériques, produits, évènements, organisations, présentes sur un ou plusieurs tokens, et à les remplacer par un seul token.

# 2. Présentation des plateformes d'analyse linguistique

## 2.1. Stanford Core NLP

Stanford Core NLP a été créée par Christopher D. Manning, professeur d'informatique à l'université de Stanford, Californie. La première version a été publiée fin 2010.
Il s'agit d'un module (API), utilisable pour la reconnaissance de langue dans des programmes informatiques.
Stanford est composé de plusieurs outils de reconnaissance du langage, tel que un tagger des parties de discours qui assigne à chaque mot un token (comme verbe ou nom), la reconnaissance d'entité nommée, un parseur naturel qui analyse la structure grammatical des phrases et peut identifier le sujet d'une phrase, un solveur de coréférence, un analyseur des sentiments en assignant à chaque mot des points positifs ou négatifs selon la négativité ou positivité du mot, et l'extraction d'information à partir de texte simple, en extrayant des relations entre les mots.

Selon les outils, l'analyse est basée sur des statistiques, le deep learning ou des règles. Stanford supporte actuellement plusieurs langues telles que le français, l'arabe, le chinois, l'anglais, l'allemand et l'espagnol. Il est possible de l'utiliser sur d'autres langues, mais il faut alors développer des modèles de références pour pouvoir utiliser Stanford CoreNLP. Des modèles ont été développés par des personnes extérieures à Stanford pour l'italien, le portugais et le suédois.

Pour l'outil qui nous intéresse ici, le tagger, il est développé en Java et utilise le set de tag Penn Treebank. Il est déterministe, mais utilise aussi des heuristiques pour déterminer si les points sont des fins de phrases, si les guillemets font partie d'un mot, etc....

## 2.2. NLTK

NLTK a été développé par Steven Bird, professeur à l'université de Pennsylvanie, et un de ses anciens étudiants, Edward Loper. La première version a été publiée en 2001. NLTK est un programme open source, tout le monde peut y contribuer en continuant le développement du programme, que ce soit sur son fonctionnement ou en ajoutant un corpus

de référence. Il est possible de n'utiliser que le toolkit de traitement du langage, mais NLTK fournit aussi des démonstrations graphiques, des données-échantillons, des tutoriels, ainsi que la documentation de l'interface de programmation (API).
Le toolkit contient en fait une suite de bibliothèques de traitement de texte pour la classification, la tokenisation, le stemming, le balisage, l'analyse et le raisonnement sémantique.
NLTK combine à la fois l'analyse grâce aux statistiques, mais aussi à base de règles. NLTK peut être intégré à Stanford CoreNLP pour profiter de certaines fonctionnalités de celui-ci.

# 3. Evaluation de l'analyse morpho-syntaxique

L'objectif de ce projet est d'évaluer les plateformes NLTK et Stanford, en particulier sur l'analyse morpho-syntaxique et sur la reconnaissance d'entités nommées. Pour cela nous avons utilisé ces plateforme pour analyser un corpus de textes en anglais (pos_test.txt), composé d'articles courts, généralement dans le domaine de l'économie.
Ce corpus comprend au total 481 phrases et environ 10 000 mots (incluant les ponctuations), avec quelques modifications par rapport au langage naturel : certaines construction grammaticales comme n't sont séparé du mot auquel elles sont associées (par exemple "aren't" devient "are n't").
Le corpus a été analysé séparément par NLTK et stanford, produisant à chaque étape un fichier résultat. L'évaluation se faisait en comparant ces fichiers résultats à un fichier de référence contenant le résultat exact.

Pour l'évaluation de l'analyse morpho-syntaxique, nous avons utilisé 3 métriques :

1. la précision : la proportion des items pertinents parmi l'ensemble des items proposés
   $$precision = \frac{nombre\ de\ prediction\ de\ classe\ i\ correcte}{nombre\ de\ prediction\ de\ classe\ i}$$
2. le rappel : la proportion des items pertinents proposés parmi l'ensemble des items pertinents
   $$rappel = \frac{nombre\ de\ prédiction\ de\ classe\ i\ \ correcte}{nombre\ réel\ d'élément\ de\ classe\ i}$$
3. la F-mesure : une mesure qui combine la précision et le rappel :
   $$Fmesure = 2 \times \frac{precision \times rappel}{precision + rappel}$$

Dans un cas multi-classe comme l'analyse morpho-syntaxique, la précision et le rappel globaux sont la moyenne de la précision et du rappel de chaque classe.

Nous avons remarqué qu'à cause d'une différence de découpage des mots, les fichiers résultats n'avaient pas la même taille selon la plateforme utilisée. Cela pose problème pour l'évaluation quand les fichiers résultat contient un seul mot / token par ligne (format de fichier "en colonne), on obtient alors des résultat extrêmement mauvais (cf tableau de résultat). La solution est d'utiliser un autre format pour les fichiers de résultat (format "en ligne", où tous les mots / token d'une même phrase sont sur la même ligne). Avec ce format, le script pour l'évaluation fonctionne correctement et on obtient alors des résultats bien plus plausibles.

### 3.1. Résultat pour Stanford Core NLP

| | Line file evaluation | Column file evaluation |
|---|---|---|
| Word precision | 0.7050 | 0.0239 |
| Word recall | 0.7089 | 0.0220 |
| Tag precision | 0.6216 | 0.0238 |
| Tag recall | 0.6251 | 0.0220 |
| Word F-mesure | 0.7069 | 0.0229 |
| Tag F-mesure | 0.6234 | 0.0229 |

### 3.2. Résultat pour NLTK

| | Line file evaluation | Column file evaluation |
|---|---|---|
| Word precision | 0.5389 | 0.0255 |
| Word recall | 0.5272 | 0.0236 |
| Tag precision | 0.4734 | 0.0255 |
| Tag recall | 0.4632 | 0.0236 |
| Word F-mesure | 0.5330 | 0.0245 |
| Tag F-mesure | 0.4683 | 0.0245 |

# 4. Evaluation de la reconnaissance d'entités nommées

Après l'analyse morpho-syntaxique, on s'intéresse maintenant à la reconnaissance d'entités nommées. On utilise ici un corpus déjà annoté (ne_reference.txt.conll_update.txt). Ce corpus, toujours en anglais, contient des articles sur les grandes entreprises automobiles des Etats-Unis, avec au total 430 phrases et environ 10 000 mots (avec les ponctuations), soit environ la même taille que le premier corpus.
On applique à ce corpus les NE recognizer de NLTK et Stanford séparément, et on évalue ensuite les résultat en utilisant les mêmes mesures que précédemment : précision, rappel et f-mesure. Comme précédemment, on a un problème pour l'évaluation quand le fichier est au format colonne, donc on le transforme en format ligne pour obtenir le bon résultat (cf tableaux)

### 4.1. Stanford Core NLP

| | Line file evaluation | Column file evaluation |
|---|---|---|
| Word precision | 0.9871 | 0.0202 |
| Word recall | 0.9957 | 0.0202 |
| Tag precision | 0.8961 | 0.0202 |
| Tag recall | 0.9039 | 0.0202 |
| Word F-mesure | 0.9913 | 0.0202 |
| Tag F-mesure | 0.9000 | 0.0202 |

### 4.2. NLTK

| | Line file evaluation | Column file evaluation |
|---|---|---|
| Word precision | 0.9805 | 0.0195 |
| Word recall | 0.9842 | 0.0195 |
| Tag precision | 0.8818 | 0.0195 |
| Tag recall | 0.8851 | 0.0195 |
| Word F-mesure | 0.9823 | 0.0195 |
| Tag F-mesure | 0.8835 | 0.0195 |

## 5. Points forts, limitations et difficultés rencontrées

Au vu du niveau d'analyse où nous avons été, il serait difficile pour nous de critiquer ces plateformes. Ce sont des outils linguistiques qui offrent déjà un bon nombre de modules qui ont été développés et améliorés au fur et à mesure.
On peut néanmoins effectuer un bilan des évaluations des outils POS Tagger et NER.

### 5.1. Analyse morpho-syntaxique

L'analyse morpho syntaxique donne des performances correctes que ce soit pour la plateforme Stanford ou NLTK. On remarque quand même de meilleures performances sur la plateforme Stanford.

On peut notamment expliquer ces performances moyennes de plusieurs manières:

Il y a tout d'abord un troncage des mots lors de l'extraction du corpus de référence. En effet, certains tokens sont composés de plusieurs mots ("$ 212 million" est un token) mais lors de l'extraction du texte, nous ne gardons que le premier mot (soit "$"). Ce qui peut expliquer les différences de tags entre le corpus de référence et stanford/NLTK.

Ensuite, lors de la conversion des tags lima en universels, "QUOTE" est remplacé par ".". De ce fait, même avec la version en ligne du corpus on obtient un nombre différent de lignes au final.

Les plateformes Stanford et NLTK ont des découpages différents du corpus de texte:
Le 's possessif est mal géré par Stanford et NLTK. (today's devient today / 's)
Stanford sépare les mots composés. (seven-day devient seven / - / day)
NLTK ne gère pas bien les mots abrégés. (F.H. devient F.H / .)

### 5.2. Reconnaissance d'entités nommées

De manière générale, les plateformes Stanford et NLTK offrent de bonnes performances pour la reconnaissance d'entités nommées.

La différence entre les tokens semble être dû aux caractères guillemets (" ") utilisés pour les citations. Les différents types de guillemets relevés sont: ' ' (accent aigu) ou `` (accent grave) ou " (guillemets).
Les contractions ne semblent pas être gérées par Stanford et NLTK. (l' m devient l / ' / m)
Comme pour le tagger, Stanford sépare les mots composés. (52-35 devient 52 / - / 35)

Finalement, on remarque que le corpus de référence n'a pas reconnu des entités nommées. Comme exemple, "Akerson", "Canada", "Obama", etc.. n'ont pas été reconnu comme entité nommée alors que Stanford et NLTK les reconnaissent.
Pareil aussi pour "General Motors" ou "GM" qui ne sont pas reconnus tout le temps que ce soit avec Stanford, NLTK ou dans le corpus de référence.

## 6. Organisation

Concernant l'organisation, nous avons tous plus ou moins participé à l'ensemble des tâches car chacun avait besoin de comprendre un minimum les différentes parties de ce projet.
Plus précisément, Nathan s'est plus attardé sur les TPs, Chun plus sur la partie projet, Félix sur le rapport.

## 7. Annexes

Une correction au niveau du corpus de référence "ne_reference.txt.conll_update.txt" a été faite : les lignes vides comprenaient des tabulations ce qui empêchait le bon fonctionnement de notre script, nous avons donc enlevé ces tabulations.