Linguistique de corpus

Outils et méthodes de traitement de corpus

Patrick Paroubek

LIMSI-CNRS
Dépt. CHM - Groupe LIR
Bât. 508 Université Paris XI, 91403 Orsay Cedex
pap@limsi.fr

mercredi 23 novembre 2016 / L3 - Cours 9
Un tagger en bash 1/2

Methodologie :

- Bien appréhender ce que l'on cherche à faire. Combien de paramètres en entrée, de quelle sortes, que représentent-ils ? Quel est le résultat attendu ?
- Identifier l'espace de recherche (toutes les valeurs possibles des paramètres d'entrée).
- Identifier les cas limites (les frontières de l'espace de recherche).
- Commencer par écrire un programme qui résoud un seul cas, un des plus simples.
- Puis modifier le programme pour prendre en compte un autre cas.
- et puis un autre etc.
- au bout d'un moment on a couvert tout l'espace de recherche

Pour créer un étiqueteur morpho-syntaxique, il va nous falloir un dictionnaire, constitué par exemple d'un fichier (par ex. mondico.txt) sur deux colonnes, la première contenant la forme et la seconde la liste de ses étiquettes, que l'on va constituer avec un éditeur.

```
le  PronomMasculinSingulier|ArticleDéfiniMasculinSingulier
livre NomCommunMasculinSingulier|VerbePrincipalIndicatifPrésent3PersonneSingulier
belle AdjectifQualificatifFémininSingulier
...
```

Pour interroger cet dictionnaire, nous utiliserons la commande `egrep` qui attrape les lignes contenant un motif de recherche.

```
#!/bin/bash

# le programme tagger1
# appel: ./tagger1 untexte.txt  dimaju-4.1.1 untexteavecetiquettesMS.txt

MONTEXTE=$1
MONDICO=$2
RESULTAT=$3

echo -n "" > $RESULTAT

for w in `cat $MONTEXTE | tr '\012' '\040'`
do
  TAG=`egrep $w $MONDICO`
  echo $TAG >> $RESULTAT
done
```

Le résultat est le suivant :

```
 tim>  cat montexte.txt
Methodologie :
Bien appréhender ce que l'on cherche à faire. Combien de paramètres en entrée, de quelle sorte
 tim> ./tagger1.sh montexte.txt dimaju-4.1.1 montextetagged.txt
 tim> cat montextetagged.txt

-t-on PRV:sg Binary file dimaju-4.1.1 matches
Bien ADV SBC:sg Binary file dimaju-4.1.1 matches

- ce PRV:sg -ce PRV:sg aberrance SBC:sg aberrances SBC:pl abjuratrice ADJ:sg abjuratrices ADJ
abandonnique ADJ:sg abandonniques ADJ:pl abaque SBC:sg abaques SBC:pl abdique VCJ:sg abdiquent

...etc...
```

On s'apperçoit qu'il y a un message inclus dans les sorties de *egrep* :

```
Binary file dimaju-4.1.1 matches
```

. En regardant la page du manuel pour egrep (*manegrep*) on apprend qu' il faut mettre l'option $-a$ à *egrep* si l'on veut qu'il disparaisse. Notez qu'en fonction de la configuration de votre shell ce phénomène peut ne pas se produire.

```
File and Directory Selection
    -a, --text
           Process a binary file as if it were text; this is equivalent to
           the --binary-files=text option.

    --binary-files=TYPE
           If  the  first  few  bytes of a file indicate that the file contains
           binary data, assume that the file is of type TYPE.  By default, TYPE is
           binary, and grep normally outputs either a one-line message saying that
           a binary file matches, or no message if there is no match.  If  TYPE
           is  without-match,  grep  assumes that a binary file does not match;
           this is equivalent to the -I option.  If TYPE is text, grep processes a
           binary file as if it were text; this is equivalent to the -a option.
           When processing binary data, grep may treat  non-text  bytes  as  line
           terminators;  for  example,  the  pattern  '.' (period) might not match
           a null byte, as the null byte might be treated as a line terminator.
           Warning: grep --binary-files=text might output binary garbage, which
           can have nasty side effects if the output is  a  terminal  and  if  the
           terminal driver interprets some of it as commands.
```

On s'apperçoit aussi que pour un mot donné (ici *ce*), notre patron de filtrage fourni à *egrep* est trop laxiste, car il capture d'autres lignes que celle concernant le mot ("aberrance, aberrances SBC :pl abjuratrice,...)

```
- ce PRV:sg -ce PRV:sg aberrance SBC:sg aberrances
```

Le mot à trouver dans le dictionnaire est toujours en début de ligne (donc nous allons l'indiquer avec ^) et est suivi d'un caractère tabulation (ajouté dans le patron avec un copier-coller). Par exemple :

```
egrep "^le	" mondico.txt
```

Le programme devient donc *tagger2* :

```
#!/bin/bash

# le programme tagger2
# appel: ./tagger2 untexte.txt  dimaju-4.1.1 untexteavecetiquettesMS2.txt

MONTEXTE=$1
MONDICO=$2
RESULTAT=$3

echo -n "" > $RESULTAT

for w in `cat $MONTEXTE | tr '\012' '\040'`
do
TAG=`egrep -a "^$w " $MONDICO`
echo $TAG >> $RESULTAT
done
```

Le problème de sélection est résolu (cf la ligne ce),

```
tim> head -n 10 montextetagged2.txt
.
: :
Bien ADV SBC:sg

ce PRV:sg DTN:sg PRO:sg
que SUB$ SUB REL ADV PRO:sg

cherche VCJ:sg
```

mais le premier mot (Méthodologie) a disparu. Avec le test suivant on s'apperçoit qu'il n'est pas trouvé dans le dictionnaire.

```
tim> egrep "^Methodologie "  dimaju-4.1.1
tim>
```

On va donc ajouter un teste pour afficher une étiquette "*INCONNU*" pour les mots non trouvés dans le programme *tagger*3.

```
#!/bin/bash

# le programme tagger3
# appel: ./tagger3 untexte.txt  dimaju-4.1.1 untexteavecetiquettesMS3.txt

MONTEXTE=$1
MONDICO=$2
RESULTAT=$3

echo -n "" > $RESULTAT

for w in `cat $MONTEXTE | tr '\012' '\040'`
do
TAG=`egrep -a "^$w " $MONDICO`
if [[ "$TAG" == "" ]];
then
  echo "$w INCONNU" >> $RESULTAT
else
  echo $TAG >> $RESULTAT
done
```

Le problème est résolu, mais on s'apperçoit qu'appréhendern'est pas trouvé

```
tim> cat untexteavecetiquettesMS3.txt
Methodologie INCONNU
: :
Bien ADV SBC:sg
appréhender INCONNU
ce PRV:sg DTN:sg PRO:sg
que SUB$ SUB REL ADV PRO:sg
... etc...
```

or il est dans le dictionnaire mais avec un accent différent, c'est donc un problème d'encodage :

```
 egrep hender dimaju-4.1.1  | egrep "^app"
apprhender VNCFF
apprhendera VCJ:sg
apprhenderai VCJ:sg
apprhenderaient VCJ:pl
apprhenderais VCJ:sg
apprhenderait VCJ:sg
apprhenderas VCJ:sg
apprhenderez VCJ:pl
apprhenderiez VCJ:pl
apprhenderions VCJ:pl
apprhenderons VCJ:pl
apprhenderont VCJ:pl
```

En chargeant le dictionnaire avec un navigateur WEB (e.g. mozilla) et en changeant l'encodage des caractères pour levisualiser (menu view->character encoding) on s'apperçoit que le fichiier n'est pas en UTF8 mais en ISO 8859-1 Western Europe. On le convertit avec au choix l'une des deux commandes suivantes (attention la première modifie le fichier original).

```
tim> recode latin1..utf8 utf8 dimaju-4.1.1

tim> iconv -f latin1  -t utf8 dimaju-4.1.1 > dimaju-4.1.1_utf8
```

Le problème est résolu appréhenderest trouvé

```
 tim> ./tagger3.sh montexte.txt dimaju-4.1.1_utf8 montextetagged3_utf8.txt
 tim> head -n 10 montextetagged3_utf8.txt
Methodologie INCONNU
: :
Bien ADV SBC:sg
appréhender VNCFF
ce PRV:sg DTN:sg PRO:sg
que SUB$ SUB REL ADV PRO:sg
l'on INCONNU
cherche VCJ:sg
à PREP
faire. INCONNU
...etc...
```