TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

──────── \* ───────



BÁO CÁO MÔN: TÌM KIẾM VÀ TRÌNH DIỄN THÔNG TIN

ĐỀ TÀI: Sử Dụng Solr Xây Dựng Máy Tìm Kiếm

GVHD: TS.Nguyễn Bá Ngọc

Nhóm sinh viên thực hiện:

|  |
| --- |
| Nguyễn Hữu Phú 20143449 |
| Trần Thị Hải Hà 20141322 |
| Trần Thị Hồng 20141869 |
| Phạm Đức Nhất 20143302 |
| Nguyễn Khánh Thiện 20142468 |

Hà Nội ngày 21 tháng 5 năm 2018

Mục lục

I. Mô tả bài toán…………………………………………………………… 3

1. Giới thiệu chung………………………………………………………3

2. Giới thiệu về Solr……………………………………………………..3

II. Triển khai………………………………………………………………. 6

1. Thu thập dữ liệu……………………………………………………....6

2. Cài đặt apache solr…………………………………………………....6

3 Xây dựng server web tìm kiếm cho người dùng………………………7

III. Demo…………………………………………………………………...14

IV. Đánh giá………………………………………………………………..17

1. Đánh giá độ chính xác………………………………………………..17

2. So sánh 2 collection…………………………………………………..22

V. Tổng kết…………………………………………………………………23

1. Ưu điểm……………………………………………………………….23

2. Nhược điểm…………………………………………………………...23

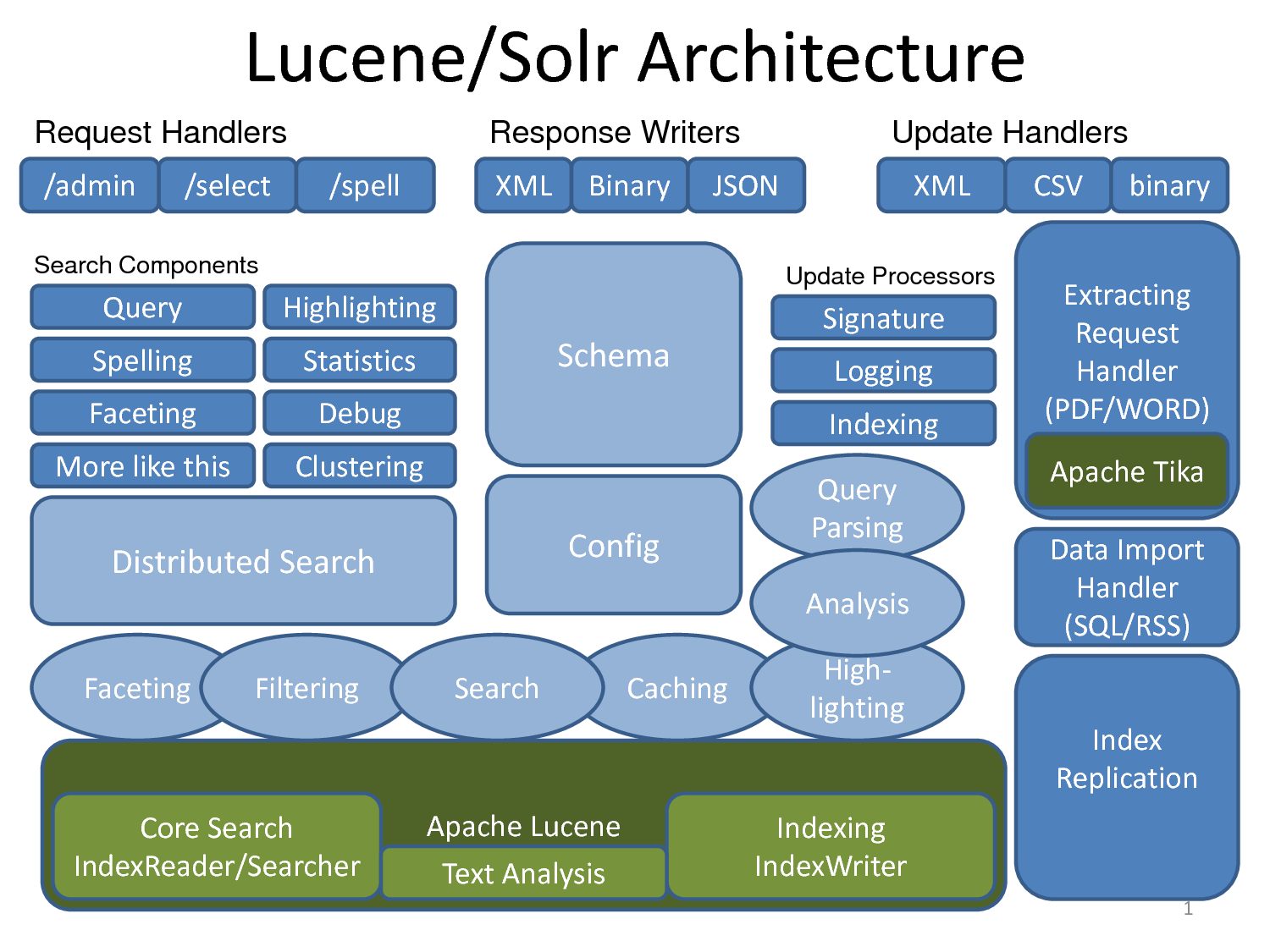
VI. Danh mục tài liệu tham khảo…………………………………………...24

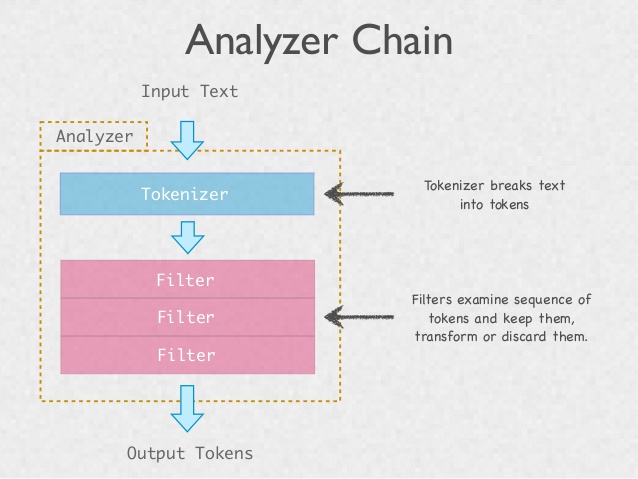
# **Mô tả bài toán**

1. Giới thiệu chung

* Hiện nay lượng dữ liệu trên internet rất lớn, tổng số website ở mức 760 triệu (2013), tổng dữ liệu trên internet khoảng 670 tỷ GB. Trong khi đó lượng dữ liệu trùng lặp, dữ liệu rác cũng khá nhiều.
* Hiện nay nhu cầu thông tin của con người này càng cao, nhu cầu về thông tin có chất lượng, không rác, không trùng lặp càng lớn. Đối mặt với những thách thức về lượng dữ liệu khổng lồ trên internet đó, cần phải có 1 hệ thống tìm kiếm để giúp người dùng có thể nhanh chóng tiếp cận với dữ liệu mình cần và dữ liệu cần phải có chất lượng cao: không rác, không trùng lặp.
* Có nhiều nền tảng opensource giúp triển khai 1 máy tìm kiếm như : Apache Solr, Elatic Search…
* Trước nhu cầu đó thì nhóm em quyết định lựa chọn Apache Solr để xây dựng một máy tìm kiếm giúp tìm kiếm:
* Nội dung tìm kiếm : Tin tức văn bản hằng ngày
* Ngôn ngữ tìm kiếm : Tiếng Việt.
* Nguồn dữ liệu : Các website tin tức
* Giao diện tương tác người sử dụng: Giao diện web.

1. Giới thiệu về Solr

* 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.
* Solr chạy như một máy chủ tìm full-text search độc lập. Nó sử dụng lõi thư viện tìm kiếm Lucene Java cho việc đánh chỉ mục full-text và tìm kiếm, và có REST (REpresentational State Transfer) như API HTTP/XML và JSON khiến nó có thể sử dụng bởi những ngôn ngữ lập trình phổ biến nhất. Cấu hình mở rộng của Solr cho phép nó có thể được thiết kế với nhiều loại ứng dụng mà không cần code Java, và nó có một kiến trúc plugin hỗ trợ chỉnh sửa nâng cao hơn.
* Tổng quát thì Solr bao gồm có nhiều thành phần (components) khác nhau bao gồm**:**
* Apache Lucene để phân tích, đánh chỉ mục tìm kiếm dữ liệu.
* Apache Tika dùng để trích xuất metadata, tìm kiếm và chỉ mục nhiều loại file document khác nhau như pdf, docx, mp3, jpg (hỗ trợ 66 file types khác nhau).
* Apache UIMA (đọc như you-eee-mah, Unstructured Information Managemen Architecture), đây cũng là một project thuộc Apache Foudnation, nó được dùng để phân tích một lượng lớn dữ liệu không có cấu trúc nhằm tìm ra được những thông tin có ích cho người dùng. Ví dụ:
* Phân tích các phim và trích xuất phụ đề rồi dựa vào đó để tìm ra diễn viên nào đóng trong phim đó.
* Tìm các bài viết, video, hình ảnh có liên quan tới chủ đề của một bài viết cụ thể nào đó.
* Apache Velocity là một Java-based template engine.
* Carrot2 (search results clustering engine) dùng để phân loại và nhóm các kết quả tìm kiếm thành những danh mục có cùng chủ đề (thematic categories).
* Một số đặc trưng của Apache Solr:
* Khả năng tìm kiếm văn bản toàn diện(Full-Text Search) giống kiểu Google.
* Chỉnh sửa để hiệu năng tốt hơn.
* Dựa trên các chuẩn mở trong giao tiếp với các hệ thống khác – XML, JSON và HTTP
* Quản trị dưới dạng giao diện HTML đơn giản
* Thống kê dưới dạng JMX
* Cấu hình đơn giản dễ dàng với định dạng XML
* Có khả năng bổ sung các phần mở rộng(plugin) mới. Ví dụ như phân tích mở rộng tiếng Việt: Bắt lỗi chính tả, bỏ dấu,…
* Cho phép highlighting kết quả tìm kiếm, như cách mà google hiện thị thông tin tóm tắt về kết quả mà ở đó câu truy vấn được in đậm
* Có thể xây dựng rất nhiều ứng dụng khác mà một trang tìm kiếm cần như: autosuggestion, spellchecking, xây dựng tagcloud, phân loại kết quả clustering (như Bing làm), trending keywords, category navigation, các kết quả liên quan, nhóm kết quả (field collapsed) …
* Cho phép scale hệ thống một cách dễ dàng khi bạn có một lượng lớn dữ liệu mà không đủ chứa trên một máy chủ hay phải phục vụ rất nhiều người dùng đồng thời.
* Solr cũng có thể dùng như CSDL NoSQL hay như cache layer, dùng cho các listing cần performance tốt.
* Solr cũng sắp hỗ trợ realtime cho phép tìm kiếm ngay kết quả sau khi index. Điều nàyđặc biệt khó khi index rất lớn. Hiện tại Solr cho phép kết quả rât nhanh, nhưng phải hy sinh thời gian index. Với dữ liệu lớn có khi bạn phải mất 30 phút chỉ để cập nhật được một tài liệu.
* Solr hỗ trợ rất nhiều công cụ để tinh chỉnh kết quả tìm kiếm, bằng tất cả các thông tin mà bạn cung cấp làm sao để kết quả trả về là tốt nhất. Ví dự như đánh trọng số các trường, click log, số lượt view, …
* Kiến trúc
* Index trong Solr

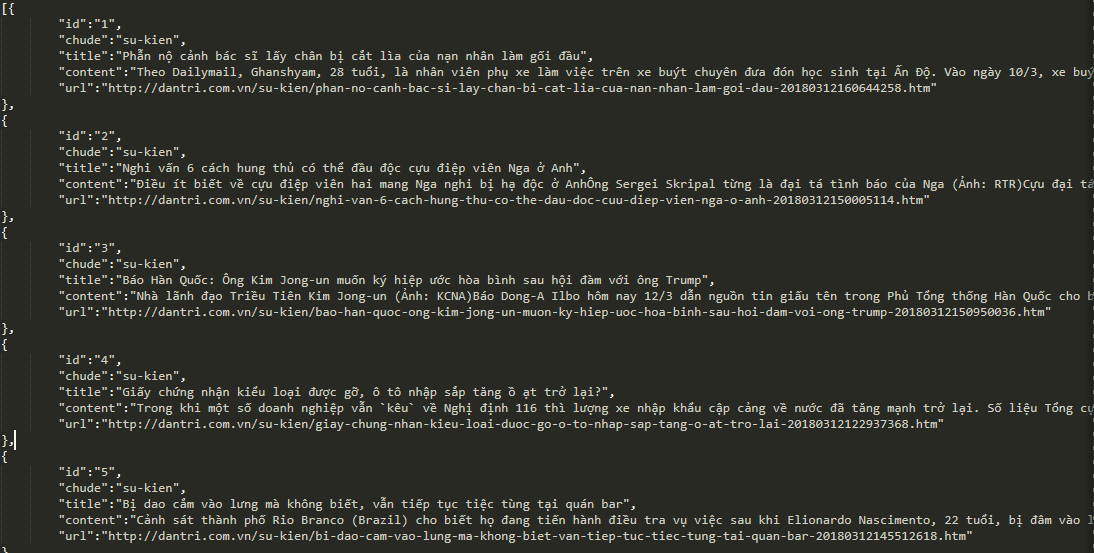


* Chức năng:
* Phân tích, đánh chỉ mục dữ liệu tìm kiếm
* Trích xuất metadata, tìm kiếm và chỉ mục nhiều loại tài liệu khác nhau : pdf, docx, mp3, jpg…
* Hỗ trợ trên tìm kiếm trên ngôn ngữ: Anh, Pháp, Thái Lan….
* Khả năng mở rộng, tùy biến cao.

# **Triển Khai**

1. Thu thập dữ liệu

* Nguồn thu thập: các bài báo trong trang <http://dantri.com.vn>
* Thư viện hỗ trợ : Jsoup của Java
* Các trường thu thập :
* Subject : Chủ đề, thể loại của loại tin đó.
* Tile : Tiêu đề của tin tức.
* Content : Nội dung của tin tức.
* URL: Đường dẫn tới trang tin tức gốc.
* Đầu ra : file json gồm 8000 tin tức.
* Làm sạch dữ liệu: Loại bỏ các ký tự đặc biệt, các thẻ html….



1. Cài đặt Apache Solr

* Cài đặt Solr với 2 collection để thuận tiện cho việc đánh giá : 1 collection( core1) với cấu hình mặc định của Solr, 1 collection (core2) đã sửa file schema: thêm cấu hình đánh index để tạo ra các term không dấu( hỗ trợ việc search không dấu), cộng với cấu hình file stopwords trỏ tới file stopword tiếng việt.
* Các bước thực hiện
* Cài đặt Apache Solr 7.2 trên ubunu 14.04. Chi tiết cài đặt : <https://lucene.apache.org/solr/guide/7_2/installing-solr.html>
* Tạo collection :

#sudo su - solr -c "/opt/solr-7.2.1/bin/solr create -c core1 -n data\_driven\_schema\_configs"

#sudo su - solr -c "/opt/solr-7.2.1/bin/solr create -c core2 -n data\_driven\_schema\_configs"

* Cấu hình schema ( áp dụng với core2) để đánh index cho dữ liệu:

+ Thêm 1 fieldType : text\_genaral\_vn

*<fieldType name="text\_general\_vn" class="solr.TextField" positionIncrementGap="100" multiValued="true">*

*<analyzer type="index">*

*<tokenizer class="solr.StandardTokenizerFactory"/>*

*<filter class="solr.StopFilterFactory" words="lang/stopwords\_vn.txt" ignoreCase="true"/>*

*<filter class="solr.LowerCaseFilterFactory"/>*

*<filter class="solr.ASCIIFoldingFilterFactory" preserveOriginal="true"/>*

*</analyzer>*

*<analyzer type="query">*

*<tokenizer class="solr.StandardTokenizerFactory"/>*

*<filter class="solr.StopFilterFactory" words="lang/stopwords\_vn.txt" ignoreCase="true"/>*

*<filter class="solr.SynonymGraphFilterFactory" expand="true" ignoreCase="true" synonyms="synonyms.txt"/>*

*<filter class="solr.LowerCaseFilterFactory"/>*

*<filter class="solr.ASCIIFoldingFilterFactory" preserveOriginal="true"/>*

*</analyzer>*

*</fieldType>*

+ Thêm 1 filter để đánh index với tiếng việt không dấu:

*<filter class="solr.ASCIIFoldingFilterFactory" preserveOriginal="true"/>*

+Tạo 1 file từ dừng tiếng việt : stopwords\_vn.txt (<https://raw.githubusercontent.com/stopwords/vietnamese-stopwords/master/vietnamese-stopwords.txt>) và cấu hình cho solr loại bỏ từ dừng từ tập từ dừng trong danh sách này.

*<filter class="solr.StopFilterFactory" words="lang/stopwords\_vn.txt" ignoreCase="true"/>*

* Tạo các field thuộc FieldType text\_genaral\_vn đã tạo ở trên ( Áp dụng với core2):
* Subject : Field này tương ứng với trường Subject trong dữ liệu thu thập.

*<field name="chude" type="text\_general\_vn" multiValued="true" indexed="true" stored="true"/>*

* Title: Field này tương ứng với trường Title trong dữ liệu thu thập.

*<field name="title" type="text\_general\_vn" multiValued="true" indexed="true" stored="true"/>*

* Content : Field này tương ứng với trường Content trong dữ liệu thu thập.

*<field name="content" type="text\_general\_vn" multiValued="true" indexed="true" stored="true"/>*

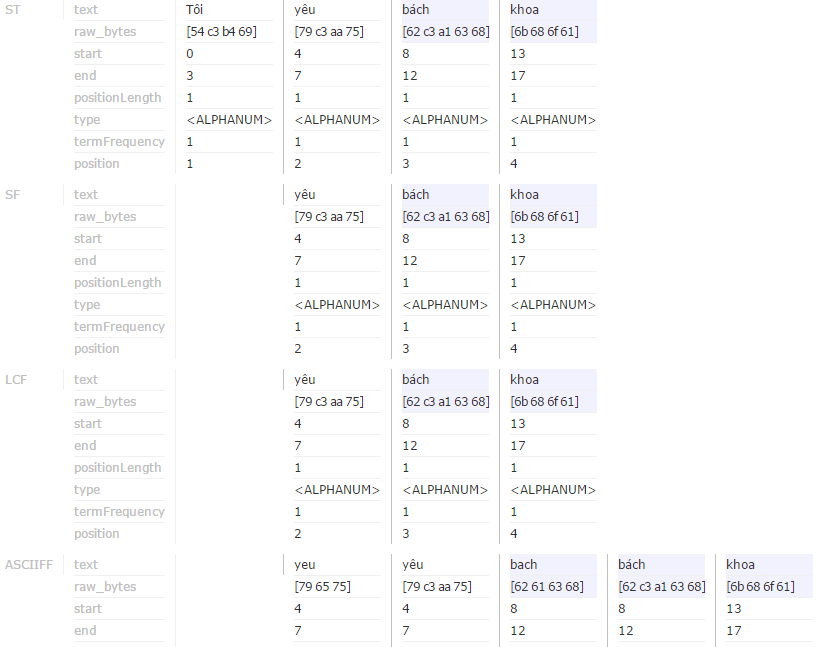
* Url : Field này tương ứng với trường Url trong dữ liệu thu thập.

*<field name="url" type="text\_general\_vn" multiValued="true" indexed="true" stored="true"/>*

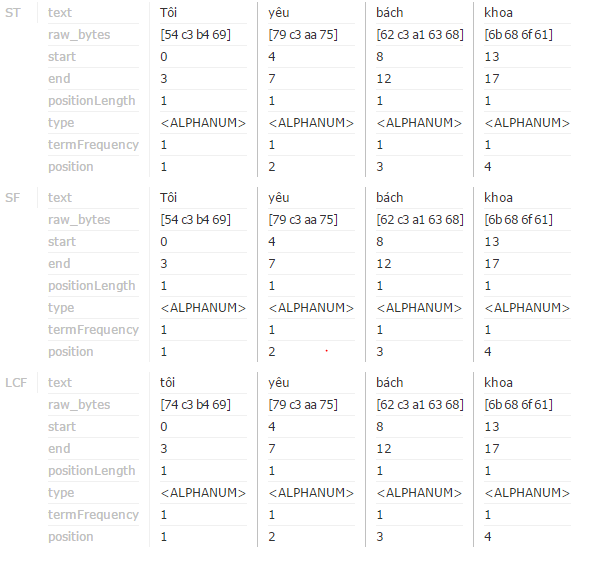
* Index giữ liệu vào Solr sử dụng post tool:

#/opt/solr-7.2.1/bin/post -c final\_collection document.json

* Kiểm tra việc hoạt động của analyzer trong việc tách từ và loại bỏ từ dừng cho index và query.

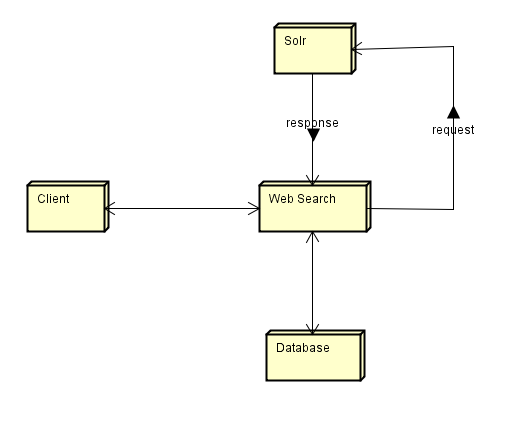


**Tách từ và loại bỏ từ dừng với core2**



**Tách từ và loại bỏ từ dừng với core1**

1. Xây dựng server web tìm kiếm cho người sử dụng
2. *Kiến trúc*



1. *Xây dựng giao diện web*

* Backend: Sử dụng Laravel Framework
* Front-end : Sử dụng templace search bootstraps.
* Giao diện gồm có : Form nhập từ khóa tìm kiếm, phần body hiển thị kết quả: title, url, content. Giao diện có hỗ trợ phân trang, hiển thị số lượng kết quả tìm thấy, thời gian truy vấn.
* Giao diện kết quả trả về có lựa chọn để người dùng chọn chỉ những chủ đề mà mình muốn hiển thị.
* Web server sẽ gọi API của Solr để trả về kết quả rồi hiển thị cho người dùng.
* Luồng hoạt động:

Từ giao diện front-end, người dùng nhập 1 câu truy vấn bất kỳ. Từ khóa sẽ được gửi tới back-end.

Back-end xử lý từ khóa và gửi từ khóa đó thông qua lời gọi API tới server Solr thông qua plugin Guzzle.

Server Solr truy vấn trên từ khóa đó và trả về cho Back-end.

Back-end đọc file Json xử lý đổ giao diện trả về cho front-end để người dùng xem.

1. *Các tính năng mới*

* Thống kê các từ khóa được truy vấn nhiểu nhất dựa trên việc lưu lại lịch sử tìm kiếm của người dùng.
* Gợi ý truy vấn :

Tác dụng: Khi người dùng gõ từ khóa thì hệ thống sẽ gợi ý ra list các từ khóa có thể phù hợp với từ khóa mà người dùng muốn truy vấn

Luồng hoạt động: Cơ sở của việc gợi ý này là dựa trên việc so sánh từ mà người dùng đang gõ với top từ khóa của hệ thống, nếu tương đồng thì sẽ gợi ý cho người dùng để người dùng không phải gõ nhiều.

Công nghệ: Sử dụng HTML5: Danh sách top từ khóa sẽ được đổ ra 1 cặp thẻ <datalist> và thuộc tính list của thẻ <input> sẽ bằng id của thẻ <datalist>

* Highlight từ khóa tìm kiếm:

Tác dụng: Giúp người dùng thấy rõ được từ khóa mình đang tìm kiếm có xuất hiện trong văn bản

Luồng hoạt động: Khi server web gọi API của Solr để lấy kết quả tìm kiếm và sau đó server web sẽ thực hiện in đậm các từ khóa tìm kiếm có trong văn bản và trả về cho người dùng.

Công nghệ: Sử dụng hàm str\_replace trong php để thay thế từ khóa tìm kiếm xuất hiện trong văn bản bằng chính từ khóa đó và thêm thẻ <b> để in đậm.

* Tìm kiếm tiếng việt không dấu:

Tác dụng: Giúp tìm kiếm được các từ khóa không dấu do người dùng nhập vào

Luồng hoạt động:Giống luồng hoạt động của truy vấn thông thường.

Công nghệ: Cấu hình Solr sử dụng class tách từ *ASCIIFoldingFilterFactory*  như ở mục 2.

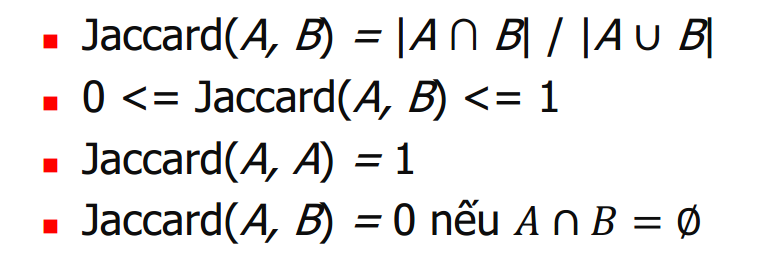
* Sửa từ khóa tìm kiếm khi người dùng gõ sai

Tác dụng: Sửa từ khóa gõ sai cho người dùng, giúp cải thiện hiệu quả tìm kiếm.

Phương pháp: Sử dụng thống kê với độ đo jaccard.

Luồng hoạt động: Khi người dùng nhập vào từ khóa truyền lên hệ thống web server, server web sẽ tính độ đo jaccard giữa từ khóa người dùng nhập với từng từ khóa trong top từ khóa, nếu độ đo jaccard nằm trong khoảng 0.55 tới nhỏ hơn 1 thì từ khóa trong top sẽ là từ khóa phù hợp. Và hệ thống sẽ tìm kiếm trên từ khóa phù hợp đó và trả kết quả cho người dùng.

Ví dụ : Từ khóa “Đại học Bách Khoa Hà Noi”, ở đây từ “Noi” viết sai. Trong danh sách thống kê từ khóa đã có từ “Đại Học Bách Khoa Hà Nội” . Hệ thống sẽ tính độ đo Jaccard giữa 2 xâu này = 0.89 nằm trong khoảng (0.55,1) thì hệ thống sẽ sửa từ khóa cho người dùng thành “Đại Học Bách Khoa Hà Nội”.



**Độ đo jaccard**

# **Demo**

* Tìm kiếm thông thường và highlight từ khóa.



* Tìm kiếm không dấu



* Gợi ý từ khóa



* Sửa lỗi từ khóa:



* Thống kê từ khóa tìm kiếm nhiều nhất trong tháng



# **Đánh giá**

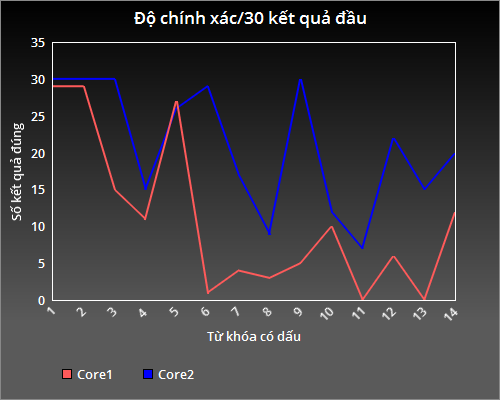
1. Đánh giá độ chính xác

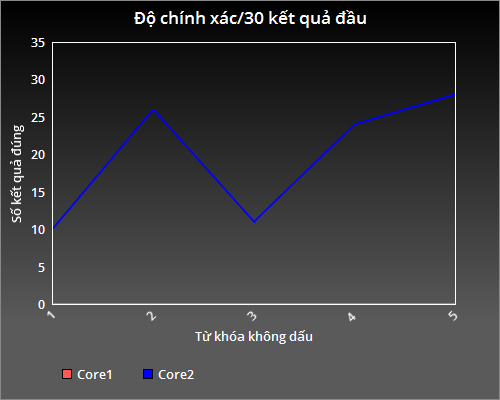
* Bảng đo độ chính xác và độ recall (core1)

|  |  |  |  |
| --- | --- | --- | --- |
| Từ khóa | Số kết quả chính xác 30 kết quả đầu tiên | Số kết quả chính xác 60 kết quả đầu tiên | Độ recall |
| U23 Việt Nam | 29 | 53 | 0.55 |
| Trạm thu phí BOT | 29 | 56 | 0.52 |
| Tổng thống Trump | 15 | 29 | 0.52 |
| Thưởng tết 2018 | 11 | 12 | 0.92 |
| Ô tô giảm giá | 27 | 46 | 0.59 |
| Arsenal Watford | 1 | 1 | 1 |
| Giá xăng dầu tăng | 4 | 6 | 0.67 |
| Lịch nghỉ Tết 2018 | 3 | 3 | 1 |
| MU Liverpool | 5 | 9 | 0.56 |
| Pháo hoa đêm giao thừa | 10 | 11 | 0.91 |
| Mỹ gặp Triều Tiên | 0 | 0 | 0 |
| Tiền ảo Bitcoin | 6 | 7 | 0.86 |
| Các đội tham dự world cup 2018 | 0 | 0 | 0 |
| Taxi truyền thống với Uber | 12 | 19 | 0.63 |
| Thay Van Nhu Cuong | 0 | 0 | 0 |
| Thoi tiet mien bac | 0 | 0 | 0 |
| Gia xang tang | 0 | 0 | 0 |
| Khong khi lanh mien bac | 0 | 0 | 0 |
| Dinh La Thang | 0 | 0 | 0 |

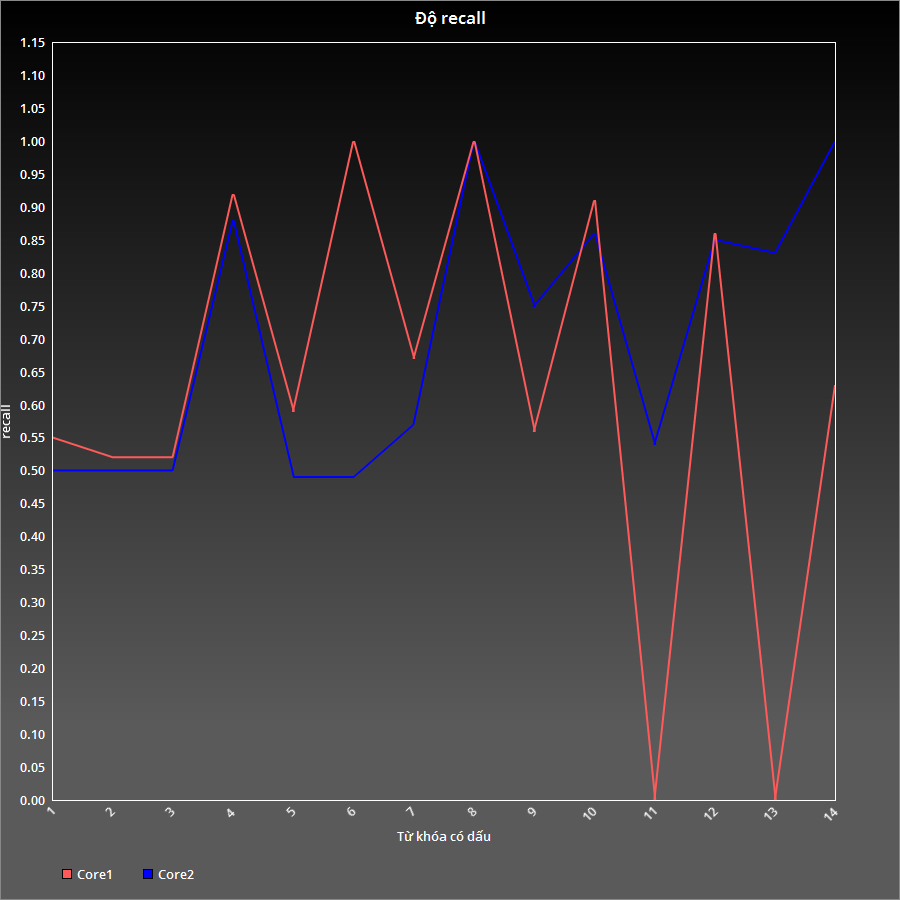
* Bảng đo độ chính xác và độ recall (core2)

|  |  |  |  |
| --- | --- | --- | --- |
| Từ khóa | Số kết quả chính xác 30 kết quả đầu tiên | Số kết quả chính xác 60 kết quả đầu tiên | Độ recall |
| U23 Việt Nam | 30 | 60 | 0.5 |
| Trạm thu phí BOT | 30 | 60 | 0.5 |
| Tổng thống Trump | 30 | 60 | 0.5 |
| Thưởng tết 2018 | 15 | 17 | 0.88 |
| Ô tô giảm giá | 26 | 53 | 0.49 |
| Arsenal Watford | 29 | 59 | 0.49 |
| Giá xăng dầu tăng | 17 | 30 | 0.57 |
| Lịch nghỉ Tết 2018 | 9 | 9 | 1 |
| MU Liverpool | 30 | 40 | 0.75 |
| Pháo hoa đêm giao thừa | 12 | 14 | 0.86 |
| Mỹ gặp Triều Tiên | 7 | 13 | 0.54 |
| Tiền ảo Bitcoin | 22 | 26 | 0.85 |
| Các đội tham dự world cup 2018 | 15 | 18 | 0.83 |
| Taxi truyền thống với Uber | 20 | 20 | 1 |
| Thay Van Nhu Cuong | 10 | 10 | 1 |
| Thoi tiet mien bac | 26 | 47 | 0.55 |
| Gia xang tang | 11 | 13 | 0.85 |
| Khong khi lanh mien bac | 24 | 31 | 0.77 |
| Dinh La Thang | 28 | 54 | 0.52 |

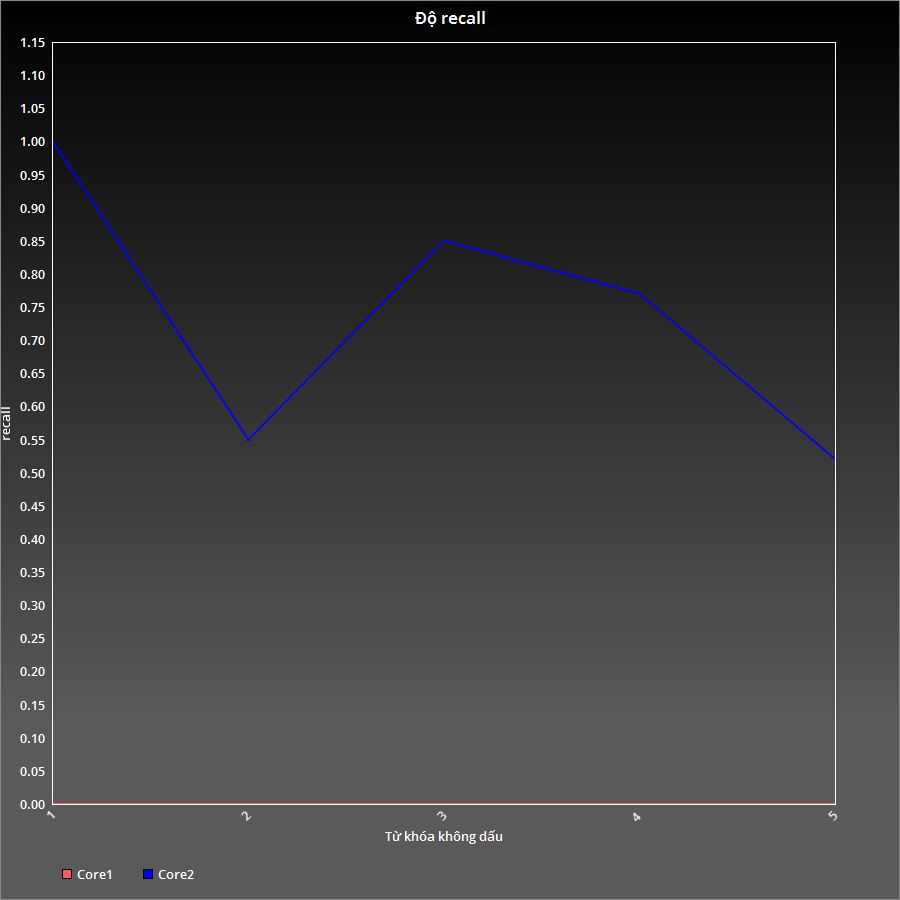
 **Biểu đồ so sánh độ chính xác 2 core1 và core2 với tiếng việt có dấu**



**Biểu đồ so sánh độ chính xác core1 và core2 tiếng việt không dấu**



**Biểu đồ so sánh độ recall core1 và core2 với tiếng việt có dấu**



**Biểu đồ so sánh độ recall core1 và core2 với tiếng việt không dấu**

1. So sánh 2 collection core1 và core2

* Core1 chỉ áp dụng tách từ theo khoảng trắng và dấu câu nên kích thước index, không có loại bỏ từ dừng tiếng việt nên kích thước index sẽ lớn vì phải đánh index cho cả các từ dừng.
* Do chỉ tách từ theo khoảng trắng và dấu câu nên core1 sẽ làm việc tốt với việc tìm kiếm các từ khóa là tiếng việt còn với những từ khóa không dấu thì không thể làm việc được.
* Với những từ khóa tiếng việt mà chỉ cần có 1 từ thiếu 1 dấu câu thì độ chính xác sẽ giảm xuống.
* Core2 áp dụng việc loại bỏ từ dừng tiếng việt nên các từ dừng tiếng việt sẽ được loại bỏ giảm kích thước index, tăng độ chính xác khi tìm kiếm.
* Core2 áp dụng việc tách 1 từ tiếng việt thành 1 từ có dấu và 1 từ không dấu thì có thể áp dụng tìm kiếm với tiếng việt không dấu tuy kích thước của index tăng lên.
* Nếu xét về chất lượng của việc chỉ tìm kiếm trên tiếng việt có dấu và người dùng gõ đúng từ khóa thì chất lượng của core1 sẽ tốt hơn core2. Nhưng tổng thể thì thường người dùng sẽ nhập thiếu từ khóa, từ khóa không dấu nên tổng thể thì core2 sẽ áp dụng thưc tế hơn core1.

# Tổng kết

1. Ưu điểm

* Thời gian truy vấn ngắn, kết quả trả về có độ chính xác và độ recall khá tốt nên có thể tích hợp hệ thống tìm kiếm này vào các hệ thống có sẵn giúp hỗ trợ việc tìm kiếm dễ dàng hơn.
* Giao diện hệ thống thân thiện, dễ sử dụng.
* Hệ thống gồm nhiều tính năng như: tìm kiếm với tiếng việt không dấu, sửa từ khóa, highlight giúp dễ dàng cho người dùng, cải thiện kết quả tìm kiếm.

1. Nhược điểm

* Việc tách từ trong Solr đang dừng lại ở việc tách từ theo dấu câu, khoảng trắng chứ chưa theo ngữ nghĩa nên độ chính xác không cao.
* Danh sách từ khóa phổ biến trong cơ sở dữ liệu còn ít do hệ thống chỉ ở mức demo chưa public ra ngoài nên chưa lấy được nhiều từ khóa nên việc gợi ý từ khóa chưa đạt hiệu quả tối đa.
* Danh sách từ khóa phổ biến ít nên việc sửa lỗi từ khóa còn đạt hiệu quả thấp, nhiều từ khóa viết sai mà không được sửa. Hệ số Jaccard còn chưa tối ưu nhất.



* Chưa sửa được lỗi chính tả tiếng việt.

# **Danh mục tài liệu tham khảo**

1. <https://lucene.apache.org/solr/guide/7_2/installing-solr.html>
2. <https://raw.githubusercontent.com/stopwords/vietnamese-stopwords/master/vietnamese-stopwords.txt>
3. <http://butchiso.com/2013/08/tim-hieu-ve-apache-solr.html>
4. <https://www.digitalocean.com/community/tutorials/how-to-install-solr-on-ubuntu-14-04>
5. <https://laravel.com/>
6. <https://bootsnipp.com/snippets/AlRxx>
7. <https://bootsnipp.com/snippets/bpWm9>