Master Informatique

Rapport du projet d'initiation à la recherche

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

Année 2018-2019

Ètudiants : Alex Ginestra
Christopher Klein

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

Décharge de responsabilité

Nous soussignons Alex Ginestra et Christopher Klein, déclarons avoir pris connaissance de la charte des examens et notamment du paragraphe spécifique au plagiat.
Je suis pleinement conscient(e) que la copie intégrale sans citation ni référence de documents ou d'une partie de document publiés sous quelques formes que ce soit (ouvrages, publications, rapports d'étudiant, internet etc...) est un plagiat et constitue une violation des droits d'auteur ainsi qu'une fraude caractérisée.
En conséquence, je m'engage à citer toutes les sources que j'ai utilisées pour produire et écrire ce document.

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 Bruno Guillaume, Yves Lepage, Jean Lieber et Emmanuel Nauer pour leur accueil et leur sympathie.

Sommaire

Table des matières

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 franÃ§aise 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

Le TAL (traitement automatique des langues) ou TALN (traitement automatique du langage naturel) est un domaine dont le but est de créer des outils de traitement de la langue naturelle pour diverses applications. Le traitement automatique du langage naturel couvre de très nombreuses disciplines de recherche. Ces applications servent notamment dans le domaine de la syntaxe, de la sémantique, du traitement de la parole ou encore dans le domaine de la fouille de données.

Les premiers travaux remontent en 1950, pendant la guerre froide, ou Alan Turing expose un test, communément appelé le «test de Turing», qui vise à mesurer la capacité d'un programme informatique à communiquer avec un humain. Ce test permet de mesurer «l'intelligence» d'une machine. Depuis, le traitement automatique des langues a bien évolué et est devenu un domaine pluridisciplinaire alliant la linguistique, l'informatique, l'intelligence artificielle pour créer des applications de plus en plus complexes nous permettant de simplifier nos tâches quotidiennes.

Un correcteur automatique de la langue est un exemple d'outil utilisant des domaines du traitement automatique des langues. Il utilise des discipline de recherche variés tel que l'analyse syntaxique, la correction orthographique et d'autre plus spécifique tel que la délimitation de phrase ou la morphologie. Un outil tel que celui-ci est un monstre de conception et de développement et les meilleurs correcteurs automatique connues à ce jour ont été développés par des entreprises internationales comme Microsoft, Google ou encore Apple.

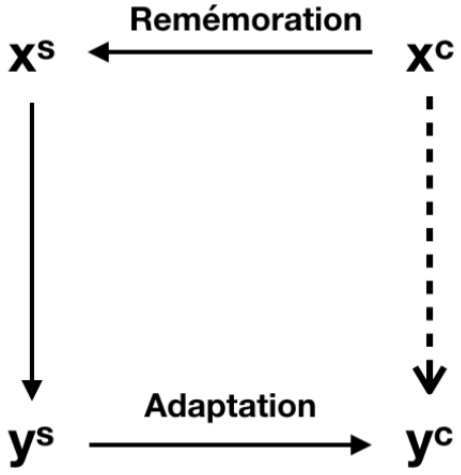
Contrairement aux cas décrit précédemment, notre sujet consiste à créer un système permettant la correction automatique de phrases franÃ§aise en s'appuyant sur le raisonnement à partir de cas (RàPC). Pour répondre à cette problématique, nous allons tout d'abord introduire le raisonnement à partir de cas et expliquer comment il va nous permettre de corriger des phrases. Puis nous aborderons une partie plus pratique ou nous détaillerons les différentes étapes permettant de résoudre notre problématique. Nous parlerons ensuite des résultats obtenus suite à nos développements et finirons par introduire les suites possibles à notre projet.

1 Analyse du problème et organisation

1.1 Raisonnement à partir de cas

Le raisonnement à partir de cas (noté RàPC) est un raisonnement semblable a celui des humain qui consiste à résoudre des problèmes de la vie quotidienne s'appuyant sur des expériences semblable rencontrés par le passé. Grâce aux expériences passées déjà résolues, nous arrivons à en déduire une solution qui automatiquement s'ajoute à notre expérience. La reproduction de cet exemple un grand nombre de cas nous permettra d'avoir une expérience telle que nous arriverons à déduire une solution de n'importe quel problème.

Nous voyons donc qu'il y a deux principales étapes dans ce raisonnement. Premièrement, il y a ce qu'on appellera la «remémoration», qui est le rapprochement entre notre problème actuel (que nous appellerons problème cible) et un problème qui a déjà été résolu dans le passé (qu'on nommera problème source). La seconde étape s'appelle l'»adaptation». Ce principe permet d'obtenir une solution de notre problème source en modifiant la solution du problème cible.



Nous voyons donc qu'il est possible de résoudre des problèmes grâce au raisonnement à partir de cas. Néanmoins, pour pouvoir les résoudre, nous devons avoir des problèmes similaire déjà résolus. Cet ensemble de couples (problème, solution) résolu s'appelle une base de cas. En plus du couple (problème, solution), une explication est ajoutée et permet d'améliorer l'étape de la remémoration.

Voici un exemple de résolution de problème grâce au raisonnement à partir de cas : Voici le problème que nous rencontrons (problème cible) : «Tu n'as pas manger». Nous avons dans notre base de cas un couple tel que celui-ci : «Il a recommencer» est le problème source et «Il a recommencé» est la solution du problème source. La remémoration va donc rapprocher notre problème cible du problème source et en utilisant la solution du problème source ainsi que l'explication, l'Adaptation va pouvoir nous fournir une solution de notre problème cible qui sera «Tu n'as pas mangé».

Pour parvenir à résoudre notre problématique qui est la correction automatique du franÃ§ais à l'aide du raisonnement à partir du cas, il nous faut donc une base de cas qui permette en théorie de corriger toutes les erreurs possible du franÃ§ais. En prenant en compte tous les types d'erreurs possibles (grammaire, orthographe, conjugaison, etc.), il y aurait un nombre immense de cas à définir dans notre base, ce qui est impossible à construire à la main. Le but de notre projet est donc de construire une base de cas à l'aide d'outils diverses et variés qui serait totalement automatisée.

1.2 Travail existant

Une première piste nous a été fournie par un groupe d'étudiants de L3 composé de M. Giang, M. Levy, M. Ly, qui avaient travaillé sur un projet intitulé corrector. Le projet consistait à faire un site internet capable d'apporter une correction à une phrase fausse donnée en entrée. Cette correction devait se faire à l'aide d'une base de cas qui pouvait s'enrichir avec des interactions humaines (utilisateur/administrateur du site).

Notre début d'étude a donc été guidé par les moyens mis en oeuvre pour effectuer un remplissage automatique de leur base de cas initiale, et plus particulièrement un : les corpus de WiCoPaCo. Le site WiCoPaCo met en libre accès des fichiers au format XML contenant des phrases, ou parties de phrases avec une correction effectuée ainsi qu'un éventuellement commentaire laissé par l'auteur de la correction. Ces fichiers sont le résultat des corrections faites par les administrateurs des pages Wikipédia, ce qui nécessite une correction étant donné que les contenu des pages est apporté par des utilisateurs. Les fichiers en question contiennent donc des centaines de milliers de cas composés de la manière suivante : le groupe de phrase avant modification avec la mise en évidence de la faute, suivi du même groupe de phrase avec la correction apportée également mise en évidence.

L'intérêt principal de ces fichiers étant l'énorme masse de données qu'ils contiennent, nous permettant ainsi d'en extraire un grand nombre de cas. Cependant, même si cette solution semble être idéale et simple à mettre en place, il s'avère qu'elle est loin d'être parfaite. Car sur ces corrections, une partie étant des corrections de contenu, une autre étant des reformulations, et bien d'autres types de corrections n'étant pas des erreurs de franÃ§ais mais sont pourtant contenues dans ces fichiers. La problématique de l'épuration de cette énorme masse de données se pose donc.

Face à ce problème, le groupe d'étudiants de L3 avaient mis en place un script python qui prenait un fichier XML en entrée et produisait en fichier CSV en sortie. Le script s'occupait aussi de la suppression de certains cas : les retours en arrière. Il ne retenait donc pas les cas qui étaient des retours sur correction, c'est à dire lorsque le correcteur transformait une phrase A en phrase B, puis transformait à nouveau la phrase B en phrase A.

C'est donc en reprenant cette base de travail que nous avons débuté notre projet, dans l'optique de pouvoir épurer cette énorme masse de données à l'aide de filtres pour obtenir uniquement des cas de corrections de langue.

1.3 Mise en place du projet

Pour mettre en place notre projet, nous avons donc grandement utilisé le travail déjà effectué par nos collègues qui nous ont précédés. Nous avons décidé d'utiliser l'énorme quantité de cas que nous fournissait les fichiers XML de WiCoPaCo pour créer notre base de cas automatique. Ces fichiers regroupant plus de 200 000 cas, elle serait suffisamment conséquente pour couvrir un maximum d'erreur de franÃ§ais. Néanmoins, comme cela a ete dit ci dessus, nous ne pouvons pas seulement transformé ces fichiers directement en une base de cas car un grand nombre de cas ne sont pas utilisable. Nous devons donc reprendre le travail qui a été fait en amont et continué à filtrés les cas jusqu'à obtenir un ensemble de cas correctes.

à la suite d'une réflexion sur le développement de notre projet, nous avons décidé de ne pas reprendre les travaux effectués par les étudiants précédent pour plusieurs raisons : premièrement, bien que nous comprenions l'idée directrice du développement, nous n'avions pas toutes les subtilités pour comprendre parfaitement le code. De plus, nous avions dans l'optique d'implémenter plusieurs filtres et que la création de ceux-ci soit facile. C'est pourquoi nous nous sommes résolues a utilisé le langage orienté objet Java pour le développement de notre application car c'est un langage que nous avons eu l'habitude de cÃ´toyer au cours de nos études, qui permet de lire et d'écrire des fichiers facilement et qui permet, une fois le projet structuré, une implémentation simple et rapide de nouvelles fonctionnalités.