# Phần 1: Tổng quan

## 1. Tên chương trình

Phân tích dữ liệu các trang báo online.

## 2. Mục đích

Chương trình được xây dựng với mục đích:

* Tìm hiểu các thông tin từ dữ liệu thu thập được từ các trang báo online.
* Đưa ra được một số nhận định từ dữ liệu

## 3. Công nghệ sử dụng

Các công nghệ sử dụng:

* Spark với thư viện PySpark (v2.4.5)
* Zeppelin Notebook làm editor (v0.9.0)
* GitHub để quản lý mã nguồn
* Ứng dụng chạy trên nền Docker

## 4. Phân Công công việc

Công việc được phân chia cho các thành viên trong nhóm như sau:

* **Vũ Đình Nguyên:**
  + Thu thập dữ liệu
  + Tiền xử lý dữ liệu
  + Cấu hình phân cụm
* **Hoàng Thị Minh Nhất:**
  + Đưa ra các phân tích tổng quan cho dữ liệu
* **Hoàng Văn Công:**
  + Phân tích một số thông tin khác từ dữ liệu

# Phần 2: Thiết kế hệ thống

## 1. Thu thập dữ liệu

### a. Công cụ và nguồn dữ liệu.

* Dữ liệu được thu thập bằng công cụ Scrapy với Python
* Các nguồn báo được thu thập là:
  + 24h
  + Afamily
  + DanTri.vn
  + VNExpress
  + <Nguyên điền tiếp..>
* Kích thước dữ liệu: <TODO>
* Số lượng bài viết: <TODO>

### b. Mô tả dữ liệu

Các bài báo được thu thập với các thông tin sau:

|  |  |  |
| --- | --- | --- |
| Name | Type | Desc |
| \_id | string | ID của bài báo, dùng để phân biệt các bài báo với nhau.  ID được xây dựng bằng cách <Nguyên tiếp tục> |
| headline | string | Tiêu đề (title) của bài báo |
| thumbnail | string | Hình ảnh thu nhỏ của bài báo (nếu có) |
| description | string | Subtitle của bài bá, 1 câu ngắn gọn để mô tả bài báo |
| type | string | Loại của bài báo: Article, <Nguyên Điền tiếp Xem còn loại gì nữa không?> |
| date\_published | integer | Thời gian đăng bài báo.  Thời gian được tính bằng Timestamp (Unix epoch time) |
| date\_modified | integer | Thời gian bài báo được chỉnh sửa / update.  Thời gian được tính bằng Timestamp (Unix epoch time) |
| author.name | string | Tên tác giả |
| content | string | Nội dung bài báo |
| keywords | string | Danh sách các từ khoá của bài báo |
| category | string | Category của bài báo |
| language | string | Ngôn ngữ của bài báo: vi, en,... |
| word\_count | integer | <Xem count này là count của cái gì?> |
| url | string | Link đến bài báo |
| like | integer | Số lượng like của bài báo |
| share | integer | Số lượng share của bài báo |
| seen | integer | Số lượng seen của bài báo |
| organization | string | Nguồn của bài báo |

Dưới đây là schema của Data Frame (data\_df.printSchema())

|  |
| --- |
| root  |-- \_id: string (nullable = true)  |-- headline: string (nullable = true)  |-- thumbnail: string (nullable = true)  |-- description: string (nullable = true)  |-- type: string (nullable = true)  |-- date\_published: integer (nullable = true)  |-- date\_modified: integer (nullable = true)  |-- author\_type: string (nullable = true)  |-- author.name: string (nullable = true)  |-- publisher: string (nullable = true)  |-- content: string (nullable = true)  |-- images: string (nullable = true)  |-- keywords: string (nullable = true)  |-- category: string (nullable = true)  |-- language: string (nullable = true)  |-- geo\_place\_name: string (nullable = true)  |-- geo\_region: string (nullable = true)  |-- geo\_position: string (nullable = true)  |-- word\_count: integer (nullable = true)  |-- url: string (nullable = true)  |-- like: integer (nullable = true)  |-- share: integer (nullable = true)  |-- seen: integer (nullable = true)  |-- organization: string (nullable = true)  : string (nullable = true) |

## 

## 2. Tiền xử lý dữ liệu

Có một số thông tin trong dữ liệu thu thập về cần được tiền xử lý để cho quá trình phân tích dữ liệu được dễ dàng hơn.

### a. Xử lý số lượng like, seen, share

* **Desc**: Trong quá trình thu thập dữ liệu, số lượng like, seen, share sẽ bị gán bằng -1 nếu không get được số lượng hoặc số lượng bằng 0. Do vậy cần chuyển hết các thông số này về 0, nếu như có giá trị âm.
* **Xử lý**:

|  |
| --- |
|  |

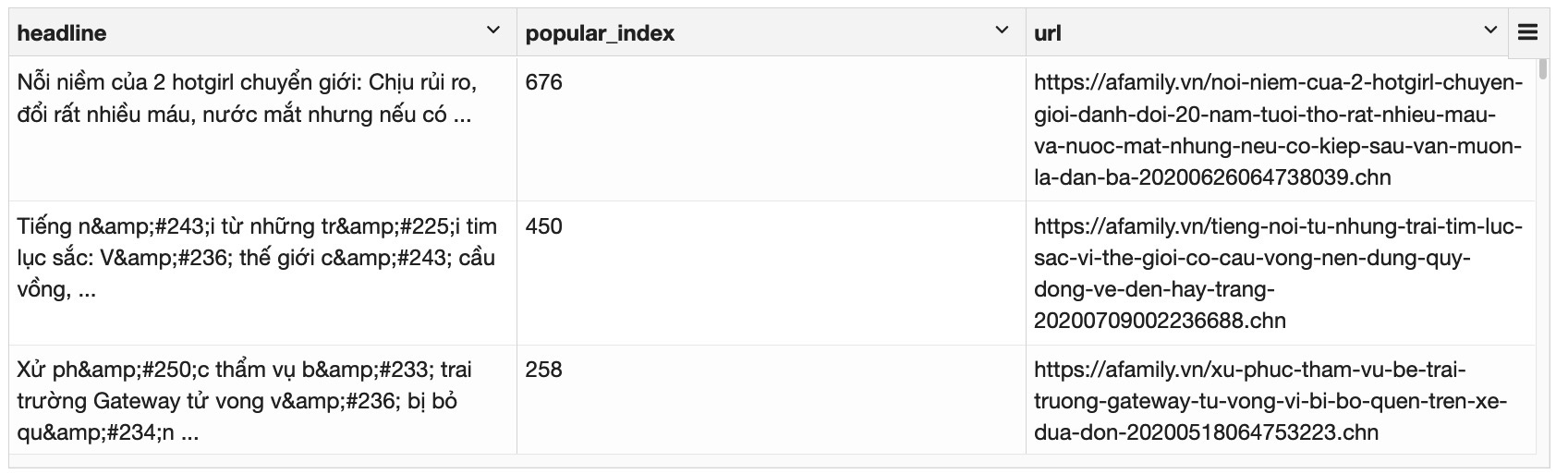
* **Kết quả**: OK

### b. Xử lý like, seen, share với popular\_index

* **Desc**: Do cả 3 tham số này đều thể hiện độ phổ biến của bài báo, nên chúng ta sẽ gộp 3 thông tin này vào làm một, và gọi nó là “*popular\_index”*. Phần xử lý dữ liệu phía sau khi cần xác định độ phổ biến của dữ liệu ta chỉ cần dựa vào chỉ số “*popular\_index*” thay vì làm việc với cả 3 chỉ số trên.
* **Xử lý**:

|  |
| --- |
| %spark.pyspark  data\_df = data\_df.withColumn('popular\_index', data\_df.like + data\_df.seen + data\_df.share).orderBy('popular\_index', ascending=False)  temp = data\_df.select('headline', 'popular\_index','url')  z.show(temp) |

* **Kết quả**: OK

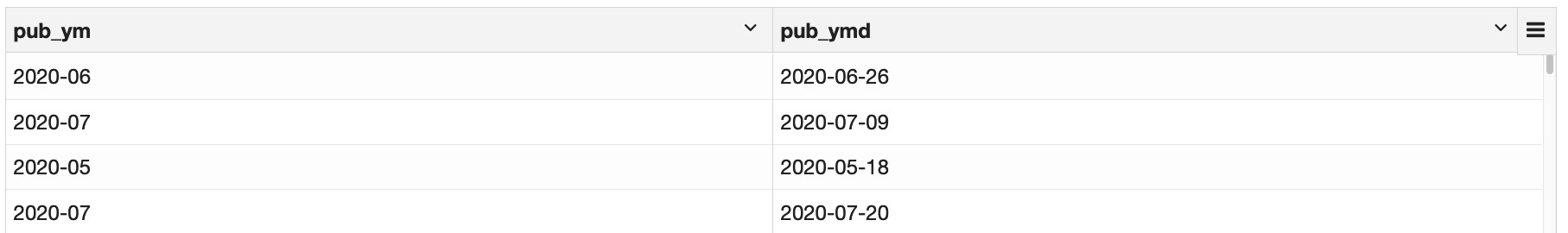


### c. Xử lý date\_published

* **Desc**: Do thời gian đang được lưu trữ dưới dạng số timestamp, khó để xử lý cũng như để theo dõi. Do vậy sẽ chuyển timestamp về format thời gian thông thường yyyy-MM-dd (cho những truy vấn chỉ quan tâm đến tháng và năm) và yyyy-MM (cho những truy vấn quan tâm để cả năm, tháng và ngày).
* **Xử lý**:

|  |
| --- |
| data\_df = data\_df.withColumn('pub\_ym', F.date\_format(data\_df.date\_published.cast(dataType=T.TimestampType()), "yyyy-MM"))  data\_df = data\_df.withColumn('pub\_ymd', F.date\_format(data\_df.date\_published.cast(dataType=T.TimestampType()), "yyyy-MM-dd")) |

* **Kết quả**: OK



### d. Xử lý category [TODO]

* **Desc**: Do mỗi một trang báo sẽ có một cách phân chia category khác nhau, nên chúng ta cần gộp những category giống hoặc gần giống nhau vào làm 1 để data được đồng nhất.
* **Xử lý**: Các category sau sẽ được gộp vào làm 1

|  |
| --- |
|  |

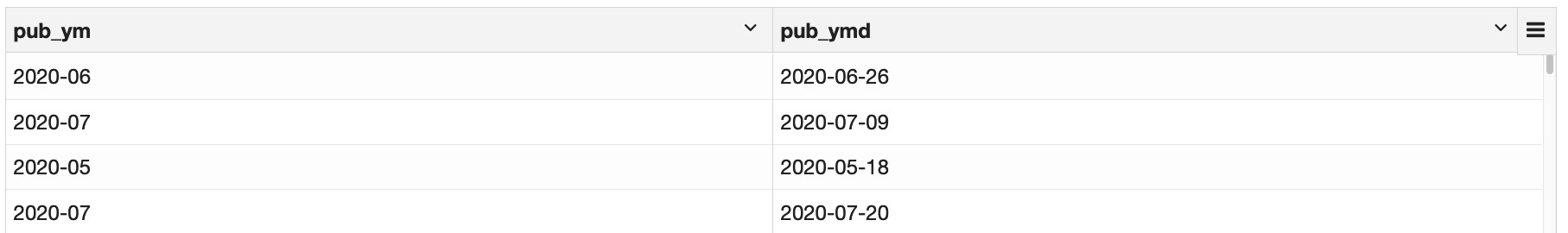
* **Kết quả**: OK

### a. template

* **Desc**: a.
* **Xử lý**:

|  |
| --- |
| b |

* **Kết quả**: OK



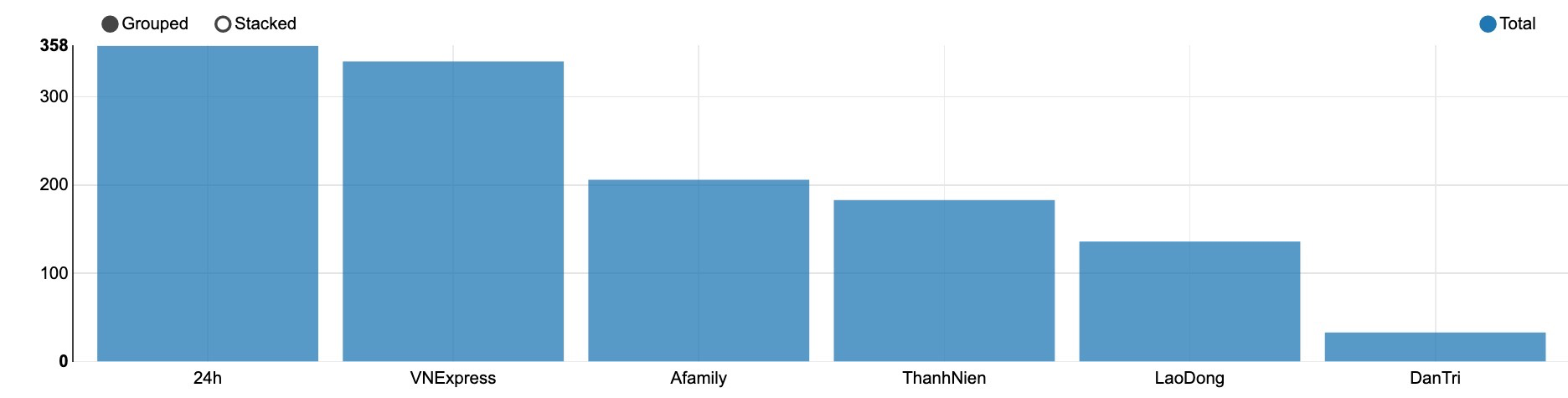
## 3. Một số thống kê từ dữ liệu

### a. Thống kê về nguồn bài báo [Nhất: Update lại với data mới ]

* **Desc**: Thống kê số lượng bài báo theo nguồn.
* **Code**:

|  |
| --- |
| %spark.pyspark  temp = (data\_df.groupBy("organization")  .agg(F.count("\_id").alias("Total"))  .orderBy("Total", ascending=False))  z.show(temp) |

* **Kết quả**:



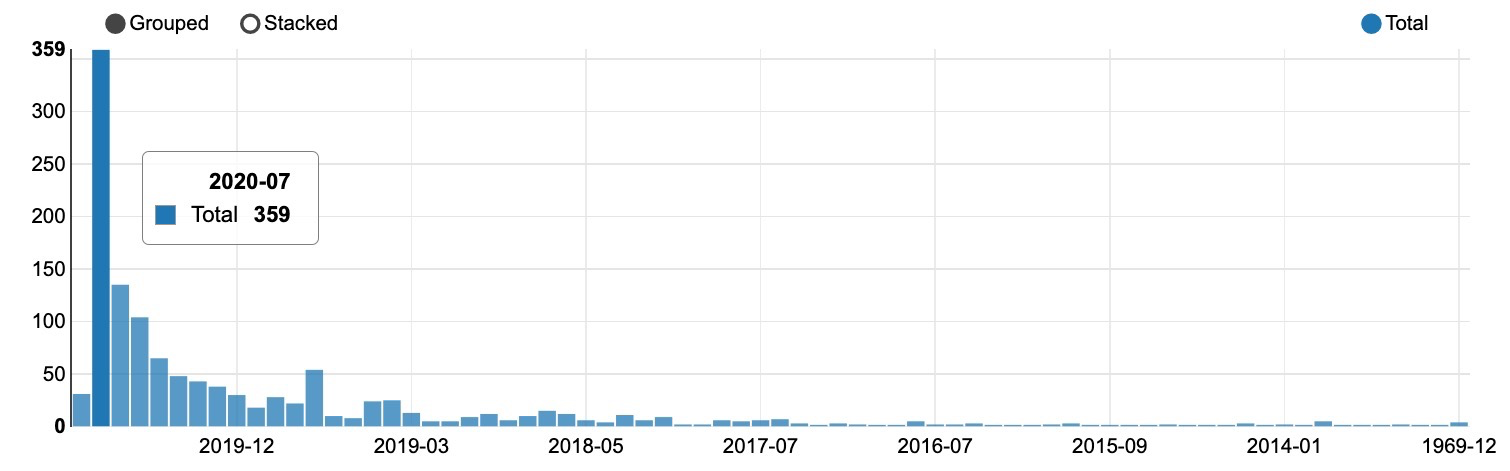
* **Nhận xét**:
  + Số lượng bài báo nhiều nhất đến từ trang: 24h và VN Express.
  + Phần lớn nguyên nhân đến từ sự khác biệt này đến từ phần thu thập dữ liệu.
  + Việc đưa ra nhận xét này để cho chúng ta có cái nhìn tổng quan về dữ liệu.

### b. Thống kê số lượng bài viết theo tháng. [Nhất: Update lại với data mới ]

* **Desc**: Thống kê số lượng bài viết theo tháng.
* **Code**: Dựa vào trường “pub\_ym” đã được tiền xử lý trước đó.

|  |
| --- |
| %spark.pyspark  temp = (data\_df.groupBy("pub\_ym")  .agg(F.count("\_id").alias("Total"))  .orderBy("pub\_ym", ascending=False))  z.show(temp) |

* **Kết quả**:



* **Nhận xét**:
  + Data cần gần với hiện tại thì càng nhiều, có 2 nguyên nhân:
    - Quá trình crawl data được thực hiện dựa theo các bài viết mới nhất rồi tới các bài viết trước đó. (crwal theo số trang)
    - Ngoài ra còn một phần do số lượng bài viết cũng ngày càng nhiều lên.
  + Có một số bài viết trường hợp đặc biệt timestamp = 0, nên nó rơi vào trường hợp 1969-12.

### c. Thống kê category [Nhất: Update lại với data mới ]

* **Desc**: Thống kê số lượng bài viết theo category
* **Code**:

|  |
| --- |
| %spark.pyspark  temp = (data\_df.groupBy("category")  .agg(F.count("\_id").alias("Total"))  .orderBy("Total", ascending=False))  z.show(temp) |

* **Kết quả**:

|  |
| --- |
|  |

* **Nhận xét**:
  + Số lượng bài viết chiếm tỷ trọng lớn nhất thuộc category: Sức khoẻ (17%) và Làm đẹp (15%)
  + Như vậy là ta thấy về tổng quan thì hầu như mọi người khá quan tâm về vấn đề sức khoẻ và làm đẹp.
  + Ta sẽ xem xét kỹ hơn theo từng nguồn trang một phía sau đây.

### d. Thống kê category dựa theo nguồn trang [Nhất: Update lại với data mới ]

* **Desc**: Thống kê số lượng bài viết theo category theo từng nguồn trang. Để xem trang nào có xu hướng đăng bài viết theo category nào.
* **Code**:

|  |
| --- |
| %spark.pyspark  temp1 = (data\_df.where(data\_df.organization == '24h')  .groupBy("category")  .agg(F.count("\_id").alias("Total"))  .orderBy("Total", ascending=False))  z.show(temp1) |

* **Kết quả**:

|  |  |  |
| --- | --- | --- |
| 24h: | VNExpress | DanTri: |
|  |  |  |
|  |  |  |

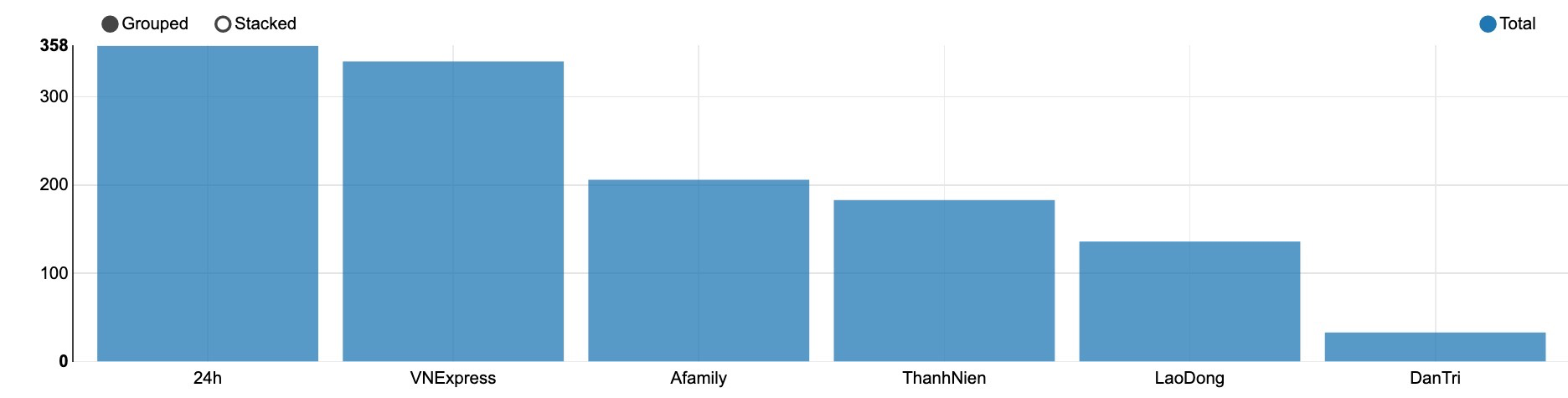
* **Nhận xét**:
  + Các trang sau có thiên hướng đưa bài về các tin chính trị xã hội:
  + Các trang sau có thiên hướng đưa bài về sức khỏe, gia đình:
  + Các trang sau có thiên hướng đưa bài về các tin tức giải trí:

### a. Ttemplate [Nhất: Update lại với data mới ]

* **Desc**: T
* **Code**:

|  |
| --- |
| %XYZ |

* **Kết quả**:



* **Nhận xét**:
  + ABC

## 4. Cài đặt phân cụm