INTRODUCTION AU TALN

Préparation au travail Pratique

12 janvier 2014

Philippe Langlais

felipe@iro.umontreal.ca

RALI
Dept. Informatique et Recherche Opérationnelle
Université de Montréal

PLAN

1. Compter les mots dans un texte

2. Scripter en Shell

rali

COMPTER LES MOTS DANS UN TEXTE

COMPTER DES MOTS DANS UN CORPUS

- Un mot dans un **corpus** est appelé une **occurrence** (ou très souvent un **token**). C'est la réalisation d'un **type** particulier.

```
Ca c' est pour moi , le plus beau et le
plus triste paysage du monde . C' est le même
paysage que celui de la page précédente , mais
je l' ai dessiné une fois encore pour bien vous
le montrer .
```

- Il y a 43 occurrences dans ce corpus (en comptant les signes de ponctuation), mais il y a seulement 34 types (en distinguant majuscule/minuscule ; 33 sinon).
- 75% de ces types ont une **fréquence** de 1 dans ce corpus.

rali

FAUT-IL UTILISER UN "VRAI" LANGAGE DE PROGRAMMATION ?

C++ (ou autre langage du même type) :

```
cat corpus | frequence
```

Où `frequence` est un programme utilisant une structure de donnée de type hash-map. Voici à quoi ça peut ressembler avec la STL :

```
  while (cin >> s) {
    it = mots.find(s);
    if (it == mots.end())
       mots.insert(make_pair(s,1));
    else
       ++(it->second);
  }
```

rali

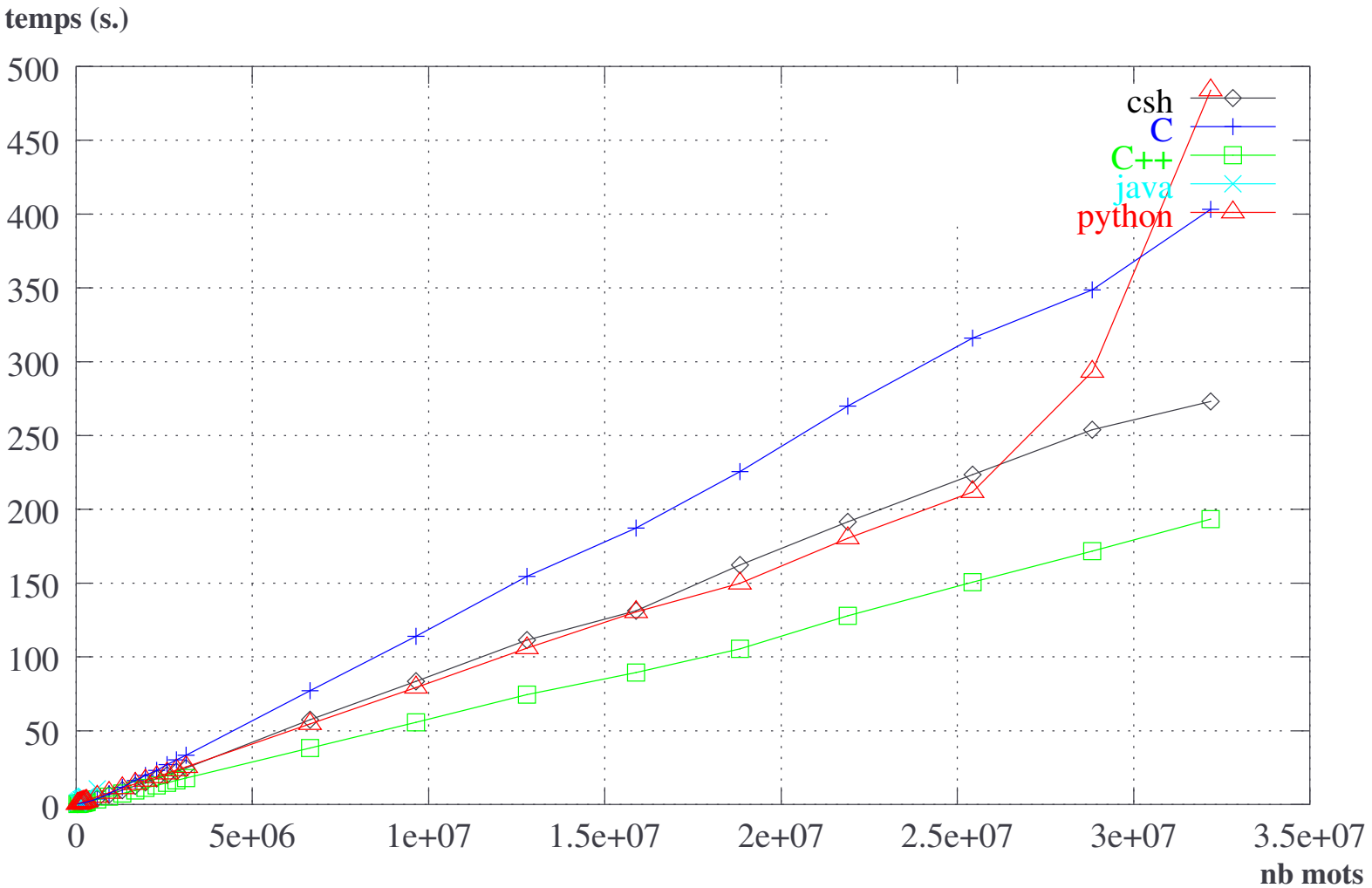
FAUT-IL UTILISER UN "VRAI" LANGAGE DE PROGRAMMATION ?

Langage de commande (shell scripts) :

```
1  cat corpus | sort | uniq -c | sort -k1,1n
```

rali

FAUT-IL UTILISER UN "VRAI" LANGAGE DE PROGRAMMATION ?



rali

TYPES, TOKENS ET FAITS

- 60 000 entrées dans le Petit Robert
- 75 000 dans le grand Robert.
- Vocabulaire moyen d'un individu : $\sim$ 5 000 mots

- Google (mars 2006) :

  > "We processed 1,011,582,453,213 words of running text and are publishing the counts for all 1,146,580,664 five-word sequences that appear at least 40 times. There are 13,653,070 unique words, after discarding words that appear less than 200 times."

http://googleresearch.blogspot.com/2006/08/all-our-n-gram-are-belong-to-you.html

rali

LOI DE ZIPF : $F \times R \approx CST$

mot	r	f	$f \times r$	mot	r	f	$f \times r$
the	1	1836714	1836714	done	200	16313	3262600
{sent}	2	1639250	3278500	children	300	10188	3056400
.	3	1383044	4149132	ensure	400	7880	3152000
is	10	504770	5047700	corporation	500	6414	3207000
not	20	221350	4427000	turner	1000	2915	2915000
à	30	133279	3998370	damage	2000	1204	2408000
?	40	98435	3937400	withdrawn	3000	658	1974000
all	50	81796	4089800	finances	4000	408	1632000
's	60	65562	3933720	neighbourhood	5000	282	1410000
those	70	53489	3744230	opposes	7000	153	1071000
his	80	46108	3688640	momentum	8000	117	936000
so	90	40810	3672900	forecasting	10000	73	730000
per	100	36099	3609900	rambled	50000	2	100000

- f = fréquence, r = rang

rali

SCRIPTER EN SHELL

- dans un **terminal** :

`ls`	liste le contenu du répertoire courant
`cd rep`	se déplace dans le répertoire `rep`
`mkdir rep`	crée le répertoire `rep` dans le répertoire courant
`cd`	se déplace dans le répertoire **home**
`pwd`	indique le répertoire courant

- beaucoup de commandes, certaines complexes :

```
1  find . -name "*.cpp" -exec grep -i 'funcZ(' {} \;
```

rali

```
1 echo "bonjour" | rev
```

`echo` affiche une chaine

`rev` renverse le texte qu'on lui donne, ligne à ligne

`c1 | c2` l'entrée de `c2` est la sortie de `c1`

```
1 echo "bonjour" | rev >! a
```

`c1 >! file` la sortie de `c1` est **redirigée** dans le fichier `file` (qui est écrasé s'il existe)

`c1 >> file` la sortie de `c1` est **redirigée** dans le fichier `file` (après le contenu existant)

rali

% man tr

```
TR(1)                                                                User Commands

NAME
       tr - translate or delete characters

SYNOPSIS
       tr [OPTION]... SET1 [SET2]

DESCRIPTION
       Translate, squeeze, and/or delete characters from standard input, writing to standard output.

       -c, -C, --complement
              use the complement of SET1

       -d, --delete
              delete characters in SET1, do not translate

       -s, --squeeze-repeats
              replace each input sequence of a repeated character that is listed in SET1 with a single occurrence of that character

       -t, --truncate-set1
              first truncate SET1 to length of SET2

       --help display this help and exit

       --version
              output version information and exit

       SETs are specified as strings of characters.  Most represent themselves.  Interpreted sequences are:

       \NNN   character with octal value NNN (1 to 3 octal digits)
```

rali

COMMANDE TR

```
% head -n 1 zola1.txt
```

Onze heures venaient de sonner à la Bourse, lorsque Saccard entra chez Champeaux, dans la salle blanc et or, dont les deux hautes fenêtres donnent sur la place.

```
% head -n 1 zola1.txt | tr ' ' '\012'
```

Onze
heures
venaient
de
sonner
à
la
Bourse,
lorsque ...

COMMANDE GREP

```
% head -n 60 zola1.txt | tr ' ' '\012' | grep -i ven
```

venaient
venu
venait
souvent
venu
vendre
venir

- `grep` recherche une chaine (ou expression régulière) dans l'entrée standard
- toutes les occurrences des chaines sont affichées sur la sortie standard

rali

COMMANDE UNIQ

```
% head -n 200 zola1.txt | tr ' ' '\012' | grep -i "^ven" | sort | uniq -c
```

```
1 venaient
5 venait
1 vend
1 vendez,
2 vendre
2 venir
1 vent,
1 vente
1 ventre
2 venu
1 venue
```

- `sort` tri un fichier (beaucoup d'options disponibles)
- `uniq` élimine les séquences de chaines identiques (l'option `-c` permet de compter les répétitions)

rali

COMMANDE SORT

```
% head -n 200 zola1.txt | tr ' ' '\012' | grep -i "^ven" | sort | uniq -c | sort -k1,1nr
```

```
5 venait
2 vendre
2 venir
2 venu
1 venaient
1 vend
1 vendez,
1 vent,
1 vente
1 ventre
1 venue
```

- l'option `-k1,1n` donnée à `sort` demande le tri de la première colonne selon un critère numérique (`n`) et en ordre inverse (`r`) qui par défaut est croissant

rali

COMMANDE AWK

- permet d'introduire des programmes C en shell
- un tutoriel est nécessaire pour maîtriser cette commande
- on peut s'en servir pour afficher une colonne particulière (ici la seconde)

```
% head -n 200 zola1.txt | tr ' ' '\012' | grep -i
"^ven" | sort | uniq -c | sort -k1,1nr |
awk 'print $2'
```

venait
vendre
venir
venu
venaient
vend ...

rali

- afficher une ligne sous contrainte
  (ici la première colonne doit être supérieure à 2)

```
% head -n 1000 zola1.txt | tr ' ' '\012' | grep
-i "^ven" | sort | uniq -c | sort -k1,1nr |
awk '$1 > 2 {print $0}'
```

```
25 venait
7 venir
6 venu
4 Vendôme,
3 venaient
3 venant
3 vendre
3 vente
3 ventre
```

TANNÉS DE TAPER DES COMMANDES À CHAQUE FOIS ?

- Créez un **script** !!!

% cat myscript

```
1  #!/bin/csh -f
2
3  set in = $1
4
5  cat $in | \
6        awk 'BEGIN {ok=0} \
7            /DEBUT DU FICHIER/ {ok = 1} \
8            /FIN DU FICHIER/ {ok=0} \
9            {if (ok>10) print $0; else if (ok) ok++}' \
10         | tr  ' ' '\012' | tr '[:upper:]' '[:lower:]'
```

- `chmod u+x myscript`
- `./myscript zola1.txt`

rali