# Lab 2 : Wikipedia

Big Data Analysis

Quentin Vaucher, André Neto da Silva, Sylvain Renaud

Hes·so
Haute Ecole Spécialisée
de Suisse occidentale
Fachhochschule Westschweiz
University of Applied Sciences and Arts
Western Switzerland

March 23, 2019

# Contents

# 1 Introduction

# 2 Attempt #1 : naive ranking

## 2.1 List of language ranked using naive ranking

| Rank | Language | Number of article |
|---|---|---|
| 1 | Java | 2017 |
| 2 | JavaScript | 1738 |
| 3 | C# | 850 |
| 4 | CSS | 554 |
| 5 | C++ | 555 |
| 6 | Python | 545 |
| 7 | PHP | 452 |
| 8 | MATLAB | 324 |
| 9 | Perl | 300 |
| 10 | Ruby | 287 |
| 11 | Scala | 161 |
| 12 | Haskell | 128 |
| 13 | Objective-C | 112 |
| 14 | Clojure | 60 |
| 15 | Groovy | 55 |

## 2.2 Processing time using naive ranking

Processing Part 1: naive ranking took **32125 ms**.

# 3 Attempt #2 : ranking using inverted index

## 3.1 List of language ranked using inverted index

| Rank | Language | Number of article |
|---|---|---|
| 1 | Java | 2017 |

| Rank | Language | Number of article |
|---|---|---|
| 2 | JavaScript | 1738 |
| 3 | C# | 850 |
| 4 | CSS | 554 |
| 5 | C++ | 555 |
| 6 | Python | 545 |
| 7 | PHP | 452 |
| 8 | MATLAB | 324 |
| 9 | Perl | 300 |
| 10 | Ruby | 287 |
| 11 | Scala | 161 |
| 12 | Haskell | 128 |
| 13 | Objective-C | 112 |
| 14 | Clojure | 60 |
| 15 | Groovy | 55 |

### 3.2 Processing time using inverted index

Processing Part 2: ranking using inverted index took **5965 ms**.

## 4 Attempt #3 : ranking using reduceByKey

### 4.1 List of language ranked using reduceByKey

| Rank | Language | Number of article |
|---|---|---|
| 1 | Java | 2017 |
| 2 | JavaScript | 1738 |
| 3 | C# | 850 |
| 4 | CSS | 554 |
| 5 | C++ | 555 |
| 6 | Python | 545 |
| 7 | PHP | 452 |

| Rank | Language | Number of article |
|---|---|---|
| 8 | MATLAB | 324 |
| 9 | Perl | 300 |
| 10 | Ruby | 287 |
| 11 | Scala | 161 |
| 12 | Haskell | 128 |
| 13 | Objective-C | 112 |
| 14 | Clojure | 60 |
| 15 | Groovy | 55 |

### 4.2 Processing time using reduceByKey

Processing Part 3: ranking using reduceByKey took **2847 ms**.

## 5 Comparison

The final result is the same for all three attempts. Processing time varies.

| Attempt | Method | Processing time (ms) |
|---|---|---|
| #1 | Naive | 32125 |
| #2 | Inverted index | 5965 |
| #3 | reduceByKey | 2847 |

Best performer is attempt #3 with reduceByKey option.

## 6 Full output

```
1 List((Java,2017), (JavaScript,1738), (C#,850), (CSS,555), (C++,554), (
    Python,545), (PHP,452), (MATLAB,324), (Perl,300), (Ruby,287), (Scala
    ,161), (Haskell,128), (Objective-C,112), (Clojure,60), (Groovy,55))
2 List((Java,2017), (JavaScript,1738), (C#,850), (CSS,555), (C++,554), (
    Python,545), (PHP,452), (MATLAB,324), (Perl,300), (Ruby,287), (Scala
    ,161), (Haskell,128), (Objective-C,112), (Clojure,60), (Groovy,55))
3 List((Java,2017), (JavaScript,1738), (C#,850), (CSS,555), (C++,554), (
    Python,545), (PHP,452), (MATLAB,324), (Perl,300), (Ruby,287), (Scala
```

```
    ,161), (Haskell,128), (Objective-C,112), (Clojure,60), (Groovy,55))
Processing Part 1: naive ranking took 32125 ms.
Processing Part 2: ranking using inverted index took 5965 ms.
Processing Part 3: ranking using reduceByKey took 2847 ms.
```