Alexis Couvreur

Projet Langage Naturel

Le dialecte de Yoda

# Description du problème, exemples d’entrée/sortie

## Description du problème

Le but est de fournir par écrit une phrase normalement construite en Français au programme, celui-ci devra être en mesure de pouvoir de rendre cette phrase dans le dialecte de Yoda (cf. Wikipedia). Sa façon de parler est devenue mythique : elle utilise la syntaxe objet, sujet, verbe, totalement à l'inverse de la plupart des langues indo-européennes, qui suivent la syntaxe sujet, verbe, objet. Cependant, pour que certaines phrases restent compréhensibles, ce ne sont souvent que les auxiliaires qui sont déplacés. La formation des phrases est une résultante d'une figure stylistique appelée anastrophe.

## Exemples d’entrée/sortie

Voici des exemples d’entrées et de sorties :

* « Tu dois manger du riz. » « Manger du riz, tu dois. »
* « J’aimerais aller à la piscine. » « À la piscine, aller, j’aimerais. »
* « Il faut faire attention aux orages. » « Aux orages, attention il faut faire. »
* « Entre deux mots, il faut choisir le moindre » « Entre deux mots, choisir le moindre, il faut »
* « Bonjour. » « Bonjour. »
* « abcde » « abcde »
* « Il te reste encore beaucoup à apprendre. » « Beaucoup encore il te reste à apprendre. »
* « Visqueux ? Boueux ? Je vis ici ! » « Visqueux ? Boueux ? Ici je vis ! »
* « Il est ton père » « Ton père, il est. »
* « De consulter les hologrammes de sécurité, une grande souffrance te causera. »
* « Fie-toi à ton intuition, tu finiras par le trouver. » « À ton intuition fie-toi, par le trouver tu finiras. »
* « Nous sommes en danger. » « En danger, nous sommes. »
* « Vous devez partir. » « Partir, vous devez »

# Recherche bibliographique des solutions existantes

## L’anastrophe

L'anastrophe est une figure de style, dite de « construction », qui consiste en une inversion de l'ordre habituel des mots d'un énoncé pour créer un effet de langue raffiné.

Dans la langue Française, qui est qualifiée de langue SVO (sujet-verbe-objet) on transforme donc une phrase normale en anastrophe tel que :

Sujet – Verbe – Objet devient Objet – Sujet – Verbe

## Analyseurs syntaxiques libres

Plusieurs analyseurs libres sont à disposition tel que **TreeTagger**, **Berkeley parser**, **Standford parser** ou encore **UDpipe**.

* **TreeTagger** est un outil permettant d’annoter un texte avec sa partie du discours et des informations sur le lemme. **TreeTagger** est utilisée pour l’Allemand, l’Anglais, le Français, l’Italien, etc. ce qui en fait un point fort étant donné qu’il sera possible d’utiliser la partie du discours pour identifier correctement le pattern des phrases et ainsi pouvoir agir en conséquence.
* **Berkeley parser** est un outil qui permet d’écrire et de trier des arbres résultant d’une entrée tokenizée. Beaucoup d’options supplémentaires sont présentes mais ne contient pas de grammaire française par défaut (seulement Anglais, Allemand et Chinois). Il est possible de faire apprendre une grammaire avec une « treebank ». Etant donnée l’entrée de mon programme (phrases courte) cet outil fournit beaucoup plus que ce qui est nécessaire et semble plus complexe à mettre en place.
* **Standford parser** est un outil d’analyse de langage naturel qui travaille sur la structure grammaticale dans des phrases et fournit des informations s’il s’agit du sujet, de l’objet ou alors du verbe. **Standford parser** est un analyste probabiliste, il va se baser sur la connaissance sur la probabilité du langage pour sortir des situations ambigües.
* **UDPipe** est un outil de tokenization, qui tag et affiche les lemmes. Il fournit le résultat sous plusieurs formes tel que le texte tokenisé, dans un tableau ou encore dans un arbre.

# Tests de logiciel libres existants

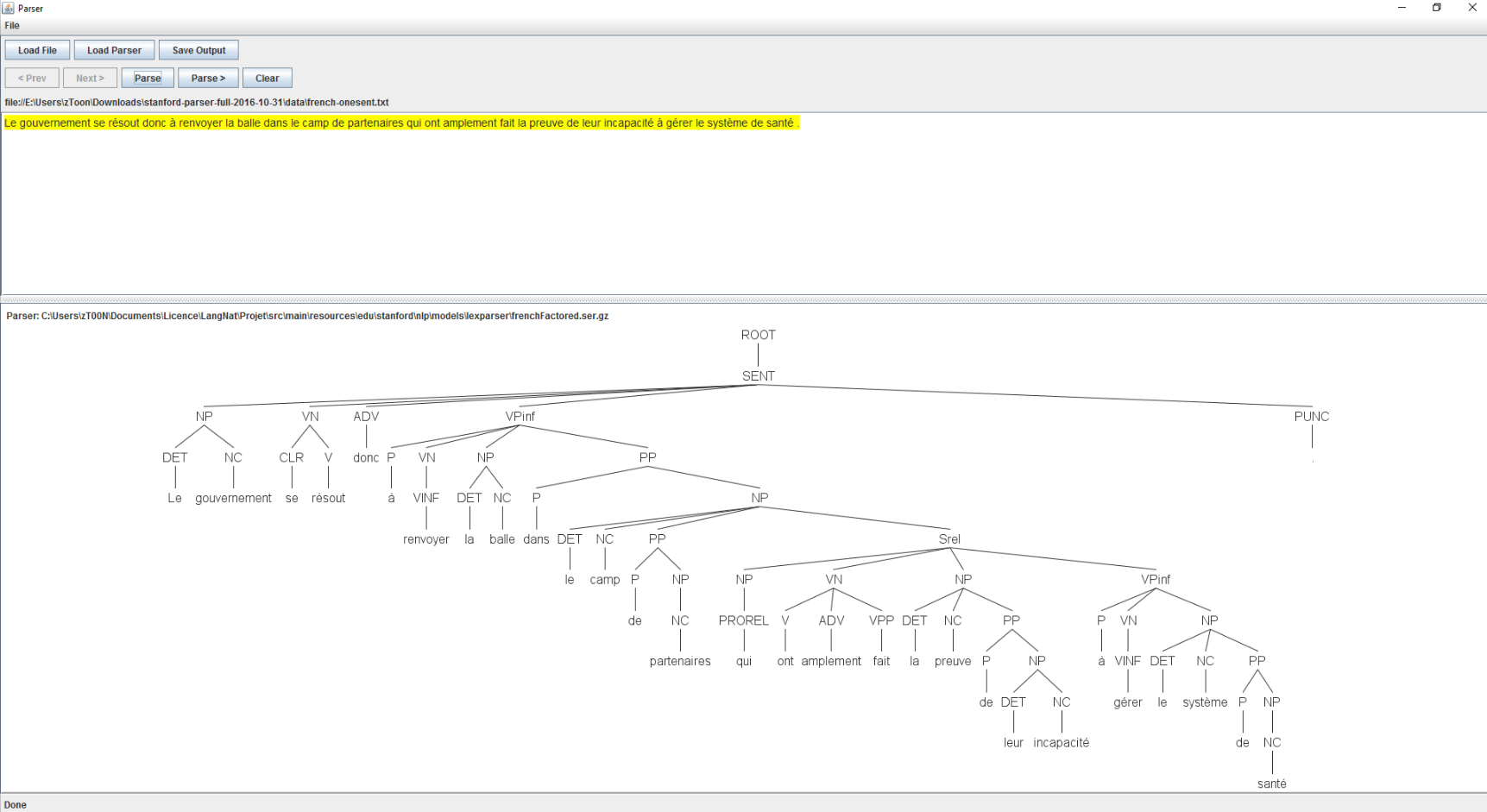
Un site Anglophone proposant le service de transformer des phrases Anglophones dans le dialecte de Yoda existe à l’adresse suivante : <http://www.yodaspeak.co.uk/> ; l’output est souvent accompagné d’interjections donnant un rendu plus semblable à celui de Yoda.

L’utilisation de **TreeTagger** est réalisable très rapidement à travers Java, il suffit d’abord d’installer **TreeTagger** et d’ensuite utiliser un binder tel que **TT4J**, qui dispose d’un répertoire maven ce qui rend l’implémentation plus simple.

|  |  |  |
| --- | --- | --- |
| Il | PRO:PER | il |
| est | VER:pres | Ãªtre |
| ton | DET:POS | ton |
| père | NOM | père |
| . | SENT | . |

Voici un exemple d’entrée/sortie lors de l’utilisation de l’analyseur **TreeTagger** à travers Java :

En négligeant les problèmes d’encodage en sortie, on peut facilement constater comment transformer cette phrase en un anastrophe grâce aux informations données. Le pronom « il » est le sujet, le verbe « est » est le verbe et l’objet est « ton père » ; ainsi s’il on transforme la phrase SVO en OSV on obtient : ton père il est. Il suffit de mettre les majuscules au bon endroit et de séparer la phrase par des virgules pour qu’elle soit correcte.

L’utilisation de **Standford parser** nous donne un arbre contenant toutes les informations nécessaires tel que :

# Cahier des charges pour l’implémentation du prototype

Le but est de réaliser un programme Java prenant en entrée un texte et renvoyant ce texte dans le dialecte de Yoda. Seul les structures de la forme SVO seront modifiées, les autres phrases / morceaux de phrases resteront inchangés et retournés tel quel.

Le texte en entrée aura une longueur maximale de 300 caractères, si celle-ci est supérieur le programme retournera seulement jusqu’à la dernière phrase trouvée avant le 300ème caractère. Le texte est exclusivement en Français, mais vu l’étendu des analyseurs syntaxiques il est envisageable d’implémenter l’Anglais étant donné que celui-ci utilise la même typologie syntaxique (SVO). Toute forme non comprise restera inchangée (faute de frappe, mot non reconnu, etc.).

Le texte en retour sera décomposé comme il se doit pour former une phrase correcte, l’anastrophe proposant très souvent des tournures de phrase nécessitant la séparation de celle-ci par une ou plusieurs virgules. (« Il est ton père. » : « Ton père, il est »)

La présentation du programme est pour le prototype seulement en ligne de commande, des améliorations futurs peuvent être envisagées comme ajouter une interface graphique, ou alors transformer le programme en un webservice (SOAP / REST) pouvant être utilisé par la suite sur un site web. La suite du projet sera donc potentiellement pourvue d’un site web de test.