DoIT

TÀI LIỆU MÔ TẢ KỸ THUẬT

9/2017

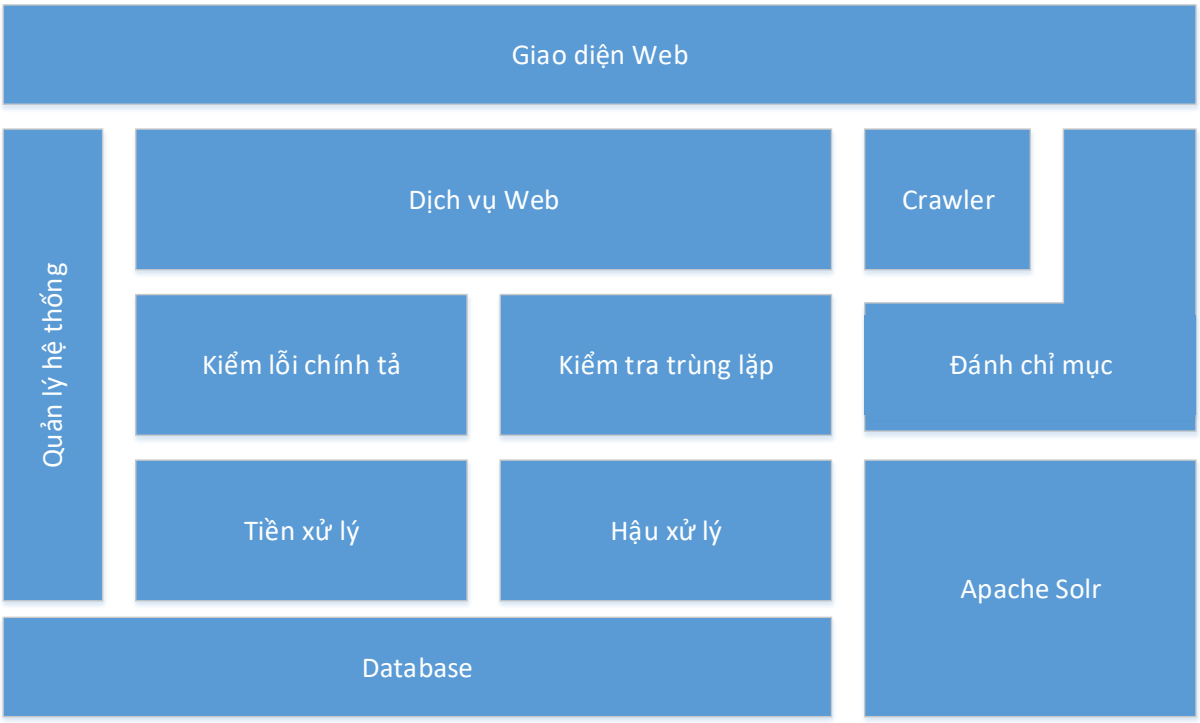
MỤC LỤC

Tài liệu này trình bày một số điểm kỹ thuật cơ bản của hệ thống DoIT. Phần đầu của tài liệu trình bày về kiến trúc tổng thể của hệ thống. Sau đó, các kỹ thuật chính sử dụng cho hai chức năng kiểm lỗi chính tả và phát hiện trùng lặp sẽ được mô tả. Hai phần cuối của tài liệu trình bày về các phần mềm nền tảng và công cụ được sử dụng để xây dựng DoIT và cấu hình phần cứng tham khảo để triển khai hệ thống.

1 Kiến trúc hệ thống

Phần này sẽ trình bày về các mô đun chính trong hệ thống và các bước xử lý chính khi một tài liệu được người dùng gửi lên hệ thống.

1.1 Các mô đun chính



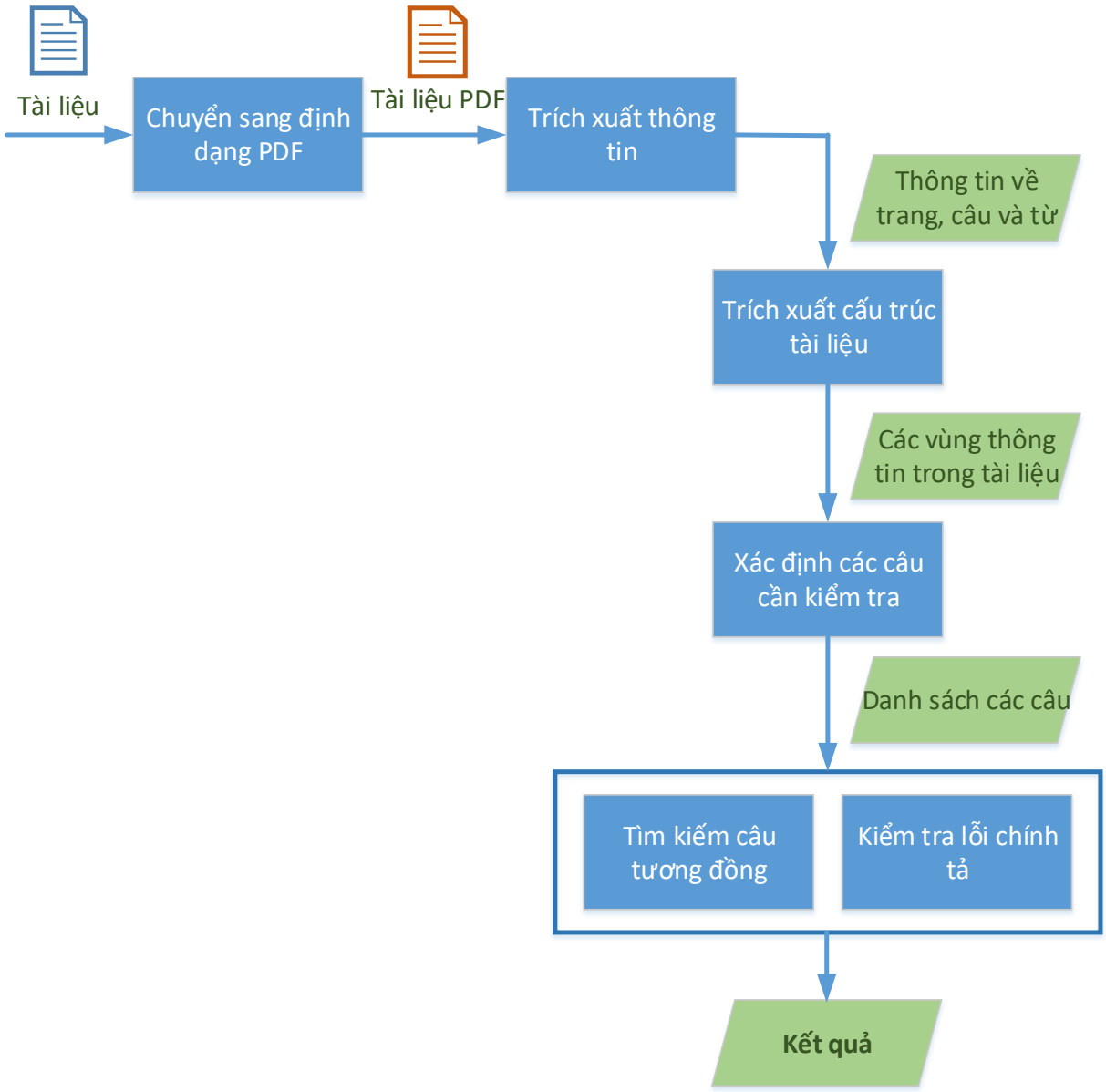
Hình 1: Kiến trúc hệ thống

Hình 1 ở trên mô tả kiến trúc của hệ thống đã được xây dựng. Người dùng cuối sử dụng hệ thống thông qua trình duyệt Web. Mô đun Dịch vụ Web cung cấp API để

phần ứng dụng Web sử dụng các chứ năng của hệ thống. Việc xây dựng các chức năng dưới dạng dịch vụ Web sẽ làm cho hệ thống dễ dàng có các loại phần mềm khách khác nhau và cho phép các hệ thống khác có thể kết nối đến. Hai mô đun quan trọng nhất của hệ thống là Kiểm lỗi chính tả và Kiểm tra trùng lặp. Mô đun Tiền xử lý nhận các file văn bản với các định dạng khác nhau, phân tích và trích xuất thông tin về nội dung, bố cục và siêu dữ liệu (ví dụ như: tác giả, tên luận văn…) để chuẩn bị cho việc kiểm lỗi chính tả và kiểm tra trùng lặp. Mô đun Hậu xử lý tổng hợp kết quả, chuẩn bị các thông tin hướng dẫn/khuyến cáo cho người dùng sau khi việc kiểm lỗi chính tả, kiểm tra trùng lặp được thực hiện xong. Crawler là mô đun thu thập dữ liệu từ Internet. Các website thu thập được sẽ được đánh chỉ mục vào Apache Solr. Mô đun Quản lý hệ thống cung cấp các chức năng liên quan đến các khía cạnh chung trong hệ thống như tài khoản người dùng, văn bản, quản lý cấu hình Apache Solr …

1.2 Các bước xử lý chính

Các bước xử lý khi một tài liệu được người dùng gửi lên hệ thống được mô tả trong Hình 2. Khi người dùng tải lên hệ thống một tài liệu để kiểm tra trùng lặp và lỗi chính tả, tài liệu này sẽ được chuyển thành dạng PDF nhằm thống nhất cách xử lý về sau. Tài liệu ở định dạng PDF này sẽ được phân tích để trích xuất từ, câu, và trang (khối Trích xuất thông tin). Thông tin được trích xuất bao gồm cả định dạng của từ và tọa độ vị trí của các phần tử này. Dựa trên những dữ liệu này, meta-data của tài liệu (gồm tác giả, tiêu đề, và một số thông tin khác) sẽ được trích xuất. Khối trích xuất cấu trúc tách và đánh dấu các khối nội dung khác nhau trong tài liệu. Sau bước xử lý này, chúng ta sẽ biết được các khối như trang tiêu đề, mục lục, các đề mục, các đoạn nội dung. Trong bước tiếp theo, danh sách các câu được xem là nội dung của tài liệu sẽ được trích xuất và được thực hiện kiểm tra chính tả và tương đồng. Kết quả kiểm tra sẽ được sử dụng để đánh dấu và chuẩn bị cho việc hiển thị.



Hình 2. Các bước xử lý chính

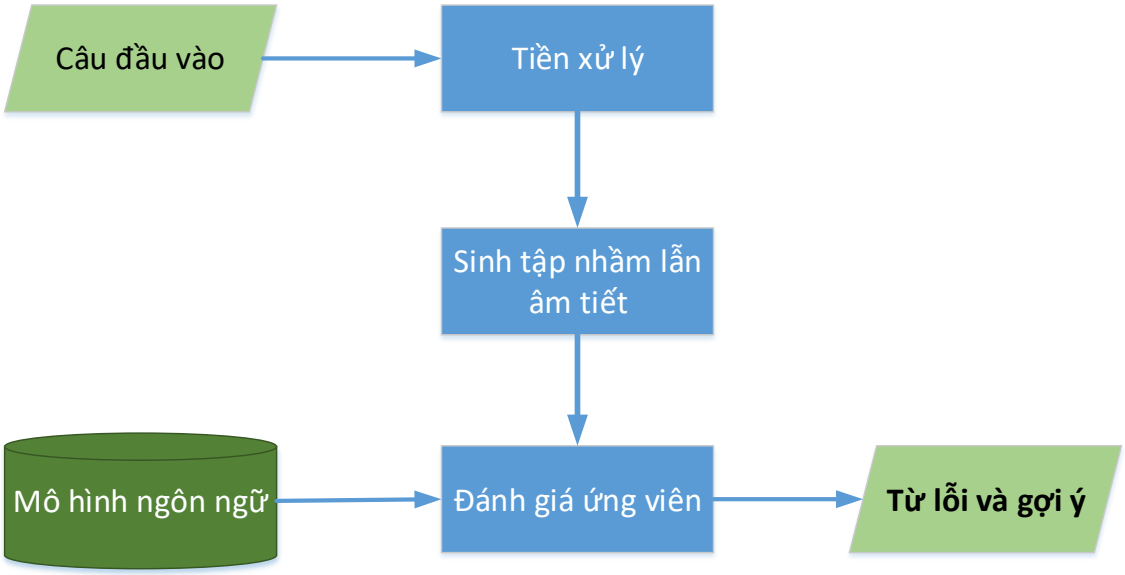
2 Chức năng kiểm lỗi chính tả

Mô đun kiểm lỗi chính tả được chia thành hai chức năng con là phát hiện lỗi và gợi ý sửa lỗi. Chức năng phát hiện lỗi tìm kiếm các âm tiết bị lỗi chính tả trong văn bản. Chức năng sửa lỗi sẽ đưa ra các gợi ý sửa chữa và tự động lựa chọn phương án hợp lý nhất. Lỗi chính tả trong Tiếng Việt được chia thành hai loại chính: âm tiết sai chính tả

không tồn tại trong từ điển Tiếng Việt và âm tiết sai chính tả do ngữ cảnh. Trong mô đun này chúng tôi chủ yếu tập trung vào phần âm tiết sai chính tả do ngữ cảnh. Những âm tiết này tồn tại trong từ điển Tiếng Việt nhưng không phù hợp với văn bản (Ví dụ: trong câu "Cuốn xách này rất hay", từ "xách" mang ý nghĩ là mang, vác theo đã bị dùng sai, từ chính xác cần được dùng ở đây là từ "sách").

Trong hệ thống DoIT, chúng tôi sử dụng mô hình ngôn ngữ N-gram làm hướng tiếp cận chính. Đồng thời, phân đoạn từ (word segmentation) và khoảng cách Levenstein được sử dụng để hỗ trợ đánh giá ứng viên tốt nhất.

Hình 3 mô tả các mô đun chính trong phân hệ kiểm lỗi chính tả. Phần sau sẽ trình bày các chức năng của từng mô đun.



Hình 3: Kiến trúc tổng quát của phân hệ kiểm lỗi chính tả

2.1 Mô đun tiền xử lý

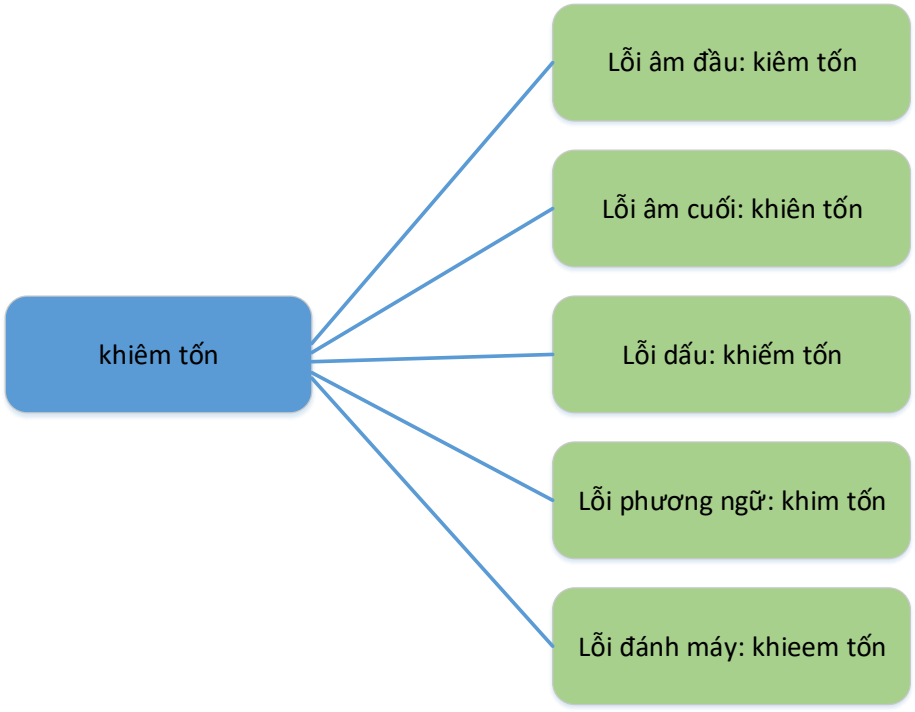
Mô đun tiền xử lý thực hiện việc phân tích từ tố, chuẩn hóa một số loại từ tố đặc biệt:

- Tách dấu câu, phân biệt các từ tố, loại bỏ các yếu tố gây nhiễu: phân tách dấu câu gắn với từ, tách câu thành các từ riêng biệt, loại bỏ các ký tự đặc biệt như ký tự cảm xúc,…

- Loại bỏ vấn đề lặp ký tự và từ viết tắt: dựa vào các quy tắc của Tiếng việt và tập từ điển để sửa chữa những yếu tố này, thông thường, đối với nguyên âm tiếng việt không xuất hiện quá 2 lần, phụ âm chỉ xuất hiện 1 lần.
- Nhóm các loại từ tố đặc biệt vào các lớp như lớp Số, lớp Ngày tháng, lớp Tên, … và gán nhãn đặc trưng: với từng lớp từ đặc biệt sẽ được gán nhãn tương ứng để thuận tiện trong việc sử dụng, ví dụ lớp Số - NUMBER, lớp Ngày tháng – DATE, …

2.2 Mô đun sinh tập nhầm lẫn âm tiết

Tập nhầm lẫn của một âm tiết *s* là tập hợp các âm tiết mà mỗi âm tiết trong tập đó có thể bị nhầm lẫn thành *s* (có mối quan hệ về chính tả). Có rất nhiều nguyên nhân nhầm lẫn như đánh máy, phát âm (phương ngữ) ở người hay do đặc trưng của các hệ nhận dạng chữ. Một chương trình kiểm lỗi chính tả đi kèm với các hệ soạn thảo văn bản cần sử dụng các tập nhầm lẫn do đánh máy và phát âm hợp lại.



Hình 4: Các lỗi thường gặp

Nhầm lẫn do đánh máy sai

Đây là kiểu nhầm lẫn phổ biến. Thường có 3 loại lỗi nhầm lẫn do đánh máy như sau:

- Chèn: nhầm "run" thành "rung" (chèn thêm chữ cái 'g')
- Xóa: nhầm "chung" thành "cung" (xóa chữ 'h')
- Thay thế: nhầm "bán" thành "ván" (thay thế chữ 'b' thành chữ 'v')

Một âm tiết có thể bị biến đổi sau những lần biến đổi liên tiếp của 1 trong 3 phép biến đổi trên. Ví dụ: "nầm" thành "năm" (xóa dấu huyền và thanh thế 'â' thành 'ă'). Trên thực tế, âm tiết chỉ thường nhầm lẫn sau 1 phép biến đổi.

Nhầm lẫn do phát âm

- Khi chữ viết phân biệt âm tiết mà phát âm theo một phương ngữ nào đó lại không phân biệt. Cho nên với những phương ngữ khác nhau có những vấn đề chính tả khác nhau. Trong khi với người nói phương ngữ miền Bắc có các vấn đề chính tả "viết ch- hay tr-", "viết -iêu hay -ươu", v.v. thì với người nói phương ngữ miền Nam lại có các vấn đề "viết -n hay -ng", "viết dấu hỏi hay dấu ngã", v.v.
- Khi chữ viết phân biệt âm tiết mà phát âm tiếng Việt ngày nay không còn phân biệt, đó là trường hợp "viết d- hay gi-", một vấn đề chính tả chung cho mọi miền trong cả nước.

Bảng 1 dưới đây liệt kê các loại lỗi phát âm thường gặp.

Bảng 1: Bảng liệt kê các lỗi phát âm thường gặp

CH-	TR-		
D-	GI-		
D-	GI-	NH-	
D-	GI-	R-	
D-	GI-	V-	
HW-	NG-	QU-	W-
L-	N-		
S-	X-		
-C	-T		
-N	-NG		

-AI	-AY	
-EM	-ÊM	
-ÊCH	-ÊT	
-IÊM	-IM	
-IÊU	-IU	
-IÊU	-ƯƠU	
-OAI	-OI	
-OM	-ÔM	-ƠM
Hỏi	ngã	
Ngã	Nặng	

Ví dụ:

- Âm tiết: “dam” → Tập nhầm lẫn: “giam”
- Âm tiết: “nhòm” → Tập nhầm lẫn: “nhồm”, “nhờm”
- Âm tiết: “chương” →Tập nhầm lẫn: “trương”

Một vấn đề cần quan tâm trong mô đun này là tập âm tiết nhầm lẫn có thể bùng nổ về số lượng. Điều này sẽ làm giảm tốc độ xử lý của mô đun khi dùng trong thực tế. Để giảm kích thước của tập nhầm lẫn này, mô đun sử dụng từ điển Tiếng Việt và tần suất xuất hiện của âm tiết như một công cụ đắc lực. Việc này sẽ làm giảm đáng kể số lượng các âm tiết không phù hợp. Chúng tôi sử dụng hai điều kiện để kiểm soát các ứng viên được tạo ra: thứ nhất, nếu ứng viên không tồn tại trong từ điển Tiếng Việt sẽ bị loại bỏ, thứ hai, sử dụng tần suất xuất hiện của cụm bigram để loại bỏ các ứng viên không có khả năng để sửa. Nếu từ đang xét là w_0, ứng viên sẽ là w'_0, từ kế tiếp là w_1 và từ kế sau là w_{-1}. Ta có:

$$freq1 = bigram\_frequency(w_{-1}, w_0)$$

$$freq1' = bigram\_frequency(w_{-1}, w'_0)$$

$$freq2 = bigram\_frequency(w_0, w_1)$$

$$freq2' = bigram\_frequency(w'_0, w_1)$$

Nếu $freq1 > freq1'$ and $freq2 > freq2'$ thì mô đun sẽ loại bỏ ứng viên w'_0.

Tóm lại, mô đun sinh tập nhầm lẫn âm tiết có nhiệm vụ là sinh ra tất cả ứng viên thay thế có thể có của một âm tiết đầu vào.

2.3 Mô đun đánh giá ứng viên

Chức năng của mô đun này là xác nhận âm tiết đang được xét có phải là lỗi chính tả hay không? Nếu đúng là lỗi, dựa vào tập âm tiết nhầm lẫn mô đun sẽ chọn lựa ứng viên phù hợp nhất cho việc thay thế. Giả sử từ tại vị trí w_0, ta cần xác định xem từ đó có phải lỗi và nếu là lỗi thì từ đúng (có thể) là thế gì. Chúng tôi sử dụng ngữ cảnh xung quanh từ cần xem xét với bán kính là 2, tức là hai từ kế tiếp và hai từ kế sau w_0. Vậy ta sẽ có: hai từ kế tiếp lần lượt là w_2, w_1 , hai từ kế sau sẽ là w_{-2}, w_{-1} và ứng viên thay thế sẽ là w'_0. Chúng ta cần xác định được giá trị xác xuất sau:

$$P(w_0 | w_{-2}, w_{-1}, w_1, w_2)$$

Sử dụng phương pháp nội suy chúng ta có được:

$$P(w_0 | w_{-2}, w_{-1}, w_1, w_2) \approx f(P(w_0 | w_{-1}, w_1), P(w_0 | w_1, w_2), P(w_0 | w_{-1}, w_{-2}))$$

Để tính toán khả năng này, mô đun áp dụng một đại lượng SC được cấu thành từ ba yếu tố bao gồm có: N-gram, phân đoạn từ (word segmentation) và khoảng cách Levenshtein. N-gram đại diện cho mối quan hệ của ứng viên với ngữ cảnh xung quanh vị trí cần xét. Phân đoạn từ đại diện cho khả năng cấu thành cụm từ có nghĩa trong câu. Khoảng cách Levenshtein thể hiện mức độ tương đồng giữa ứng viên thay thế và âm tiết ban đầu. Biểu thức trên được thể hiện như sau:

$$TotalScore(w'_0) = SC1 \times SC2 \times SC3$$

- $SC1$ thể hiện mối quan hệ của w_{-2}, w_{-1}, w_0
- $SC2$ thể hiện mối quan hệ của w_{-1}, w_0, w_1
- $SC3$ thể hiện mối quan hệ của w_0, w_1, w_2

Mỗi đại lượng SC được tính như công thức sau:

$$SC = a \times sc_{ngram} + b \times sc_{ws} + c \times sc_{ld}$$

Trong đó:

- sc_{ngram} : đại lượng N-gram của ứng viên và ngữ cảnh
- sc_{ws} : đại lượng word segmentation của ứng viên và ngữ cảnh

- sc_{ld} : đại lượng khoảng cách Levenshtein của ứng viên và âm tiết đang xét
- a , b , c : tham số
- $a + b + c = 1$

Với giá trị sc_{ngram} :

$$sc_{ngram} = f(bigram, trigram)$$

Với giá trị sc_{ws}:

$$sc_{ws} = \begin{cases} 1 & trigram \in Dictionary \\ 0.3 & bigram \in Dictionary \\ 0 & trường\ hợp\ khác \end{cases}$$

Với giá trị sc_{ld}:

$$sc_{ld} = \frac{Maxlen\ (w'_0, w_0) - Levenshtein\ distance\ (w'_0, w_0)}{Maxlen\ (w'_0, w_0)}$$

Giá trị $TotalScore$ sẽ được so sánh với một giá trị ngưỡng. Nếu giá trị này thấp hơn ngưỡng thì việc thay thế được xem là không cần thiết (nghĩa là không có lỗi). Bằng việc lựa chọn giá trị đại lượng cao nhất lớn hơn ngưỡng, chúng ta sẽ xác định được vị trí sai cũng như từ gợi ý.

3 Chức năng kiểm tra trùng lặp

Phần này mô tả tổng quan về chức năng kiểm tra trùng lặp của hệ thống. Hai nội dung được trình bày bao gồm cơ sở dữ liệu sử dụng cho chức năng kiểm tra trùng lặp và cách tính độ tương đồng.

3.1 Cơ sở dữ liệu

Hệ thống DoIT dùng hai nguồn dữ liệu để kiểm tra sự trùng lặp: dữ liệu từ Internet và dữ liệu nội sinh (như luận văn, báo cáo,… của các trường, cơ quan).

Tài liệu trên mạng Internet

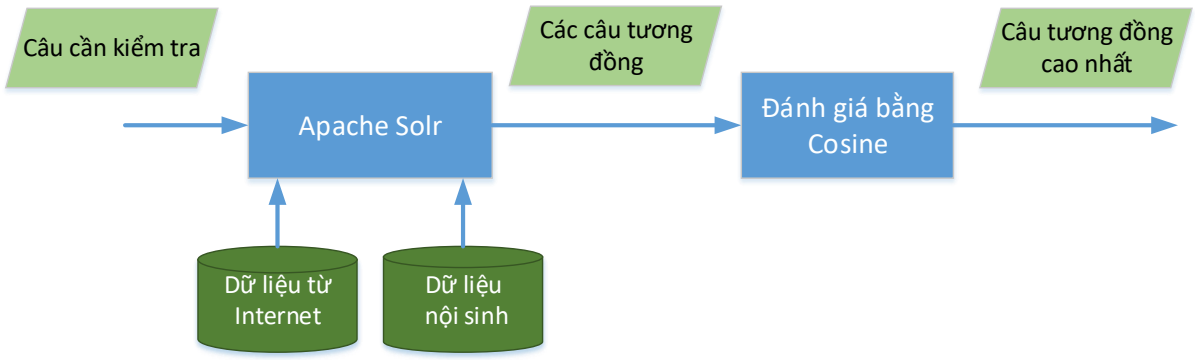
Mô đun thập dữ liệu từ Internet được xây dựng dựa trên Scrapy. Chúng tôi sử dụng các thuật toán xử lý ngôn ngữ tự nhiên và trích xuất thông tin để lấy dữ liệu văn bản từ bất kỳ nguồn nào có trên Internet. Hệ thống cũng cho phép người dùng nhập thêm vào các website để tự động phân tích và lấy thông tin văn bản từ các nguồn đó lưu vào trong cơ sở dữ liệu để làm nguồn dữ liệu cho việc tìm kiếm.

Tài liệu nội bộ

Tài liệu văn bản nội bộ của các trường, cơ quan thường được lưu dưới dạng file văn bản PDF. Trong trường hợp tài liệu không ở dạng PDF, thư viện LibreOffice sẽ được sử dụng để chuyển tài liệu thành định dạng PDF. Chúng tôi dùng thư viện PdfBox[1] và các thuật toán xử lý ngôn ngữ tự nhiên để đọc và trích xuất nội dung siêu dữ liệu (meta-data) và chia ra thành các thành phần như tiêu đề, mục lục, nội dung chương,... sau đó lưu vào trong cơ sở dữ liệu.

3.2 Đánh giá tương đồng

DoIT sử dụng Apache Solr[2] - một thư viện quản lý cơ sở dữ liệu và tìm kiếm hiệu suất cao được viết bằng Java- để kiểm sự tương đồng giữa các câu (một cách chính xác hơn là kiểm tra sự tương đồng giữa một câu trong tài liệu được kiểm tra và các câu trong CSDL). Khi nhận một chuỗi ký tự lớn để tìm kiếm, Apache Solr sẽ tiến hành phân tích chuỗi này thành các phần và tìm kiếm các câu có chứa các từ phần này. Kết quả do Apache Solr trả về là danh sách các câu với độ tương đồng giảm dần. Những câu này được tiếp tục đánh giá bằng độ đo Cosine và chọn câu có độ tương đồng cao nhất (Hình 5).



Hình 5: Đánh giá tương đồng bằng cách sử dụng Apache Solr

Phương pháp sử dụng độ đo Cosine đánh giá sự tương đồng của hai chuỗi ký tự bằng việc vector hóa hai chuỗi ký tự đó thành hai vector trong không gian và tính toán cosine góc giữa hai vector. Sau đó, giá trị này được tổng hợp thành độ tương đồng của đoạn văn bản, chương văn bản và cuối cùng là tổng hợp cho toàn bộ văn bản.

[1] https://pdfbox.apache.org/
[2] https://lucene.apache.org/solr/

4 Các phần mềm nền tảng và công cụ

Sản phẩm DoIT được phát triển dựa trên các phần mềm nền tảng và công cụ chính được liệt kê trong Bảng 2.

Bảng 2: Các phần mềm nền tảng và công cụ

STT	Phần mềm	Chức năng	URL
1	Spring Web MVC	Nền tảng giúp xây dựng website, hỗ trợ tốt cho phát triển các nghiệp vụ phức tạp, yêu cầu hiệu năng đáp ứng cao và bảo mật chặt chẽ.	https://spring.io/
2	Apache Solr	Nền tảng cung cấp chức năng đánh chỉ mục dữ liệu lớn và tìm kiếm nhanh chóng.	http://lucene.apache.org/solr/
3	Apache PDFBox	Thư viện cung cấp các hàm xử lý tài liệu PDF.	https://pdfbox.apache.org/
4	LibreOffice API	Nền tảng các phần mềm văn phòng, cung cấp hàm chuyển đổi các định dạng văn bản khác nhau.	https://api.libreoffice.org/
5	Scrapy	Nền tảng giúp thu thập và trích xuất dữ liệu trên internet.	https://scrapy.org
6	MongoDB	Lưu trữ bản ghi không cấu trúc. Phục vụ việc lưu trữ đường dẫn internet về thông tin meta-data.	https://www.mongodb.com/
7	Laravel	Framework PhP, được sử dụng để xây dựng phần Web	https://laravel.com/

5 Cấu hình phần cứng tham khảo

Cấu hình phần cứng phù hợp phụ thuộc vào số lượng người dùng, tần suất sử dụng hệ thống, dung lượng tài liệu được kiểm tra,... Hiện nay hệ thống đang được triển khai trên máy chủ có cấu hình như sau:

- Hệ điều hành: CentOS/Ubuntu Server
- CPU: Intel Xeon E5-2690
- Bộ nhớ: 32GB
- Đĩa cứng: 16TB

6 Tổng kết

Tài liệu này trình bày tổng quan về khía cạnh kỹ thuật của hệ thống DoIT với các nội dung chính bao gồm kiến trúc cơ bản của hệ thống và các bước xử lý chính trong hai chức kiểm lỗi chính tả và phát hiện trùng lặp. Các phần mềm nền tảng và công cụ chính được sử dụng trong hệ thống cũng được liệt kê.