Algorithme de correction d'orthographe

La performance de l'algorithme de correction orthographique repose sur des propositions (de termes corrigés) que l'on cherche à optimiser. L'algorithme lui-même va s'appuyer sur des probabilités, au niveau de l'entrée des caractères par l'utilisateur, tout comme au niveau de la possibilité qu'un mot choisi en particulier soit fréquemment utilisé dans la langue française, et soit ainsi plus "populaire" et par là plus "probable" qu'un autre.

Positionnement thématique (étape 1)

INFORMATIQUE (Informatique pratique), MATHEMATIQUES (Analyse), MATHEMATIQUES (Mathématiques Appliquées).

Mots-clés (étape 1)

Mots-Clés (en français)	**Mots-Clés** (en anglais)
Correcteur	*Spell Checker*
Analyse de données Bayésienne	*Autocorrection*
Distance de Levenshtein	*Bayesian Data Analysis*
Probabilité	*Optimization*
Base de données	*Big Data*

Bibliographie commentée

Les logiciels de correction d'orthographe sont devenus omniprésents dans l'utilisation informatique quotidienne. Ils sont en effet utilisés par les moteurs de recherche internet (Google, Yahoo, Bing), par les programmes de messagerie (réseaux sociaux, logiciels intégrés dans les téléphones portables), ainsi que sur la plupart des logiciels d'écriture (Microsoft Word, Google Drive, etc).

Le premier algorithme de correction de texte - dit "correcteur" - a vu le jour en 1971 dans un laboratoire d'intelligence artificielle à l'université de Stanford. Écrit par Ralph Gorin, un étudiant local, il proposait des mots alternatifs à ceux rentrés par l'utilisateur, (si ceux tapés n'apparaissaient pas dans sa base de données), en cherchant des mots ne différant que d'une lettre de l'entrée. L'algorithme fut vite répandu à l'échelle internationale via l'ARPAnet(*) ouvrant la voie à un outil aujourd'hui indissociable des programmes de texte.

La plupart des approches, lors de la création d'un correcteur, consistent à s'appuyer sur des "documents sources"[2], contenant des phrases ainsi que des mots orthrographiés de façon exacte, voire sur un ensemble de règles (grammaticales, de structure, etc)[4], de statistiques d'apparition des mots[5], ou bien sur tous ceux là à la fois, rendant les algorithmes "multi-tâches"[2] . Plus complexes, certains correcteurs, par exemple celui du moteur de recherche Google[3], n'utilisent pas

directement de telles bases de données mais s'appuyent sur un modèle d'erreurs statistiques constamment mis-à-jour via Internet.

Notamment pour des raisons de concurrence commerciale, les logiciels utilisant des correcteurs orthographiques jadis spécialisés ont eu tendance à les faire évoluer vers une couverture plus globale du type d'erreurs possibles, les rendant ainsi "hybrides". C'est le cas de Microsoft Word/Office, qui est ainsi passé d'un originel correcteur orthographique "simple" à un correcteur multi-tâches[1].

Néanmoins, et ce sera le nerf de notre travail, un correcteur prenant en compte un grand nombre de paramètres, par exemple à la fois la typographie, la grammaire, ainsi que les structures des phrases, pourra perdre en exactitude (du fait d'un trop grand nombre de résultats) sur des fonctions ciblées : s'il suffit d'une correction de la typographie d'un mot entré par l'utilisateur, un algorithme qui ne se spécialise pas dans cette fonction risque d'afficher de nombreux résultats parasites[4] (ce que montrent par ailleurs au quotidien les moteurs de recherche du type Bing, Yahoo). Il convient de noter qu'un trop grand nombre de paramètres affecte également la complexité du programme, et ainsi son temps d'exécution[2], ce qui est clairement prohibitif dans le cadre du respect des cahiers des charges auxquels sont soumis les divers programmes.

(*) (acronyme anglais de « Advanced Research Projects Agency Network », est le premier réseau à transfert de paquets - de données informatiques - créé aux Etats-Unis. Le projet date de 1966, mais n'aboutit réellement qu'à partir de 1971).

Problématique retenue

Nous souhaitons, dans notre travail, créer un algorithme de correction uniquement orthographique aussi performant que possible pour un nombre limité de lettres (nous nous limiterons à six lettres).

Objectifs du TIPE

Afin de répondre à la problématique posée, nous nous intéresserons avant tout à l'utilisation de l'expression régulière sur Python, tout en essayant d'ajouter des améliorations conditionnelles à l'algorithme final et en cherchant une utilisation optimale de nos bases de données.

Abstract

The goal of our work was to create a functional spell checker. To achieve that, we sought a database containing all the French words correctly written, then created our own database made of over 3 million words, consisting in french sentences, which was supposed to allow us to find out which words are the most frequently used ; then, we created a main algorithm (supported by smaller ones) editing the words entered by the user, and that way, looking for the ones that were the most similar to them in the databases. We finally tried to optimize it with additional conditions.

Références bibliographiques

[1] MSDN Archive, 2009 : "Un correcteur contextuel français dans Office 2010" : *https://blogs.msdn.microsoft.com/correcteurorthographiqueoffice/2009/07/15/un-correcteur-contextuel-franais-dans-office-2010/*

[2] Google Inc, Bangalore, India : "How Difficult is it to Develop a Perfect Spell-checker? A Cross-linguistic Analysis through Complex Network Approach" : *http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=52A3B869596656C9DA285DCE83A0339F?doi=10.1.1.146.4390&rep=rep1&type=pdf*

[3] Google inc, Australia, 2009 : "Using the Web for Language Independent Spellchecking and Autocorrection" : *http://static.googleusercontent.com/media/research.google.com/en/us/pubs/archive/36180.pdf*

[4] Andrew R.Golding (MERL - Mitsubishi research laboratory) & Dan Roth (Department of computer science, University of Illnois) : "A winnow-Based approach to context sensitive spelling correction" : *http://l2r.cs.uiuc.edu/~danr/Papers/spellJ.pdf*

[5] Andrew Gelman, director of the Applied Statistics Center at Columbia University : "Discrete examples : genetics and spell checking" : *http://www.stat.columbia.edu/~gelman/stuff_for_blog/spelling.pdf*