TP3 - Etiquetage morpho-syntaxique

Programmation pour le TAL - M1 SDL & SCA

7 mars 2012

Pour chacun des exercices, sauvegardez le script Python dans un fichier qui a comme nom *exo < numéro de l'exercice >.py* puis mettre tous les fichiers dans une archive .zip qui a comme nom *TP3_<nom de l'étudiant>_<prénom de l'étudiant >.zip*. A la fin de la séance, envoyez l'archive en fichier attaché à l'adresse *perrier@loria.fr* en mettant comme objet du message *programmation pour le TAL : TP3*.

1 Introduction à l'étiquetage morpho-syntaxique à la Brill

L'étiquetage morpho-syntaxique est une opération qui s'insère dans la chaîne d'analyse du langage naturel et qui est très utile pour lever les ambiguïtés lexicales qu'il contient. Elle consiste à attacher à chacun des mots d'un corpus une étiquette donnant des informations morpho-syntaxiques sur ce mot : sa catégorie grammaticale mais aussi d'autres informations comme par exemple le mode pour un verbe. Un jeu d'étiquettes est préalablement choisi soigneusement puis tout le problème est d'attacher à chaque mot du corpus l'étiquette correcte (on suppose alors qu'il n'y a qu'une étiquette possible par mot).

On veut écrire un étiqueteur morpho-syntaxique à la Brill. Un tel étiqueteur est fondé sur des règles apprises à partir d'un corpus dont on sait que l'étiquetage est correct. Les phases de construction sont les suivantes :

1) A partir du corpus d'apprentissage étiqueté C, on construit un lexique L des mots qu'il contient associés à leur étiquette la plus fréquente.

   Choisissons un corpus d'apprentissage artificiel :
   C = ([('la', 'DET'), ('belle', 'ADJ'), ('ferme', 'V'), ('la', 'DET'), ('porte', 'N'), ('.', 'PONCT'), ('la', 'DET'), ('belle','ADJ'), ('fille', 'N'), ('est', 'COP'), ('ferme','ADJ'), ('.', 'PONCT'), ('une','DET'), ('belle','N'), ('la','PRO'), ('porte','V'), ('.', 'PONCT'), ('une','DET'), ('belle','ADJ'), ('fille','N'), ('la','PRO'), ('porte','V'), ('.', 'PONCT'), ('il','PRO'), ('la','PRO'), ('porte','V'), ('.', 'PONCT')])

   Le lexique obtenu est alors :
   L = {'est' : 'COP', 'la' : 'PRO', 'belle' : 'ADJ', 'porte' : 'V', '.' : 'PONCT', 'fille' : 'N', 'une' : 'DET', 'ferme' : 'ADJ', 'il' : 'PRO'}

2) A l'aide du lexique L, on ré-étiquette le corpus d'apprentissage et on obtient un corpus C_0, qui contient bien entendu des erreurs.

   Dans notre exemple, nous obtenons :
   C_0 = *[('la', 'PRO'), ('belle', 'ADJ'), ('ferme', 'ADJ'), ('la', 'PRO'), ('porte', 'V'), ('.', 'PONCT'), ('la', 'PRO'), ('belle', 'ADJ'), ('fille', 'N'), ('est', 'COP'), ('ferme', 'ADJ'), ('.', 'PONCT'), ('une', 'DET'), ('belle', 'ADJ'), ('la', 'PRO'), ('porte', 'V'), ('.', 'PONCT'), ('une', 'DET'), ('belle', 'ADJ'), ('fille', 'N'), ('la', 'PRO'), ('porte', 'V'), ('.', 'PONCT'), ('il', 'PRO'), ('la', 'PRO'), ('porte', 'V'), ('.', 'PONCT')]*

   Il comporte 6 erreurs. Pour corriger ces erreurs, on va chercher des règles de modification des étiquettes en fonction de leur contexte droit ou gauche, ou des deux.

3) En essayant toutes les possibilités, on cherche la règle R_0 qui corrige le maximum d'erreurs dans C_0. On l'applique et on obtient un corpus C_1.

   Une règle se présente comme un quadruplet : (étiquette à modifier, nouvelle étiquette, contexte gauche, contexte droit). Le contexte gauche (droit) est l'étiquette qui doit être immédiatement à gauche (à droite) de l'étiquette à modifier pour que la règle s'applique. Ce contexte peut être indéfini (on peut l'indiquer par le symbole $*$).

   Dans notre exemple, la règle R_0 est : ('PRO', 'DET', '$*$', 'ADJ')

4) On itère cette opération à partir de C_1 jusqu'à ce que le nombre d'erreurs tombe au-dessous d'un certain seuil. On obtient ainsi une liste de règles $R_0, R_1, \ldots, R_n$.

5) On considère maintenant le corpus qu'il s'agit d'étiqueter. On commence par l'étiqueter à l'aide du lexique L (cela sous-entend que tous les mots du corpus sont dans le lexique) puis on lui applique dans l'ordre les règles $R_0, R_1, \ldots, R_n$. Le résultat est le corpus étiqueté qu'on obtient après l'application de la règle R_n.

2 Construction d'une boite à outils pour un étiqueteur à la Brill

Les exercices suivants consistent à construire une boite à outils de fonctions destinées à réaliser cet étiqueteur. Il est nécessaire de tester chaque fonction au fur et à mesure. Vous pourrez la tester sur l'exemple présenté dans la section précédente mais vous pouvez la tester sur un corpus plus gros, le corpus de Brown. Le chapitre 5 *Categorizing and Tagging Words* du tutoriel de NLTK vous indiquera comment récupérer ce corpus étiqueté.

Le premier exercice va permettre de réaliser la phase 1 de l'algorithme décrit à la section précédente.

Exercice 2.1 *Ecrivez une fonction* apprendre_etiquettes(corpus) *qui a pour paramètre un corpus étiqueté* corpus *et qui retourne un lexique donnant pour chaque mot l'ensemble des étiquettes rencontrées avec le nombre de leurs occurrences. Le corpus se présente sous forme d'une liste de couples (mot, étiquette) et le lexique sous forme d'un dictionnaire dont chaque entrée est elle-même un dictionnaire.*

Ecrivez ensuite une fonction choix_etiquette(liste_etiquette) *qui a pour paramètre un dictionnaire dont les clés sont des étiquettes et les valeurs correspondantes un nombre d'occurrences et qui retourne l'étiquette la plus fréquente dans le dictionnaire. Quand il y a plusieurs solutions possibles, on choisit la première qui arrive.*

A partir des deux fonctions précédentes, écrivez une fonction apprendre_lexique(corpus) *qui a pour paramètre un corpus étiqueté* corpus *et qui retourne un lexique donnant pour chaque mot l'étiquette la plus fréquente selon ce corpus.*

L'exercice suivant permet d'initialiser l'étiquetage d'un corpus et de comparer des corpus étiquetés (étape 2 de l'algorithme).

Exercice 2.2 *Ecrivez une fonction* reetiqueter_corpus(corpus,lexique) *qui a pour paramètre un corpus étiqueté* corpus *et un lexique de mots étiquetés* lexique *et qui retourne un nouveau corpus étiqueté où les étiquettes du corpus initial ont été remplacées par celles fournies par le lexique.*

Ecrivez une fonction comparer(corpus_reference, corpus_test) *qui permet de comparer un corpus étiqueté à tester par rapport à un corpus de référence avec les mêmes mots et avec des étiquettes correctes. La fonction doit retourner le nombre d'erreurs dans le corpus à tester.*

L'exercice qui suit s'attache à la détermination et à l'application des règles de ré-étiquetage (étape 3 de l'algorithme).

Exercice 2.3 *Ecrivez une fonction* appliquer_regle(corpus,ancienne_etiquette, nouvelle_etiquette, contexte_gauche, contexte_droit) *qui a pour paramètre un corpus étiqueté* corpus *et une règle de ré-étiquetage exprimée par les 4 paramètres suivants. Elle retourne un nouveau corpus où chaque occurrence de* ancienne_etiquette *est remplacée par* nouvelle_etiquette *quand elle est précédée de l'étiquette* contexte_gauche *et suivie de l'étiquette* contexte_droit*. On peut remplacer un seul des deux contextes par * qui indique que celui-ci est ignoré.*

Ecrivez une fonction choisir_regle(corpus_reference, corpus_test) *qui retourne la règle de modification d'étiquettes sous forme d'un quadruplet qui permet de corriger un maximum d'erreurs dans* corpus_test *,compte tenu de* corpus_reference.

On termine par deux fonctions, la première qui consiste à apprendre les règles à partir du corpus de référence (étape 4 de l'algorithme), et la seconde qui consiste à les appliquer à un corpus brut (étape 5 de l'algorithme).

Exercice 2.4 *Ecrivez une fonction* generer_regles(corpus_reference, corpus_test, seuil) *qui retourne une liste de règles permettant de corriger le corpus étiqueté* corpus_test *pour que le taux d'erreurs par rapport au nombre d'étiquette tombe au-dessous de* seuil*, compte tenu du corpus de référence* corpus_reference.

Ecrivez une fonction etiqueter_corpus(corpus_test, lexique, liste_regles) *qui prend en paramètres un corpus brut* corpus_test*, un lexique de mots étiquetés* lexique *permettant d'initialiser l'étiquetage du corpus brut et une liste de règles de ré-étiquetage à appliquer* liste_regles*. La fonction retourne un corpus étiqueté après application des règles.*