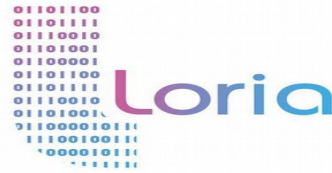




UNIVERSITÉ  
DE LORRAINE



André GIANG

## Rapport de stage

# Un moteur d'inférences pour la correction à partir de cas de phrases en français

**Du 9 avril au 6 juin 2018**

Responsable de stage :

Bruno GUILLAUME

Yves LEPAGE

Jean LIEBER

Emmanuel NAUER

**Licence 3 informatique**



# INTRODUCTION

De nos jours, nous sommes aidés et entourés de beaucoup d'outils qui nous permettent de modifier et corriger nos fautes d'orthographe. Cependant aujourd'hui et de tout temps nous avons toujours eu de nombreuses déformations et « raccourcis » grammaticaux sans même s'en rendre compte le but de ce stage est d'arriver à corriger une phrase incorrecte grammaticalement.

Par exemple « e suis sur Nancy » devrait être « Je suis à Nancy ».

Pour ce faire nous allons utiliser le raisonnement à partir de cas (RàPC) qui consiste à résoudre un problème grâce à une base de cas qui contient des couples (problème, correction du problème).

Voici un exemple,

Notre phrase à corriger  
J'aime pas les araignées.

Un couple qui servira à la correction du problème  
J'aime pas nager.      Je n'aime pas nager.

Notre correction à partir du couple dans la base  
J'aime pas les araignées.      Je n'aime pas les araignées.

Le RàPC se déroule en deux étapes :

- Remémoration : Choix d'un cas dans la base jugé ressemblant au problème demandé.
- Adaptation : Correction et modification du problème demandé en une correction à partir du cas choisit.

Il y a aussi une troisième étape qui est celle de l'apprentissage. Ici elle se traduit par l'ajout de nouveaux cas dans la base permettant ainsi de nouvelles ou meilleures correction. C'est un outil qui peut donc apprendre très vite si beaucoup de cas lui sont donnés, dans notre groupe ce sera M Levy qui se chargera de créer un script permettant la récupération de cas trouvé sur le Web et notre trinôme M Ly se chargera de créer la base de donnée ainsi qu'un site vitrine donnant l'accès au moteur au reste du monde.

Nous allons voir plus en détails la conception et le développement du correcteur de phrases à partir des travaux d'Yves Lepage et de ses collaborateurs.  
En effet ceux-ci ont développés un outil qui utilise le raisonnement à partir de cas qui permet de traduire une phrase d'une langue à une autre.

Nous allons nous servir de ces travaux afin de développer en Python un outil capable de proposer une correction d'une phrase cible en français, en s'appuyant sur le principe de raisonnement à partir de cas.

Nous allons vous présenter le déroulement du développement du moteur de correction de phrases en plusieurs parties:

- L'explication et le choix des outils utilisés
- La phase d'analyse et compréhension des travaux d'Yves Lepage

Les parties suivantes seront les différentes étapes du développement du moteur

- Première version de L'adaptation
- L'amélioration de l'algorithme de l'adaptation
- Réflexion sur la remémoration
- Remémoration

Pour finir sera présenté les difficultés rencontrés et apports éventuels du moteur

- Les difficultés rencontrés
- Les améliorations éventuelles et futures
- Conclusion

## Choix et conception

Dans le cadre du stage j'ai développé le moteur avec le langage Python dans sa version 2. Ce choix s'est fait assez naturellement, en effet les travaux d'Yves Lepage étaient en Python 2.7 et dans un soucis de cohérence nous avons décidé de garder le même langage de programmation.

N'ayant encore jamais développé sous Python j'ai fallu apprendre ce nouveau langage.

Une fois cela fait et décidé j'ai récupéré et analysé le code de la maquette qu'Yves Lepage a développée.

Sa maquette permet de traduire une phrase d'une langue à une autre à partir du RàPC et d'analogie.

Une analogie est une association d'idée entre deux ou plusieurs objets de nature différente.

Quelques analogies :

Avion : voler :: Voiture : rouler

Faim : manger :: Soif : boire

J'aime pas les marshmallow : Je n'aime pas les marshmallow :: J'aime pas nager dans le lac du connemara : Je n'aime pas nager dans le lac du connemara

Un exemple du fonctionnement de la maquette d'Yves Lepage:

They love apples.      Ils aiment les pommes.

Le fonctionnement est un peu différent il n'utilise pas qu'une base de cas mais aussi un dictionnaire de mot.

Phrase à traduire

They love apples.

- Remémoration

Cas dans la base

They love pears.      Ils aiment les poires.

- Adaptation

Mot à traduire

pear | apple    poire | pomme

Remplacement de poires par pommes

Ils aiment les poires.    Ils aiment les pommes.

Résultat

They love apples.      Ils aiment les pommes.

## Développement

Nous avons repris le code de la maquette d'Yves Lepage et l'avons modifié et adapté selon nos besoins et fonctionnement.

Dans notre cas nous n'allons pas avoir besoin du dictionnaire nécessaire au traducteur car nous corrigeons les fautes dans la même langue.

Une interface en ligne de commande a été ajoutée afin d'être plus facile à comprendre et mieux indiquer les marches à suivre pour tester le moteur.

```
gandre@gandre-HP-Notebook:~/workspace/StageL3/MOTEUR/Fwd Implementation dune maquette du modèle de Na
gao/maquettes/maquettes/test$ python Test.py
Rentrez la phrase à corriger : J'aime pas nager.
Voulez vous entrer une correction ? 0/n : 0
Rentrez la phrase source fausse : J'aime pas manger.
Rentrez la phrase source corrigée : Je n'aime pas manger.
Bs J'aime pas nager.

# Translating sentence: J'aime pas nager.
#      J : e n : 'aime pas nager. :: J'aime pas manger. : J'aime pas nager. : Je n'aime pas manger.

RESULTAT J'aime pas nager.      Je n'aime pas nager. | J'aime pas manger.
```

Une première version du moteur a été développé.

Celle-ci ne comporte pour l'instant que la 2eme étape du raisonnement à partir de cas l'adaptation, et donc ne s'occupe pas encore de choisir les cas optimaux s'ils existent qui fait partie de la remémoration.

## Algorithme Adaptation:

Rentrer la phrase à corriger

Choix de la base de cas à utiliser

Tri croissant des cas par longueur d'édition (LCS)

Parcours de chaque cas

Calcul des préfixe suffixe commun (problème cible , problème source)

si oui (ressemblance dans la fin et/ou le début)

Calcul préfixe suffixe commun problème source, sol ( source)

récupération de la sous chaîne correctrice à insérer :

J | **e n** | 'aime pas manger.

récupération de la sous chaîne à remplacer dans le problème source :

J |  | 'aime pas manger.

Découpage dans problème cible

Chaîne de la position 0 à position préfixe + longueur préfixe

sous chaîne à insérer 'e n'

Chaîne du reste jusqu'à la fin

retourne le triplet

**Je n**'aime pas nager.