# Rapport du projet de traduction automatique de la langue

## Sommaire

[Rapport du projet de traduction automatique de la langue 1](#_Toc34604061)

[Sommaire 1](#_Toc34604062)

[Objectif du projet 1](#_Toc34604063)

[Evaluation des plateformes d’analyse linguistique 2](#_Toc34604064)

[Analyse morpho-syntaxique 2](#_Toc34604065)

[Reconnaissance d’entités nommées 2](#_Toc34604066)

[Contribution des membres du groupe 2](#_Toc34604067)

[Sullivan Honnet 2](#_Toc34604068)

[Jules Vittone-Burnel 2](#_Toc34604069)

## Objectif du projet

L’objectif affiché de ce projet était d’évaluer les résultats des analyses linguistiques réalisées par divers outils et d’en déduire lesquels étaient les plus efficaces dans quelle situation et éventuellement de proposer des pistes d’améliorations intéressantes pour ces outils.

L’objectif plus profond, abritait par ce projet, était de nous pousser à nous intéresser un domaine que nous rencontrions pour la première fois, celui du traitement automatique des langues et, par extension, la linguistique de manière générale.

## Evaluation des plateformes d’analyse linguistique

### Analyse morpho-syntaxique

Figure 1 : Résultat de l'analyse morpho-syntaxique de stanford

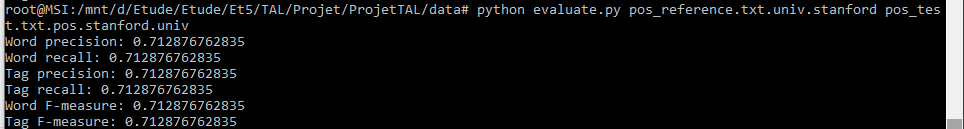
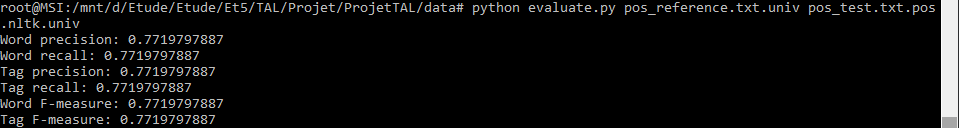


Figure 2 : Résultat de l'analyse morpho-syntaxique de nltk

### Une image contenant capture d’écran, intérieur, assis, objet Description générée automatiquement avec une confiance faibleReconnaissance d’entités nommées

Figure 3 : Résultat de la reconnaissance d'entités nommées de nltk

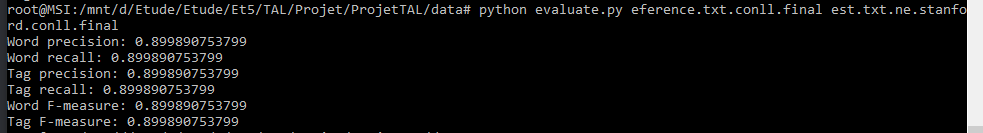
La méthode du modu de NLTK pour la reconnaissence d’entités nommées est de créer des arbres avec les noms et les adjectifs qui y sont liés, cette méthode permet de récupérer facilement les différentes entités nommées mais pour placer les tags conll sur les différents mots il faut faire une analyse des différents arbres et ensuite taggé en conséquence. De plus le module de reconnaissence utilise de nombreux autres modules ce qui augmente les incertitudes et donc les erreurs d’approximations ce qui un certain niveau d’inprécision (12%) lors de l’évaluation.

Figure 4 : Résultat de la reconnaissance d'entités nommées de stanford

La plateforme de stanford pour la reconnaissance des entités nommées fait un reconnaissance en faisant un comparaison avec un dictionnaire de mots qui lui sert à faire la distinction entre les différentes entités du texte, l’aventage de cette méthode c’est que l’étude ce fait rapidement mais l’analyse ne prend pas en compte des mots qui lierais différente entités nommées comme par exemple dans la référence : « The B-ORG United I-ORG B-LOC States I-ORG I-LOC 's I-ORG largest I-ORG car I-ORG manufacturer I-ORG General I-ORG Motors I-ORG ». Qui devient avec stanford : «  The O United B-LOC States I-LOC 's O Largest O car O manufacturer O General B-ORG Motors I-ORG ». On remarque bien que le mot « The » est bien reconnu comment faisant partie de l’entité nommée dans la référence alors que stanford non, ces différences explique pourquoi il y a une baisse de la précision ;

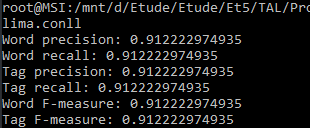


Figure 5 : Résultat de la reconnaissance d'entités nommées de lima

## Contribution des membres du groupe

### Sullivan Honnet

Il eut la charge de l’utilisation des outils nltk et stanford, il s’est chargé de générer les fichiers à partir de ces outils.

De plus il s’est occupé du traitement des fichier python pour la reconnaissance des entités nommées pour les trois outils : lima, nltk et stanford.

Le but des différents fichier est de récupérer les résultats des outils pour pouvoir par la suite faire la comparaison avec les fichiers de référence ici le fichier ne\_reference.txt.conll. Le premier problème étant les sauts de ligne au milieu des phrases qui était traiter différemment selon l’outil par exemple stanford considère le « . » comme caractère pour faire un saut de ligne et concaténait alors les morceau de phrase ce qui fait que le fichier qui en résultait ne correspondait donc que pour les premiers mots de la première ligne. On a donc eu le parti pris de concaténer les phrases et de sauter des lignes qu’après un « . ».

### Jules Vittone-Burnel

Il s’est occupé de l’utilisation et de la génération du fichier lima et d’écrire le script principal. Ce script a pour rôle de formater tous les fichiers de départ pour obtenir des fichiers utilisables par la suite par le programme evaluate.py. L’une des grosses difficultés dans cette partie fut que chaque fichier suivait sa propre nomenclature et ses propres régles et qu’il n’existait pas d’uniformité entre le fichier de départ fourni et les fichiers obtenus par nltk, stanford et lima.

Le problème se présenta pour l’analyse morpho-syntaxique d’abord par le fichier source qui contenait de nombreuses lignes comportant plusieurs mots pour un seul tag alors que tous les autres programmes donnaient une équivalence un mot, un tag. Ce problème fut résolu en découpant les groupes de mots d’une ligne en plusieurs lignes contenant un mot et le tag associé auparavant au groupe de mots.