GRAC: GRAmmar Checker

Maxime Biais

2 février 2005

1 Introduction

Le projet GRAC a pour but de créer un correcteur grammatical open source indépendant de la langue. Après avoir exposer brièvement quelques généralités sur la correction grammaticale, nous expliquerons comment nous entendons mener notre projet. Le but de GRAC est de proposer une correction basée sur l'apprentissage et donc indépendante d'un langage particulier.

2 A propos de la correction grammaticale

La correction grammaticale est un véritable enjeux dans le traitement du langage naturel et génère beaucoup d'enthousiasme de la part des maisons d'édition de logiciels car le marché en est très demandeur. Cependant le correcteur grammatical infaillible n'a pas encore vu le jour.

2.1 Des fautes pour êtres corrigées

On distingue plusieurs types d'erreurs dans un texte :

- Les fautes d'orthographe correspondent à l'utilisation d'un mot non reconnu, c'est à dire que ce mot n'est pas présent dans le dictionnaire utilisé. Nous reviendrons sur la notion de dictionnaire et son impacte sur le correcteur grammatical
- Les fautes de grammaire correspondent à l'utilisation erronnée d'un mot présent dans le dictionnaire. Ce peut être, entre autre, une faute d'accord avec un participe passé, l'emploi du mauvais genre ou nombre.
- Les fautes de sens correspondent aux fautes sémantiques, c'est à dire à une utilisation involontaire d'un mot à la place d'un autre, ou d'un mot hors contexte.

On l'aura compris seules les fautes d'orthographe et de grammaire sont corrigées de nos jours, avec plus ou moins d'efficacité. Les fautes de sens ne sont quant à elles pas détectées d'autant plus qu'elles font intervenir la notion de contexte. En effet l'emploi d'un mot inopportun peut-être volontaire dans un contexte précis.

2.2 Des fautes d'ortographe à la correction grammaticale

L'orthographe est liée dans un texte tapuscrit à un dictionnaire de mots. La qualité du correcteur orthographique depend directement de ce document car il doit contenir un maximum de mots. Pourtant on remarque plusieurs probleme sur ce point :

- Les bases lexicales sont incomplètes, particulièrement lorsque l'on utilise des mots techniques.
- Les mots composés et les expressions sont mal gérées, particulièrement lorsqu'il n'y a pas de trait d'union

– En cas de mots inconnnus, les mots proposés en correction sont trop nombreux et pas toujours proposés dans un ordre pertinent.

Néanmoins les vérificateurs orthographiques ont été adoptés par la plupart des traitements de texte. L'insuffisance au niveau des correcteurs grammaticaux n'est pas due aux correcteurs orthographique dans un tout premier lieu mais à leur fiabilité très médiocre. C'est cette piêtre fiabilité qui fait renoncer bon nombre d'utilisateur à leur utilisation. En effet les utilisateurs solides en grammaire ont peur, à raison, que le correcteur grammitical leur ajoute des fautes. Tandis que les utilisateurs faibles en grammaire n'ont aucune confiance et moyen de se rendre compte de la pertinence des corrections.

2.3 Dilemne

Corriger l'expression écrite apparaît comme une vraie difficultée à laquelle il est très difficile de répondre. Au-delà des erreurs classiques d'accord, de genre, de nombre, on se retrouve confronté à ce qu'a voulu dire l'auteur et on commence à toucher au sens. En effet suivant le contexte, certaines formes peuvent être acceptées, tandis qu'elle ne le seront pas dans un autre. On pense particulièrement à l'emploi d'un type de langage particulier (familier, courant, soutenu) mais on peut aussi s'interroger sur la coutûme par rapport aux règles.

Ainsi par exemple, doit-on réclamer un indicatif ou un subjonctif dans une proposition débutant par "après que" ? L'Academie et les puristes réclament l'indicatif, mais l'usage lui préfère depuis longtemps le subjonctif et une phrase comme "après qu'il est allé à l'usine " paraît suspecte.

Enfin, le correcteur grammatical ne doit pas se substituer à l'apprentissage de la grammaire mais doit être un outil de formation perpetuelle de maîtrise de la langue. C'est pourquoi il est nécessaire pour un correcteur grammatical professionnel de fournir des messages clairs et utiles rappelant les règles élémentaires et justifiant la correction de l'erreur.

3 Fonctionnement général de GRAC

L'architecture de GRAC est illustré par la figure 1. Le principe de la phase d'execution est simple :

– On prend le texte en entrée, on le passe dans un analyseur lexical pour le découper en mot et en phrase.
– On utilise un Part Of Speech Tagger pour affecter à chaque mot du texte qui va subir la correction un tag qui correspond à sa nature. Par exemple dans la phrase : “Je suis un ours”, on affectera au mot “je” le tag : “sujet singulier”, au mot “suis” le tag : “verbe première personne singulier”.
– Après cette phase, il suffit de vérifier que les tags qui se suivent dans une phrase suivent des règles de grammaires que l'on aura définit.

Pour utiliser GRAC il faut d'abord lancer une phase d'apprentissage dont le but est double :

1. Créer la base de connaissances du Part Of Speech Tagger probabiliste.
2. Créer une base de règles de grammaire automatiquement déduites d'un corpus sans fautes.

4 Part Of Speech Tagger : POST

Le but du Part Of Speech Tagger est d'etiquetter chaque mot du texte en entrée par ses caractéristiques grammaticales. Le POST fonctionne en plusieurs phases :

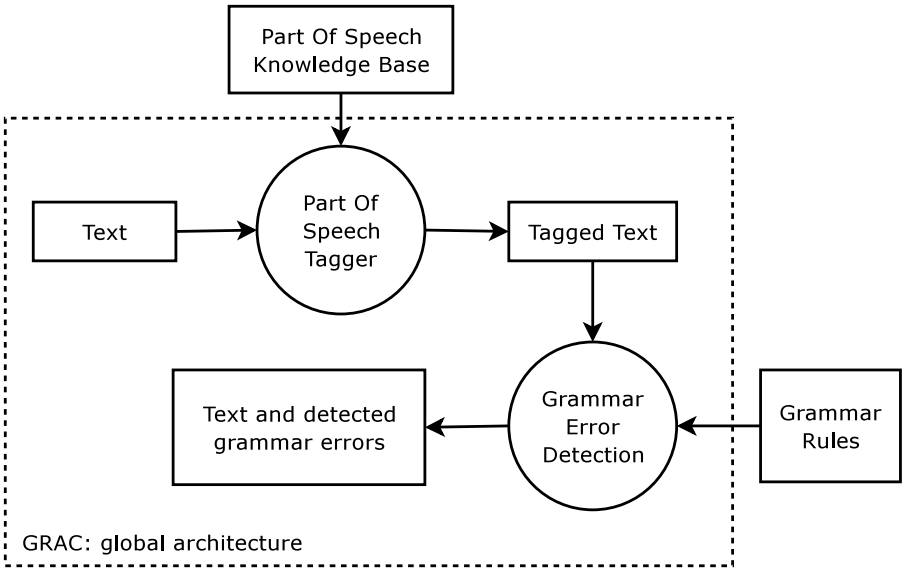

FIG. 1 – Execution de GRAC.

1. On pose les tags sur tout les mots qui ne posent pas de problème d'ambiguité. Par exemple le mot "artichauts" est un nom commum masculin pluriels et ce quelque soit le contexte dans lequel il est utilisé.
2. On utilise le POST probabiliste pour déterminer les mots ambiguës. Il utilise deux bases de connaissances :
   (a) une recense chaque mot associé aux probabilités qu'il soit etiquetté de tel ou tel tag. Par exemple le mot "chasse" est utilisé comme verbe avec une probabilité de 0.7 et comme nom avec une probabilité de 0.3. Les probabilités sont bien sur différentes suivant le corpus d'apprentissage utilisé.
   (b) une base qui contient toutes les possibilités d'associer 3 tags à la suite. Par exemple un nom commun est précédé d'un article et suivit d'un adjectif avec une probabilité de 0.6. Ces probabilités sont encore déterminés à partir du corpus d'apprentissage.

   Les deux bases sont associées pour déterminer le tag le plus probable à utiliser sur les mots qui ne sont pas encore étiquettés.
3. Il est possible que tout les mots du textes ne soit pas présent dans la première base. C'est à ce moment qu'intervient le POST basé sur des règles qui va utiliser différentes règles dépendantes de la langue pour déterminer le tag des mots inconnus.

L'architecture des POSTs sont illustrés dans la figure 2.

4.1 POST basé sur des règles

Parfois un mot est inconnu dans le dictionnaire des mots étiquetés et il nous est nécessaire de déduire son type pour pouvoir améliorer la correction globale.En effet il vaut mieux avoir un minimum de mots dont le type est inconnu. Pour cela on tente d'étudier l'orthographe du mot et particulièrement sa terminaison et d'en tirer des généralités. Par exemple un mot terminant par -isme est a coup sur un nom commun masculin.

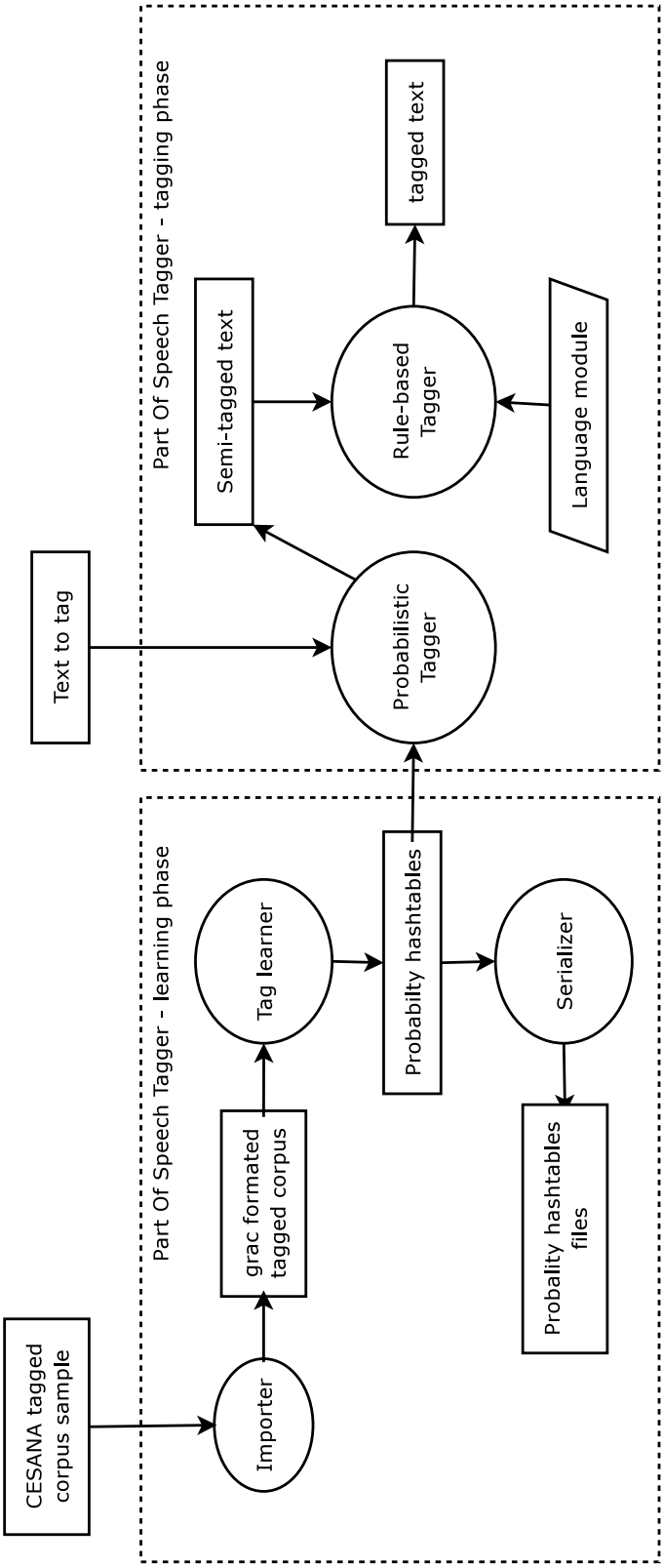

FIG. 2 – Architecture du POST.

Cette étude est très dépendante de la langue et on ne trouve pas toujours de règles génériques, particulièrement en francais où les mots sont souvent distinctifs et pas formés à partir d'une base et d'un suffixe générique. En anglais en revanche, on peut distinguer quelques règles sympathiques :

- Si la terminaison du mot est de la forme -dom, -ment, -tion, -sion, -ance, -ence, -er, -or, -ist, -ness, -iciy, alors on peut prédire un nom commun sigulier
- Si le mot termine en -ly alors on peut prédire un adverbe.
- Si le mot se termine en -ive, -ic, -al, -able, -y, -ous, -ful, -less on peut prédire un adjectif.
- Si le mot termine en -ize, -ise, -ate on peut prédire une forme verbale à l'infinitif.

Il n'est pas aussi aisé en francais de réaliser ce type de reconnaissance mais nous pouvons tout de même répertorier les quelques régles suivantes :

- Si la terminaison du mot est de la forme -isme, alors on peut prédire un nom commun singulier masculin.

5 Grammar Error Detection : GED

Cette partie n'est pas encore codée dans GRAC. Les sections suivantes présentent 2 façon complémentaires de détecter des erreurs de grammaires.

5.1 GED Probabiliste

Le GED Probabiliste ressemble au POST probabiliste. La phase d'apprentissage se base sur un corpus sans fautes de grammaires que l'on va etiquetter. Ensuite on va garder l'enchaînement de chaque phrase ou du moins de morceaux de phrases. Un enchaînement est une suite de tags, on peut aussi les appeler des règles de grammaires déduites.

Une fois ces enchaînements appris, l'execution est très simple. On passe chaque phrase du texte à corriger dans chacune des règles de grammaires déduites et si un phrase ne rentre dans aucune de ces règles, c'est qu'elle est sûrement erronées.

5.2 GED basé sur des règles

Le principe d'execution est identique que le GED probabiliste. La différence est que les règles ne sont pas déduites d'un corpus mais écrites à la main. Cette fois on peut déterminer des règles "fausses", c'est à dire des fautes de grammaire courantes que l'on voit souvent. L'interêt de ce type de règles est que l'on peut présenter une solution. Par exemple on fera une règle : Les articles pluriels sont toujours suivit d'un adjectif pluriel ou d'un nom commun pluriels. Si jamais une phrase qui contient un article pluriel ne passe pas dans cette règles, on pourra indiquer à l'utilisateur qu'il faut soit singulariser l'article, soit mettre au pluriel l'adjectif ou le nom commun suivant.

6 Divers

6.1 Avancement du projet

Pour le moment seul l'analyseur lexicale et les 2 types de POST ont été codés. Il nous reste donc toute la partie GED. Notre principal problème est de trouver un corpus étiquetté en Français de préférence assez gros (plus d'un million de mots). Pour le moment nous travaillons sur un simple

échantillon du corpus CESANA. Celui-ci suffit pour tester le POST mais ne nous permet pas de réaliser une base de connaissance assez importante pour que GRAC soit un jour exploitable.

Pour rendre GRAC utilisable il nous faut donc :
- Trouver des corpus etiquettés et des corpus sains (sans fautes de grammaire et d'orthographe) dans un maximum de langue.
- Écrire le module de detection d'erreur grammaticale probabiliste ainsi que le module de detection d'erreur grammaticale basé sur des règles.

6.2 Licenses

Tous le codes de GRAC est soumis à la GPL quand à la documentation et aux explications techniques ils sont soumis à la FDL. Ces deux licences sont disponible à cette adresse : `http://www.gnu.org/licenses/licenses.html`

6.3 Hébergement du projet

GRAC est hébergé sur SourceForge : `http://sourceforge.net/projects/grac/`.

6.4 Note

Ce document n'a été soumis à aucun détecteur d'erreurs grammaticales.