

Auteur 5.1 : **Dufour Emilien**

Groupe : Drapp Thibault, Badiane Issa, Dufour Emilien

5 janvier 2018

# PLP rapport de TP

## Exercice 5

Question 5.1 : TF-IDF par **Emilien DUFOUR**

### Objectif :

Nous devons réaliser un programme Map-Reduce calculant le TF-IDF de chaque mot d'un ensemble de documents.

### Méthode :

Soit  $TFIDF_{i,j}$ , le TFIDF du mot  $m_i$  du document  $d_j$  du corpus  $C$ .

$$TFIDF_{i,j} = \text{wordcount}(m_i, d_j) / \text{card}(d_j) * \log_{10}(\text{card}(C) / \text{card}(\{d \in C \mid m_i \in d\}))$$

On effectue une série de 3 Map-Reduce sur notre Corpus.

	Key	Value
<b>Map 1</b>	[mot i, doc j]	1
<b>Shuffle 1</b>	[mot i, doc j]	{1, 1, ..., 1}
<b>Reduce 1</b>	[mot i, doc j]	Wordcount <sub>i,j</sub>
<b>Map 2</b>	doc j	[mot i, wordcount <sub>i,j</sub> ]
<b>Shuffle 2</b>	doc j	{[mot 1, wordcount <sub>1,j</sub> ], ..., [mot N, wordcount <sub>N,j</sub> ]}
<b>Reduce 2</b>	[mot i, doc j]	[Wordcount <sub>i,j</sub> , lengthdoc <sub>j</sub> ]
<b>Map 3</b>	mot i	[doc j, wordcount <sub>i,j</sub> , lengthdoc <sub>j</sub> ]
<b>Shuffle 3</b>	mot i	{[doc 1, wordcount <sub>i,1</sub> , lengthdoc <sub>1</sub> ], ..., [doc M, wordcount <sub>i,M</sub> , lengthdoc <sub>M</sub> ]}
<b>Reduce 3</b>	[mot i, doc j]	[Wordcount <sub>i,j</sub> , lengthdoc <sub>j</sub> , nbrDocHavingMot <sub>i</sub> ]
	—>	[TF-IDF <sub>i,j</sub> ]

## Résumé et remarques importantes :

On applique donc en cascade 3 fois l'architecture java pour les jobs MapReduce.

La complexité réside dans la gestion des clefs composites et dans l'implémentation de plusieurs boucles sur les mêmes « valeurs » dans le Reducer.

- Pour les clefs composites, on crée donc des classes dont les attributs sont les éléments de nos vecteurs clefs. Ces classes implémentent l'interface

« WritableComparable<CompositeKey> ».

- Pour les boucles dans le Reducer : il s'agit de générer à l'issue du Reducer plus de clefs que celles reçues. C'est le cas par exemple du Reducer du round2 : on émet des couples clef : <mot i, doc j> valeur : <wordcount ij , tailleDoc j> alors que le Mapper émettait clef : <doc j> valeur : <mot i, wordcount ij> (voir schéma ci-dessus).

Il faut donc que le Reducer fasse deux boucles sur le même objet « Iterator ». Pour cela on crée un cache qui contiendra les « valeurs » et que l'on remplit dans la première boucle. Il faudra faire attention à remplir le cache en créant des copies des objets contenus dans «Iterator», afin de ne pas faire pointer les objets de notre cache vers le même objet.

## Résultats :

Voici les 20 mots ayant le TF-IDF le plus élevé, classés en ordre décroissant.

```
output4 — nano part-r-00000 — 141x30
GNU nano 2.0.6 File: part-r-00000
Buck file:/home/cloudera/workspace/IO_TFIDF/inputs/callwild 0.002387172231519686
dogs file:/home/cloudera/workspace/IO_TFIDF/inputs/callwild 5.778472444931353E-4
Thornton file:/home/cloudera/workspace/IO_TFIDF/inputs/callwild 5.589814331982783E-4
myself file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 5.4486411878595E-4
Buck's file:/home/cloudera/workspace/IO_TFIDF/inputs/callwild 4.168078484868517E-4
Spitz file:/home/cloudera/workspace/IO_TFIDF/inputs/callwild 4.0733494283942326E-4
Francois file:/home/cloudera/workspace/IO_TFIDF/inputs/callwild 3.978620371919948E-4
John file:/home/cloudera/workspace/IO_TFIDF/inputs/callwild 3.59970414602281E-4
sled file:/home/cloudera/workspace/IO_TFIDF/inputs/callwild 3.59970414602281E-4
Buck, file:/home/cloudera/workspace/IO_TFIDF/inputs/callwild 3.126058863651387E-4
dogs, file:/home/cloudera/workspace/IO_TFIDF/inputs/callwild 2.9366007507028186E-4
Friday file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 2.8233867973453773E-4
shore, file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 2.5261881870984956E-4
- file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 2.3775888819750548E-4
Perrault file:/home/cloudera/workspace/IO_TFIDF/inputs/callwild 2.3682264118571116E-4
However, file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 2.3280557802672408E-4
Hal file:/home/cloudera/workspace/IO_TFIDF/inputs/callwild 2.2734973553828271E-4
God file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 2.2042230259977066E-4
me. file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 2.1546899242898932E-4
me; file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 2.1546899242898932E-4

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

On a d'abord calculé le wordcount de toutes les paires (**mot i** , **doc j**).

```

output1 — nano part-r-00000 — 142x35
GNU nano 2.0.6 File: part-r-00000

'we've file:/home/cloudera/workspace/IO_TFIDF/inputs/callwild 1
'whoa! file:/home/cloudera/workspace/IO_TFIDF/inputs/callwild 1
'why file:/home/cloudera/workspace/IO_TFIDF/inputs/callwild 1
('gas' file:/home/cloudera/workspace/IO_TFIDF/inputs/callwild 1
(I file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 4
(Points file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 1
(September, file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 1
(a file:/home/cloudera/workspace/IO_TFIDF/inputs/callwild 1
(a file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 1
(gas file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 4
(at file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 1
(cakes file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 1
(first) file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 1
(for file:/home/cloudera/workspace/IO_TFIDF/inputs/callwild 1
(for file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 9
(for, file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 1
(from file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 1
(happily file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 1
(he file:/home/cloudera/workspace/IO_TFIDF/inputs/callwild 1
(ifI file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 1
(it file:/home/cloudera/workspace/IO_TFIDF/inputs/callwild 2
(it file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 1
(of file:/home/cloudera/workspace/IO_TFIDF/inputs/callwild 1
(or file:/home/cloudera/workspace/IO_TFIDF/inputs/callwild 2
(or file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 1
(out file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 1
(seventy file:/home/cloudera/workspace/IO_TFIDF/inputs/callwild 1
(so file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 2
(still file:/home/cloudera/workspace/IO_TFIDF/inputs/callwild 1
(such file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 1

^G Get Help ^O WriteOut ^R Read File ^V Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^N Next Page ^U UnCut Text ^T To Spell

```

Ensuite on calcule le nombre de mots par document et on l'associe aux couples (**mot i** , **doc j** , **wordcount ij**).

```

output2 — nano part-r-00000 — 142x35
GNU nano 2.0.6 File: part-r-00000

tin file:/home/cloudera/workspace/IO_TFIDF/inputs/callwild 1 31778
cheerful. file:/home/cloudera/workspace/IO_TFIDF/inputs/callwild 1 31778
timid file:/home/cloudera/workspace/IO_TFIDF/inputs/callwild 2 31778
know.' file:/home/cloudera/workspace/IO_TFIDF/inputs/callwild 1 31778
ranging; file:/home/cloudera/workspace/IO_TFIDF/inputs/callwild 1 31778
know. file:/home/cloudera/workspace/IO_TFIDF/inputs/callwild 1 31778
know.' file:/home/cloudera/workspace/IO_TFIDF/inputs/callwild 1 31778
cheer file:/home/cloudera/workspace/IO_TFIDF/inputs/callwild 1 31778
cheeks. file:/home/cloudera/workspace/IO_TFIDF/inputs/callwild 1 31778
cheek. file:/home/cloudera/workspace/IO_TFIDF/inputs/callwild 1 31778
went, file:/home/cloudera/workspace/IO_TFIDF/inputs/callwild 5 31778
ranging file:/home/cloudera/workspace/IO_TFIDF/inputs/callwild 1 31778
know, file:/home/cloudera/workspace/IO_TFIDF/inputs/callwild 1 31778
ranged file:/home/cloudera/workspace/IO_TFIDF/inputs/callwild 1 31778
checked file:/home/cloudera/workspace/IO_TFIDF/inputs/callwild 4 31778
know file:/home/cloudera/workspace/IO_TFIDF/inputs/callwild 17 31778
instructing file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 1 121547
zenith, file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 1 121547
zee, file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 1 121547
youth, file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 1 121547
youth file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 1 121547
yourself file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 1 121547
yours, file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 1 121547
yourn file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 1 121547
your file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 39 121547
young. file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 1 121547
young, file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 1 121547
young file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 34 121547
you? file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 5 121547
you;" file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 1 121547

^G Get Help ^O WriteOut ^R Read File ^V Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^N Next Page ^U UnCut Text ^T To Spell

```

Ensuite on calcule le nombre de documents contenant un certain mot. En associant ces quantités aux couples ( mot  $i$  , doc  $j$  , wordcount  $ij$  , tailleDoc  $j$ ), on détermine les couples ( **mot  $i$  , doc  $j$  , TFIDF  $ij$**  ).

```

output3 — nano part-r-00000 — 142x35
GNU nano 2.0.6 File: part-r-00000
A file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 4.953310170781363E-6
"Alas! file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 2.4766550853906816E-6
"Al file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 7.429965256172046E-6
"Am file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 2.4766550853906816E-6
"And file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 2.4766550853906816E-6
"And, file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 2.4766550853906816E-6
"Are file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 2.4766550853906816E-6
"Ay, file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 2.4766550853906816E-6
"Be file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 2.4766550853906816E-6
"Between file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 2.4766550853906816E-6
"But file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 2.4766550853906816E-6
"But, file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 2.4766550853906816E-6
"But," file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 7.429965256172046E-6
"Call file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 7.429965256172046E-6
"Can file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 2.4766550853906816E-6
"Captain, file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 2.4766550853906816E-6
"Do file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 2.4766550853906816E-6
"Eatee file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 2.4766550853906816E-6
"For file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 7.429965256172046E-6
"For," file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 4.953310170781363E-6
"Friday, file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 1.733658559773477E-5
"Gentlemen," file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 2.4766550853906816E-6
"Go file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 2.4766550853906816E-6
"God file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 2.4766550853906816E-6
"Governor," file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 4.953310170781363E-6
"Ha!" file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 2.4766550853906816E-6
"Hark file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 2.4766550853906816E-6
"Have file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 2.4766550853906816E-6
"He file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 4.953310170781363E-6
"Here file:/home/cloudera/workspace/IO_TFIDF/inputs/defoe-robinson-103.txt 2.4766550853906816E-6

[ Read 17837 lines ]
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell

```

Enfin on sélectionne les 20 couples ( **mot  $i$  , doc  $j$  , TFIDF  $ij$**  ) ayant les plus forts TFIDE.