UNIVERSITÉ D'AVIGNON ET DES PAYS DE VAUCLUSE

M2 ILSEN – 2019/20

UE Ingénierie du document et de l'information

UCE Indexation & recherche

Vincent Labatut

TP 5 | Correction

1 Occurrence des termes

Exercice 1

```
private static Map<String,Integer> countTerms(List<Token> tokens)
{  Map<String,Integer> result = new HashMap<String, Integer>();

   for(Token token: tokens)
   {  String term = token.getType();
      Integer count = result.get(term);
      if(count==null)
         count = 0;
      count++;
      result.put(term, count);
   }

   return result;
}
```

Exercice 2

```
private static void writeCounts(Map<String,Integer> counts, String fileName) throws
    FileNotFoundException, UnsupportedEncodingException
{  File file = new File(fileName);
   FileOutputStream fos = new FileOutputStream(file);
   OutputStreamWriter osw = new OutputStreamWriter(fos,"UTF-8");
   PrintWriter writer = new PrintWriter(osw);

   for(Entry<String,Integer> entry: counts.entrySet())
   {  String term = entry.getKey();
      Integer count = entry.getValue();
      writer.println("\""+term+"\","+count);
   }

   writer.close();
}
```

Exercice 3

```
public static void processCorpus() throws FileNotFoundException,
    UnsupportedEncodingException
{  List<Token> tokens = new LinkedList<Token>();
   Map<String,Integer> counts;
   long startTotal = System.currentTimeMillis();

   // tokénization
   {  System.out.println("Tokenizing corpus");
      long start = System.currentTimeMillis();
      Tokenizer tokenizer = new Tokenizer();
      tokenizer.tokenizeCorpus(tokens);
      long end = System.currentTimeMillis();
```

```
12          System.out.println(tokens.size()+" tokens were found,
13                                          duration="+(end-start)+" ms\n");
14      }
15
16      // normalisation
17      {  System.out.println("Normalizing tokens");
18         long start = System.currentTimeMillis();
19         Normalizer normalizer = new Normalizer();
20         normalizer.normalizeTokens(tokens);
21         long end = System.currentTimeMillis();
22         System.out.println(tokens.size()+" tokens remaining after normalization,
23                                          duration="+(end-start)+" ms\n");
24      }
25
26      // décompte
27      {  System.out.println("Counting terms");
28         long start = System.currentTimeMillis();
29         counts = countTerms(tokens);
30         long end = System.currentTimeMillis();
31         System.out.println("There are "+counts.size()+" distinct terms in the
32                                     corpus, duration="+(end-start)+" ms\n");
33      }
34
35      // enregistrement
36      {  String outFile = FileTools.getTermCountFile();
37         System.out.println("Recording counts in "+outFile);
38         long start = System.currentTimeMillis();
39         writeCounts(counts,outFile);
40         long end = System.currentTimeMillis();
41         System.out.println("Counts recorded, duration="+(end-start)+" ms\n");
42      }
43
44      // affichage du nom du fichier des mots vides
45      String stopWordsFile = FileTools.getStopWordsFile();
46      System.out.println("Stop-words file: " + stopWordsFile+"\n");
47
48      long endTotal = System.currentTimeMillis();
49      System.out.println("Total duration="+(endTotal-startTotal)+" ms\n");
50  }
```

Exercice 4

Comme on l'a déjà vu lors de la construction de l'index, on a un total de 7 612 885 tokens dans `wp`, si on ne filtre pas les mots-vides. En filtrant les 75 mots-vides les plus fréquents, cela ramène ce décompte à 4 251 506, soit à peu près -44%.

En cours, on avait obtenu -38% pour les 30 premiers mots-vides, et -52% pour les 150 premiers, donc on peut considérer que ce résultat est dans la fourchette attendue (pour ce qui concerne l'espace occupé par l'index). Notez que les valeurs vues en cours concernaient un corpus de textes en anglais.

Exercice 5

Pour le corpus `springer`, qui est en anglais et qui est plus petit que `wp`, on obtient 1 121 433 postings sans filtrage, et 666 925 en se débarassant des 75 premiers mots-vides, soit -41%. La valeur est là aussi proche de la référence du cours.

Si on effectue la racinisation, alors bien sûr, le nombre de tokens sans filtrage des mots-vides ne change pas (1 121 433). En revanche, le nombre avec filtrage diminue légèrement plus : 645 316, soit -42%. Cela s'explique par le fait que les tokens sont distribués autrement, puisqu'il y a moins de termes différents quand on racinise. En cours aussi, on observait peu de différences

dues à la racinisation, pour ce qui est du nombre de tokens.

Exercice 6

Il faudrait faire un test statistique pour être vraiment rigoureux, mais il semblerait, visuellement, qu'il s'agisse bien d'une loi de puissance dans les deux cas. En particulier, on observe que les graphiques linéaires (à gauche) présentent de très fortes proportions de mots de faible fréquence et seulement quelques mots de très forte fréquence ; tandis que les graphiques log-log (à droite) correspondent plus ou moins à une ligne droite. Cela confirme donc bien aux remarques faites en cours à propos de la distribution des mots dans une collection de textes.

2 Traitement des mots vides

Exercice 7

```
private void loadStopWords() throws FileNotFoundException,
    UnsupportedEncodingException
{  String fileName = FileTools.getStopWordsFile();
   File file = new File(fileName);
   FileInputStream fis = new FileInputStream(file);
   InputStreamReader isr = new InputStreamReader(fis,"UTF-8");
   Scanner scanner = new Scanner(isr);
   while(scanner.hasNext())
   {  String term = scanner.next();
      stopWords.add(term);
   }

   scanner.close();
}
```

Exercice 8

```
public String normalizeType(String string)
{  String result = string;

   // on passe en minuscules
   result = result.toLowerCase();

   // on supprime les signes diacritiques
   result = java.text.Normalizer.normalize(result, Form.NFD)
                   .replaceAll("\\p{InCombiningDiacriticalMarks}+", "");

   // on renvoie null en cas de chaîne vide
   if(result.isEmpty())
      result = null;
   // sinon on tente de raciniser
   else if(stemmer!=null)
      result = stemmer.stemType(result);

   // on renvoie null si on a un mot vide
   if(result!=null && stopWords.contains(result))
      result = null;

   return result;
}
```

3 Application et comparaison

Exercice 9

```
Tokenizing corpus...
7612885 tokens were found, duration=14559 ms

Normalizing tokens...
4251506 tokens remaining after normalization, duration=13269 ms

Building index...
Sorting tokens...
4251506 tokens sorted, duration=5176 ms
Filtering tokens...
2019952 tokens remaining, corresponding to 142717 terms, duration=1056 ms

Building posting lists...
2019952 postings listed, lexicon=ARRAY, duration=560 ms
There are 142717 entries in the index, token list=LINKED, duration=6793 ms

Total duration=34622 ms
```

Le nombre de tokens après normalisation est bien sûr plus faible (4 251 506 vs. 7 612 885), puisqu'on ignore certains d'entre eux (les non-mots). Par conséquent, il y a aussi moins de tokens distincts après filtrage (2 019 952 vs. 2 177 994), cependant la différence est moins importante, car on a supprimé les types les plus fréquents. Enfin, le nombre de termes diminue également (142 717 vs. 142 792), mais seulement de 75, ce qui correspond au nombre de non-mots qu'on avait listés.

Du point de vue du traitement lui-même, on va marginalement plus vite (34 622 vs. 36 723 ms, soit -9%) et le fichier est légèrement plus petit (25 607 956 octets vs. 27 190 325 octets, soit -9%). L'amélioration réelle porte sur la qualité des résultats.

Exercice 10

Sans traiter les mots-vides, on obtient un affichage de la forme suivante :

```
Loading the index
Index loaded, duration=78767 ms

Processing request "Projet Apache développé en plusieurs langages"
Query processed, duration=33 ms
Result: 11 document(s)
[204, 510, 880, 1099, 1271, 1281, 1299, 1845, 2048, 2232, 2343]
Files:
[119c82b5-ac36-44c6-9f3e-1b0111639132.txt...]
Et en les traitant, on a :
Loading the index
Index loaded, duration=74335 ms

Processing request "Projet Apache développé en plusieurs langages"
Query processed, duration=21 ms
Result: 12 document(s)
[204, 366, 510, 880, 1099, 1271, 1281, 1299, 1845, 2048, 2232, 2343]
Files:
[119c82b5-ac36-44c6-9f3e-1b0111639132.txt...]
```

Le traitement est légèrement plus rapide sans les mots vides, mais ce n'est pas flagrant (et pour plus de fiabilité, il faudrait faire une évaluation à plus grande échelle, en considérant un grand nombre de requêtes différentes). On observe que dans le deuxième cas, on récupère un document de plus : c'est celui de docID 366, qui correspond au fichier `20df6cba-c474-436d-90d3-9c2798faa6b9.txt` (article Wikipedia portant sur OpenSearchServer). Il ne contient pas un ou plusieurs des mots-vides présents dans la requête, ce qui explique qu'il ait été renvoyé par le 2ème moteur mais pas par le 1er.

Exercice 11

Le détail des performances obtenu est le suivant. En filtrant les mots-vides mais sans faire de racinisation, on a :

```
PRECISION  RECALL   F_MEASURE
0.00       0.00     0.00       Arthroscopic treatment of cruciate ligament injuries
0.00       0.00     0.00       Complications of arthroscopic interventions
0.00       0.00     0.00       Pathophysiology and prophylaxis of arthrofibrosis
0.00       0.00     0.00       HIV Epidemiology, Risk Assessment
0.00       0.00     0.00       Patient-controlled analgesia indications and limits
0.00       0.00     0.00       Priming with non-depolarizing muscle relaxants
0.71       0.62     0.67       Complications after laparoscopic cholecystectomy
0.00       0.00     0.00       Heparin induced thrombocytopenia, diagnosis and...
0.50       0.17     0.25       Diagnostic in Lyme disease
0.29       0.83     0.43       Treatment of acute myocardial infarction
1.00       0.11     0.19       Catheter ablation and cardiac mapping
1.00       0.06     0.12       Diagnostic approach in injuries of the shoulder
0.00       0.00     0.00       Differential diagnosis in infertility
0.00       0.00     0.00       Approach of the correction of deformities in...
0.21       0.65     0.31       Treatment of squamous cell carcinoma
1.00       0.07     0.12       Associated diseases with insulin dependent...
0.69       0.41     0.51       Therapy in chronic low back pain
0.37       0.86     0.52       Treatment of ventricular tachycardia
0.33       0.05     0.08       Indication for implantable cardioverter...
0.00       0.00     0.00       Diagnostic in acute and chronic myocarditis
0.50       0.07     0.12       Cause of dysphagia
0.67       0.14     0.24       Treatment of sensorineural hearing loss (SNHL)
1.00       0.12     0.22       Complications of surgical repair of aortic aneurysm
1.00       0.06     0.12       Treatment of psychosomatical patients
0.00       0.00     0.00       New approach in cruciate ligament surgery
----------------------------------------
0.37       0.17     0.16
```

Et en filtrant les mots-vides tout en faisant la racinisation, on a :

```
PRECISION  RECALL   F_MEASURE
0.33       0.14     0.20       Arthroscopic treatment of cruciate ligament injuries
0.00       0.00     0.00       Complications of arthroscopic interventions
0.00       0.00     0.00       Pathophysiology and prophylaxis of arthrofibrosis
0.00       0.00     0.00       HIV Epidemiology, Risk Assessment
0.00       0.00     0.00       Patient-controlled analgesia indications and limits
0.00       0.00     0.00       Priming with non-depolarizing muscle relaxants
0.78       0.88     0.82       Complications after laparoscopic cholecystectomy
1.00       0.07     0.13       Heparin induced thrombocytopenia, diagnosis and...
0.50       0.17     0.25       Diagnostic in Lyme disease
0.29       0.83     0.43       Treatment of acute myocardial infarction
1.00       0.11     0.19       Catheter ablation and cardiac mapping
1.00       0.06     0.12       Diagnostic approach in injuries of the shoulder
0.80       0.17     0.29       Differential diagnosis in infertility
0.00       0.00     0.00       Approach of the correction of deformities in...
0.19       0.88     0.31       Treatment of squamous cell carcinoma
1.00       0.07     0.12       Associated diseases with insulin dependent...
0.69       0.41     0.51       Therapy in chronic low back pain
0.33       0.91     0.49       Treatment of ventricular tachycardia
0.25       0.05     0.08       Indication for implantable cardioverter...
0.00       0.00     0.00       Diagnostic in acute and chronic myocarditis
0.75       0.20     0.32       Cause of dysphagia
0.67       0.14     0.24       Treatment of sensorineural hearing loss (SNHL)
1.00       0.12     0.22       Complications of surgical repair of aortic aneurysm
```

```
0.29        0.75      0.42       Treatment of psychosomatical patients
0.00        0.00      0.00       New approach in cruciate ligament surgery
-------------------------------------

0.43        0.24      0.21
```

La Table 1 résume les résultats obtenus. Comme observé précédemment, il y a une amélioration des performances avec la racinisation seule, et avec le filtrage des mots-vides seul. Quand on combine les deux, l'amélioration est encore plus marquée. Cependant, les performances restent faibles : l'index nécessite d'autres améliorations.

Racinisateur	Filtrage m-v	Précision	Rappel	F-mesure
Non	Non	0.38	0.10	0.14
Oui	Non	0.46	0.14	0.18
Non	Oui	0.37	0.17	0.16
Oui	Oui	0.43	0.24	0.21

Table 1. Performances obtenues pour le corpus `springer` avec/sans filtrage des mots-vides et avec/sans racinisation.