TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN CÔNG NGHÊ THÔNG TIN VÀ TRUYỀN THÔNG

Báo cáo bài tập lớn môn
Tìm kiếm và trình diễn thông tin

Xây dựng hệ thống tìm kiếm bài báo VNexpress

Giáo viên hướng dẫn: Thầy Nguyễn Bá Ngọc

Sinh viên thực hiện:

1. Phạm Đức Tuệ MSSV: 20164432
2. Phạm Duy Tiên MSSV: 20164038
3. Hồ Xuân Cường MSSV: 20160537

Hà nội, ngày 11/06/2020

Tóm tắt nội dung

Internet ngày càng phát triển ngày càng phát triển xu hướng tìm kiếm thông tin là nhu cầu tất yếu. Ta càng thấy được tầm quan trọng và tính thiết yếu khi nhìn vào những ông lớn như Google, Bing của Microsoft, … Trong khuôn khổ bài tập lớn môn học IT4853, chúng em đã cùng nhau tìm hiểu và xây dựng hệ thông tìm kiếm đơn giản thông tin báo VNxperess hàng ngày. Báo cáo gồm các phần:

1. Đặt vấn đề
2. Phương pháp tiếp cận: Dựa trên các công cụ crawler của python và search platform lucene solr. Trình bày về phương pháp thực hiện, một số vấn đề gặp phải và phương pháp giải quyết
3. Kết quả đạt được và hướng phát triển

Mục lục

Nội dung

Phần 1: Đặt vấn đề

Từ xa xưa, loài người cổ đại đã phải trang bị rất nhiều kĩ năng để phục vụ cho việc sinh tồn: Săn bắn, hái lượm, leo trèo,... Mà trong đó, **tìm kiếm** là một trong những kĩ năng sống còn của con người. Theo dòng thời gian, với sự xuất hiện của chữ viết và sách, việc lưu trữ và tìm kiếm lại trở thành một nhu cầu thiết yếu.

Vào những năm 90, một nghiên cứu chỉ ra rằng phần lớn mọi người sẽ thích tra cứu thông tin từ người khác hơn là sử dụng các hệ thống tìm kiếm CNTT. Tất nhiên, trong thời gian đó, để đặt vé máy bay, người ta vẫn phải tìm gặp các công ty dịch vụ. Mặc dù vậy, khi bước sang thế kỉ 21, với những cải tiến đột phá từ các hệ thống tìm kiếm để cải thiện kết quả tìm kiếm và trải nghiệm người dùng, Web Search đã trở thành một tiêu chuẩn và là một nguồn đáng tin cậy cho việc tìm kiếm thông tin.

Thuật ngữ Information Retrieval có thể mang nghĩa rất rộng. Khi di mua hàng, bạn lấy thẻ tín dụng từ trong ví ra để có thể nhập mã thẻ thanh toán, đó cũng là một dạng của Information Retrieval. Tuy nhiên, ở khía cạnh học thuật, Information Retrieval được định nghĩa là:

Information Retrieval là hoạt động tìm kiếm tài liệu có bản chất phi cấu trúc (**unstructured**) như văn bản, hình ảnh, video,.. sao cho phù hợp (**relevant**) với một nhu cầu thông tin (**information need**) nào đó, từ một tập hợp dữ liệu lớn (**large collections**).

Trong một bài toán IR **điển hình**, đầu vào là:

- Một bộ ngữ liệu (**corpus**) các tài liệu văn bản
- Một câu truy vấn (**query**) của người dùng dưới dạng văn bản

Đầu ra:

- Một tập xếp hạng (**ranked list**) các văn bản mà được cho là phù hợp (**relevant**) với câu truy vấn (**query**)

Phần 2: Phương pháp tiếp cận

1. Đề xuất mô hình tìm kiếm thông tin

2.1.1 Search Engine là gì?

Search Engine (Web Search Engine) - tiếng Việt gọi là Công cụ Tìm kiếm - là một hệ thống dùng để tìm kiếm thông tin trên mạng.

Hiểu đơn giản thì Search Engine là trang web mà tại đó, người dùng gõ từ hoặc cụm từ muốn tìm hiểu vào khung tìm kiếm để được thấy các kết quả là những trang web, hình ảnh, video, địa chỉ, bản đồ, tài liệu,.v.v... liên quan đến điều mà họ cần tìm.

Những từ hay cụm từ mà người dùng gõ được gọi là từ khóa (keyword). Các kết quả được hiển thị liên quan đến từ khóa đó được sắp xếp theo một thứ tự cụ thể, được quyết định bằng thuật toán chuyên biệt của loại Search Engine mà người dùng đang sử dụng

2.1.2 Cơ chế hoạt động.

Dù khác nhau về thuật toán sắp xếp nhưng hầu hết các Search Engine đều có một phương thức hoạt động giống nhau. Cơ chế làm việc của một Search Engine gồm có ba bước: **crawling** (thu thập dữ liệu), **indexing** (sắp xếp dữ liệu vào kho) và **retrieval** (truy xuất dữ liệu).

- **Crawling**

Mọi Search Engine đều phải tiến hành giai đoạn cơ bản này - thu thập dữ liệu từ mọi trang web trên internet. Đầu tiên, các công cụ tìm kiếm sẽ **truy cập vào một trang web bất kỳ để quét và lấy dữ liệu của trang đó**. Sau đó, nó sẽ men theo các link (đường dẫn) trong trang để tiếp cận các trang liên quan khác. Nhờ vậy, toàn bộ các trang web trên internet sẽ được ghi nhận vào hệ thống của Search Engine.

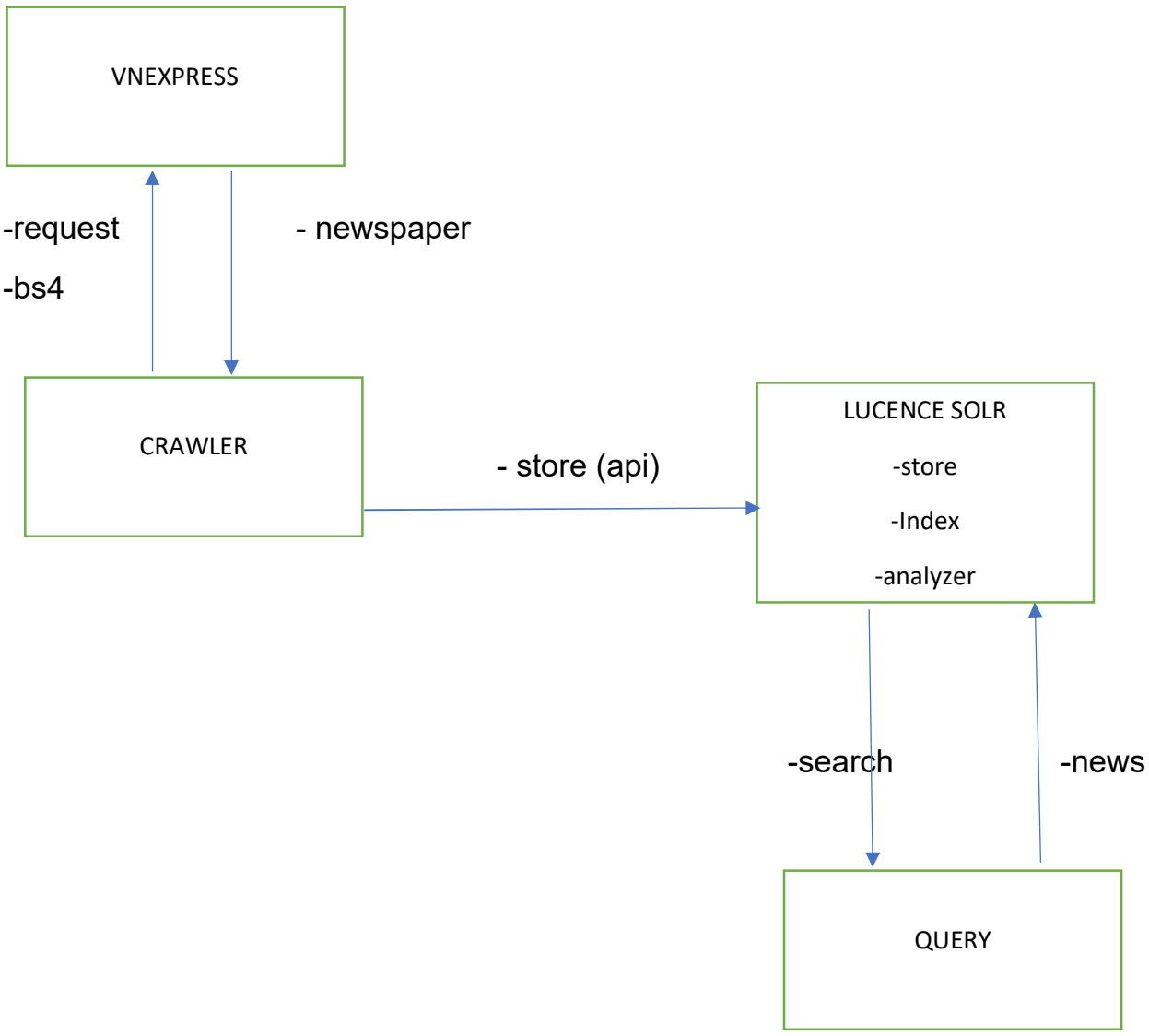
- **Indexing**

Quá trình indexing diễn ra ngay lập tức và song song với bước crawling trên. Khi indexing, các Search Engine sẽ sắp xếp lại dữ liệu đã có được vào trong kho phần cứng lưu trữ của mình**.** Với Google Search, đó là một siêu bộ nhớ gồm hàng chục ngàn ổ cứng với dung lượng tổng lên đến hàng petabyte (1 petabye = 10 tỷ gigabyte). Mọi thông tin được lưu trữ ở đây để sẵn sàng được trích xuất ngay khi có bất cứ ai nào gõ một thứ gì đó vào khung search.

- **Retrieval**

Khi nhận yêu cầu tra cứu của người dùng, các Search Engine sẽ thực hiện truy xuất thông tin đã lưu trong cơ sở dữ liệu, thực hiện sắp xếp các kết quả tìm được và hiển thị danh sách câu trả lời cho chúng ta**.** Các Search Engine dựa trên 2 tiêu chí để đánh giá thứ tự của các kết quả tìm kiếm: sự liên quan và độ phổ biến. Các kết quả tra cứu liên quan đến yêu cầu của bạn được ưu tiên nhất, sau đó mới xét đến độ phổ biến của từng kết quả.

2.1.3 Mô hình tìm kiếm thông tin đề xuất



2. Xây dựng crawler

Công cụ crawler em lựa chọn ở đây là BeautifulSoap (bs4) & request thay vì Scapy, chúng ta hay cụng đi vào so sánh giữa hai công cụ trên với bài toán hiện tại.

2.2.1 So sánh BeautifulSoap và Scapy

- Scapy là một Web-spider hay là một Framework cho Web scraper, khi input là một start url cho việc crawler và số các URLs để scrapy crawl, scapy tự động đi theo các liên kết trong trang web.

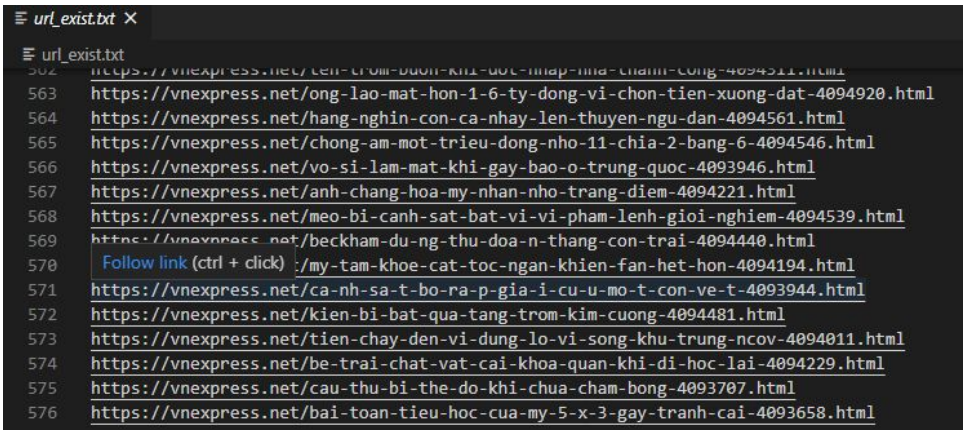
- Trong khi beautifulSoup là một thư viện (parse libray) . Lấy dữ liệu HTML từ trang web muốn crawler sau đó parse. Việc đi theo các liên kết như thế nào lập trình viên quy định. Về cơ bản BeautifulSoup có thể thực hiện crawler như Scapy (tuy hạn chế hơn).
- Với bài toán chỉ là crawler nội dung cũng như tiêu đề của VNexperss , việc lựa chọn BeautifulSoup thay vì Scapy vì một số lý do như:
    - Là thư viện nhẹ, dễ tiếp cận, sử dụng nhanh
    - Cộng đồng hỗ trợ lớn, document rõ rang
    - Phù hợp vơi bài toán

2.2.2 Phương pháp thực hiện Crawler.

- Mô hình chung:
    - Sử dụng request để lấy HTML của trang Web muốn crawler
    - Dùng Beautifulsoup để parse và lấy dữ liệu text về

2.2.3 Vấn đề gặp phải

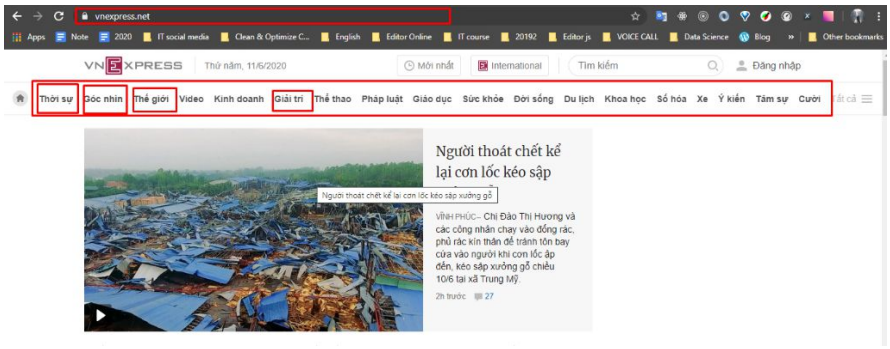
- Việc tạo cronjob hằng ngày đảm bảo việc luôn crawler được những bài viết mới
- Vấn đề trùng lặp: Việc crawler hằng ngày có thể dẫn tới việc các bài đăng có thế giống nhau: Vì định danh mỗi bài viết luôn được Vnexperss để trên 1 url => Lưu lại các url đã crawler



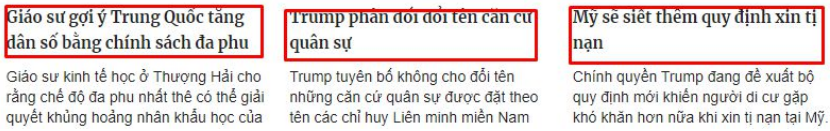
2.2.4 Sơ đồ:

Đi vào trang web gốc (vnexpress.vn)

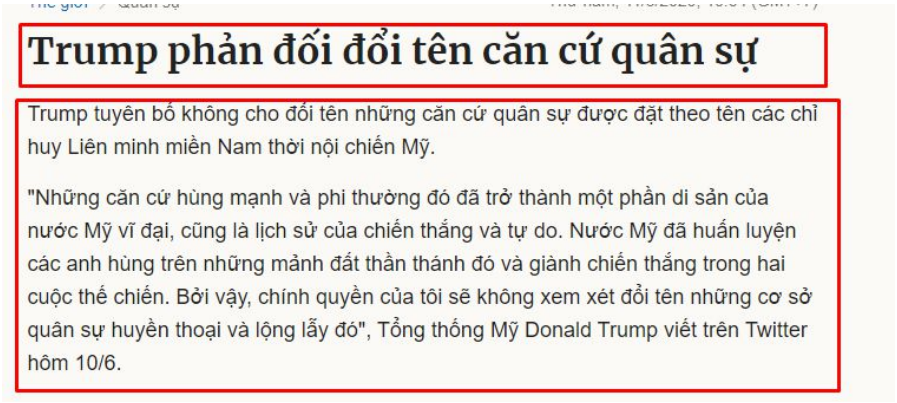
Đặt lệnh hàng ngày thực hiện crawler



Đi vào các trang



Lấy thông tin bài đăng



Trump phản đối đổi tên căn cứ quân sự

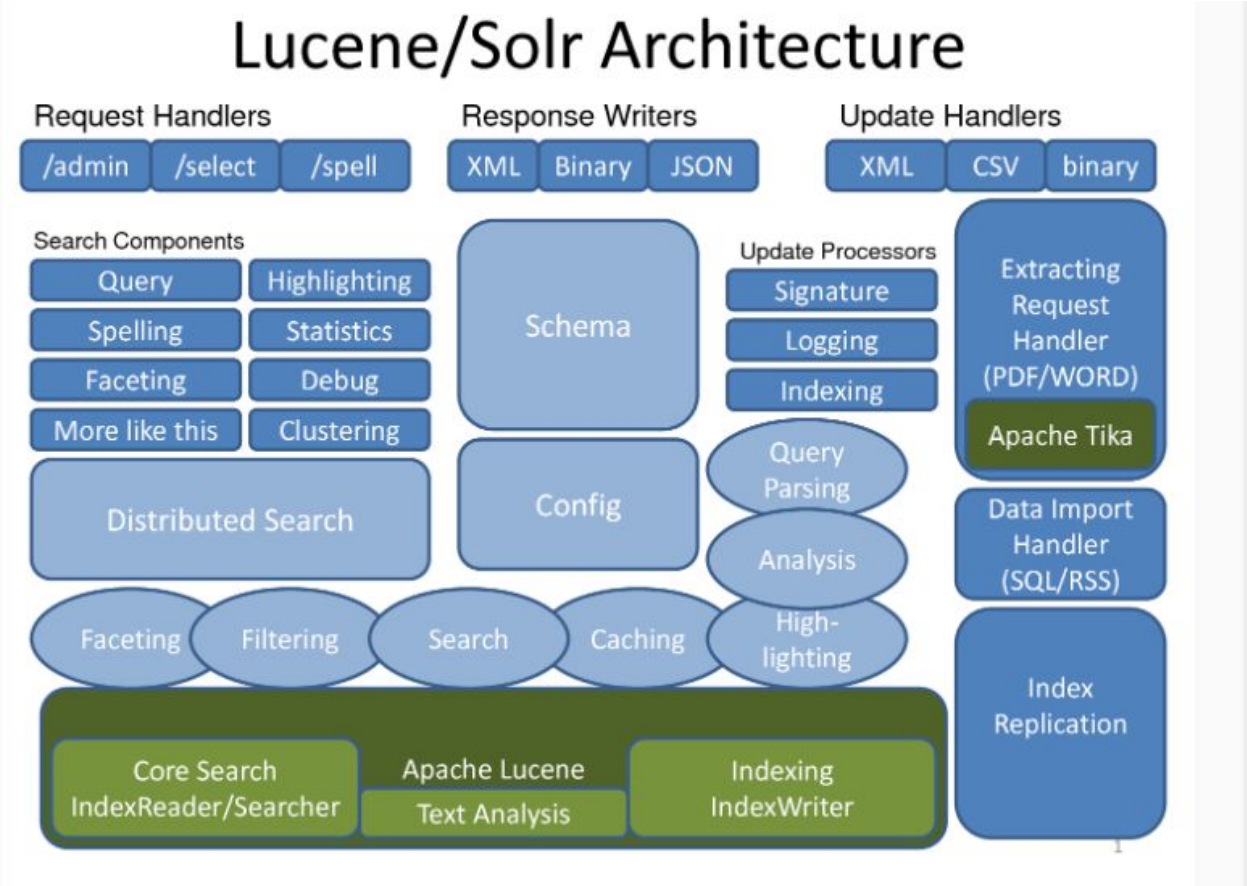
Trump tuyên bố không cho đổi tên những căn cứ quân sự được đặt theo tên các chỉ huy Liên minh miền Nam thời nội chiến Mỹ.

"Những căn cứ hùng mạnh và phi thường đó đã trở thành một phần di sản của nước Mỹ vĩ đại, cũng là lịch sử của chiến thắng và tự do. Nước Mỹ đã huấn luyện các anh hùng trên những mảnh đất thần thánh đó và giành chiến thắng trong hai cuộc thế chiến. Bởi vậy, chính quyền của tôi sẽ không xem xét đổi tên những cơ sở quân sự huyền thoại và lộng lẫy đó", Tổng thống Mỹ Donald Trump viết trên Twitter hôm 10/6.

3. Apache Lucene Solr

2.3.1 Apache lucene Solr là gì ?

Apache Solr là một open source full-text search platform dựa trên Apache Lucence. Lucene là một thư viện được viết bằng Java dùng để phân tích, đánh chỉ mục (indexing) và tìm kiếm thông tin được phát triển đầu tiên bởi Doug Cutting vào năm 2000. Cutting đồng thời cũng là tác giả của Hadoop lúc ông đang làm việc cho Yahoo vào năm 2005.



- **Request handler**: Xử lý yêu cầu (tìm kiếm, cập nhật chỉ mục), sử dụng bộ api của apache solr
- **Search Component**: Thực hiện tác vụ như kiểm tra lỗi chính tả, tìm kiếm, highlighting thực hiện ngay sau bước handler
- **Query parser**: TIến hành phân tích cú pháp câu truy vấn => chuyển sang dạng àm solr hiểu
- **Response Writer**: Dữ liệu tìm kiếm trả về dưới dạng JSON, XML hoặc Binary dựa vào yêu cầu bài toán
- **Analyer/Tokenizer**

2.3.2 Phân tích cú pháp trong Lucence Solr (Analyze/Tokenizer)

Một số điểm nổi bật của lucence solr là:

- Hỗ trợ cấu hình mềm dẻo
- Kiến trúc cho phép mở rộng
- Hỗ trợ nhiều ngôn ngữ

Analyzer

- Bộ phân tích cú pháp thực hiện trong hai ngữ cảnh. Trong quá trình index, khi mà trường dữ liệu được tạo, kết quả các token sau quá trình analyzer được lưu vào solr. Và Quá trình truy vấn kết quả phân tích được bộ analyzer phân tích và sau đó tìm kiếm

```
<fieldType name="nametext" class="solr.TextField">
  <analyzer type="index">
    <tokenizer class="solr.StandardTokenizerFactory"/>
    <filter class="solr.LowerCaseFilterFactory"/>
    <filter class="solr.KeepWordFilterFactory" words="keepwords.txt"/>
    <filter class="solr.SynonymFilterFactory" synonyms="syns.txt"/>
  </analyzer>
  <analyzer type="query">
    <tokenizer class="solr.StandardTokenizerFactory"/>
    <filter class="solr.LowerCaseFilterFactory"/>
  </analyzer>
</fieldType>
```

- Đối với những truy vấn dạng (Prèĩ, Wildcard, Regex …) Input đầu vào không giống ngôn ngữ tự nhiên, Synonyms và stop word không được áp dụng. Analyzer phải áp dụng dạng mutiterrn analyzer

```
<analyzer type="multiterm">
  <tokenizer class="solr.KeywordTokenizerFactory" />
</analyzer>
```

- Tokenizers

Tokenizers chuyển dữ liễu text sang dạng token (từ có nghĩa). Trong hầu hết trường hợp Tokenizer bỏ đi kí tự khoảng trắng (không hoàn toàn đúng vd: Tiếng việt)

- Filters

Đầu vào là một Token đầu ra đã được chuyển sang dạng tương ứng với config. Ví dụ: touppercase, lowercase, loại bỏ từ dừng, chuyển từ ngữ về dạng nguyên thủy infinity,…

2.3.3 Indexing

Sau quá trình Analyzer, những Token còn lại sẽ được đánh index để phục vụ quá trình fulltext search trong Solr. Vì Solr là một fulltext search library nên việc cấu solr xóa hoặc sửa đi những thông tin nếu nó cần thiết nhằm phục vụ truy vấn cũng như giảm kích thước chỉ mục. Chỉ mục solr dùng là TF-IDF

- TF (term frequency): Số lần xuất hiện t trong văn bản d
- IDF (inverse document frequency):

$$idf(t) = \log(N/df_t)$$

df: tần suất văn bản của từ t

2.3.4 Scoring

Quá trình score của solr dựa trên nhân lucene. Chúng ta cùng đi vào tìm hiểu cách lucence tính toán điểm:

Quá trình tính toán dựa trên sự kết hợp giữa VSM và Boolean Model. Boolean Model thu hẹp những văn bản cần tìm kiếm, solr có cải tiến mô hình Boolean và fuzzy searching, nhưng chủ yếu vẫn là VSM

Về VSM (Vector Space Model): Trong VSM tài liệu và câu truy vấn là vector m chiều với mỗi vị trí trong ma trận là trọng số tf-idf. Độ tương đồng cosine giữa 2 vector

$$\text{cosine-similarity}(q,d) = \frac{V(q) \cdot V(d)}{|V(q)|\,|V(d)|}$$

VSM Score

Tf được tính:

$$\text{tf}(t \text{ in } d) = \text{frequency}^{1/2}$$

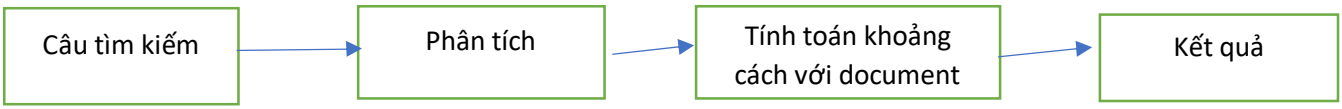
Idf được tính bởi

$$\text{idf}(t) = 1 + \log\left(\frac{\text{numDocs}}{\text{docFreq}+1}\right)$$

⇨ Tuy nhiên lucene có cải tiến công thức để giải quyết nhiều hơn vấn đề ví dụ trong 1 văn bản nhiều paragraph giống nhau

2.3.5 Searching

- Quá trình tìm kiếm trên solr được cung cấp bằng bộ api có sẵn



Tuy nhiên do bộ phân tích của solr không có tiếng việt:

⇨ Giải phát là sử lý trước bằng công VnTokenier để tách từ sau đó loại bỏ các lừ dùng sẵn có rồi mới thực hiện truy vấn cũng như index vào solr

Stopwords:

stopwords.txt

```
1   a_lô
2   a_ha
3   ai
4   ai_ai
5   ai_nấy
6   ai_đó
7   alô
8   amen
9   anh
10  anh_ấy
11  ba
12  ba_ba
13  ba_bản
14  ba_cùng
```

Câu truy vấn và dữ liệu lưu vào solr:

```
clean = clean_data("Cô gái 20 tuổi, phổi đông đặc, thoát chết nhờ được
"thay cả hai lá phổi, trong ca ghép đầu tiên thuộc
" loại này cho bệnh nhân Covid-19")
```

- Kết quả:

```
(python38) C:\IR-IT4853>python ./crawler/vnexpress/vnexpress.py
Cô gái 20 phổi đông_đặc thoát chết thay hai lá phổi ca ghép bệnh_nhân Covid - 19
```

Phần 3: Kết quả và hướng tiếp cận

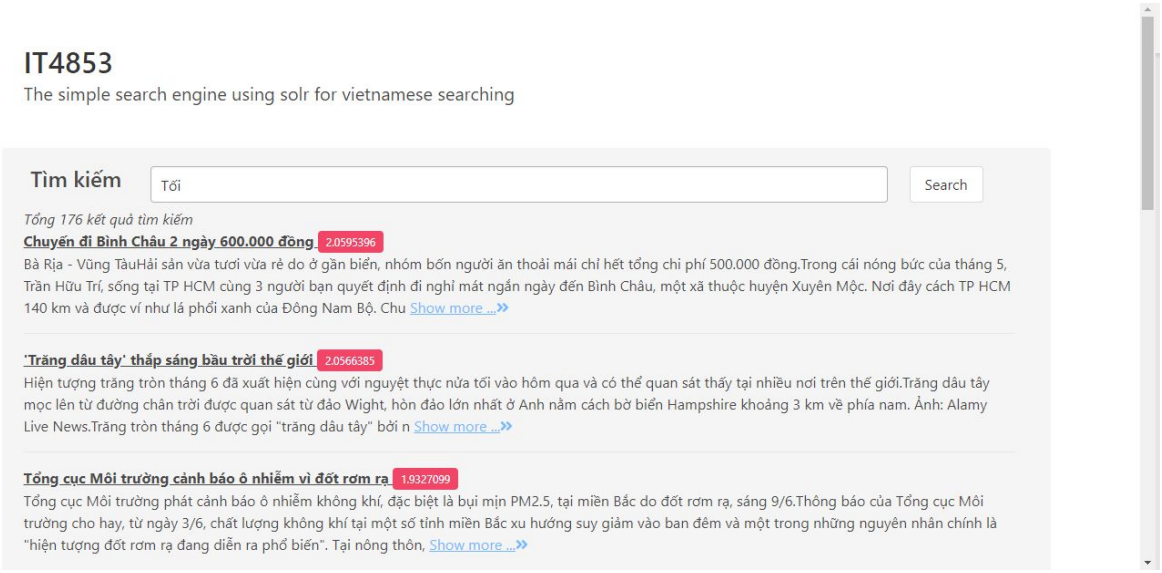
Chương trình chúng em gồm 2 phần: Web crawler và Web search

3.1 **Web crawler**: Tìm kiếm bài đăng VNEXPRESS sau đó lưu vào solr

Dữ liệu crawler gồm 3 phần title, content, url


```
{
  "title":["Chịu tang cha ở chốt chống dịch biên giới"],
  "url":["https://vnexpress.net/chiu-tang-cha-o-chot-chong-dich-bien-gioi-4085609.html"],
  "id":"759931e9-5b61-4509-b631-cd06902a82c7",
  "content":["Gần ba giờ sáng, thiếu uý Bùi Quang Huy, đồn Biên phòng cửa khẩu Lóng Sập (Sơn La) nhận điện thoại của mẹ báo ti
  "_version_":1664368085262401536},
{
```


3.2 **Web search**: Hiện thị giao diện kết quả



3.3 Hướng tiếp cận

- Qua bài tập lớn chúng em đã hiểu được sơ bộ một Information Retrieval là gì và đã xây dựng thành công một Search Engine cho riêng mình.
- Vì thời gian có giới hạn nên cũng chưa thể làm tốt một số vấn đề như: Sử dụng Scrapy thay cho BeautifulSoap, Tích hợp hỗ trợ tìm kiếm từ đồng nghĩa tăng hiệu quả cho truy vấn

Tài liệu tham khảo

[1] Giáo trình Tìm kiếm và trình diễn thông tin

https://github.com/bangoc/IT4853, của Thầy Nguyễn Bá Ngọc

[2] Documents của luncene Solr

https://lucene.apache.org/solr/guide/8_5/

[3] Docments của luncene – Scoring

https://lucene.apache.org/core/2_9_4/scoring.html#Understanding%20the%20Scoring%20Formula

[4] Một số tài liệu internet khác