

**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI  
VIỆN CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG**

-----o0o-----



**BÁO CÁO MÔN HỌC:  
LƯU TRỮ VÀ XỬ LÝ DỮ LIỆU LỚN**

**Đề tài:**

**THU THẬP, LƯU TRỮ VÀ XỬ LÝ DỮ LIỆU BÁO ĐIỆN TỬ  
SỬ DỤNG SPARK, HADOOP VÀ ELK STACK**

**Giảng viên hướng dẫn: TS. Đào Thành Chung**

**Nhóm sinh viên thực hiện: Zootopia**

1	Đàm Trọng Tuyên	20173463
2	Ngô Việt Hoàng	20173142
3	Nguyễn Thị Nhung	20173294

**Hà Nội, tháng 12, năm 2020**



## Mục lục

Danh mục bảng.....	i
Danh mục hình ảnh.....	i
Tóm tắt.....	1
1 Giới thiệu.....	1
2 Hệ thống .....	2
2.1 Tổng quan hệ thống.....	2
2.2 Cấu hình hệ thống.....	3
3 Thu thập và lưu trữ dữ liệu.....	4
3.1 Dữ liệu .....	4
3.2 Thu thập dữ liệu .....	4
3.3 Tích hợp và làm sạch dữ liệu .....	5
3.4 Kết quả.....	5
4 Tìm kiếm và trực quan hóa dữ liệu.....	6
4.1 Cài đặt và lưu trữ.....	6
4.2 Kết quả thực hiện .....	7
5 Xử lý và phân loại dữ liệu.....	10
5.1 Chuẩn bị dữ liệu .....	11
5.1.1 Tách từ Tiếng Việt.....	11
5.1.2 Chuẩn hóa câu .....	11
5.1.3 Vector hóa .....	12
5.1.4 Chia tập dữ liệu.....	13
5.2 Xây dựng mô hình phân loại văn bản.....	13
5.3 Kết quả đạt được .....	13
6 Tổng kết và hướng phát triển .....	14
7 Tài liệu tham khảo .....	15

## Danh mục bảng

Bảng 1: Kết quả thu thập dữ liệu: .....	5
---	---

## Danh mục hình ảnh

Hình 1: Tổng quan hệ thống .....	2
Hình 2: Kết quả cài đặt Hadoop .....	3
Hình 3: Kết quả cài đặt Spark .....	4
Hình 4: Kết quả cài đặt Elasticsearch .....	4
Hình 5: Luồng xử lý thu thập dữ liệu .....	4
Hình 6: Luồng crawl dữ liệu .....	5
Hình 7: Minh chứng dữ liệu lưu trong HDFS .....	5
Hình 8: Luồng xử lý lưu trữ, trực quan dữ liệu với ELK Stack .....	6
Hình 9: Cấu hình file config_news.conf cho Logstash .....	7
Hình 10: Dữ liệu được lưu trữ trong cụm Elasticsearch .....	7
Hình 11: Phân bố dữ liệu trên các máy .....	8
Hình 12: Phân bố dữ liệu theo nguồn .....	8
Hình 13: Phân bố dữ liệu trang soha.vn theo tháng .....	9
Hình 14: Phân bố dữ liệu trang soha.vn theo ngày trong tháng .....	9
Hình 15: Luồng xử lý và xây dựng mô hình phân loại văn bản. ....	10
Hình 16: Thống kê dữ liệu đầu vào cho bài toán phân loại văn bản .....	11
Hình 17: Luồng chuẩn hóa dữ liệu .....	12
Hình 18: Chạy job huấn luyện mô hình phân loại văn bản trên Spark .....	13
Hình 19: Các job đã được chạy xong .....	14

### **Tóm tắt**

Trong thời đại hiện nay, mỗi ngày có lượng lớn dữ liệu được tạo ra và được truyền tải đi trên thế giới. Việc thu thập, lưu trữ và xử lý dữ liệu đóng vai trò quan trọng để các quốc gia, tổ chức và cá nhân có được lợi thế phát triển và động lực để vươn lên. Trong báo cáo này, nhóm triển khai một hệ thống đơn giản nhằm thực hiện quá trình thu thập, lưu trữ và xử lý dữ liệu ở quy mô nhỏ. Đối tượng dữ liệu được thử nghiệm trong báo cáo là các tin tức trên một số trang tin tức lớn tại Việt Nam. Sau quá trình thu thập và lưu trữ, nhóm thực hiện làm sạch dữ liệu và đưa ra một số đặc điểm thống kê của tin tức. Dữ liệu sau khi xử lý được trích xuất lấy các đặc trưng của văn bản để xây dựng mô hình phân loại văn bản.

### **1 Giới thiệu**

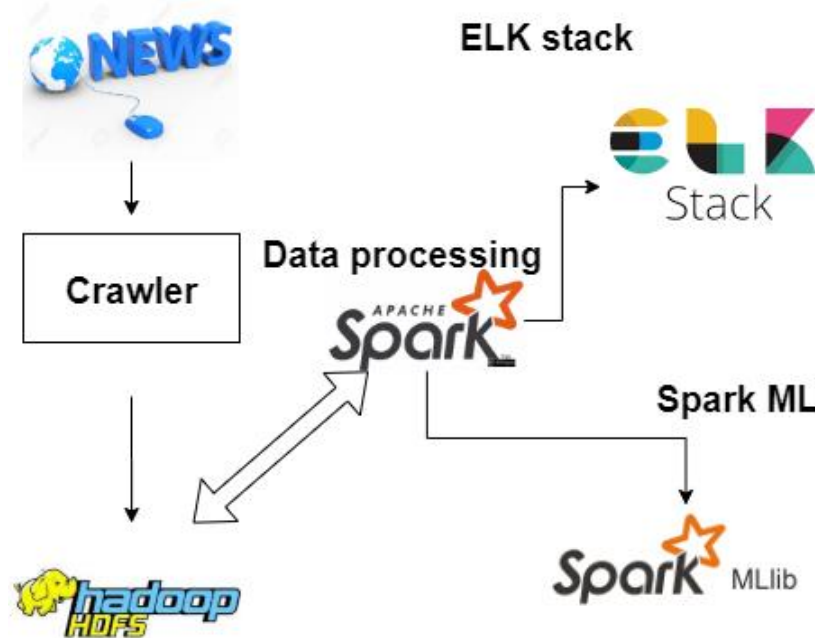
Công nghệ ngày càng phát triển mang lại cho cuộc sống con người nhiều chuyển biến tích cực. Mỗi ngày, hoạt động của con người tạo ra rất nhiều dữ liệu. Công nghệ sử dụng những dữ liệu này để tối ưu và chuyên biệt hóa, sau đó mang đến nhiều tiện ích hơn cho con người. Chính dựa vào công nghệ, con người có thể làm được nhiều thứ hơn, kết nối nhiều hơn và khám phá nhiều hơn, từ đó tạo ra nhiều dữ liệu hơn. Vòng quay của dữ liệu lại được tiếp tục. Một công ti bán lẻ, khi nắm bắt được thông tin về nhu cầu sử dụng, có thể thay đổi chiến lược kinh doanh của mình có phù hợp. Một tờ báo, khi biết được độc giả của mình muốn gì, có thể mang đến những tin tức mà họ quan tâm phù hợp. Vì lẽ đó, dữ liệu chính là nguồn tài nguyên quan trọng trong sự phát triển của xã hội. Các quốc gia, tổ chức và cá nhân nắm được và lấy được tri thức từ dữ liệu sẽ có lợi thế phát triển lớn.

Trong khuôn khổ môn học, báo cáo trình bày về những công việc nhóm đã thực hiện để xây dựng một mô hình đơn giản có chức năng thu thập thông tin các bài báo từ một số trang báo trực tuyến tại Việt Nam, lưu trữ và xử lý các dữ liệu thu được. Dữ liệu được lưu trữ phân tán trên nhiều máy, có chức năng sao lưu để có thể truy cập được khi một số máy mất kết nối. Quá trình xử lý dữ liệu bao gồm làm sạch dữ liệu, thống kê một số đặc điểm của tin tức, trích xuất đặc trưng để xây dựng mô hình phân loại văn bản.

## 2 Hệ thống

### 2.1 Tổng quan hệ thống

Hệ thống được xây dựng gồm bốn phần với các chức năng nhằm thu thập, xử lý, lưu trữ và trực quan hóa dữ liệu tin tức thu thập từ internet.



Hình 1: Tổng quan hệ thống

Thành phần của hệ thống bao gồm:

- **Bộ phận thu thập dữ liệu:** bộ phận này sử dụng Scrapy, một công cụ thu thập dữ liệu trên internet, với nhiệm vụ lấy dữ liệu từ các trang báo điện tử.
- **Bộ phận lưu trữ:** hệ thống lưu trữ dữ liệu vào Hadoop dưới dạng Hadoop File System (HDFS) để có thể lưu dữ liệu phân tán và có chức năng sao lưu, đảm bảo truy cập được khi một số máy mất kết nối.
- **Bộ phận xử lý và tích hợp dữ liệu:** Từ dữ liệu đã được lưu trong HDFS, Spark được sử dụng để xử lý và làm sạch dữ liệu. Dữ liệu đã làm sạch lại được lưu vào HDFS.
- **Bộ phận biểu diễn dữ liệu:** Dữ liệu đã được làm sạch được tổng hợp và lưu vào Elasticsearch thông qua Logstash phục vụ tìm kiếm thông tin và trực quan hóa dữ liệu. Việc trực quan hóa dữ liệu sẽ được Kibana đảm nhiệm.

Hệ thống trên đạt được các yêu cầu sau đối với một hệ thống dữ liệu lớn:

- **Khả năng mở rộng:** hệ thống sử dụng Hadoop, Spark, ELK Stack đều là những thành phần có khả năng mở rộng cao và dễ dàng.

- **Tính bền vững của dữ liệu:** Dữ liệu được lưu trữ trong HDFS và Elasticsearch đều được lưu trữ phân tán nên tính ổn định cao kể cả khi có máy bị mất kết nối.
- **Tốc độ tính toán cao:** việc sử dụng Spark giúp đơn giản hóa các tác vụ tính toán trên dữ liệu phân tán, khiến việc xử lý đạt được hiệu năng cao trong môi trường phân tán.
- **Khả năng trực quan hóa dữ liệu:** dữ liệu được trực quan sử dụng Kibana giúp cho việc quản lý, trực quan dữ liệu được dễ dàng.

## 2.2 Cấu hình hệ thống

Hệ thống được cài đặt trên ba máy tính với cấu hình chi tiết như sau:

- Máy 1:
  - o Hostname: vostro2420
  - o Hệ điều hành: Ubuntu 18.04
  - o Chip: Core i5-3230M, 4 cores.
  - o RAM: 8GB DDR3
- Máy 2:
  - o Hostname: thinkpadt530
  - o Hệ điều hành: Ubuntu 18.04
  - o Chip: Core i5-3320M, 4 cores.
  - o RAM: 6GB DDR3
- Máy 3:
  - o Hostname: asus
  - o Hệ điều hành: Window 10 Pro
  - o Chip: i5-6198U
  - o RAM: 8GB DDR3

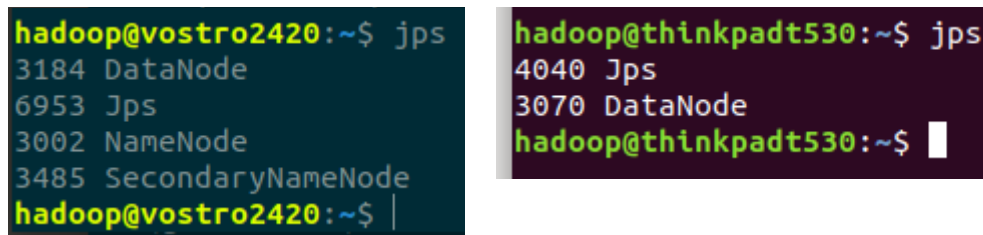
Các thành phần của hệ thống được cài đặt cụ thể như sau:

- Hadoop:

Vì lý do kỹ thuật không thể kết nối ba máy, nên Hadoop chỉ được cài đặt trên hai máy chạy Ubuntu, trong đó một máy vừa là datanode, vừa là master.

Replication = 2.

Chunk size = 64MB



```
hadoop@vostro2420:~$ jps
3184 DataNode
6953 Jps
3002 NameNode
3485 SecondaryNameNode
hadoop@vostro2420:~$ |

hadoop@thinkpadt530:~$ jps
4040 Jps
3070 DataNode
hadoop@thinkpadt530:~$
```

Hình 2: Kết quả cài đặt Hadoop

- Spark:

Cài đặt trên hai máy chạy Ubuntu

```
hadoop@vostro2420:/opt/spark$
5427 Worker
5496 Jps
5274 Master
hadoop@vostro2420:/opt/spark$

hadoop@thinkpadt530:~$ jps
5699 Jps
5210 Worker
hadoop@thinkpadt530:~$
```

Hình 3: Kết quả cài đặt Spark

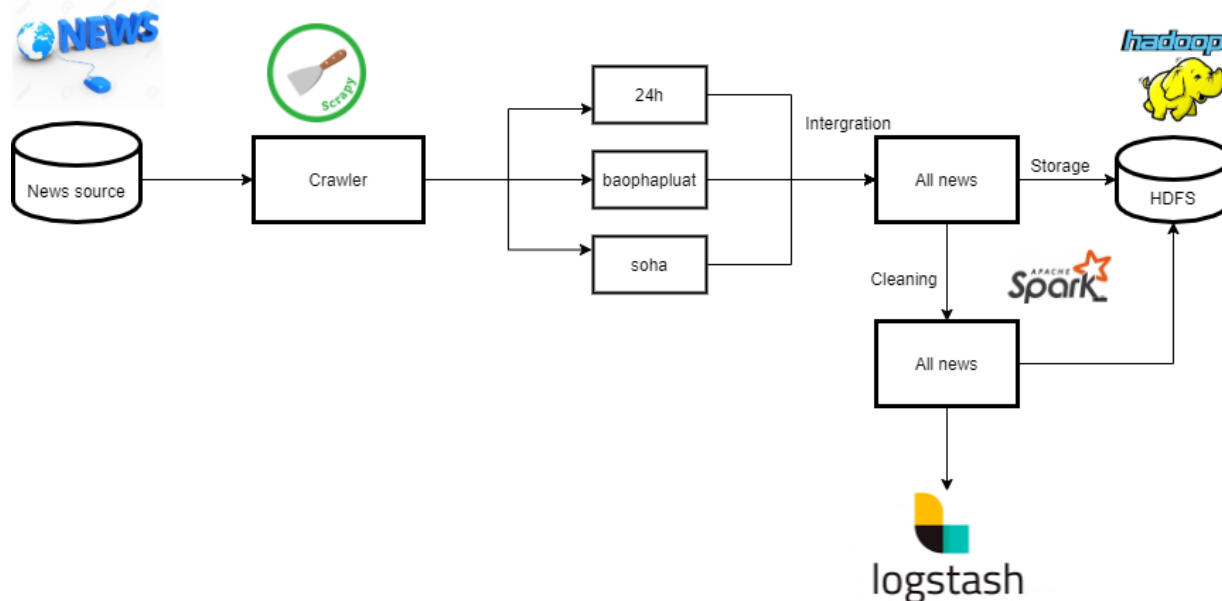
- ELK Stack:

Cài đặt trên cả ba máy.

```
timestamp cluster status node.total node.data shards pri relo init unassign pending_tasks |
08:37:57 mycluster green 3 3 24 12 0 0 0 0
```

Hình 4: Kết quả cài đặt Elasticsearch

### 3 Thu thập và lưu trữ dữ liệu



Hình 5: Luồng xử lý thu thập dữ liệu

#### 3.1 Dữ liệu

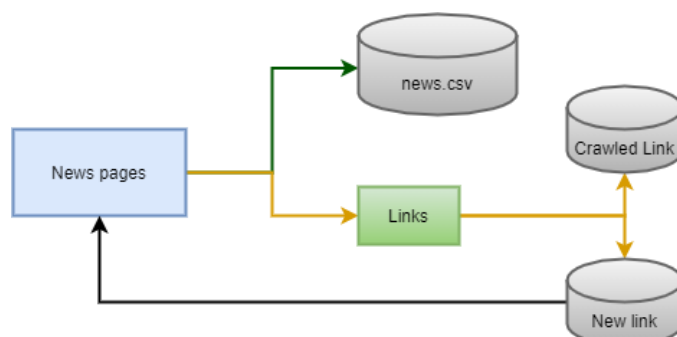
Dữ liệu được thu thập từ ba trang báo trực tuyến **soha.vn**, **baophapluat.vn** và **24h.com**. Những bài viết trên ba trang này đa dạng về thể loại và cấu trúc bài viết.

#### 3.2 Thu thập dữ liệu

Cấu trúc của mỗi trang báo khác nhau nên cần xây dựng crawler riêng cho từng trang web. Công cụ được nhóm sử dụng là Scrapy, một framework python giúp cho việc crawl được



song song hóa, với khả năng chịu lỗi tốt. Tuy nhiên, đầu vào của Scrapy là danh sách các đường dẫn (link) cần crawl nên cần phải thu thập link và xử lý trước khi truyền vào cho scrapy.



Hình 6: Luồng crawl dữ liệu

Với mỗi trang web, việc trích xuất các trường thông tin được xử lý theo đường dẫn Xpath của thành phần tương ứng trên trang HTML.

### 3.3 Tích hợp và làm sạch dữ liệu

Quá trình làm sạch dữ liệu cần phải thực hiện các công đoạn:

- Loại bỏ các trang lỗi khi crawl (trang không có nội dung, bài báo không có tiêu đề, id bài báo không đúng chuẩn, xử lý các trường dữ liệu bị thiếu).
- Chuẩn hóa dữ liệu về định dạng cần lưu trữ.

Dữ liệu sau khi làm sạch được lưu dưới dạng CSV và lưu trữ trong Hadoop.

### 3.4 Kết quả

Tốc độ thu thập dữ liệu: 800-900 trang/ 1 phút.

Nguồn	Link đã duyệt qua	Số lượng bài báo	Dung lượng (MB)
24h.com	~800.000	555143	802
Baophapluat.vn	~100.000	95690	314
Soha.vn	~300.000	196497	613

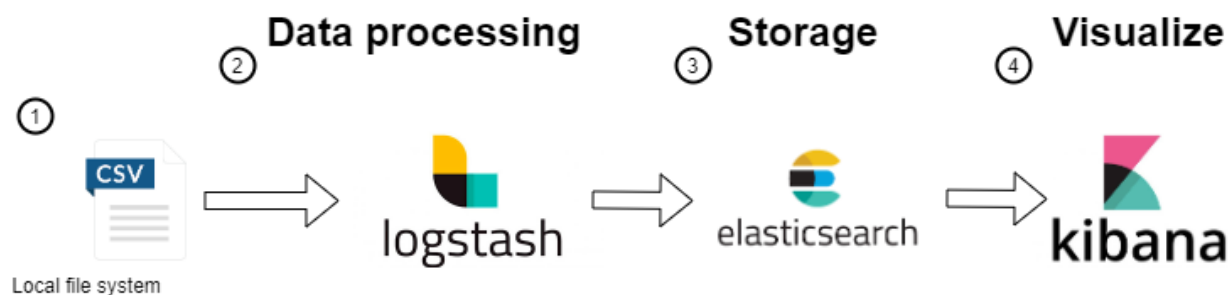
Bảng 1: Kết quả thu thập dữ liệu:

Kết quả dữ liệu sau khi được lưu vào HDFS được thể hiện trong hình sau:

Owner	Group	Size	Last Modified	Replication	Block Size	Name
hadoop	supergroup	1.61 GB	Dec 21 16:08	2	64 MB	all_news_cleaned.csv
hadoop	supergroup	802.19 MB	Dec 21 15:56	2	64 MB	news_24h.csv
hadoop	supergroup	314.21 MB	Dec 21 16:00	2	64 MB	news_laws.csv
hadoop	supergroup	612.84 MB	Dec 21 15:59	2	64 MB	news_soha.csv

Hình 7: Minh chứng dữ liệu lưu trong HDFS

#### 4 Tìm kiếm và trực quan hóa dữ liệu



Hình 8: Luồng xử lý lưu trữ, trực quan dữ liệu với ELK Stack

ELK Stack là một nhóm phần mềm nguồn mở, dựa trên Elastic cho phép tìm kiếm, phân tích, thể hiện trực quan các log thu thập được từ các nguồn, các log này ở bất kỳ định dạng nào, ELK là trung tâm phân tích log. Trung tâm log có nhiệm vụ xác định các vấn đề phát sinh trên các server, các ứng dụng thay cho việc truy cập trực tiếp vào log của từng server, từng ứng dụng. ELK gồm các thành phần chính:

- Elasticsearch - máy chủ lưu trữ và tìm kiếm dữ liệu
- Logstash - thành phần xử lý dữ liệu, sau đó nó gửi dữ liệu nhận được cho Elasticsearch để lưu trữ
- Kibana - ứng dụng nền web để tìm kiếm và xem trực quan các logs
- Beats - gửi dữ liệu thu thập từ log của máy đến Logstash

Trong báo cáo này, ELK được sử dụng để lưu trữ dữ liệu liên quan tới bài báo đã thu thập được.

##### 4.1 Cài đặt và lưu trữ

Dữ liệu sau khi được xử lý bằng Spark được lưu lại thành một file duy nhất ở định dạng CSV. Elasticsearch lưu trữ dữ liệu dưới dạng JSON nên thông thường nếu muốn đẩy dữ liệu vào Elasticsearch ta cần một bước chuyển từ dữ liệu dạng CSV sang JSON.

Hướng tiếp cận cơ bản nhất có thể kể đến chính là chuyển dữ liệu từ CSV sang JSON và dùng bulk API của Elasticsearch để đẩy dữ liệu vào cụm Elastic.

Tuy nhiên, phương pháp này tồn tại nhược điểm lớn đó là phải cần thêm một bước tạo file JSON, dung lượng file JSON bị giới hạn nhỏ hơn 2GB. Chính vì vậy, Logstash được sử dụng làm giải pháp thay thế.

Với Logstash, chỉ cần cấu hình đầu vào và đầu ra dữ liệu (địa chỉ Elasticsearch master) Logstash sẽ xử lý phân đọc và đẩy dữ liệu vào cụm Elasticsearch.

```

1 input {
2   file {
3     path => "../..../all_news_cleaned.csv"
4     start_position => "beginning"
5   }
6 }
7 filter {
8   csv {
9     columns => ['content', 'news_id', 'sapo', 'title', 'topic',
10    'url', 'sub_topic']
11   }
12 }
13 output {
14   elasticsearch {
15     hosts => ["192.168.0.111:9200"]
16     index => "zootopia_news"
17   }
18 }

```

Hình 9: Cấu hình file config\_news.conf cho Logstash

Khởi chạy Logstash đẩy dữ liệu vào cụm Elasticsearch:

```
bin/logstash -f config/config_news.conf
```

Elasticsearch làm nhiệm vụ lưu trữ, xử lý tìm kiếm thông tin dưới dạng full text search. Điều này giúp tìm kiếm bài báo theo nội dung một cách dễ dàng.

Tiếp theo, để trực quan hóa dữ liệu, giúp cho việc quản lý lưu trữ và tìm kiếm các bài báo một cách dễ dàng báo cáo sử dụng Kibana, một công cụ khác trong bộ công cụ ELK Stack. Để giúp Kibana xác định nguồn dữ liệu, chỉ cần cấu hình lại địa chỉ **host** trong file **config/kibana.yml** về địa chỉ master của cụm Elastic.

## 4.2 Kết quả thực hiện

Kết quả thực hiện được thể hiện trong các hình từ Hình 10 đến Hình 14.

Pretty

Raw

Preview

Visualize

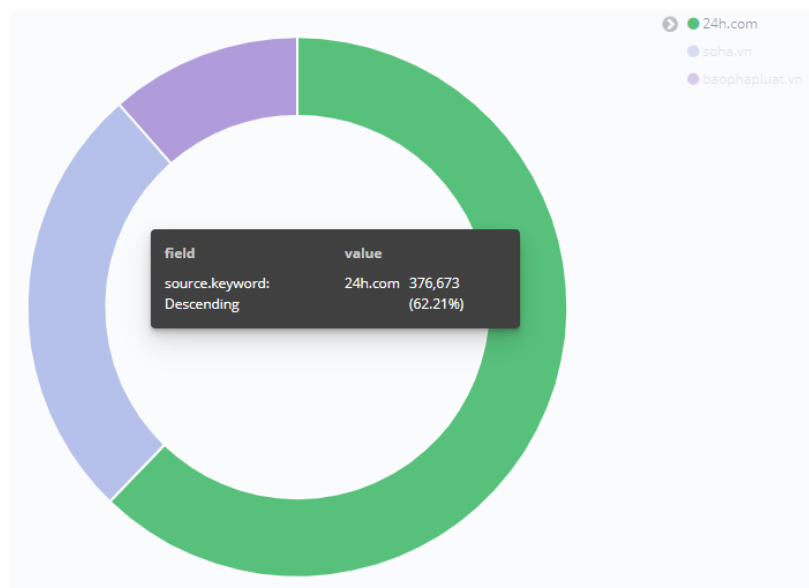
Text

1	health	status	index	uuid	pri	rep	docs.count	docs.deleted	store.size	pri.store.size
2	green	open	zootopia_news	vmYbKdCPTfyTE0Sgojx5QA	5	1	605551	0	4gb	2gb
3	green	open	.kibana_1	BccQ7kL_TuWj2AtPLqbIwx	1	1	5	0	42.4kb	21.2kb
4	green	open	zootopia	G0TVaOmcSOSw1Z1WMVUszA	5	1	386163	0	525.1mb	262.8mb
5	green	open	.kibana_task_manager	VpeNX-cFR9iInk5vXpaXdg	1	1	2	0	25.1kb	12.5kb
6										

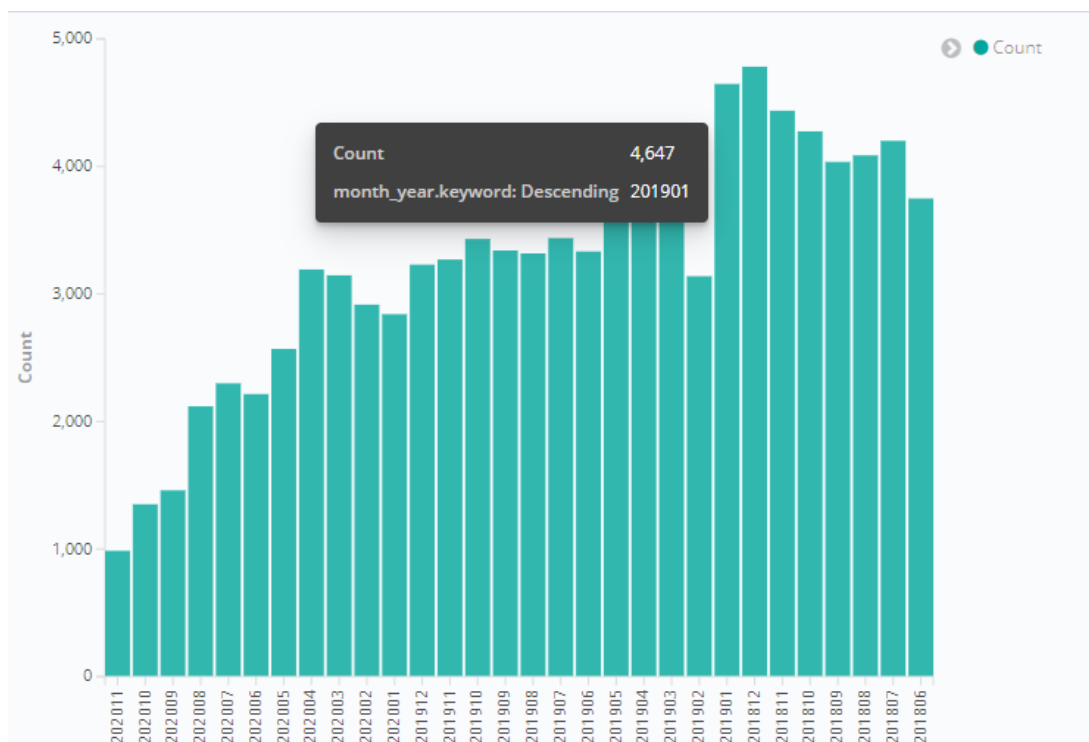
Hình 10: Dữ liệu được lưu trữ trong cụm Elasticsearch

index	shard	pri	rep	state	docs	store	ip	node
zootopia	3	r		STARTED	77651	52.7mb	192.168.0.113	node3
zootopia	3	p		STARTED	77651	52.2mb	192.168.0.112	node2
zootopia	1	p		STARTED	76792	52.5mb	192.168.0.113	node3
zootopia	1	r		STARTED	76792	52.6mb	192.168.0.112	node2
zootopia	2	p		STARTED	77200	52.7mb	192.168.0.111	node1
zootopia	2	r		STARTED	77200	52.7mb	192.168.0.112	node2
zootopia	4	p		STARTED	77371	52.3mb	192.168.0.111	node1
zootopia	4	r		STARTED	77371	52.2mb	192.168.0.113	node3
zootopia	0	p		STARTED	77149	52.4mb	192.168.0.111	node1
zootopia	0	r		STARTED	77149	52.4mb	192.168.0.112	node2
zootopia_news	3	r		STARTED	120855	410.6mb	192.168.0.113	node3
zootopia_news	3	p		STARTED	120855	411.8mb	192.168.0.112	node2
zootopia_news	1	p		STARTED	121415	415.5mb	192.168.0.111	node1
zootopia_news	1	r		STARTED	121415	414.1mb	192.168.0.112	node2
zootopia_news	2	p		STARTED	120915	412.9mb	192.168.0.113	node3
zootopia_news	2	r		STARTED	120915	411.3mb	192.168.0.112	node2
zootopia_news	4	p		STARTED	121321	412.7mb	192.168.0.111	node1
zootopia_news	4	r		STARTED	121321	412.5mb	192.168.0.113	node3
zootopia_news	0	r		STARTED	121045	405.3mb	192.168.0.111	node1
zootopia_news	0	p		STARTED	121045	408.2mb	192.168.0.112	node2
.kibana_task_manager	0	r		STARTED	2	12.5kb	192.168.0.111	node1
.kibana_task_manager	0	p		STARTED	2	12.5kb	192.168.0.113	node3
.kibana_1	0	p		STARTED	5	21.2kb	192.168.0.111	node1
.kibana_1	0	r		STARTED	5	21.2kb	192.168.0.113	node3

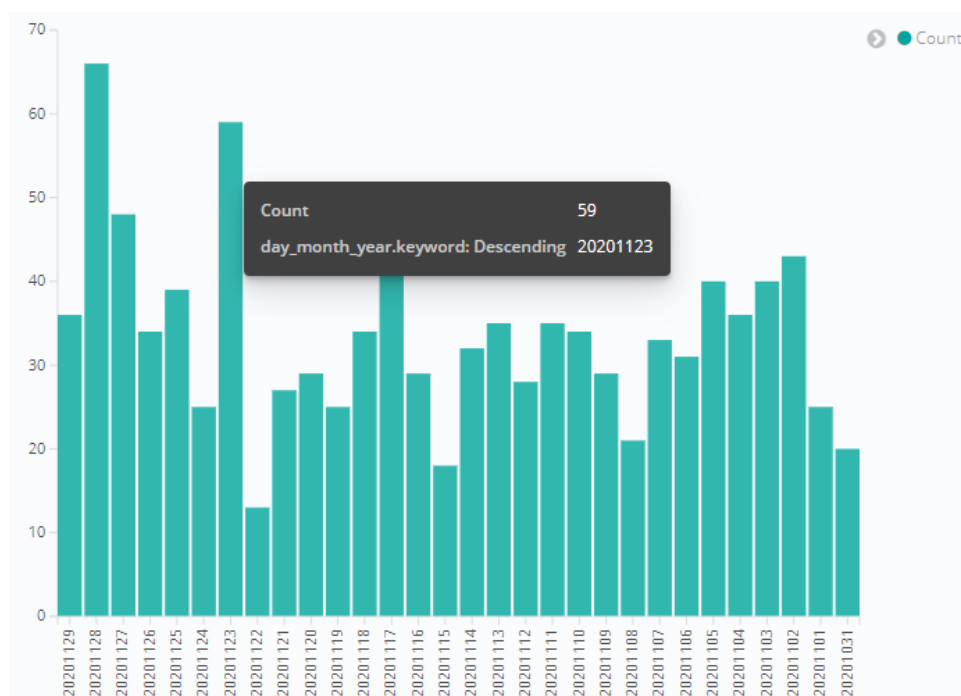
Hình 11: Phân bố dữ liệu trên các máy



Hình 12: Phân bố dữ liệu theo nguồn

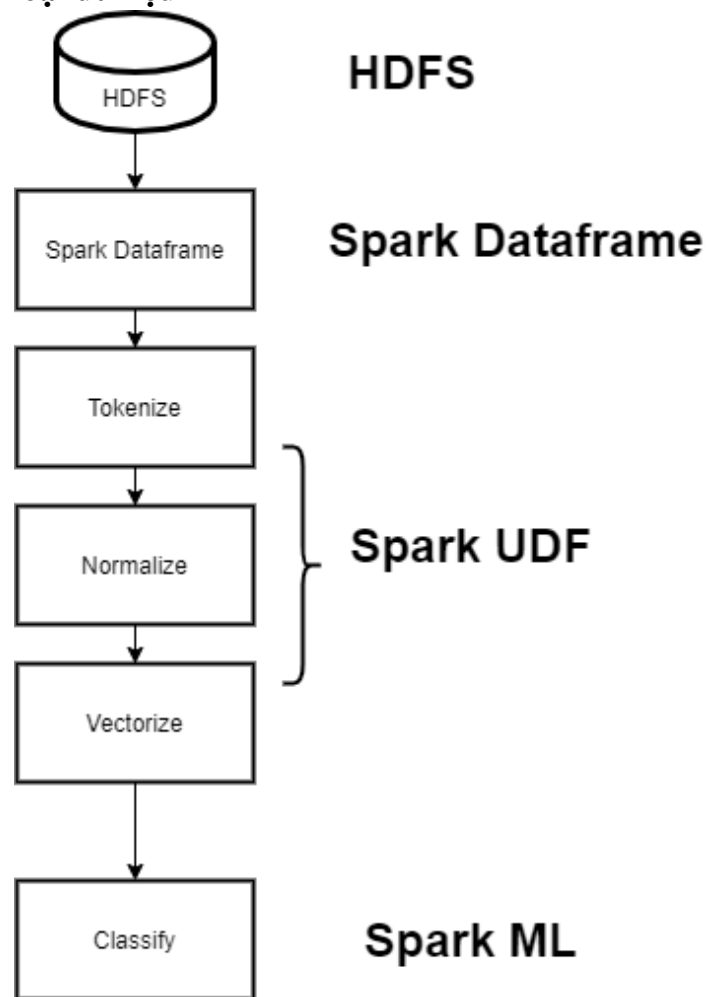


Hình 13: Phân bố dữ liệu trang soha.vn theo tháng



Hình 14: Phân bố dữ liệu trang soha.vn theo ngày trong tháng

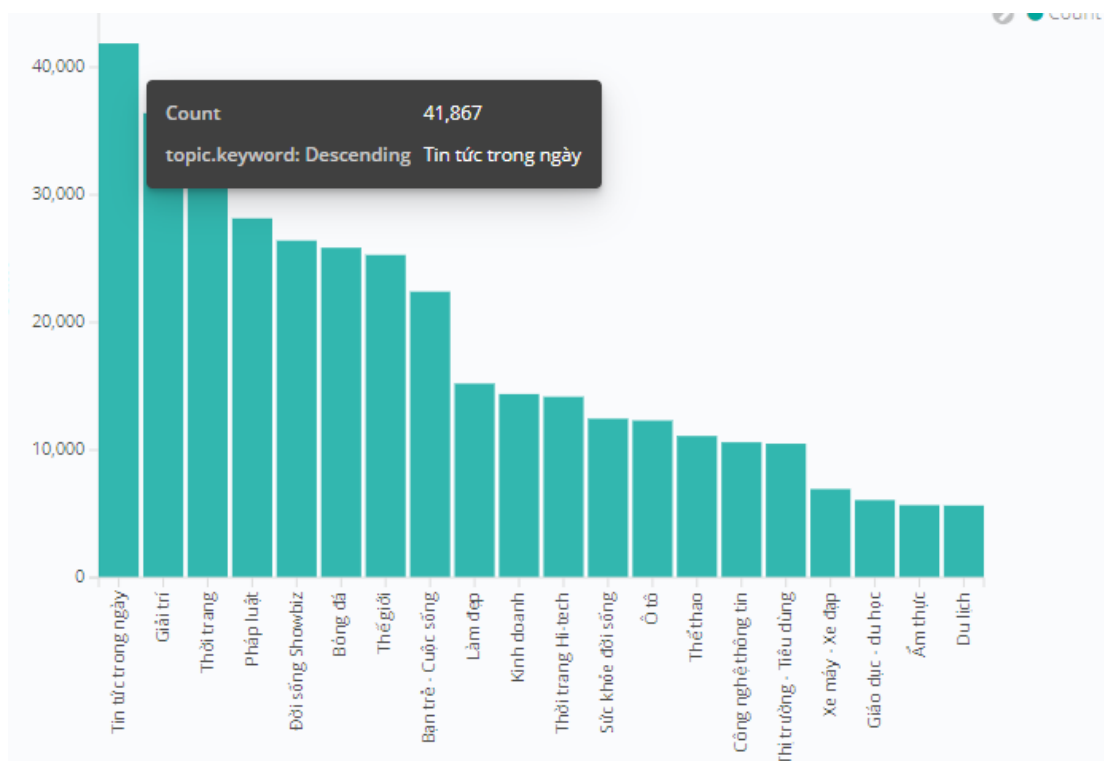
## 5 Xử lý và phân loại dữ liệu



Hình 15: Luồng xử lý và xây dựng mô hình phân loại văn bản.

Bộ phận xử lý dữ liệu có nhiệm vụ chuẩn hóa dữ liệu theo định dạng cần thiết để đưa vào mô hình học máy. Luồng xử lý dữ liệu được thể hiện trên hình Hình 15.

Dữ liệu được sử dụng cho bài toán phân loại văn bản được lấy từ trang **24h.com** với 20 chủ đề có số lượng bài báo thu thập được lớn nhất. Hình 16 cho thấy các chủ đề có số lượng lớn nhất và số lượng bài tương ứng được biểu diễn trên đồ thị sử dụng Kibana.



Hình 16: Thống kê dữ liệu đầu vào cho bài toán phân loại văn bản

## 5.1 Chuẩn bị dữ liệu

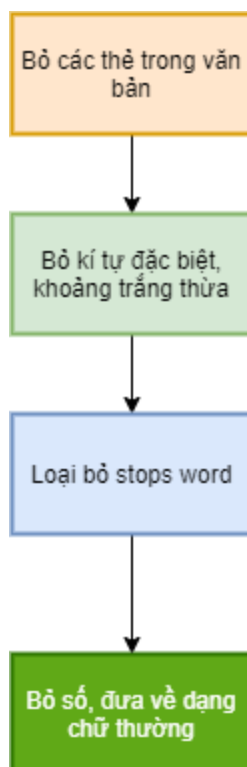
### 5.1.1 Tách từ Tiếng Việt

Trong Tiếng Việt, từ cấu tạo từ một hay nhiều tiếng. Các từ tạo bởi một tiếng có thể ghép với nhau để tạo thành từ khác. Ý nghĩa của từ theo đó cũng thay đổi trong từng văn cảnh khác nhau. Ví dụ trong cụm “Giới chính trị” thì “chính trị” là một từ, khác nghĩa với việc “chính” và “trị” tách rời nhau. Do đó quá trình xử lý tách từ cần quan tâm đến cấu tạo và ngữ nghĩa của từ, cũng như văn cảnh của câu. Vì vậy, việc xử lý tách từ Tiếng Việt là rất quan trọng.

Báo cáo sử dụng thư viện Pyvi để tách từ. Đây là một sản phẩm của nhóm nghiên cứu đến từ Đại học Bách khoa Hà Nội [1].

### 5.1.2 Chuẩn hóa câu

Bước chuẩn hóa thực hiện theo các bước trên Hình 17.



Hình 17: Luồng chuẩn hóa dữ liệu

Đầu tiên, tất cả thẻ HTML (ví dụ: **bold**) còn lại trong văn bản sẽ bị loại bỏ. Tiếp theo, bỏ đi các ký tự đặc biệt không phải chữ cái và khoảng trắng thừa. Cuối cùng là bỏ chữ số và đưa chuỗi đầu vào về dạng chữ thường. Các thao tác chuẩn hóa sẽ được thực hiện thông qua việc sử dụng các biểu thức chính quy (regular expression) cũng như các phương thức của String trong python.

Tuy nhiên, trong Spark, để có thể thực hiện một hàm tự định nghĩa một cách song song trên nhiều máy, cần đưa hàm về dạng **udf** function. Để tương thích với việc sử dụng hàm remove stops word của SparkML, ở bước tiếp theo, kiểu dữ liệu trả về của hàm chuẩn hóa dữ liệu tự định nghĩa sẽ phải là ArrayType.

Với danh sách stops word đầu vào, StopWordsRemover của Spark ML Feature sẽ giúp loại bỏ stops word có trong chuỗi đầu vào.

### 5.1.3 Vector hóa

Sử dụng hai phương pháp cơ bản là CountVectorizer và TfidfVectorizer để mã hóa dữ liệu nội dung bài báo đầu vào. Đầu vào của các Class này là SparkDataFrame với tên cột dữ liệu và tên cột dữ liệu mã hóa dưới dạng Count hay Tfidf tương ứng. Vì số lượng từ lớn, nên để đảm bảo hiệu năng trong quá trình huấn luyện, báo cáo sử dụng 10000 từ có tần suất xuất hiện lớn nhất để mã hóa cho chuỗi đầu vào tương ứng với tham số vocabSize của CountVectorizer, TfidfVectorizer là 10000.



Với nhãn đầu vào ở dạng Text, ta cần mã hóa sang dạng số để mô hình có khả năng hiểu được. Spark ML hỗ trợ việc đó thông qua việc sử dụng StringIndexer. Tương tự với việc mã hóa nội dung, đầu vào của StringIndexer cũng là SparkDataFrame, tên cột chứa nhãn bài báo và tên cột đầu ra mã hóa tương ứng.

### 5.1.4 Chia tập dữ liệu

Dữ liệu đầu vào (biểu diễn dưới dạng SparkDataFrame) được chia thành hai tập train-test với tỉ lệ 7-3 tương ứng bằng cách sử dụng phương thức randomSplit, hàm này trả về hai tập train-test tương ứng. Kết quả thu được, tập train có 160001 văn bản và tập test có 69007 văn bản.

### 5.2 Xây dựng mô hình phân loại văn bản

Mô hình Machine Learning được lựa chọn là Naïve Bayes, một mô hình xác suất cơ bản cho phân loại dữ liệu. Để đánh giá chất lượng mô hình, sử dụng Multiclass Classification Evaluator trong module *evaluate* của *spark.ml* với metrics là “accuracy”. Đầu vào của hàm đánh giá là kết quả trả về từ hàm *transform* của model Naïve Bayes.

### 5.3 Kết quả đạt được

Một sơ hình ảnh về quá trình chạy:

**Spark Jobs (?)**

User: deserteagle735  
Total Uptime: 51 min  
Scheduling Mode: FIFO  
Active Jobs: 1  
Completed Jobs: 7

▶ Event Timeline

▼ Active Jobs (1)

Job Id ▼	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
7	collectAsMap at MulticlassMetrics.scala:61 collectAsMap at MulticlassMetrics.scala:61 (kill)	2020/12/21 21:08:38	1.2 min	0/2	0/14 (4 running)

Page: 1 1 Pages. Jump to 1 . Show 100 items in a page. Go

▼ Completed Jobs (7)

Job Id ▼	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
6	collect at NaiveBayes.scala:192 collect at NaiveBayes.scala:192	2020/12/21 20:33:17	35 min	2/2	207/207
5	collect at StringIndexer.scala:204 collect at StringIndexer.scala:204	2020/12/21 20:32:52	24 s	2/2	8/8
4	top at CountVectorizer.scala:236 top at CountVectorizer.scala:236	2020/12/21 20:32:51	0.3 s	1/1 (1 skipped)	7/7 (7 skipped)
3	count at CountVectorizer.scala:233 count at CountVectorizer.scala:233	2020/12/21 20:20:44	12 min	2/2	14/14

Page: 1 1 Pages. Jump to 1 . Show 100 items in a page. Go

Hình 18: Chạy job huấn luyện mô hình phân loại văn bản trên Spark

URL: spark://192.168.0.111:7077  
 Alive Workers: 2  
 Cores in use: 8 Total, 0 Used  
 Memory in use: 11.3 GiB Total, 0.0 B Used  
 Resources in use:  
 Applications: 0 Running, 3 Completed  
 Drivers: 0 Running, 0 Completed  
 Status: ALIVE

▼ Workers (2)

Worker Id	Address	State	Cores	Memory	Resources
worker-20201221200300-192.168.0.111-40751	192.168.0.111:40751	ALIVE	4 (0 Used)	6.6 GiB (0.0 B Used)	
worker-20201221200328-192.168.0.113-42885	192.168.0.113:42885	ALIVE	4 (0 Used)	4.6 GiB (0.0 B Used)	

▼ Running Applications (0)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
----------------	------	-------	---------------------	------------------------	----------------	------	-------	----------

▼ Completed Applications (3)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
app-20201221201919-0002	Zootopia-ML	8	2.0 GiB		2020/12/21 20:19:19	deserteagle735	FINISHED	1.4 h
app-20201221201444-0001	Zootopia-clean-data	8	2.0 GiB		2020/12/21 20:14:44	deserteagle735	FINISHED	1.5 min
app-20201221200332-0000	Zootopia-clean-data	4	2.0 GiB		2020/12/21 20:03:32	deserteagle735	FAILED	0.4 s

Hình 19: Các job đã được chạy xong

Thời gian tiền xử lý: 1.5 phút

Thời gian huấn luyện và kiểm thử: 1.4 giờ

Chất lượng mô hình: 73.84 % cho tập test.

## 6 Tổng kết và hướng phát triển

Báo cáo đã dựng nên một hệ thống thực hiện hầu hết các giai đoạn trong xử lý dữ liệu: thu thập, xử lý, lưu trữ và quản lý, trực quan hóa.

Hệ thống trên hoạt động đảm bảo các tính chất cần có của hệ thống dữ liệu lớn: tính sẵn sàng cao, tính mở rộng cao, tính bền vững của dữ liệu và khả năng quản lý dữ liệu.

Về ứng dụng, có thể tích hợp vào các hệ thống khác như: phát hiện tin giả, gợi ý tin tức. Hệ thống cũng có thể sử dụng độc lập trong ứng dụng tìm kiếm bài báo theo nội dung, theo tiêu đề, theo thời gian hay có thể gợi ý tin tức theo chủ đề.

Tuy hệ thống đã hoạt động một cách ổn định nhưng tính tự động hóa của hệ thống chưa thực sự cao: dữ liệu thu thập vẫn chưa được xử lý tự động, thời gian thực và vẫn cần lưu dữ liệu Local (phần Logstash). Do vậy trong tương lai, nhóm định hướng sẽ phát triển thêm module xử lý dữ liệu thời gian thực (sử dụng Spark Streaming hoặc Kafka); dữ liệu đầu vào của Logstash sẽ sử dụng luôn trong HDFS mà không cần thông qua file Local (sử dụng Hive hoặc thông qua webhdfs).

## **7 Tài liệu tham khảo**

- [1] T. V. Trung, "Python Vietnamese Toolkit," [Online]. Available: <https://github.com/trungtv/pyvi>.