APPRENTISSAGE ET ALIGNEMENT AUTOMATIQUES DE LANGUES SANS CONNAISSANCES PRÉALABLES

École des Ponts

ÉCOLE NATIONALE DES PONTS ET CHAUSSÉES
DÉPARTEMENT IMI

Darius AFCHAR HERAVI MOGHADDAM
Fatine BOUJNOUNI
Matthieu TOULEMONT

1 INTRODUCTION

Principe

Partant d'un texte connu, on cherche à trouver **automatiquement** la traduction d'un mot dans un texte de langue inconnue en se basant sur ses différentes apparations. L'idée est que à l'échelle d'un texte, les mots apparaissent aux mêmes endroits dans toutes les langues.

Objectifs

En traitant les apparitions des mots sous forme de signaux, on cherche à associer chaque mot d'un texte au signal de la forme la plus proche dans l'autre texte.

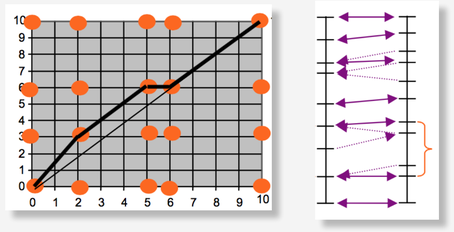
Hypothèses et difficultées

Il ne peut exister d'alignement parfait que si chaque mot est traduit de manière unique en un autre mot dans une autre langue, c'est-à-dire, s'il existe une **bijection** entre les deux langues considérées. Cela est bien évidemment faux, le style littéraire, les synonymes, les déclinaisons verbales empêchent un alignement parfait.

Nous avons cherché à palier le problème et à quantifier la qualité de notre traduction.

2 ALIGNEMENT

Pour quantifier la ressemblance entre les vecteurs d'apparitions des mots, on utilise un **algorithme de dilatation temporelle (DTW)**. Il s'agit d'une technique classique d'alignement de deux séquences de longueurs différentes qui minimise la distortion.



Algorithme complet

On **associe** tout d'abord chaque mot du premier texte à un **ensemble probable de candidats** du second texte. En première approche, on a choisi les mots qui apparaissent un nombre proche de fois dans les deux textes.

Puis on calcule les distances entre chaque mot et chaque candidats grâce au DTW. On trouve celui qui **minimise cette distance**.

Enfin on analyse la qualité de la distance pour savoir si le mot est bien traduit ou non.

Premiers résultats

Avec un alignement simple, on obtient **25%** de mots correctement traduits en moyenne. L'algorithme est notamment incapable d'aligner les verbes qui peuvent se décliner de multiples façon dans une langue mais n'avoir qu'une forme dans une autre.

En associant les morphènes proches d'un même texte avec une **distance de Jaro-Winkler** on améliore un peu ce résultat.

3 CLASSIFICATION MORPHOLOGIQUE

Après avoir effectué l'alignement, nous voulions avoir **la possibilité de traduire un mot par un groupe de mots dans l'autre langue**. Nous nous sommes d'abord tournés vers la méthode employé par John Goldsmith dans le logiciel *Linguistica*, qui emploie le concept de *Minimum Description Length* de Rissanen. À la suite de multiples résultats et un échange de mail avec l'intéressé, nous avons décidé de changer de méthode.

Le manque de temps nous a forcé à utiliser des heuristiques plutôt que d'implémenter d'autres approches développées par des chercheurs.

Hypothèses

- On n'étudie que les mots de longueur supérieure à quatre

- La taille des suffixes varient suivant la longueur du mot, on fixe le maximum à cinq

- On ne connait pas la langue du texte

Principe

i. Encoder le texte dans un Patricia Trie

ii. Trouver les suffixes et les coupures dans l'arbre

iii. Pour chaque mot du texte créer un ensemble de segmentations

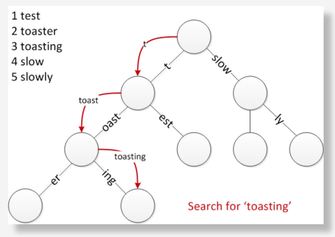
iv. Calcul des indices de vraisemblances pour chaque segmentation, en prenant en compte l'étape 2

$$(-1) * log(\frac{[radical_i + suffixe_k]}{[radical_i] * [suffixe_k]}) \quad (1)$$

v. Ne garder que la segmentation la plus vraisemblable pour chaque mot

vi. Créer les signatures

```
('a', 'e', 'er') :
 confirma | confirm
 répliqua | répliqu
 accepta | accept
 accepter | accept
 réplique | répliqu
 confirmer | confirm
 accepte | accept
 remua | remu
 remue | remu
 remuer | remu
 confirme | confirm
 répliquer | répliqu
```

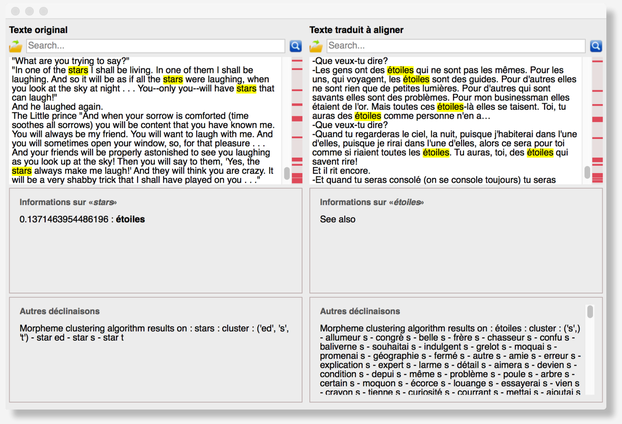


4 DÉVELOPPEMENT

Architecture & Technologie

L'architecture MVC était adaptée à notre problématique, notre algorithme étant contenu dans la partie modèle et l'interface générée par le framework **PyQt4** dans la vue et le contrôleur.

Nous avons utilisé le gestionnaire de projet **GitHub** pour coordonner le développement des différentes parties du projet.

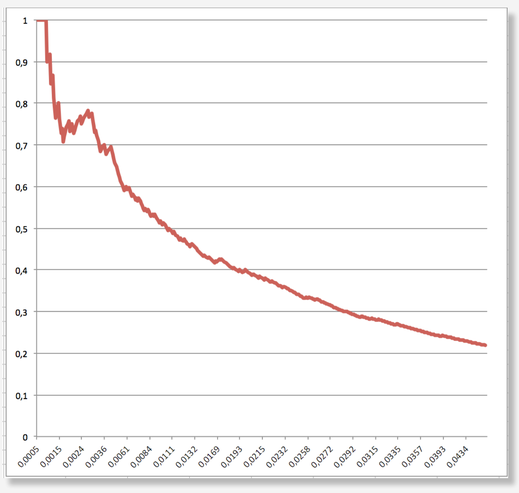


5 ANALYSE DES RÉSULTATS

Nous avons testé notre logiciel à l'aide de plusieurs textes de longueur variable (e.g. *Le Petit Prince* et *Les Misérables*).

Pertinence de la traduction

Nous avons trié les mots traduits en fonction de leur distance moyenne et compté le pourcentage de mots justes que nous obtenions avec l'augmentation de celle-ci, afin de pouvoir indiquer à l'utilisateur une estimation de la pertinence de la traduction.



Pourcentage de mots justes en fonction de la distance par occurence

6 LIMITES & AMÉLIORATIONS

Complexité

Un livre moyen comportant plusieurs milliers de mots, trouver l'alignement entre deux devint rapidement très long. Nous avons donc toujours fait en sorte d'avoir une complexité minimale à chaque opération et mis en place un système de sauvegarde des prétraitements.

Classification morphologique

La méthode que nous avons appliquée n'est pertinente que pour les langues latines et l'anglais. Ceci à cause de l'hypothèse forte sur la taille des suffixes. Si nous avions eu plus de temps, nous aurions sans doute essayé l'approche de Mathias Creutz et Krista Lagus visant à maximiser la probabilité a posteriori des segmentations. Cette approche est non-supervisée et se libère des hypothèses que nous avons effectuées.

Alignement automatique

Malgré les heuristiques et les apprentissage de la morphologie, les langues sont chargées d'exceptions et de lois propres, elles n'obéissent pas à des règles mathématiques générales. Difficile donc d'espérer faire mieux en terme de résultats sans connaissances sur les langues considérées.

Remerciements : Renault MARLET, Xavier CLERC