Autocomplétion lors d'une saisie

J'ai pour objectif d'écrire un algorithme qui, au cours de la saisie d'un texte, devine le mot que l'utilisateur souhaite écrire ensuite.

Il s'agit donc pour l'algorithme de faire le choix optimal parmi tous les mots qui pourraient suivre, avec la contrainte d'un temps d'exécution court, en assimilant l'utilisateur à une source aléatoire de mots.

C'est une première étude sur la compression de texte qui, en lien avec la théorie de l'information, m'a mené à ce nouveau sujet.

Positionnement thématique (étape 1)

INFORMATIQUE (Informatique pratique), INFORMATIQUE (Informatique Théorique), MATHEMATIQUES (Autres).

Mots-clés (étape 1)

Mots-Clés (en français)	**Mots-Clés** (en anglais)
Autocomplétion	*Autocompletion*
Algorithmique du texte	*Text algorithmic*
Théorie de l'information	*Information theory*
Langage naturel	*Natural language*
Probabilités	*Probabilities*

Bibliographie commentée

Les sciences informatiques, apparues au cours de la Seconde Guerre Mondiale, ont rapidement nécessité une base théorique mathématique, notamment dans le but de quantifier les performances théoriques d'un programme. Cette base a été notamment apportée avec la théorie de l'information et de la communication développée par C. Shannon et W. Weaver en 1944 [1]. Cette théorie a établi toutes les notions clés [2] du traitement et de la communication des informations que nous utilisons aujourd'hui. À partir des années 1970, les besoins croissants de l'humanité en systèmes informatiques efficaces ont permis le développement rapide de la discipline, suivant les besoins des utilisateurs.
L'accès au grand public des systèmes informatiques de messagerie (e-mails et SMS notamment) a demandé l'essor d'une branche en particulier : l'algorithmique du texte [3]. Celle-ci a créé les outils nécessaires à l'analyse et au traitement des langues naturelles.

Si certaines opérations comme la compression de texte ou la localisation de motifs ont été (et sont encore) étudiées sous tous les angles, d'autres comme l'autocomplétion [4] (c'est-à-dire la prédiction de la suite d'un texte à partir de ce qui en a déjà été saisi) semblent encore être laissées à la simple réalisation pratique [5] sans fondement théorique spécifique.

En effet, outre l'évidente étude de la complexité d'un algorithme d'autocomplétion, il faut

déterminer une méthode pour quantifier son efficacité, et évaluer l'écart entre les performances réelles et l'efficacité maximale théorique. Celle-ci repose a priori sur les caractéristiques de la langue utilisée (dans notre cas, le français), que l'on sait décrire avec une précision souvent satisfaisante par des modèles tels que la loi de Zipf [7] (qui prédit la fréquence d'apparition d'un mot dans une langue, d'après son rang dans le classement des plus récurrents).

Il est judicieux de noter qu'il existe plusieurs types d'autocomplétion [6] : la prédiction de mots (deviner le mot suivant un mot donné), la complétion de mots (deviner la fin d'un mot de façon isolée) et la multi-prédiction de mots (deviner la fin d'un mot en s'aidant du mot précédent). Nous nous concentrerons sur la prédiction de mots.

Problématique retenue

Après avoir parcouru la littérature en théorie de l'information, nous observons donc des lacunes sur l'étude de l'autocomplétion. Il va donc s'agir pour moi de réaliser un programme d'autocomplétion et d'entreprendre son étude théorique, établissant ainsi les outils permettant d'étudier ses performances.

Objectifs du TIPE

Je cherche à écrire un programme Python d'autocomplétion, fonctionnant à l'aide d'une base de données représentant la loi de succession des mots en français. Mes objectifs de travail sont donc :
- constituer cette base de données ;
- écrire le programme d'autocomplétion ;
- évaluer la complexité temporelle de ce programme, en raison de la taille atteinte par la base de données ;
- étudier les performances de prédiction du programme.

Abstract

As digital communication develops, it has become essential to study the theory of processes like autocompletion. Thus I wrote a program that takes a word and predicts the three words which are more likely to follow, with an experimental probability of 17% that the actual following word is suggested. This uses a database that I constituted with another program that I wrote. After studying the theory behind word succession using Zipf's law, I was able to calculate the theoretical probability of successful prediction, which appears to be 50%, but is probably overestimated due to the limited size of my database.

Références bibliographiques

[1] Claude Shannon : A Mathematical Theory of Communications, 1948

[2] Wikipédia : Théorie de l'information, consultée le 12/09/2016 :
https://fr.wikipedia.org/wiki/Th%C3%A9orie_de_l'information

[3] Maxime Crochemore, Christophe Hancart, Thierry Lecroq : Algorithmique du texte, Vuibert, 2001

[4] Wikipédia : Complètement, consultée le 19/09/2016 :

https://fr.wikipedia.org/wiki/Compl%C3%A8tement

[5] Stackoverflow : Algorithm for autocomplete ?, consultée le 26/09/2016 :
http://stackoverflow.com/questions/2901831/algorithm-for-autocomplete

[6] Polypredix™, consultée le 19/09/2016 :
http://www.assistiveware.com/fr/innovation/polypredixtm

[7] Wikipédia : Loi de Zipf, consultée le 30/01/2017 : *https://fr.wikipedia.org/wiki/Loi_de_Zipf*