Master Informatique

Rapport du projet d'Initiation à la recherche

sujet : Constitution d'une base de cas de corrections du français

Année 2018-2019

Étudiants : Alex Ginestra
Christopher Klein

Équipe : Orpailleur et Sémagramme
Encadrants : Jean Lieber, Bruno Guillaume,
Yves Lepage et Emmanuel Nauer

Décharge de responsabilité

Remerciements

Tout d'abord, nous tenons à remercier toute l'équipe pédagogique du département informatique de la faculté des sciences et technologies pour les quatre années de formation en Master informatique.
J'adresse également mes remerciements à toute l'équipe Orpailleur et Sémagramme ainsi qu'aux encadrants Jean Lieber, Bruno Guillaume, Yves Lepage et Emmanuel Nauer pour leur accueil et leur sympathie.

Sommaire

Rappel du sujet

Problématique de recherche :
Le raisonnement à partir de cas (RàPC) est un raisonnement hypothétique (en général) qui consiste à résoudre un nouveau problème (le problème cible, noté cible) en s'appuyant sur une base de cas, un cas étant la représentation d'un épisode de résolution de problème. On appelle cas source un élément de la base de cas. Souvent, on représente un cas source simplement par un couple (srce, sol(srce)) : srce est un problème source, sol(srce) est une solution de ce problème source. Un modèle du processus de RàPC classique comprend deux étapes d'inférence :
Remémoration : un cas source (srce, sol(srce)) jugé similaire au problème cible (par exemple, sur la base d'une distance entre problèmes) est sélectionné.
Adaptation : La solution sol(srce) de ce cas est modifiée en une solution candidate sol(cible) de cible.

La correction de phrases est la problématique de la transformation d'une phrase incorrecte (en particulier, grammaticalement) en une phrase corrigée (nous choisirons la langue française dans ce travail, même si la problématique existe dans toutes les langues). Un cas de correction de phrase est donc un couple (srce, sol(srce)) où srce est une phrase incorrecte et sol(srce) une correction de srce. Par exemple, on a les deux cas :
srce1 = Tu as pas mangé. sol(srce1) = Tu n'as pas mangé.
srce2 = Il a recommencer. sol(srce2) = Il a recommencé.
L'adaptation se fait par des techniques de raisonnement par analogie : la solution sol(cible) est solution d'une équation analogique « srce est à sol(srce) ce que cible est à y ». Par exemple, l'adaptation de (srce2, sol(srce2)) à cible = Tu as manger. consiste à résoudre Il a recommencer. est à Il a recommencé. ce que Tu as manger. est à x qui a pour solution, avec la relation d'analogie utilisée dans le projet, x = Tu as mangé., proposition de solution proposée par le système.

Sujet :

Comme pour tout système à base de connaissances, la qualité d'un système de RàPC dépend de celle de son moteur d'inférences mais également de la qualité de sa base de connaissances, en particulier de sa base de cas. Une bonne base de cas doit avoir plusieurs qualités. Les cas sources doivent être corrects. Elle doit avoir une bonne couverture (et permettre de résoudre correctement une proportion importante de cas). Elle devrait ne pas être trop redondante (certains cas différents correspondent à la même correction).

Pour cela, on pourra consulter les mouchards d'édition de Wikipédia pour en extraire des listes de fautes grammaticales ou orthographiques typiques, ainsi que des sites de dictées ou d'orthographe. Il faudra mettre en place les outils de collecte automatiques, paramétrables en fonction des sites.

Une autre piste est la mise en place d'un jeu interactif avec un but. Le but est de faire corriger des phrases fautives par les joueurs. La phrase corrigée devrait émerger de la majorité des propositions de correction. Les phrases fautives pourraient être extraites de listes d'exemples fautifs, ou produites automatiquement à partir de patrons prédéfinis ou par application de l'analogie sur des cas déjà collectés.

Une courte étude bibliographique sur la maintenance de base de cas permettra de suggérer des pistes pour l'acquisition d'une bonne base de cas. Il faudra mettre en place une méthode pour cela, qui pourra s'appuyer sur les sites mentionnés ci-dessus.

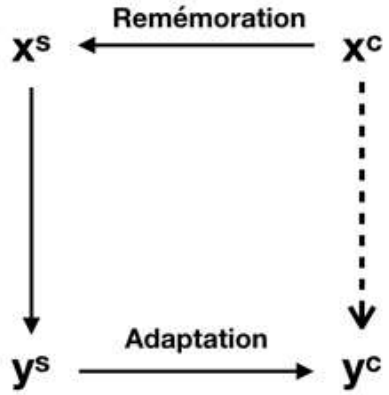
Introduction

Aujourd'hui, il est possible de trouver de nombreux correcteurs concernant la langue française. Malheureusement, dû à la complexité de notre belle langue (nombreuses règles grammaticales, syntaxe variable, ...), ces derniers s'avèrent être imparfaits. Une première approche que l'on pourrait qualifier de "naturelle", consisterait à coder de manière brut chaque règle de français. Cependant, cette solution se trouve être des plus complexes à cause des indénombrables exceptions en tout genre que contient le français. C'est pourquoi, nous abordons une approche différente de celle-ci. Nous allons travailler sur la confection d'un correcteur de français qui s'appuiera sur une base de cas, et qui tentera par divers moyens de corriger les erreurs grâce à cette dernière. Pour ce faire nous utiliserons la méthode suivante:

A : B :: C : D

Qui se traduit par **A** est à **B**, ce que **C** est à **D**.

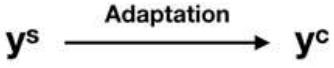
Voici le schéma du raisonnement à partir de cas (RàPC). Grâce à ce système, nous arrivons à déduire la solution à des problèmes à l'aide de différents outils : une base de cas, rassemblant un large panel de cas ainsi que la "Remémoration" et l' "Adaptation", permettant respectivement de rapprocher un cas d'un autre et de créer une solution a un problème à partir d'une solution plus ou moins similaire.



Plus précisément, nous disposons d'une base de cas, qui est en soit un regroupement de plusieurs cas. Un cas se caractérise par un problème (notée x), de sa solution (notée y) ainsi que d'une explication qui permet de passer du problème à la solution.

Dans un second temps, nous avons un problème auquel nous n'avons pas de solution existante. Nous l'appellerons problème cible (noté $\mathbf{x}^c$). Le principe du raisonnement à partir de cas est de rapprocher ce problème cible d'un problème qui se situe dans notre base de cas et pour lequel nous avons une solution. Ce problème s'appellera cas cible (noté $\mathbf{x}^s$). Cette étape s'appelle "Remémoration".

Grâce aux différences et aux similitudes entre notre problème cible et notre cas cible ainsi qu'à l'explication permettant de passer du cas cible au cas solution (noté $\mathbf{y}^s$), nous allons être capables de créer une solution de notre problème cible, appelé solution cible (notée $\mathbf{y}^c$). Cette étape est appelée "Adaptation".



I) Analyse du problème et organisation

A) Description du sujet

Pour parvenir à un système s'apparentant au schéma décrit en introduction, deux points sont essentiels: le moteur d'inférence, et la base de connaissances. Et, c'est au cours de cet UE qu'est l'initiation à la recherche, que nous tenterons d'aider les auteurs de ce sujet concernant la création d'une base de cas conséquente, et correcte.

Comme celle-ci se doit être de taille importante pour le bon fonctionnement de notre système, elle se remplira de manière automatisée. Notre approche dans notre initiation à la recherche est de choisir une des plus grosse base de cas disponible sur internet en libre accès, WiCoPaCo, qui recueil plusieurs milliers de corrections de fautes du site mondialement connu Wikipedia dans des fichiers au format XML.

Néanmoins, cette immense base de cas regroupant plus de 200 000 cas ne peut pas être utilisée telle qu'elle, car elle contient des milliers de cas utilisables : par exemple, des cas ne seront pas intéressant pour notre travail, d'autres n'auront pas d'explication assez concrètes pour être utilisé. Certain cas ne seront même pas corrigés alors que d'autres seront corrigés alors qu'ils ne sont pas faux.

Notre objectif est alors l'épuration de grosses masses de données. Pour ce faire, nous allons mettre en place plusieurs méthodes dans l'optique de "nettoyer" le contenu des fichiers de WiCoPaCo pour avoir en résultat final une base de cas correct et utilisable pour le raisonnement à partir de cas.

B) Organisation et travail existant.

(manque le travail existant, reprise du travail des étudiant de L3)

Pour débuter ce projet d'envergure, il nous fallait définir un plan de travail: Christopher devait s'occuper de la détection d'erreur, pendant qu'Alex commençait à se pencher sur le remplissage de la base de cas. Concernant la détection d'erreur, le choix eût été d'utiliser l'outil grew.

Christopher se chargeait donc de l'installer, ce qui n'était pas une mince affaire au vu des dépendances de ce dernier, afin de l'utiliser de manière automatisée en local. L'idée aurait été, par le biais de scripts, de l'exécuter sur une grosse masse de données. D'abord pour s'assurer de la validité de ces dernières, mais aussi pour pouvoir effectuer des analyses statistiques sur les fichiers utilisés ainsi que sur la résultante après traitement.

En parallèle de cela, Alex commençait un début d'étude sur le contenu des corrections tirées du site WiCoPaCo, en reprenant d'une part les travaux effectués par d'autres étudiants sur le fichier en question, mais aussi par une analyse personnelle de ce dernier.

Les travaux effectués l'année passée contenaient des informations sur la nature des fichiers, leurs structures et leurs contenus. Dans le répertoire de travail se trouvait également un scripte en python qui effectuait un filtrage sur ce type de fichier.

Suite à la reprise des travaux effectués couplé avec une analyse complémentaire par notre groupe de la composition de ces fichiers, la décision de développer un nouveau programme fut prise à l'unanimité par les membres du groupe. Le programme développé en java (langage parfaitement maîtrisé par notre groupe) permettrait l'extraction du contenu des balises XML des fichiers de WiCoPaCo, via des librairies existantes, pour en restituer une partie du contenu dans un format CSV.

C) Contraintes du sujet

(explication des contraintes du sujet)

II) Réflexion et développement

A) Description du logiciel développé

1. Description des fichiers traités

...

2. Fonctionnement du logiciel

Le choix de développer le logiciel en Java s'appuie sur les nombreux avantages que ce dernier fournit : nombreuses API et documentation en ligne, langage orienté objet, détections d'erreurs faciles. Son objectif est d'extraire et traiter les données contenues dans des fichier XML provenant du site WiCoPaCo.

Dans un premier temps, le logiciel en question utilisait un système de filtres permettant d'accepter ou rejeter un contenu précis correspondant aux critères désirés. Puis après une analyse des résultats, un autre type de filtre fut mis au point. Ceux-ci permettant d'épurer les cas acceptés au préalable. Ici, leur utilité étant de réduire la taille du cas en supprimant les données inutiles. Par exemple si un cas comporte 5 phrases avec une correction dans une seule phrase de ce groupe, alors le second type de filtre supprimera les 4 phrases non concernées par la correction.

Une fois le filtrage effectué, le logiciel crée un fichier CSV constitué de trois colonnes :

1. Le cas non corrigé
2. Le cas corrigé
3. Le commentaire laissé par le correcteur

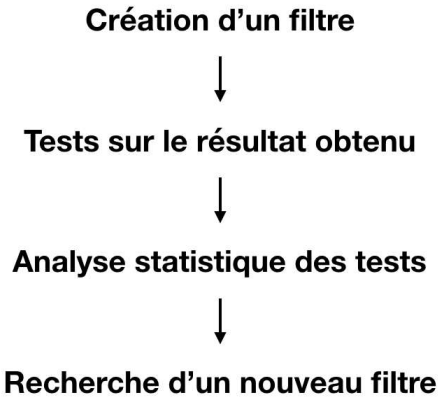
/---*

Puis, un fonctionnalité "caster" a été ajoutée au programme après plusieurs séries de tests et d'analyse des résultats. Le caster ayant pour but de ne garder que la phrase, ou partie cible de la phrase qui contient la correction car souvent le fichier tiré du site contient un ensemble de phrases au sein duquel se trouve une correction sur une seule phrase.

Ensuite, il y a eu la mise en place de plusieurs filtres tel que celui permettant d'éviter les retour sur correction : souvent utilisé pour de la paraphrase, ou encore celui sur des modifications de chiffres : une erreur sur une date, un salaire, un nombre de personnes.

Enfin une fois l'installation de grew terminée, il fut possible de tester les résultats obtenus après l'application des filtres et casters nous permettant ainsi la mise en place d'un protocol de travail :

*---/



B) Mise en place de différents filtres

1. Filtre sur les numéros

2. Filtre sur caractères spéciaux

3. ...

III) Résultats

A) Statistiques

B) Résultat final

C) Ouverture

Conclusion

Annexes

Bibliographie

Liens utiles:
https://fr.wikipedia.org/wiki/Wikipédia:Liste_de_fautes_d%27orthographe_courantes

bibliothèque d'extraction de connaissance xml en java:
http://blog.paumard.org/cours/xml/chap01-introduction-api-java.html#d0e86

lien d'installation de grew :
http://grew.fr/parsing/

Glossaire

Base de cas :
Filtre :

Déclaration contre le plagiat

Résumé