Manipulations avec le « taggeur » de Brill

L'« étiquetage » ou *tagging* d'un texte consiste à associer à chaque mot (au sens large) d'un texte une étiquette (très souvent une catégorie syntaxique, ou partie du discours) puisée dans une liste finie (voir figure 1), de sorte que le texte résultant puisse constituer l'entrée d'une analyse syntaxique, ou d'un système d'indexation, ou plus généralement d'un système de TAL quelconque.

La plupart des « étiqueteurs » sont fournis avec lexique et règles morpho-syntaxiques. Celui conçu par Éric Brill est différent. Son principe : à partir d'un corpus préalablement étiqueté (à la main), le système de Brill construit un « étiqueteur » par apprentissage. Plus précisément, l'apprentissage consiste à déduire un **lexique**, un ensemble de **règles lexicales** (pour deviner l'étiquette d'un mot nouveau), et un ensemble de **règles contextuelles**.

Ces trois ensembles d'informations constitués, l'« étiqueteur » est capable d'étiqueter un nouveau corpus, avec, vous le verrez, plus ou moins de succès.[1]

Manipulation

La première opération va donc consister en l'étiquetage à la main d'un corpus. Le corpus dont nous allons partir est accessible dans `~p71ii038/ouvert/corpusBrill`. Afin de simplifier la tâche, et de gagner du temps, ce corpus a déjà subi un certain nombre de transformations nécessaires à la bonne prise en compte par les outils de Brill :

- « Tokenisation » : des espaces ont été insérés comme séparateur autour des ponctuations ou dans divers cas particuliers.
- Mise en forme avec une phrase par ligne.
- Étiquetage de la ponctuation, et de divers mots non ambigus.

Corpus

- Créer un répertoire de travail, par exemple `brill` : dans la fenêtre « shell », taper `mkdir brill`, puis s'y placer : `cd brill`.
- Copier le fichier de corpus mentionné plus haut :
  `cp␣~p71ii038/ouvert/corpusBrill␣corpus.tag`.
- Lancer l'éditeur de texte *asedit* : `asedit &`. Cet éditeur crée une nouvelle fenêtre, avec les menus classiques d'un éditeur.
- Ouvrir le fichier `corpus.tag` sous *asedit*, et le réduire jusqu'à environ 6 ou 7 lignes, si possible prises au hasard dans le texte. Ce sont ces lignes qui seront étiquetées. Pour compter le nombre de mots de ce texte, l'enregistrer, et, dans la fenêtre de « shell », taper `wc -w corpus.tag`. S'il fait plus de 500 mots, le réduire encore.
- Sous *asedit*, étiqueter le corpus, en vous inspirant des étiquettes apparaissant dans la figure 1. **Le jeu d'étiquettes est un des paramètres affectant l'efficacité du taggeur.** Il faut être très soigneux en procédant à cet étiquetage, non seulement pour des raisons évidentes de cohérence, mais encore parce que l'apprentissage n'est pas très robuste, et peut échouer sur l'absence d'une étiquette ou d'un espace.
- Deux commandes utiles pour faire une vérification :
  `cat␣corpus.tag␣|␣tr␣'␣'␣'\012'␣|␣grep␣-v␣'/'`
  `cat␣corpus.tag␣|␣tr␣'␣'␣'\\012'␣|␣sort␣-u␣|␣more`

Une fois le corpus étiqueté, on paut procéder à l'apprentissage lui-même, qui va constituer dans le répertoire courant un certain nombres de fichiers, que vous êtes invités à regarder :

1. En vrai grandeur — 300 000 mots, par exemple —, on atteint facilement 95 % de réussite.

`ListeMots`	Liste des mots du corpus
`ListeMotsTags`	Liste des mots et de leur étiquette, tirée d'une moitié du corpus étiqueté
`ListePaires`	Liste des mots deux à deux, exploitée par l'algorithme d'étiquetage
`LexiqueBase`	Lexique (mots+étiquette) constitué pour la seconde phase de l'apprentissage
`ReglesLex`	Règles lexicales. Cf. figure 2
`ReglesCtxt`	Règles contextuelles. Cf. figure 3
`LexiqueFinal`	Lexique constitué pour servir par la suite

APPRENTISSAGE

– Récupérer le sh-script correspondant, qui s'appelle `apprend`, et se trouve dans le même répertoire que le corpus :
  `cp␣~p7lii038/ouvert/apprend␣.`
– Lancer la commande avec comme argument le nom du corpus étiqueté :
  `aprend␣corpus.tag`
– Suivre les affichages (un peu bavards).

Vous pouvez maintenant tester l'étiqueteur sur un autre corpus. La commande correspondant utilisera les fichiers `LexiqueFinal`, `ListePaires`, `ReglesLex` et `ReglesCtxt`.

TAGGAGE

– Constituer, à partir du fichier `corpusBrill`, un corpus de test, non étiqueté. Soit `essai.brut` le fichier correspondant.
– Récupérer le sh-script correspondant, qui s'appelle `etiq`, et se trouve dans le même répertoire que le corpus :
  `cp␣~p7lii038/ouvert/etiq␣.`
– Lancer la commande, qui placera le résultat du taggage dans `essai.tag` :
  `etiq␣essai.brut␣essai.tag`
– Observer le résultat...

FIG. 1 - *Exemple de corpus étiqueté*

```
L'/Art usine/N ,/, qui/PRel devrait/V être/Vinf implantée/PPs à/Prep Eloyes/NP (/( Vosges/NP
)/) représente/V un/Art investissement/N d'/Prep environ/Adv 3,7/Num milliards/N de/Prep yens/N
(/( 148/Num milliards/N de/Prep francs/N )/) ./.
Elle/Pro fabriquera/V ,/, dans/Prep un/Art premier/Adj temps/N ,/, le/Art produit/V liquide/N
qui/PRel entre/V dans/Prep le/Art processus/N des/Art photocopies/N ainsi/Adv que/Adv des/Art
pièces/N détachées/Adj pour/Prep la/Art filiale/N de/Prep Minolta/NP en/Prep RFA/NP ./.
```

FIG. 2 - *Exemples de règles lexicales*

`NN s fhassuf 1 NNS 302`	*Change the tag from NN to NNS if the word has the suffix "s".*
`ed hassuf 2 VBN 120`	*Change the tag to VBN if the suffix is "ed" (regardless of current tag) (note the difference between fhassuf and hassuf)*
`ly addsuf 2 JJ 57`	*Change the tag to JJ if adding the suffix "ly" results in a word.*
`- char JJ 42`	*Change the tag to JJ if the character "-" appears in the word.*
`NN to fgoodright VB 2`	*Change the tag from NN to VB if the word is ever seen immediately to the right of the word "to".*
`un deletepref 2 JJ 6`	*Change the tag to JJ if deleting the prefix "un" results in a word.*

FIG. 3 - *Exemples de règles contextuelles*

`NN VB PREVTAG TO`	*Change a tag from NN to VB if the previous tag is TO*
`VBP VB PREV1OR2OR3TAG MD`	*Change a tag from VBP to VB if one of the 3 previous tags is MD*

Annexe : sh-scripts

```
#! /bin/sh

SOURCE=/usr/local/RULE_BASED_TAGGER_V1.14/Utilities
LEARNER=/usr/local/RULE_BASED_TAGGER_V1.14/Learner_Code

if test $# = 1
then
  TAGGE=$1
else
  TAGGE=corpus.tag
fi

Courant=`pwd`
P1=/tmp/BrillP1.$$
P2=/tmp/BrillP2.$$
P2Brut=/tmp/BrillP2Brut.$$
BRUT=/tmp/BrillBrut.$$

# Création du fichier brut par dé-taggage
cat $TAGGE |
        perl $SOURCE/tagged-to-untagged.prl \
        > $BRUT

cat $BRUT |
        perl $SOURCE/wordlist-make.prl |
        sort +1 -rn |
        awk '{print $1}' \
        > ListeMots

echo "Liste des mots complète créée. (ListeMots)"

echo "Découpage du corpus en deux. ($P1)"
cat $TAGGE |
        perl $SOURCE/divide-in-two-rand.prl $P1 $P2

echo "Création d'une liste de mots à partir de la première partie."
cat $P1 |
        perl $SOURCE/word-tag-count.prl |
        sort +2 -rn \
        > ListeMotsTags
echo "Liste des mots étiquetés créée. (ListeMotsTags)"

cat $BRUT |
        perl $SOURCE/bigram-generate.prl |
        awk '{print $1, $2}' \
        > ListePaires
echo "Liste de paires de mots créée. (ListePaires)"

echo "Création des règles lexicales..."
perl $LEARNER/unknown-lexical-learn.prl ListeMots ListeMotsTags ListePaires\
        300 ReglesLex
echo "Fichier de règles lexicales créé. (ReglesLex)"

echo "Fin de la première partie. Tapez <Enter>."
read bidon
```

```
cat $P1 |
        perl $SOURCE/make-restricted-lexicon.prl \
        > LexiqueBase
echo "Lexique (mots taggés) d'apprentissage créé. (LexiqueBase)"

echo "On \"dé-tagge\" le deuxième demi-corpus..."
cat $P2 | perl $SOURCE/tagged-to-untagged.prl > $P2Brut

echo "On utilise la seconde moitié du corpus étiqueté..."
echo "que l'on commence par tagger naïvement. (fichier Naif)"
(cd /usr/local/RULE_BASED_TAGGER_V1.14/Bin_and_Data ;
tagger $Courant/LexiqueBase $P2Brut \
        $Courant/ListePaires $Courant/ReglesLex /dev/null\
        -w $Courant/ListeMots -S\
        > $Courant/Naif)

echo "Et maintenant on apprend les règles contextuelles,"
echo "en comparant Naif et $P2".

(cd /usr/local/RULE_BASED_TAGGER_V1.14/Bin_and_Data ;
contextual-rule-learn $P2 $Courant/Naif $Courant/ReglesCtxt $Courant/LexiqueBase)
echo "Fichier de règles contextuelles créé. (ReglesCtxt)"

echo "Pour finir, on crée le lexique final. (LexiqueFinal)"
cat $TAGGE |
        perl $SOURCE/make-restricted-lexicon.prl \
        > LexiqueFinal

rm -f /tmp/Brill*.$$
```

```
#! /bin/sh

if test $# = 2
then
  TAGGE=$2
  BRUT=$1
else
  echo "Usage : $0 <corpus brut> <corpus taggé>"
  exit 1
fi

Courant=`pwd`

(cd /usr/local/RULE_BASED_TAGGER_V1.14/Bin_and_Data ;
tagger $Courant/LexiqueFinal $Courant/$BRUT \
        $Courant/ListePaires $Courant/ReglesLex $Courant/ReglesCtxt \
        > $Courant/$TAGGE)
```