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

BÁO CÁO MÔN HỌC
THU THẬP VÀ XỬ LÝ DỮ LIỆU LỚN

THU THẬP VÀ PHÁT HIỆN TIN GIẢ VỀ BỆNH NHÂN COVID BẰNG SPARK STREAMING VÀ ĐỒ THỊ TRI THỨC

Giảng viên hướng dẫn: TS. Đỗ Bá Lâm
TS. Đào Thành Chung

Nhóm thực hiện đề tài:

Học viên	**Mã HV**
Lê Tiến Thành	CB190211
Nguyễn Quốc Khánh	CB190231

Hà Nội – 2020

MỤC LỤC

MỤC LỤC 2
DANH MỤC CÁC HÌNH VẼ 3
DANH MỤC CÁC KÝ HIỆU VÀ TỪ VIẾT TẮT 4
Phân công công việc 5
CHƯƠNG 1 Đặt vấn đề 6
1.1. Phân tích tổng quan đề bài 6
1.2. Phạm vi triển khai 6
1.3. Khó khăn chung 7
1.4. Kịch bản sử dụng 8
CHƯƠNG 2 Mô hình hệ thống đề xuất 10
2.1. Mô hình tổng quan 10
2.2. Bộ thu thập dữ liệu (Crawler) 11
2.2.1. Crawler để thu thập nguồn tin chính thống 12
2.2.2. Crawler để thu thập dữ liệu chưa được kiểm chứng 14
2.3. Trích xuất đối tượng từ dữ liệu ngôn ngữ tiếng Việt 17
2.3.1. Các vấn đề khi xử lý ngôn ngữ tự nhiên 17
2.3.2. Mô hình tổng quát 19
2.3.3. Mô đun tiền xử lý 21
2.3.4. Mô đun tìm mối liên hệ giữa các thực thể 22
2.3.5. Mô đun tách object 25
2.3.6. Tương tác cơ sở dữ liệu neo4j 27

2.3.7. Mô đun phát hiện tin giả 28
CHƯƠNG 3 Quy trình cài đặt hệ thống và các kịch bản demo 30
3.1. Quy trình cài đặt 30
CHƯƠNG 4 Tổng kết và hướng phát triển 32
4.1. Tổng kết 32
4.2. Định hướng phát triển 32
TÀI LIỆU THAM KHẢO 32

DANH MỤC CÁC HÌNH VẼ

Hình 2-1 Mô hình tổng quan 10
Hình 2-2 Mô hình thu thập xử lý nguồn tin chính thống 12
Hình 2-3 Mô hình thu thập và xử lý lượng lớn các dữ liệu báo chí chưa được kiểm duyệt (lưu ý, thứ tự hoạt động được viết theo số thứ tự từ 1-5 như trên) 14
Hình 2-4 Mô hình xử lý NLP thông thường 20
Hình 2-5 Mô hình xử lý NLP trong bài toán 21
Hình 2-6 Ví dụ về mối liên hệ giữa các nút bệnh nhân 22
Hình 2-7 Liên kết giữa các nút bệnh nhân từng cùng một chuyến bay 23
Hình 2-8 Thông tin của 1 nút bệnh nhân 25
Hình 2-9 Đồ thị tri thức COVID-19 28

DANH MỤC CÁC KÝ HIỆU VÀ TỪ VIẾT TẮT

Từ viết tắt	Ý nghĩa
CNTT	Công nghệ thông tin
CSDL	Cơ sở dữ liệu
CSV	Comma Separated Values (là một loại định dạng văn bản)

Phân công công việc

Học viên	Công việc được giao	Mức độ hoàn thành
Lê Tiến Thành	Thiết kế và triển khai các luồng dữ liệu (thu thập, tích hợp và phân luồng dữ liệu)	Hoàn thành.
Nguyễn Quốc Khánh	Phân tích và thiết kế các mô đun xử lý ngôn ngữ tiếng Việt. Xây dựng mô đun trích xuất các đối tượng thông tin trong tin tức. Xây dựng cơ sở dữ liệu neo4j phục vụ xử lý đồ thị tri thức và các API tương tác liên quan.	Hoàn thành.

CHƯƠNG 1 Đặt vấn đề

1.1. Phân tích tổng quan đề bài

Đề bài được giao được nhóm hiểu cơ bản như sau. Đề tài xoay quanh chủ đề nóng hổi trong xã hội hiện nay với các thông tin chưa được kiểm chứng tràn lan trên các báo mạng không chính thống, các mạng xã hội . Với hiểu biết hữu ích từ khóa học *IT5427-Tích hợp và xử lý dữ liệu*, nhóm cần thiết kế và triển khai một mô hình đơn giản của hệ thông thu thập dữ liệu lớn và xử lý dữ liệu lớn, giải quyết vấn đề tin giả tràn lan kể trên. Hệ thống sẽ bao gồm các thành phần tương ứng với các chức năng cấp thiết của một hệ thống thu thập phân tích và phát hiện tin giả như sau:

- Bộ thu thập dữ liệu (chính thống hoặc không chính thống) bằng cách tìm kiếm và trích xuất các website báo chí.
- Công cụ đọc trích xuất các thực thể từ dữ liệu thu thập được dưới dạng text.
- Công cụ để từ dữ liệu đã trích xuất từ thông tin trên các báo chính thông nên đồ, xây dựng nên đồ thị tri thức.
- Công cụ để kiểm chứng các dữ liệu mới trên các kênh thông tin không chính thống, đánh giá xem thông tin là thật hay giả dựa vào đồ thị tri thức đã xây dựng.

Với mỗi thành phần của hệ thống, đề bài lại có những nhánh nhỏ khác. Phần tiếp theo nêu các hướng mà nhóm lựa chọn để hoàn thiện việc xác định cụ thể nhất phạm vi mà hệ thống của nhóm được xây dựng và triển khai.

1.2. Phạm vi triển khai

- Về lựa chọn ngôn ngữ triển khai, nhóm lựa chọn tiếng Việt. Lý do đầu tiên đây là tiếng mẹ đẻ, nhóm muốn tạo ra sản phẩm tuy nhỏ nhưng có thể áp dụng cho thực tế trong nước. Ngoài ra, sử dụng ngôn ngữ tiếng Việt khi hầu như không có thư viện hỗ trợ xử lý cũng là một thách thức nhóm em muốn chinh

phục. Lý do thứ 3 là việc xử lý các tin trong nước bọn em sẽ dễ dàng biết được đâu là nguồn tin chính thống, và đâu là nguồn tin có xác suất giả cao. Từ đó, tạo được đồ thị tri thức một cách chính xác nhất.

- Về lựa chọn phạm vi tin tức, nhóm làm theo gợi ý của giảng viên hướng dẫn là làm về chủ đề COVID-19.
- Về các loại thực thể và mối quan hệ giữa các thực thể, nhóm lựa chọn giới hạn phạm vi là các bệnh nhân mắc COVID-19 tại Việt Nam, các phương tiện vận tải, vị trí các nơi mà bệnh nhân COVID-19 tại Việt Nam đã đi qua. Đồng thời nhóm cũng trích xuất xem bệnh nhân nào đã khỏi bệnh, hoặc ai vẫn còn trong tình trạng nguy kịch, ai đã tử vong, vv … Thông tin này được trích từ nguồn thông tin chính thống của trang thông tin Bộ Y Tế Chính phủ Việt Nam tại https://ncov.moh.gov.vn/dong-thoi-gian. Nhóm lựa chọn thông tin phạm vi các thông tin này vì chúng có tính ứng dụng cao. Đồ thị tri thức về các thông tin này sẽ giúp người sử dụng hệ thống hình dung được bức tranh toàn cảnh về tình hình diễn biến dịch bệnh, dễ dàng hơn trong việc đưa ra quyết định nên đi hoặc tránh địa điểm nào, hoặc bản thân có nằm trong diện gần gũi với các bệnh nhân F0 hay không. Đồ thị này khi đem đi so sánh với các nguồn tin không chính thống có thể kiểm chứng được thông tin nào sai một cách tự động mà không cần phải ghi nhớ hết khối lượng thông tin về hàng trăm bệnh nhân.
- Đề bài có nêu hệ thống cần tích hợp thêm với các đồ thị tri thức mở hiện có như Dbpedia. Tuy nhiên, về thời gian và nhân lực có hạn, nhóm chưa thể tích phân tích và triển khai yêu cầu này.
- Đề bài yêu cầu hệ thống có tính năng phát hiện tin giả, sử dụng đồ thị tri thức. Nhóm đã thực hiện được tính năng này, cụ thể nêu ở phần sau.

1.3. Khó khăn chung

- Khó khăn về tiếng Anh: là người Việt nên nhiều cấu trúc ngữ pháp phức tạp hay các cụm từ lóng trong các bài báo nước ngoài bọn em khó để hiểu và phân

tích. Ngoài ra các bài báo nước ngoài thường có tính khái quát rộng về nhiều mặt cũng như các khu vực, các quốc gia khác nhau. Vì vậy, với một người ở Việt Nam, bọn em rất khó tìm được các vùng giao nhau giữa các tạp chí (các bài báo cùng nói về 1 vấn đề)

- Khó khăn về tiếng Việt: Tuy là tiếng mẹ đẻ nhưng có nhiều cụm ngữ pháp khó. Ngoài ra, có một số "nhà báo" viết tin dưới dạng câu rút gọn, không tuân theo cấu trúc ngữ pháp chung nên hầu như không thể xử lý như ngôn ngữ thông thường. Tổng thể lại thì hầu như rất ít thư viện hỗ trợ xử lý tiếng Việt. Mạnh nhất là thư viện underthesea nhưng chỉ dừng ở mức phân tích từ loại.

- Khó khăn về kích thước dữ liệu: Đây là môn học máy với dữ liệu lớn, tuy nhiên khi xác định được phạm vi đề bài như trên và triển khai thử, nhóm phát hiện ra rằng tuy số lượng tin tức về các bệnh nhân COVID-19 trên trang thông tin https://ncov.moh.gov.vn/dong-thoi-gian cũng không nhỏ, xấp xỉ 1000 bản ghi, tuy nhiên việc xử lý số lượng dữ liệu này chưa lên đến mức phải ứng dụng các kỹ thuật tích hợp xử lý dữ liệu lớn như BatchProcessing và in memory MapReduce trong ApacheSpark. Nói cách khác, sau khi triển khai thử trên đầu vào trên, chỉ cần sử dụng code đơn luồng cũng có thể xử lý hết dữ liệu chính thống trong vòng không quá 5 phút. Vậy nên, nhóm giải quyết vấn đề thiếu dữ liệu bằng cách chuyển cách đặt vấn đề. *Thay vì việc tích hợp và xử lý dữ liệu lớn trên tập dữ liệu chính thống, nhóm chúng tôi xây dựng hệ thống để có thể scalable khi phân tích và xử lý dữ liệu chưa được kiểm chứng vì nguồn dữ liệu chưa được kiểm chứng là vô hạn.*

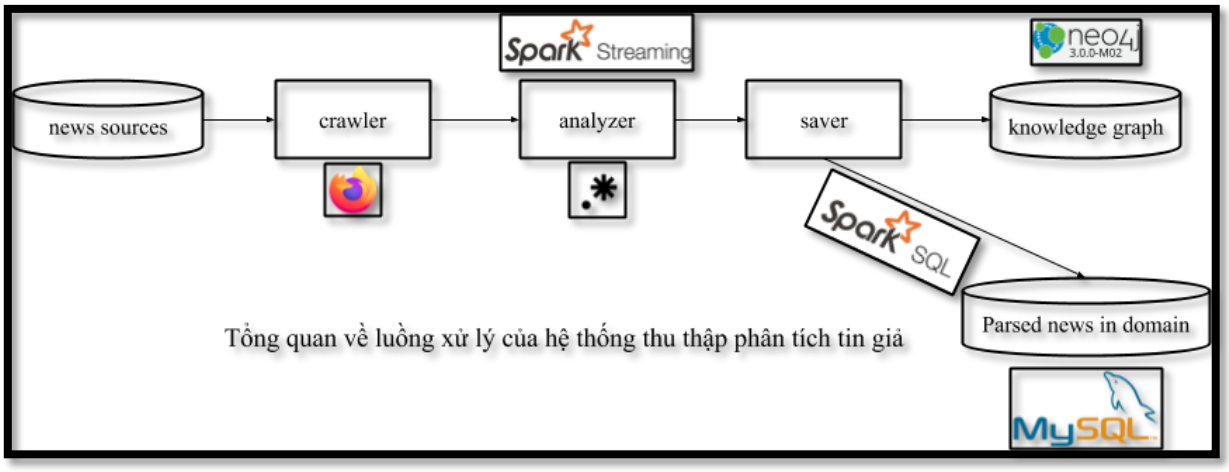
1.4. Kịch bản sử dụng

Hệ thống phát hiện tin giả này rất hữu ích cho các cơ quan quản lý, hành pháp, ví dụ như cảnh sát trật tự người phải điều tra tin giả. Công cụ có thể tự động học được một đồ thị tri thức chứa thông tin chính xác, trung thực, và xác minh tự động một mệnh đề đưa ra là đúng hay sai.

Các đơn vị liên quan nêu trên có thể đưa các mệnh đề vào kiểm tra thủ công, ví dụ "BN 91 đã tử vong". Không cần phải ghi nhớ và tìm kiếm thông tin để xác minh quá nhiều, các điều tra viên cũng có thể nhanh chóng xác minh được một bài đăng trên mạng xã hội là thật hay giả trong một tích tắc. Công cụ sẽ làm giảm thời gian, chi phí một cách đáng kể cho công việc điều tra thủ công. Hơn nữa, công cụ có thể được sử dụng ở chế độ tự động dò quét, tìm kiếm và streaming những bài đăng mới, xác minh các bài đăng đó trong thời gian thực. Nói tóm lại, cả hai kịch bản sử dụng trên đều góp phần giúp đơn vị hành pháp nhanh chóng lọc được một lượng lớn thông tin với chi phí tiết kiệm nhất.

CHƯƠNG 2 Mô hình hệ thống đề xuất

2.1. Mô hình tổng quan



Hình 2-1 Mô hình tổng quan

Mô hình nhóm thiết kế được hiển thị trong Hình 1. Các ô hình chữ nhật gồm *crawler, analyzer* và *saver* lần lượt là các thành phần xử lý chính trong hệ thống. Các ô hình trụ gồm *new sources, knowledge graph* và *parsed news in domain* là các loại dữ liệu thu thập hoặc dữ liệu sinh ra sau khi hệ thống xử lý. Các hình minh họa liền kề các thành phần thể hiện các công nghệ đã được sử dụng để hỗ trợ xây dựng hệ thống này.

Cụ thể, các thành phần xử lý bao gồm:

- *Crawler (Thành phụ trách):* Là bộ lấy dữ liệu có các chức năng gồm truy cập các website, tìm các url liên quan đến chủ đề COVID-19 tại Việt Nam và trích các đoạn văn trong mã nguồn HTML gửi đến các bộ phận sau của hệ thống. Nhóm đã thử sử sụng ***scrapy***, ***requests*** trong python để thu thập dữ liệu. Cuối cùng, nhóm đi đến kết luận là sử dụng trình duyệt ***Firefox*** điều kiển bởi ***selenium*** trong ***python3*** để lấy được nhiều nhất các thông tin từ các website báo mạng không truyền thống, có nội dung động.
- *Analyzer (Khánh phụ trách, Thành tích hợp vào SparkStreaming): Là* bộ phân tích ngữ nghĩa từ những đoạn văn mà *crawler* lấy về. Từ các câu văn ngôn ngữ

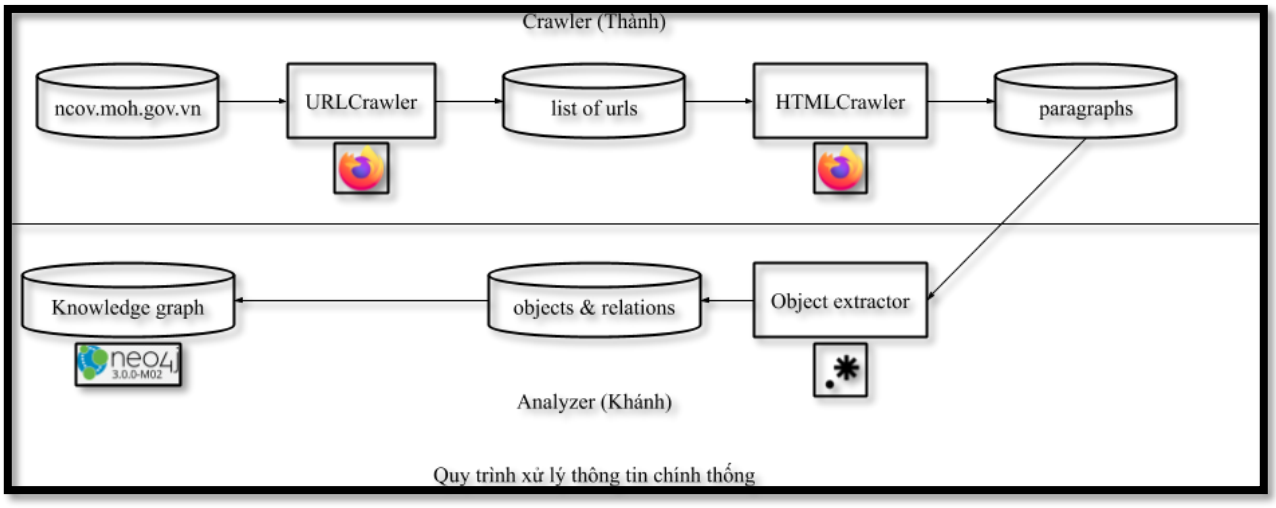
tự nhiên, sử dụng luật ngữ pháp tiếng Việt được triển khai trên nền tảng là các câu lệnh ***Regular expression*** để bắt chính xác một lượng thông tin trong phạm vi nhóm đã lựa chọn là tin tức về tình hình COVID-19 ở Việt Nam. Sau đó bộ phân tích chuyển các "*đối tượng"* hoặc *"quan hệ"* sang danh sách cạnh trên đồ thị hoặc xác minh các đối *"quan hệ"* đã phân tích được là đúng hay sai dựa vào đồ thị tri thức có sẵn. Bộ *analyzer* này cũng được tích hợp vào ***SparkStreaming***.

- *Saver (Khánh phụ trách saver cho knowledge graph, Thành phụ trách saver cho parsed news):* Là bộ phận lưu các thông tin đã phân tích được vào các cơ sở dữ liệu phù hợp. Với *dữ liệu chính thống* đã phân tích thành object, nhóm lưu lại vào cơ sở dữ liệu đồ thị ***Neo4j***. Còn với dữ liệu từ nguồn không chính thống cần xác minh thì nhóm sẽ thực hiển tìm kiếm xem *"quan hệ"* phân tích được trên nguồn không chính thống đúng hay sai, và lưu lại kết quả phân loại vào ***MySQL*** sử dụng ***SparkSQL.***

2.2. Bộ thu thập dữ liệu (Crawler)

Như đã mô tả ở phần đặt vấn đề, hệ thống chạy với 2 kịch bản khác nhau. Kịch bản thức nhất là thu thập và phân tích, sau đó lưu thông tin chính thống vào đồ thị tri thức. Kịch bản thứ hai là thu thập một lượng lớn thông tin chưa được kiểm chứng, xác minh với đồ thị tri thức đã xây dựng trên thông tin chính thống và lưu lại kết quả. Phần sau đây sẽ mô tả cụ thể thiết kế của *crawler* trong từng kịch bản nêu trên.

2.2.1. Crawler để thu thập nguồn tin chính thống



Hình 2-2 Mô hình thu thập xử lý nguồn tin chính thống

Sau tìm hiểu, nhóm lựa chọn nguồn tin chính thống bao gồm toàn bộ các dòng cập nhật trạng thái của Bộ Y tế về tình hình các bệnh nhân COVID-19 được phát hiện và chữa trị tại Việt Nam tại đường dẫn sau: https://ncov.moh.gov.vn/dong-thoi-gian. Crawler được chia làm hai thành phần nhỏ, URLCrawler liệt kê ra một hàng đợi các URL cần được truy cập và lấy mã nguồn HTML về phân tích. Sau đây là một số ví dụ của các URL mà URLCrawler lấy được khi đọc thông tin về dòng thời gian chính thống trên trang ncov.moh.gov.vn này:

- https://ncov.moh.gov.vn/web/guest/dong-thoi-gian?p_p_id=101_INSTANCE_iEPhEhL1XSde&...&_101_INSTANCE_iEPhEhL1XSde_cur=1
- https://ncov.moh.gov.vn/web/guest/dong-thoi-gian?p_p_id=101_INSTANCE_iEPhEhL1XSde&...&_101_INSTANCE_iEPhEhL1XSde_cur=2
- ...

- https://ncov.moh.gov.vn/web/guest/dong-thoi-gian?p_p_id=101_INSTANCE_iEPhEhL1XSde&...&_101_INSTANCE_iEPhEhL1XSde_cur=10

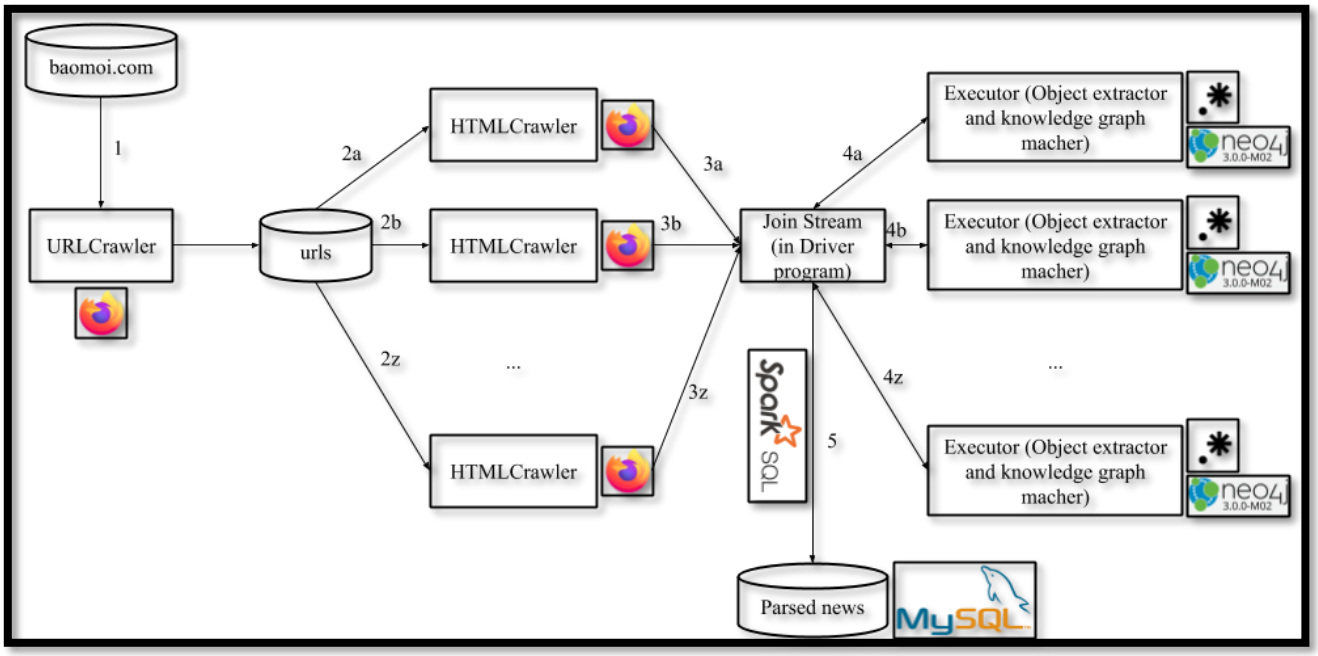
Cụ thể hơn, ban đầu nhóm triển khai Crawler dựa trên các thư viện chỉ truy cập vào web với backend như ***python3/requests*** hoặc ***python3/scrapy.*** Các thư viện này không đủ tin tưởng khi render web có tính động cao như https://baomoi.com, khó follow được khi web redirect sử dụng javascripts. Vậy nên nhóm lựa chọn điều khiển ***Firefox*** bằng ***python3/selenium*** thông qua ***geckodriver***. Ngoài ra, làm vậy còn đảm bảo cho việc các website không nhận diện bộ thu thập tin tức của nhóm là bot chứ không phải là người dùng thông thường. Nhóm cũng phải đổi User-agent để tránh nguy cơ bị chặn, dẫn đến mất mát thông tin không đáng có.

Thành phần còn lại của Crawler là HTMLCrawler. Chức năng chính của HTMLCrawler là: cho đầu vào là một URL, trả về các đoạn text được hiển thị trên website của URL ấy. HTMLCrawler sử dụng thêm thư viện ***python3/BeautifulSoup4*** để chuyển mã nguồn HTML về cấu trúc cây trong python có thể phân tích được, sau đó với các HTML tag có thể hiển thị được trên trình duyệt, HTMLCrawler trích xuất hết các đoạn văn đó và xuất cho các thành phần tiếp theo trong hệ thống để xử lý. Vấn đề phát sinh bắt buộc khiến chương trình phải xử lý theo đoạn mà không thể theo câu là vì trong tiếng Viêt, cụ thể là các bài báo thì các câu nhiều khi sử dụng thực thể ở câu trước. Ví dụ như: “Bệnh nhân này đã ngồi trên chuyến bay VN0001.” Câu này hoàn toàn vô nghĩa nếu như không có câu ở trước đề cập đến “bệnh nhân này” là bệnh nhân nào

Trong việc thu thập dữ liệu chính thống, có một phát sinh đó là dữ liệu có tính chất phụ thuộc lẫn nhau theo thời gian. Cụ thể, các thông tin về bệnh nhân được phát hiện trước thì phải được đọc và xử lý trước, các thông tin về bệnh nhân mới hơn cần phải được cập nhật vào sau. URLCrawler sắp xếp lại các URL với thời gian xa hơn lên

trước và HTML Crawler sắp xếp lại các đoạn văn ở dưới được xuất ra trước cho bộ phân tích Analyzer xử lý, đảm bảo không xảy ra lỗi này.

2.2.2. Crawler để thu thập dữ liệu chưa được kiểm chứng



Hình 2-3 Mô hình thu thập và xử lý lượng lớn các dữ liệu báo chí chưa được kiểm duyệt (lưu ý, thứ tự hoạt động được viết theo số thứ tự từ 1-5 như trên)

Nhóm đã tìm hiểu các kỹ thuật để thiết kế một hệ thống thu thập dữ liệu có tính scalability cao. Tuy nhiên, vì tài nguyên tính toán của nhóm có hạn với máy tính xách tay cá nhân cũng như thời gian và nhân lực hạn hẹp với 2 người, chúng tôi chỉ có thể triển khai mô hình với đầy đủ các thành phần. Tuy nhiên, chúng tôi sẽ vẫn nêu đầy đủ ý tưởng đã thiết kế để scale hệ thống thu thập dữ liệu này lên để xử lý khối lượng dữ liệu lớn.

Chúng tôi chọn https://baomoi.com làm trang nguồn để từ đó đào các URL liên quan. Tại bước 1 trong Hình 3, URLCrawler sẽ đọc mã nguồn của trang nói về topic COVID-19 của baomoi.com tại:

https://baomoi.com/phong-chong-dich-covid-19/top/328.epi

để từ đó lọc hết các thẻ a có chứa text hiển thị có từ khóa "covid-19" và đưa URL đọc được vào hàng đợi. Các URL lọc được lại được lấy để đọc mã nguồn, trích thẻ a có từ khóa "covid-19" đưa vào hàng đợi. Sau đó, tại bước 2 trong Hình 3, HTMLCrawler sẽ nhận các URL của URLCrawler, và đọc mã nguồn, trích các đoạn văn trong các thẻ HTML có thể hiển thị được. Cả URLCrawler và HTMLCrawler đều điều khiển một trình duyệt Firefox sử dụng thư viện python3/selenium. Tuy nhiên, có một vấn đề phát sinh là URLCrawler khi thăm một URL có thể trích thêm được hàng chục thậm chí hàng trăm URL mới. Trong lúc đó, HTMLCrawler cũng chỉ có thể duyệt được một URL. Tốc độ của URLCrawler nhanh gấp hàng chục lần so với HTMLCrawler dẫn đến việc ta phải song song hóa HTMLCrawler. Nhóm thiết kế để HTMLCrawler có thể chạy song song sử dụng python3/multithreading. Mỗi thread có thể điều khiển một process của trình duyệt Firefox mới, làm tăng được thoughput của HTMLCrawler so với URLCrawler. Đầu ra của HTMLCrawler sẽ là một chuỗi json có định dạng như sau:

{‘url’: ‘https://…’, ‘paragraph’: ‘...’}.

Sau khi các đoạn văn trong mã nguồn HTML của mỗi URL đã được tách, mỗi HTMLCrawler có thể truyền thông tin cho thành phần phía sau của hệ thống thông qua một TCP socket (bước 3 của Hình 3). Json được dump thành string, tách nhau bởi dấu xuống dòng "\n". Thành phần tiếp theo của hệ thống là chương trình Spark Streaming để phân loại thật giả các đoạn văn sử dụng đồ thị tri thức đã học được. Hiện tại, nhóm mới chỉ triển khai 1 HTMLCrawler, một TCP socket để truyền tin và một socketTextStream tại chương trình SparkStreaming để nhận dữ liệu, tuy nhiên nhóm cũng đã tìm hiểu cách tích hợp các đầu vào khác ví dụ như socketTextStream

khác hoặc Kafka stream (bằng hàm .join(otherStream, …)) một cách dễ dàng. Sau khi join, driver program của Spark Streaming có thể sử dụng dữ liệu đã join như là một DStream như với 1 stream.

Sau khi nhận dữ liệu, thành phần SparkStreaming của nhóm thực hiện những hàm sau:

- ***flatMap***, tách dòng từ dữ liệu string nhận được, đọc json trong mỗi dòng và trả về một list các dictionaries trong python3, mỗi dictionary tương ứng với một bản ghi/json/đoạn văn.
- ***map***, với mỗi dictionaries có được, thực hiện phân tích ngữ nghĩa trên trường 'paragraph' và matching với đồ thị tri thức. Trả về một dictionary có chứa thêm thông tin là thông tin có trong domain của bộ Analyzer hay không, nếu có trong domain thì thông tin có chính xác hay không.
- ***filter***, với mỗi dữ liệu đã được phân loại, chúng tôi lọc các đoạn text có trong domain để bước sau lưu lại, phục vụ người dùng trích xuất để sử dụng

Hiện tại nhóm chỉ triển khai ở dạng Spark Standalone Mode (chi tiết ở phần cài đặt phía sau). Nhóm cũng đã đọc về cách triển khai nhiều cluster của Spark trên nhiều node, để song song hóa quá trình tính toán nêu trên và tăng tốc độ hệ thống (xem bước 4 hình 3).

Cuối cùng, sau khi nhận được hết dữ liệu đã được xử lý trong batch, hệ thống sử dụng SparkSQL, cụ thể là hàm foreachRDD. Với mỗi RDD trong mỗi interval xử lý, nhóm callback để SparkSQL chuyển RDD trong mỗi batch đó thành dataframe. Sau đó, mỗi dataframe sẽ được insert vào MySQL thông qua JDBC connector. Việc này thuận tiện hơn và nhanh hơn so với dùng hàm collect RDD sau đó lại insert vào MySQL sử dụng các cách thông thường, giảm một bước convert.

Hệ thống lưu lại trường 'url', 'paragraph', 'truth' với truth là 1 nếu thông tin trong paragraph là tin thật. Các điều tra viên từ đây có thể dò lại được đoạn văn nào trong

đường link nào có thông tin sai sự thật bằng cách trích xuất MySQL. Hoặc ta có thể xây dựng frontend trích xuất và visualize dữ liệu đã xử lý.

Ngoài ra, nhóm còn gặp phải một số khó khăn trong việc lưu tiếng Việt bằng JDBC/SparkSQL. MySQL là không nhận được tiếng Việt được encode đúng cách bằng utf8mb4 encoding. Nhóm có thử add thêm option vào SparkSQL cũng vẫn gây ra lỗi không lưu được string với utf8mb4. Hơn nữa, tài liệu tham khảo cho vấn đề lưu tiếng việt từ SparkSQL tương đối khan hiếm. Vậy nên, nhóm đã xử lý bằng cách thay đổi cấu hình mặc định của MySQL tại my.cnf. Chi tiết xem tại phần cài đặt và demo.

2.3. Trích xuất đối tượng từ dữ liệu ngôn ngữ tiếng Việt

2.3.1. Các vấn đề khi xử lý ngôn ngữ tự nhiên

Các giải pháp đã đề xuất để xử lý ngôn ngữ tự nhiên và các vấn đề xử lý liên quan.

2.3.1.1. Tóm tắt văn bản

- Mục tiêu: Sử dụng các thuật toán xử lý ngôn ngữ, học máy để đưa nội dung văn bản về số lượng câu ít hơn và vẫn nêu được khái quát ý chính của văn bản.
- Ví dụ:

Tin tức:

As Xi Jinping prepared to address the World Health Assembly on Monday, it seemed like the Chinese leader might be in a vulnerable spot. More than 100 countries had signed onto a resolution calling for an indep endent probe into the origins of the coronavirus pandemic. While the lang uage in the document was thoroughly diplomatic, and did not call out any particular country, it grew out of a push by Australia to look into China's o wn failures in the initial stage of the crisis, and went against Beijing's stat ed desire for any investigation to be run by the World Health Organizatio n (WHO) itself. Chinese officials previously described Canberra's proposal as "highly irre sponsible," and accused Australian officials of undermining global efforts against the virus. But when Xi addressed the annual meeting of WHO me

```
mbers, he took a more conciliatory tone: of course China was willing to su
pport an investigation into the virus -- once the pandemic is over.
```

Tóm tắt trong 2 câu sử dụng thư viện sumy và ntlk:

```
['As Xi Jinping prepared to address the World Health Assembly on Monda
y, it seemed like the Chinese leader might be in a vulnerable spot.',
 'More than 100 countries had signed onto a resolution calling for an inde
pendent probe into the origins of the coronavirus pandemic.']
```

- Nhược điểm phát sinh:
    - Tùy theo độ dài của văn bản đầu vào, nếu để số câu rút gọn quá ít sẽ không đủ để tóm tắt ý chính trong văn bản.
    - Mất mát thông tin: về bản chất đây giống như một thể loại nén thông tin văn bản có mất mát. Ngoài các câu không được khát quát trong phần tóm tắt thì ta cũng sẽ mất các thông tin số liệu chi tiết có thể xuất hiện trong văn bản.
    - Câu văn đầu ra có thể chứa lỗi ngữ pháp: kể cả văn bản đầu vào lẫn văn bản tóm tắt đầu ra đều có thể chứa lỗi ngữ pháp. Đặc biệt các thuật toán xử lý tóm tắt ngôn ngữ tự nhiên bằng học máy hay gặp lỗi ngữ pháp. Các lỗi ngữ pháp này gây trở ngại cho quá trình tách đối tượng khỏi câu, cụ thể là quá trình phân tích cú pháp sẽ không thể thực hiện được.
- Kết luận: không sử dụng tóm tắt văn bản để xử lý tách đối tượng trong tin tức.

2.3.1.2. Gán nhãn từ loại (Part-of-Speech tagging - POS)

- Mục đích: gắn loại từ vào từng từ (cụm từ). Đây là cơ sở cho các bài toán xử lý ngôn ngữ cấp cao hơn.
- Ví dụ: “con ruồi **đậu** mâm xôi **đậu**“

⇨ con ruồi/DT đậu/ĐT mâm-xôi/DT đậu/DT

- Vấn đề phát sinh:

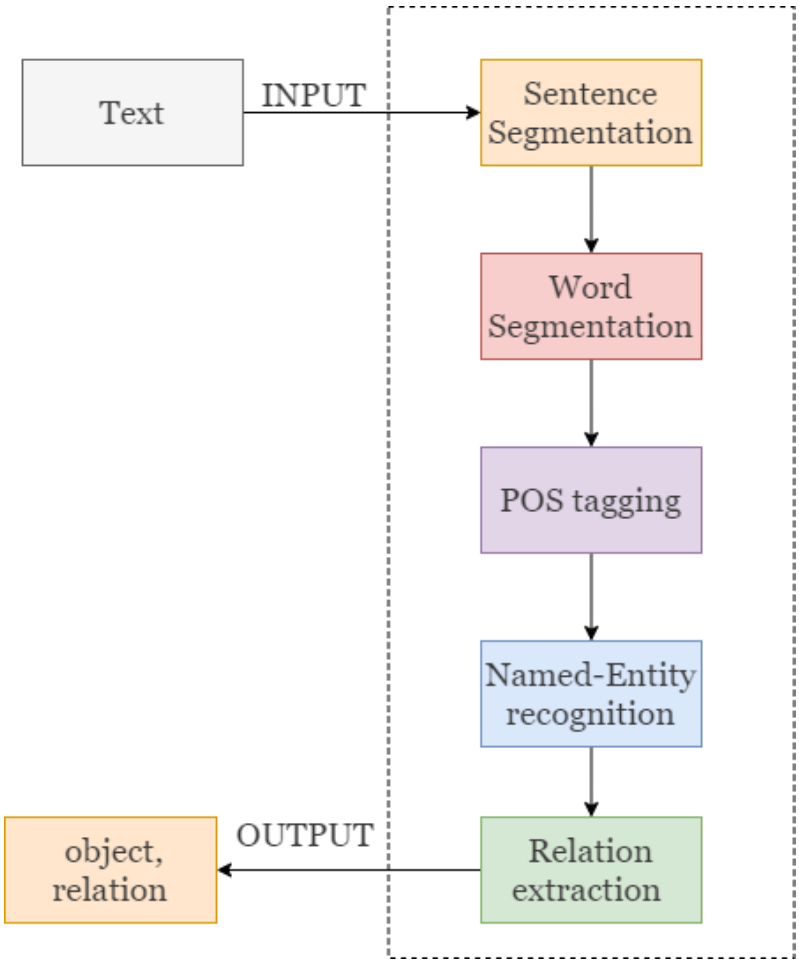
    - Nhập nhằng trong xử lý ngôn ngữ: như từ "đậu" trong câu trên, cái đầu tiên là động từ và cái sau là danh từ trong cụm danh từ. Việc nhập nhằng này gây khó khăn trong việc tìm ra
    - Cần có thư viện tiếng Việt đầy đủ cũng như tập mẫu đủ lớn để tìm ra xác suất tương ứng của mỗi loại từ của cùng một từ.
- Kết luận: sử dụng phương pháp gán nhãn từ loại với thư viện xử lý tiếng Việt có sẵn (underthesea).

2.3.1.3. Phân tích cú pháp

Tiếng Việt không có bộ thư viện nào hỗ trợ phân tích cú pháp. Kể cả có các thuật toán xử lý ngôn ngữ nhưng vẫn cần có một chuyên gia ngôn ngữ để đảm bảo tập luật chính tả đầu vào đầy đủ để đảm bảo cho thuật toán chạy ổn định và chính xác. DO vậy, trong bài toán đặt ra, chúng tôi chỉ xử lý ngôn ngữ tiếng Việt với các luật chính tả liên quan đến xử lý ngữ cảnh và chủ thể trong câu. Các vấn đề này là yêu cầu bắt buộc khi xử lý tin tức nhưng không có các tài liệu hay thư viện hỗ trợ.

2.3.2. Mô hình tổng quát

Quy trình xử lý trong một bài toán xử lý ngôn ngữ tự nhiên là

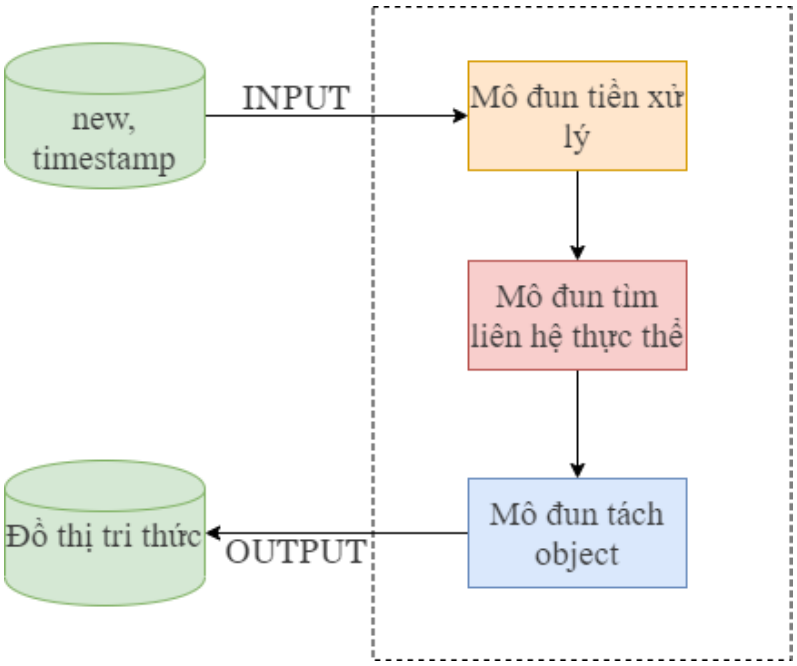


Hình 2-4 Mô hình xử lý NLP thông thường

- Sentence Segmentation: Đoạn text đầu vào được chia thành từng câu. Ta có thể hiểu là từng mệnh đề vì trong 1 số ngôn ngữ, không chỉ dấu chấm mà dấu chấm phẩy cũng dùng để ngăn cách các mệnh đề ứng với các thực thể.
- Word Segmentation: tách câu thành các từ, cụm từ mà không thể chia nhỏ hơn. Ví dụ: các từ láy như "lung_lay", "lắc_lư" không thể chia ra nhỏ hơn vì sẽ làm cụm từ trở nên mất nghĩa,
- POS tagging: đánh dấu loại từ cho từng (cụm) từ đã tách ở trên.
- Named-Entity recognition: tìm các thực thể trong mệnh đề. Ở đây thực thể là các đối tượng sự vật hiện tượng là chủ thể hoặc bị chủ thể trong câu hướng đến.

- Relation extraction: dựa theo các thực thể đã tìm được, ta tìm ra mối liên hệ giữa chúng nhằm tìm ra ý nghĩa/sự kiện mà văn bản nhắc đến.

Từ mô hình trên, áp dụng vào bài toán thực tế xử lý tin tức bệnh nhân Covid-19, nhóm em đề xuất ra mô hình xử lý tách đối tượng như sau:



Hình 2-5 Mô hình xử lý NLP trong bài toán

Từng phần trong sơ đồ sẽ được giải thích ở các phần sau.

2.3.3. Mô đun tiền xử lý

Mô đun này có chức năng xử lý các vấn đề của xâu đầu vào để đảm bảo các mô đun hoạt động ở sau không bị ảnh hưởng bởi các yếu tố bất thường không đáng có.

- Xử lý các lỗi nhập liệu: các lỗi như 2 dấu cách, thừa dấu mở ngoặc đóng ngoặc,...
- Xử lý các cụm từ viết tắt dễ gây phân tích sai như cụm “TP. HCM”. Nếu cho cụm này vào thì rất có thể mô đun đằng sau sẽ phân tích từ dấu “.” Thành hai câu vô nghĩa.

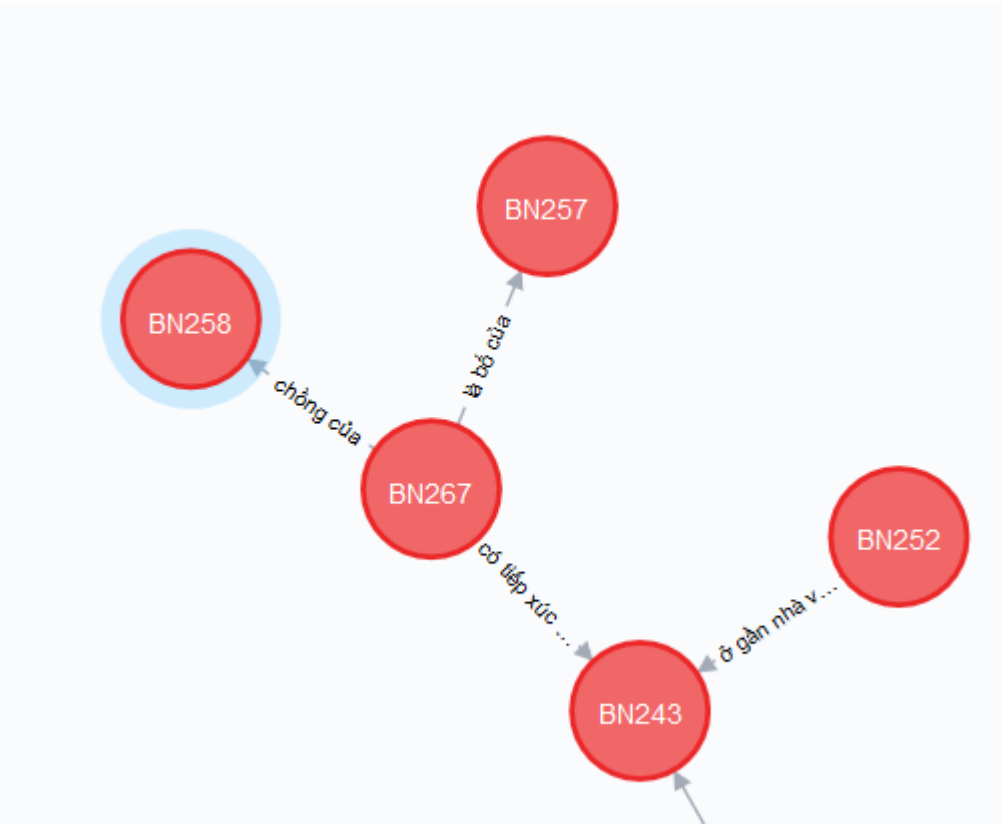
- Đổi các thực thể bệnh nhân về đúng chuẩn viết tắt "BNXX" với XX là số. Việc này giúp cho việc phân tích ở sau đơn giản hơn và tránh nhập nhằng. Ví dụ: "Bệnh nhân số 91", "BN số 91", "ca bệnh số 91",... sẽ chuyển về "BN91"

2.3.4. Mô đun tìm mối liên hệ giữa các thực thể

2.3.4.1. Mô đun tìm thực thể

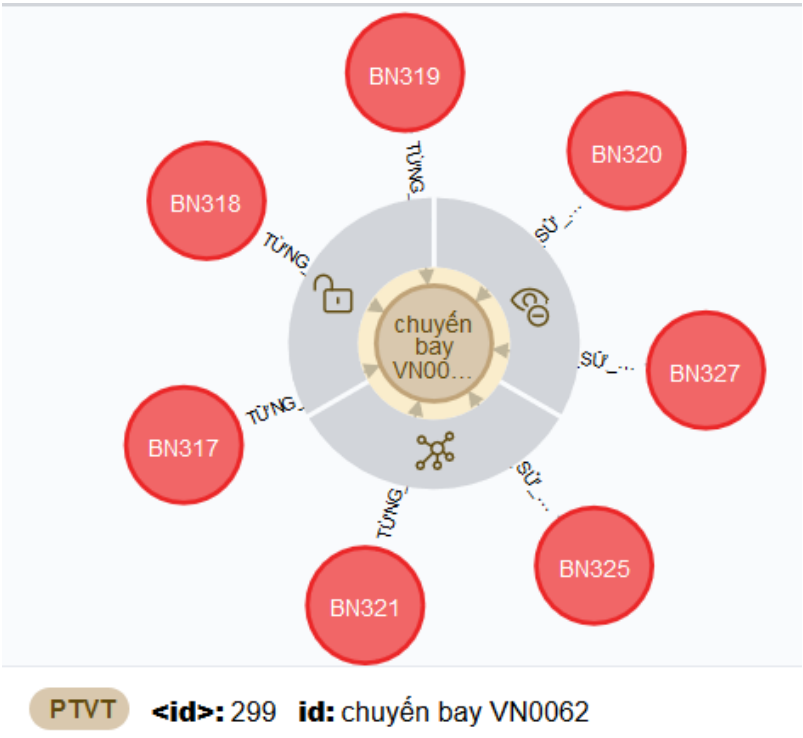
Trong bài toán đặt ra, nhóm em đề xuất ra 2 nhóm thực thể với điều kiện là các thực thể trong các nhóm này có mối liên hệ với nhau.

- Nhóm 1: các bệnh nhân đang và từng dương tính với Covid-19. Các thực thể bệnh nhân này có thể có các mối liên hệ trực với nhau hoặc cùng liên hệ với nhóm thực thể thứ 2.



Hình 2-6 Ví dụ về mối liên hệ giữa các nút bệnh nhân

- Nhóm 2: các phương tiện giao thông mà người dương tính với bệnh từng sử dụng. Đây là một môi trường kín làm lây nhiễm bệnh có thể dùng để khoanh vùng các ca nghi nhiễm.



Hình 2-7 Liên kết giữa các nút bệnh nhân từng cùng một chuyến bay

Các nhóm thực thể được xác định dựa trên các mô đun trích xuất object được tả ở dưới.

2.3.4.2. Cơ chế xác định thực thể trong ngữ cảnh ẩn thực thể

Trong ngữ pháp tiếng Việt, người ta thường tránh lặp từ khi gọi các chủ ngữ của câu ở các câu trần thuật kế tiếp.

BN267 là nam giới, 46 tuổi, xóm Hội, Hạ Lôi, Mê Linh, Hà Nội, là bố của BN 257, chồng của BN 258, có tiếp xúc gần với BN243 tại nhà ngày 20/3. Ngày 8/4, được cách ly tập trung tại Hà Nội. Ngày 13/4 bệnh nhân khởi phát với triệu chứng sốt nhẹ, mệt mỏi, đau rát họng, đau người, được lấy mẫu bệnh phẩm. Xét nghiệm ngày

14/4 cho kết quả dương tính với SARS-CoV-2. Hiện bệnh nhân được cách ly, điều trị tại Bệnh viện Bệnh Nhiệt đới Trung ương cơ sở 2.

Ví dụ như trong câu này, BN267 là chủ thể của của cả đoạn văn. Ngoài câu thứ nhất nếu trực tiếp "BN267", các câu sau người viết đã sử dụng các thủ thuật ngôn ngữ để không phải nhắc lại: Ở câu 2 là câu rút gọn, không có chủ thể và sử dụng chủ thể ở câu trước; Câu thứ 3 chỉ nêu là "bệnh nhân" và người đọc tự hiểu chủ thể ở câu trước; câu 4 và 5 tương tự như câu 2 rút gọn chủ ngữ. Ngoài ra ở câu thứ nhất, có 4 thực thể, thực thể "BN267" là thực thể chủ ngữ, các chủ thể "BN257", "BN257", "BN243" đều là các thực thể tương tác với chủ ngữ chứ không tương tác với nhau. Đây là bởi vì luật ngữ pháp tiếng Việt gộp các mệnh đề chung chủ ngữ và phân tách bởi dấu ",".

Vì vậy cơ chế xác định thực thể trong mệnh đề cần phân tích thực thể trong cả đoạn văn và tạm thời lưu lại để xử lý trong các câu kế tiếp cho đến khi gặp các chủ thể mới. Việc thay đổi thực thể trong chủ ngữ cần phân tích dưới góc độ ngữ pháp tiếng Việt.

2.3.4.3. Tìm mối liên hệ giữa các thực thể

Từ việc tách được thực thể chủ ngữ và thực thể bị hướng tới trong mệnh đề, ta có lấy liên hệ là xâu từ sau thực thể chủ ngữ đến thực thể bị hướng tới. Với trường hợp câu rút gọn thì có thể lấy từ đoạn mở đều mệnh đề cho đến thực thể bị hướng tới.

Có 3 trường hợp về thực thể mà chúng ta có thể xác định:

- Mệnh đề có 1 thực thể, thậm chí là câu rút gọn lấy chủ thể từ câu đứng trước. Trường hợp này trong mệnh đề chỉ chứa các object của thực thể chủ ngữ.

  VD: THÔNG BÁO VỀ CA BỆNH 271: Bệnh nhân nam, 37 tuổi, quốc tịch Anh, là chuyên gia của Tập đoàn Dầu khí.

- Mệnh đề có 2 thực thể A và B, ta chỉ xét trường hợp thực thể A tương tác với thực thể B. Trường hợp này lấy liên hệ như ở trên.

[A] -> [relation] -> [B]

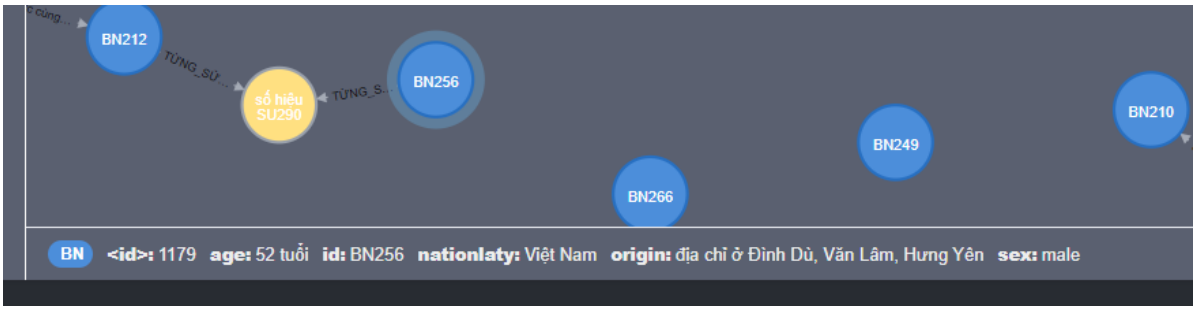
- Mệnh đề có > 2 thực thể, ta tạm thời bỏ qua trường hợp nhiều thực thể là chủ ngữ. Ở đây, thực thể A có thể tương tác với các thực thể B, C, ..

  [A] -> [relation] -> [B], -> [relation] -> [C],....

  VD: BN267 là nam giới, 46 tuổi, xóm Hội, Hạ Lôi, Mê Linh, Hà Nội, là bố của BN 257, chồng của BN 258, có tiếp xúc gần với BN243 tại nhà ngày 20/3.

2.3.5. Mô đun tách object

Thay vì phải phân tích cú pháp câu sau đó tìm ra object, chúng tôi sử dụng tập regex để tìm ra các object theo định hướng từ trước. Các regex này hoàn toàn có thể thay đổi và bổ sung để tăng độ chính xác cho tập dữ liệu trích xuất.



Hình 2-8 Thông tin của 1 nút bệnh nhân

2.3.5.1. Bộ trích xuất object từ regex

Các object trong phạm vi này là các object sau khi trích xuất có thể sử dụng luôn, ví dụ như các thông tin về mã hiệu chuyến bay, quốc tịch, quê quán, số ghế, tuổi,...

```
FLIGHT_RE= [r"(c|C)huyến bay\s{0,6}[A-Z]{1,4}\s?[0-9]{2,8}"]
NATIONLATY_RE = ["quốc tịch(.{0,1}[A-Z]\w{1,7}){1,3}",
        "quốc tịch(.{0,1}\w{1,7}){1,3}"]
ORIGIN = [r"(địa\s{1,2}chỉ|trú)\s{1,2}(tại|ở)\s{1,2}(\s|\w|,|TP.)*([A-
Z]\w{1,}",
r"(địa chỉ|trú|quê) (tại|ở)?(\s(phường|quận|thị xã|thị trấn|tỉnh|thành phố)?(\s
?\w{1,4}){1,3})"]
NUMBERSIT = ["số ghế [0-9]{1,8}[A-Z]{1,4}\s?"]
AGE = [r"[0-9]{1,3}\s{1,6}tuổi"]
```

Các thông tin này lấy sau là có thể trực tiếp đẩy vào cơ sở dữ liệu.

2.3.5.2. Bộ trích xuất object từ ngữ nghĩa

Các thông tin này có đặc thù là sau khi trích xuất bằng regex (hoặc kể cả phân tích cú pháp) đều cần thêm một bước xử lý logic nữa để đẩy vào trong cơ sở dữ liệu. Ví dụ như các regex như:

```
FEMALE = [r"(n|N)ữ"]
MALE = [r"(n|N)am", "nam giới"]

BN_RANGE = [r"CA BỆNH\s{1,6}[0-9]{1,3} - [0-9]{1,3}",
    r"Bệnh nhân\s[0-9]{1,3} - [0-9]{1,3}",
     r"Bệnh nhân số\s{1,6}[0-9]{1,3} - [0-9]{1,3}"
    ]
BNre = [r"CA BỆNH\s{1,6}[0-9]{1,3}",
    r"(b|B)ệnh nhân\s[0-9]{1,3}",
     r"(b|B)ệnh nhân số\s{1,6}[0-9]{1,3}",
    r"BN\s{0,2}[0-9]{1,3}"
    ]
DEATH = [r"(đã)?\s{1,3}(chết|khuất|ngoẻo|tử vong|mất)",
    r"(đã)\s{1,3}(khuất|mất)"
]
NEGATIVE_COVID = [r"(đã)?\s{1,3}(khỏi bệnh)"
]
```

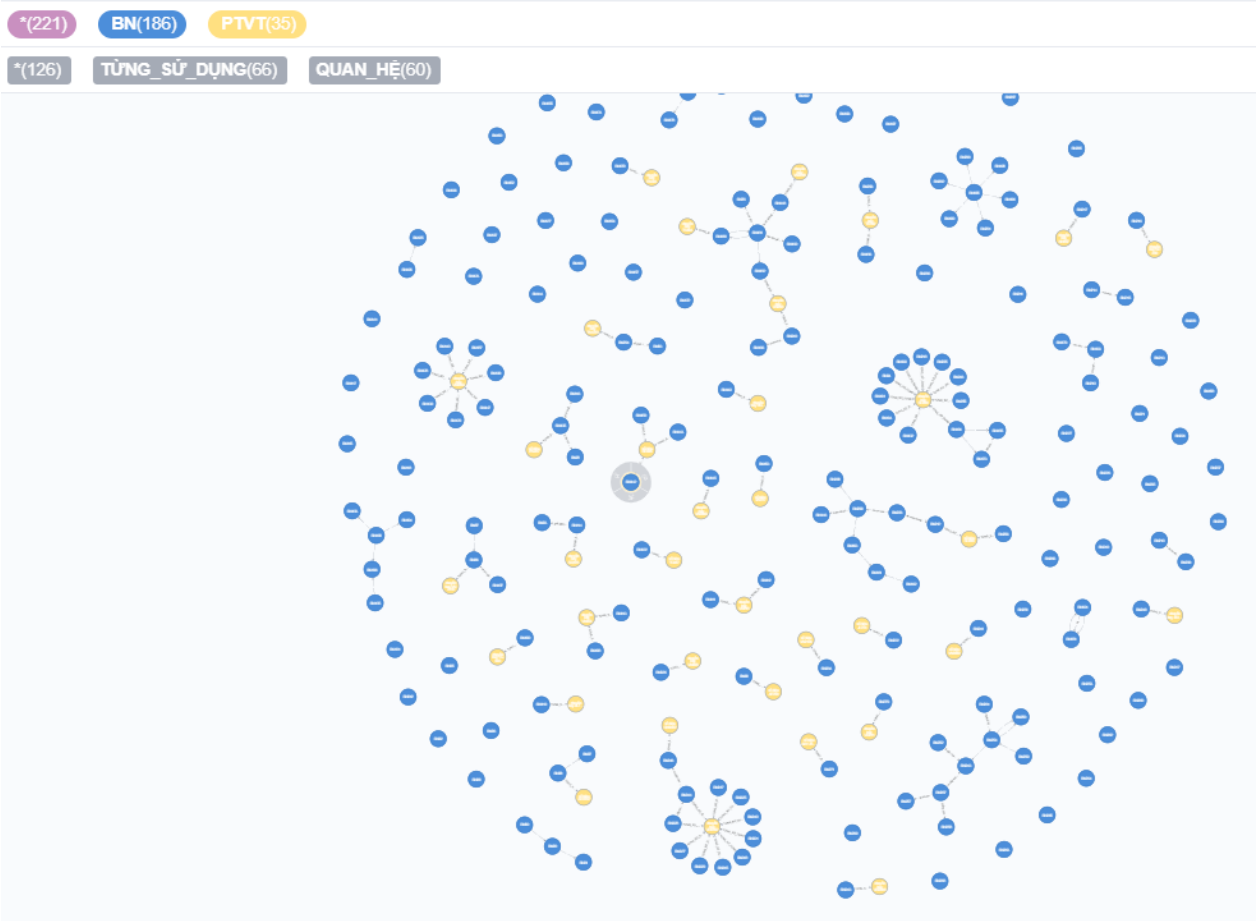
Trong đó ví dụ như các regex về giới tính, sau khi nhận ra các cụm từ như “đàn ông”, “nam giới”,... thì thay vì nhập trực tiếp vào cơ sở dữ liệu xâu này, ta xử lý bằng cách ánh xạ chúng đến kết quả là “sex: male”. Tương tự như vậy với trường trạng thái bệnh nhân, nếu phát hiện các xâu miêu tả bệnh nhân đã khỏi bệnh hay đã chết sẽ trả về trạng thái “death” hay “negative” tương ứng

2.3.6. Tương tác cơ sở dữ liệu neo4j

Neo4J là hệ quản trị cơ sở dữ liệu đồ thị được giới thiệu đầu tiên vào năm 2007 và công bố phiên bản 1.0 vào năm 2010. Neo4J là một trong những hệ quản trị cơ sở dữ liệu đồ thị được sử dụng nhiều nhất.

Nếu như cơ sở dữ liệu quan hệ như SQLServer, MySQL, Oracle để mô tả một đối tượng như MonHoc (subject) và các đặc điểm của đối tượng (properties) thì chúng mô tả bằng một bảng dữ liệu gồm nhiều cột với tên bảng là tên của đối tượng, các cột trong bảng mô tả đặc điểm của đối tượng. Mối quan hệ giữa các đối tượng được xây dựng bằng cách ghi nhận thông tin của thực thể cha vào thực thể con, ví dụ như như muốn xác định bệnh nhân nào từng sử dụng máy bay nào chúng ta cần lưu thông tin về 2 nút bệnh nhân và máy bay tương ứng, giữa 2 nút có 1 vecto nối với giá trị là số ghế mà bệnh nhân đó ngồi.

Trong bài toán đặt ra, chúng tối đã có thể thu được đồ thị sau từ trang thông tin:

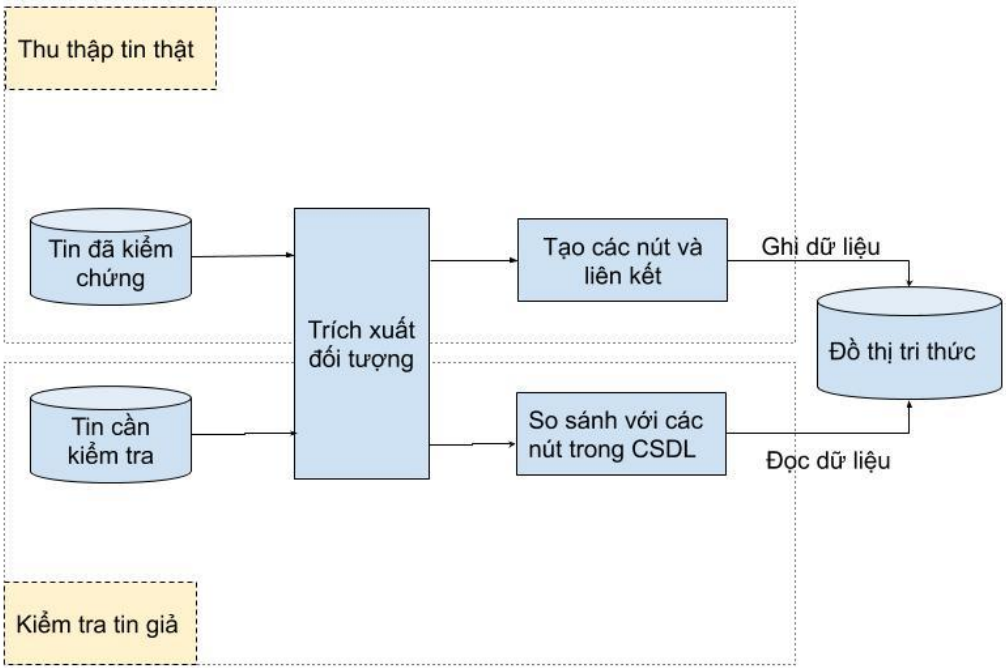


Hình 2-9 Đồ thị tri thức COVID-19

Đồ thị này thu được khoảng 200 bệnh nhân với mô đun crawler cũ. Với mô đun crawler mới, chúng tôi thu được thông tin của hơn 300 bệnh nhân.

2.3.7. Mô đun phát hiện tin giả

Giống với mô đun trích xuất từ tin thật, mô đun kiểm tra cũng sử dụng các mô đun trích xuất các đối tượng rồi so sánh với các đối tượng tương ứng của thực thể trong cơ sở dữ liệu. Các so khớp đối tượng trả về True – False cùng với giá trị gốc trong CSDL và giá trị trong tin cần kiểm tra. Tập giá trị này được lưu vào 1 mảng nhằm giúp người xem có thể kiểm tra xem đối tượng nào trong câu là sai, đối tượng nào đúng.


Thu thập tin thật
Tin đã kiểm chứng
Trích xuất đối tượng
Tạo các nút và liên kết
Ghi dữ liệu
Đồ thị tri thức
Tin cần kiểm tra
So sánh với các nút trong CSDL
Đọc dữ liệu
Kiểm tra tin giả

CHƯƠNG 3 Quy trình cài đặt hệ thống và các kịch bản demo

3.1. Quy trình cài đặt

- Cài pyspark

$ wget https://www.apache.org/dyn/closer.lua/spark/spark-3.0.0-preview2/spark-3.0.0-preview2-bin-hadoop2.7.tgz

$ tar –xf spark-3.0.0-preview2-bin-hadoop2.7.tgz

- Tải mã nguồn

$ git clone https://github.com/ltthacker/bdi_final

- Chỉnh sửa biến $SPARK_HOME trong file bdi_final/activate cho trỏ vào thư mục vừa giải nén. Ví dụ như sau:

 # Add path for pyspark

export SPARK_HOME=$HOME/Documents/course/big_data_integration/software/spark-3.0.0-preview2-bin-hadoop2.7

export PATH=$PATH:$SPARK_HOME/bin

export PYTHONPATH=$SPARK_HOME/python:$PYTHONPATH

export PYSPARK_PYTHON=python3

export SPARK_LOCAL_IP=127.0.1.1

- Cài đặt các thư viện python cần thiết

$ pip3 install -r requirements.txt --user

với requirements.txt đặt tại đường dẫn bdi_final/requirements.txt

- Tải và cài đặt geckodriver để điều khiển firefox

$ wget https://github.com/mozilla/geckodriver/releases/download/v0.26.0/geckodriver-v0.26.0-linux64.tar.gz

$ tar -xf geckodriver-v0.26.0-linux64.tar.gz

$ sudo cp geckodriver /usr/local/bin

- Nhập các biến môi trường cho Pyspark

$ cd bdi_final

$ source activate

- Khởi tạo mysql (thay đổi cấu hình MySQL trong bdi_final/config/config.yaml), sau đó

$ python3 run_saver.py

- Khởi tạo neo4j
- Chạy crawler để lấy nguồn tin chính thức xây dựng đồ thị tri thức

$ cd analyzer

$ python3 learn_analyzer.py

- Chạy crawler

$ python3 run_crawler.py

- Khởi tạo spark standalone cluster

$ cd $SPARK_HOME/sbin

$ ./start-all.sh

- Chạy bộ phân tích bằng SparkStreaming

$ spark-submit run_crawler.py --master=spark://localhost:7077

- Xem thông tin về Spark Standalone cluster trên giao diện tại localhost:8080
- Xem thông tin trên log và MySQL

CHƯƠNG 4 Tổng kết và hướng phát triển

4.1. Tổng kết

Nhóm đã hoàn thành các đề mục sau:

- Xây dựng cơ sở dữ liệu đồ thị tri thức Neo4j cùng các API tương tác.
- Thu thập và tách các trường thông tin về bệnh nhân trong tin tức mà vẫn đảm bảo đúng chủ thể trong ngữ cảnh. Các đối tượng thu được chủ yếu nhờ phương pháp logic và lọc regex.
- Xây dựng mô đun kiểm tra tin giả dựa trên đồ thị tri thức cũng như mô đun trích xuất đối tượng đã xây dựng được.

4.2. Định hướng phát triển

- Sử dụng học máy, cụ thể là model vec2text để xử lý tách đối tượng trong câu.
- Sử dụng graphframe để tăng hiệu suất xử lý data của spark.

TÀI LIỆU THAM KHẢO

[1] https://neo4j.com/developer/get-started/

[2] https://vi.wikipedia.org/wiki/Neo4J