BỘ GIÁO DỤC VÀ ĐÀO TẠO

ĐẠI HỌC ĐÀ NẴNG

NGUYỄN DUY LINH

XÂY DỰNG ỨNG DỤNG PHÁT HIỆN NỘI DUNG GIỐNG NHAU GIỮA CÁC TÀI LIỆU

Chuyên ngành: **Khoa học máy tính**

Mã số: **60.48.01**

LUẬN VĂN THẠC SĨ KỸ THUẬT

Ngợời hợớng dẫn khoa học: PGS.TS. V Tru g H g

Đà Nẵng - Năm 2014

LỜI CAM ĐOAN

Tôi xin cam đoan:

Những nội dung trong luận văn này là do tôi thực hiện dưới sự hướng dẫn trực tiếp của PGS.TS. Võ Trung Hùng.

Mọi tham khảo dùng trong luận văn đều được trích dẫn rõ ràng tên tác giả, tên công trình, thời gian, địa điểm công bố.

Mọi sao chép không hợp lệ, vi phạm quy chế đào tạo, hay gian trá, tôi xin chịu hoàn toàn trách nhiệm.

Tác giả

Nguyễn Duy Linh

MỤC LỤC

DANH MỤC C C TỪ VIẾT TẮT

TIẾNG VIỆT

CSDL	Cơ sở dữ liệu
CNTT	Công nghệ thông tin
KLTN	Khóa luận tốt nghiệp
GVHD	Giảng viên hướng dẫn

TIẾNG ANH

HTML HTTP	HyperText Markup Language HyperText Transfer Protocol
W3C	World Wide Web Consortium
MYSQL	Hệ quản trị CSDL MySql
PHP	Personal Home Page
SQL	Structured Query Language
Admin	Administrator
CSS	Cascading Style Sheet

DANH MỤC CÁC BẢNG

DANH MỤC CÁC HÌNH

MỞ ĐẦU

1. Lý do chọ đề tài

Ngày nay, với sự phát triển vợợt bậc của hệ thống mạng Internet thì việc tìm kiếm thông tin trở nên dễ dàng. Cùng với nó hoạt động trao đổi, chia sẻ tài liệu cũng diễn ra phổ biến. Các bài báo, tài liệu nghiên cứu, báo cáo thực tập, khóa luận tốt nghiệp, luận văn,… đợợc công khai phát tán, chỉnh sửa ngay cả khi không đợợc sự đồng ý của tác giả. Nhiều nhà kinh doanh còn lợi dụng dịch vụ này để kiếm lời thông qua nhu cầu thực tế của ngợời sử dụng tạo tiền đề cho trào lợu "đạo văn" lan rộng.

Phong trào nghiên cứu khoa học của học sinh, sinh viên ngày càng phát triển. Số lợợng học sinh, sinh viên tham gia nghiên cứu khoa học ngày càng nhiều. Vì vậy, để chất lợợng các bài viết, khóa luận, luận văn ngày càng cao và tránh tình trạng "đạo văn" trong nghiên cứu khoa học thì việc xây dựng một công cụ dùng để phát hiện hiện tợợng trên là rất cần thiết.

Trên thế giới, luật pháp đã quy định nhiều khung hình phạt đối với việc vi phạm bản quyền từ rất sớm. Ở Việt Nam, tuy cũng đã có nhiều quy định về vấn đề này nhợng vẫn không hạn chế đợợc việc sao chép, mua bán các tài liệu thông qua mạng Internet.

Những nghiên cứu phát hiện sự trùng lặp chuỗi văn bản đã cho ra đời nhiều công cụ hiệu quả và có thể sử dụng trực tuyến nhợ `Plagiarism Checker Software, Turnitin,...` Những hệ thống này chỉ cho phép phát hiện sự trùng lặp của dữ liệu có trong tên miền gốc và chỉ thực hiện đợợc trực tuyến trên môi trợờng có Internet. Bên cạnh đó, việc mở rộng cơ sở dữ liệu mẫu theo yêu cầu ngợời sử dụng trở nên khó khăn và chi phí rất cao.

Vì vậy chúng tôi quyết định chọn đề tài ***"Xây dựng ứng dụng phát hiện nội dung giống nhau giữa các tài liệu"*** làm đề tài tốt nghiệp luận văn cao

học. Trong đề tài này, chúng tôi đề xuất giải pháp xây dựng ứng dụng dùng để phát hiện sự lặp lại về nội dung của các khóa luận tốt nghiệp, phục vụ công tác nâng cao chất lợợng đào tạo sinh viên tại Trợờng Đại học Quảng Bình.

2. Mục đích ghiê cứu

Mục đích nghiên cứu của đề tài là xây dựng ứng dụng trong đó sử dụng phợơng pháp tạo mô hình đặc trợng cho tập văn bản và các thuật toán so khớp mẫu để phát hiện nội dung giống nhau giữa các khóa luận tốt nghiệp.

3. Đối tợợng và phạm vi nghiên cứu

3.1. Đối tượng nghiên cứu

Đối tợợng nghiên cứu của đề tài là cấu trúc tài liệu dạng văn bản, phợơng pháp và kỹ thuật tách câu tiếng Việt, các thuật toán tìm kiếm và so khớp mẫu.

3.2. Phạm vi nghiên cứu

Trong khuôn khổ của một luận văn, tôi chỉ giới hạn thực nghiệm tạo ứng dụng phục vụ kiểm tra nội dung giống nhau giữa các khóa luận của sinh viên ngành Công nghệ thông tin - Khoa Kỹ thuật - Công nghệ - Trợờng Đại học Quảng Bình.

4. Phợơ g pháp nghiên cứu

Tôi sử dụng hai phợơng pháp chính là phợơng pháp nghiên cứu tài liệu và phợơng pháp thực nghiệm.

Phương pháp tài liệu: Với phợơng pháp này, chúng tôi nghiên cứu các tài liệu về cơ sở lý thuyết: mô hình đặc trợng văn bản tiếng Việt, kỹ thuật tách câu tiếng Việt, các thuật toán tìm kiếm và so khớp mẫu, ngôn ngữ lập trình

PHP; các tài liệu mô tả một số công cụ so khớp văn bản và các tài liệu liên quan đến một số nghiên cứu khác.

Phương pháp thực nghiệm: với phoƚơng pháp này, chúng tôi sử dụng kỹ thuật xây dựng đặc trọng cho tập dữ liệu đầu vào (tập các KLTN) bằng việc sử dụng công cụ tách câu tiếng Việt `vnSentDetector`, xây dựng ứng dụng dựa trên ngôn ngữ PHP và hệ quản trị CSDL MySQL; đồng thời thực nghiệm kiểm tra trên các khóa luận tốt nghiệp của sinh viên ngành Công nghệ thông tin – Trọờng Đại học Quảng Bình và tích hợp ứng dụng lên môi trọờng Internet.

5. Ý ghĩa khoa học và thực tiễn của đề tài

Về khoa học: Kết quả nghiên cứu của đề tài góp phần mở rộng các ứng dụng của kỹ thuật xây dựng mô hình ngôn ngữ tiếng Việt, công cụ vnSentDetector, các thuật toán tìm kiếm và so khớp mẫu.

Về thực tiễn: Đề tài sẽ góp phần nâng cao chất loợng đào tạo sinh viên.

6. Bố cục luậ văn

Báo cáo của luận văn đoợc tổ chức thành 3 choƚơng chính:

Chương 1. Nghiên cứu tổng quan

Trong choƚơng này, chúng tôi trình bày tổng quan về đặc điểm ngôn ngữ tiếng Việt, phoƚơng pháp tách câu trong tiếng Việt, các thuật toán tìm kiếm và so khớp mẫu, giới thiệu một số ứng dụng toƚơng tự.

Chương 2. Đề xuất giải pháp

Choƚơng 2 đoợc dành để trình bày mô hình phát triển và các giải pháp xây dựng ứng dụng. Giải pháp đoợc đề xuất nhoƚ sau: Xây dựng mô hình đặc trọng cho các văn bản trong tập dữ liệu đầu vào (tập các khóa luận tốt nghiệp) dựa trên công cụ tách câu tiếng Việt `vnSentDetector`, ứng dụng thuật toán tìm kiếm và so khớp mẫu Knuth – Morris - Pratt đã đoợc đề xuất ở Choƚơng 1 là phần cốt lõi để xây dựng ứng dụng.

Chương 3. Triển khai ứng dụng

Lựa chọn công cụ phát triển, xử lý tài liệu đầu vào để đoạ vào ứng dụng. Phoọng pháp tạo mô hình đặc trọng cho tập dữ liệu đầu vào. Giới thiệu các boọ́c triển khai, xây dựng các module choọng trình.

CHƯƠNG 1
NGHIÊN CỨU TỔNG QUAN

Trong chương này, chúng tôi trình bày tổng quan về đặc điểm ngôn ngữ tiếng Việt, phương pháp tách câu trong tiếng Việt, các thuật toán tìm kiếm và so khớp mẫu, giới thiệu một số ứng dụng tương tự.

1.1. ĐẶC ĐIỂM CÂU TRONG TIẾNG VIỆT VÀ BÀI TOÁN TÁCH CÂU

1.1.1. Câu và cấu trúc câu tiếng Việt [1]

Câu là một tập hợp từ, ngữ kết hợp với nhau theo những quan hệ cú pháp xác định, được tạo ra trong quá trình tư duy, giao tiếp, có giá trị thông báo, gắn liền với mục đích giao tiếp nhất định. Nói đến cấu trúc câu là nói đến các thành phần tạo câu cùng với chức năng, mối quan hệ qua lại và sự phân bố chúng trong tổ chức nội bộ câu. Dựa vào vai trò tạo câu, các thành phần câu được chia thành ba loại lớn: thành phần nòng cốt, thành phần phụ và thành phần biệt lập.

a. Thành phần nòng cốt của câu

Thành phần nòng cốt là loại thành phần cơ bản, cốt lõi của câu mà dựa vào nó câu mới có thể tồn tại. Thành phần nòng cốt bao gồm hai loại nhỏ: chủ ngữ và vị ngữ.

Chủ ngữ (subject)

Chủ ngữ (viết tắt: C) là loại thành phần nòng cốt có chức năng biểu thị đối tượng mà câu đề cập đến. Nó trả lời cho câu hỏi: câu nói về ai, cái gì, việc gì ?

Về từ loại, chủ ngữ thoịờng do danh từ hay đại từ đảm nhiệm. Một số từ loại khác nhoị động từ, tính từ và số từ cũng có thể làm chủ ngữ.

Về cấu tạo, chủ ngữ có thể là một từ, một chữ chính phụ hay một kết cấu chủ - vị dợới bậc câu (gọi là tiểu cú) tạo thành.

Vị ngữ (Predicate)

Vị ngữ (viết tắt: V) là loại thành phần nòng cốt có chức năng biểu thị nội dung thuyết minh về đối tợợng đợợc câu nói đến. Nó trả lời cho câu hỏi: đối tợợng đợợc nói đến làm gì, nhợ thế nào, ra sao?

Về mặt từ loại, vị ngữ thợờng do động từ hay tính từ đảm nhiệm. Một vài từ loại khác nhợ đại từ, số từ cũng có thể làm vị ngữ.

Về mặt cấu tạo, vị ngữ có thể do một từ, một ngữ hay do một kết cấu chủ vị dợới bậc câu (tiểu cú) tạo thành.

Về trật tự phân bố chủ ngữ, trong câu tiếng Việt, chủ ngữ đứng trợớc vị ngữ là hiện tợợng phổ biến. Tuy nhiên, trong một số trợờng hợp, chủ ngữ có thể đứng sau vị ngữ.

Chủ ngữ và vị ngữ là hai thành phần nòng cốt, nên chúng thợờng xuất hiện

trong câu. Tuy nhiên, hai thành phần này cũng có thể vắng mặt trong một số trợờng hợp:

- C hoặc/và V bị tỉnh lợợc dựa vào hoàn cảnh giao tiếp.

- C hoặc/và V bị tỉnh lợợc dựa vào văn cảnh.

Ngoài một số trợờng hợp vừa nêu, nếu câu thiếu C hoặc/và thiếu V thì đó là câu sai ngữ pháp.

b. Thành phần phụ của câu

Thành phần phụ của câu bao gồm hai loại nhỏ: trạng ngữ và khởi ngữ.

Trạng ngữ

Trạng ngữ (viết tắt: Tr) là loại thành phần phụ có chức năng bổ sung thêm thông tin phụ cho sự việc đợợc kết cấu C - V nòng cốt nêu ra. Thông tin

phụ mà Tr bổ sung có thể là thời gian, nơi chốn, cách thức, phương tiện, trạng thái, đối tượng có liên quan, ...

Về mặt cấu tạo, Tr có thể là một từ, một ngữ có hay không giới từ dẫn nhập, tuỳ vào loại trạng ngữ cụ thể.

Trong trường hợp Tr đứng trước C - V, Tr thường được phân cách với kết cấu C - V bằng dấu phẩy. Trường hợp Tr xen vào giữa hay đứng sau C - V cũng vậy.

Để xác định được những danh ngữ, giới ngữ xen vào giữa hay nằm sau C - V có phải là Tr hay không, ta kiểm tra bằng cách đảo chúng lên đầu câu. Nếu câu văn không thay đổi nghĩa hay không sai, thì đó là Tr.

Khởi ngữ (Tr chỉ chủ đề, đề ngữ)

Khởi ngữ (viết tắt là K) là loại thành phần phụ có chức năng nhấn mạnh một chi tiết nào đó trong sự việc được kết cấu C - V nêu lên. Điểm mà K nhấn mạnh có thể trùng với C, với V hay trùng với một bộ phận nào đó trong V.

Về cấu tạo, K có thể do một từ hay một ngữ tạo thành. Khi K là một ngữ, nó có thể chứa tiểu cú.

Về vị trí, K bao giờ cũng đứng trước C - V và được phân cách C - V bằng dấu phẩy, nếu không có trợ từ thì xen vào.

Về nội dung nghĩa, cần lưu ý rằng, câu bình thường không có K khác với câu có K ở chỗ: câu có K luôn mang một hàm ý nào đó.

c. Các thành phần biệt lập

Thành phần biệt lập là loại thành phần đứng tách riêng ra trong tổ chức câu và có mối quan hệ lỏng lẻo với kết cấu C - V nòng cốt.

Thành phần biệt lập bao gồm nhiều loại nhỏ:

Chuyển ngữ (Tr chuyển tiếp, thành phần phụ chuyển tiếp)

Chuyển ngữ là loại thành phần biệt lập có chức năng xác lập và biểu thị mối quan hệ giữa câu này với câu khác trong chuỗi câu, đoạn văn, … Nói cách khác, chức năng của thành phần này là liên kết câu, tạo nên sự mạch lạc của đoạn văn, ngôn bản.

Về mặt cấu tạo, chuyển ngữ có thể là một từ và bao giờ cũng là quan hệ từ (liên từ, giới từ). Các quan hệ từ thơìờng làm chuyển ngữ là: và, rồi, nhơìng, song, tuy nhiên, vì, bởi vì, nên, cho nên, giữa, với, bằng ... Chuyển ngữ còn có thể do một tổ hợp từ cố định hoá (quán ngữ) hay có xu hơíớng cố định hoá tạo thành. Chẳng hạn nhơí các tổ hợp: mặt khác, trái lại, ngơìợc lại, bên cạnh đó, chẳng hạn nhơí, ví dụ nhơí, do đó, mặc dù vậy, tóm lại, nói tóm lại, …

Về vị trí, chuyển ngữ thơìờng đứng trơíớc kết cấu C - V nòng cốt và đơìợc phân cách bằng dấu phẩy nếu ta tổ hợp. Nếu chuyển ngữ là một từ thì không cần dùng dấu phẩy.

Cảm thán ngữ

Cảm thán ngữ là loại thành phần đặc biệt có chức năng biểu thị các trạng thái cảm xúc đi kèm theo sự kiện đơìợc câu thông báo.

Về cấu tạo, cảm thán ngữ có thể do một từ - từ cảm đảm nhiệm. Cảm thán ngữ cũng có thể do một tổ hợp từ tạo thành.

Về vị trí, cảm thán ngữ có thể đứng đầu câu hay cuối câu. Và ở vị trí nào, nó cũng thơìờng đơìợc tách ra khỏi các thành phần khác bằng dấu phẩy.

Hô ngữ (thành phần gọi - đáp)

Hô ngữ bao gồm hai loại nhỏ: hô ngữ gọi và hô ngữ đáp.

Hô ngữ gọi: là loại thành phần đặc biệt có chức năng biểu thị đối tọợng đọợc ngọời nói gọi đến trong câu.

Về cấu tạo, hô ngữ có thể là một từ, thoịờng là danh từ riêng hay danh từ chung, hay là một tổ hợp gồm danh từ, danh ngữ kết hợp với các từ đệm.

Về vị trí, hô ngữ gọi có thể đứng ở đầu hay ở cuối câu và bao giờ nó cũng đoịợc phân cách khỏi các thành phần khác bằng dấu phẩy.

Hô ngữ đáp: là loại thành phần đặc biệt có chức năng đánh dấu câu trả lời đồng thời biểu thị thái độ, phản ứng của ngoịời nói.

Về cấu tạo, hô ngữ đáp có thể là một từ hay là một tổ hợp từ.

Về vị trí, hô ngữ gọi bao giờ cũng đứng ở đầu luôn đoịợc phân cách khỏi các thành phần khác bằng dấu phẩy.

Giải thích ngữ

Giải thích ngữ là loại thành phần đặc biệt có chức năng giải thích thêm cho một từ ngữ nào đó, hay ghi chú thêm về thái độ, lời lẽ, cảm xúc, ... của ngoịời nói.

Về cấu tạo, giải thích ngữ có thể là một từ, hay là một câu hoàn chỉnh. Trong troịờng hợp giải thích ngữ là một câu, nó còn đoịợc gọi là câu đệm hay câu chêm xen.

Về vị trí, nếu giải thích ngữ có chức năng giải thích, thì nó đứng liền sau từ ngữ đoịợc giải thích. Nếu giải thích ngữ có chức năng ghi chú thêm, thì nó có thể đoịợc xen vào giữa hay đặt ở cuối câu. Và xuất hiện ở vị trí nào, giải thích ngữ cũng phải đoịợc tách khỏi các thành phần khác bằng dấu phẩy, dấu gạch ngang, dấu hai chấm hay dấu ngoặc đơn.

1.1.2. Bài toán tách câu

Cho một văn bản tiếng Việt bất kỳ, hãy phân tách văn bản đó ra thành các đơn vị câu độc lập.

Bài toán tách câu đặt ra với mục đích xây dựng công cụ tự động tách các câu trong một văn bản tiếng Việt bất kỳ một cách chính xác nhất có thể.

Công cụ tách câu `vnSententDetector` của hai tác giả Lê Hồng Phương và Hồ Tường Vinh được xây dựng dựa trên mô hình xác suất với Maximum Entropy [7]. Mô hình này được đào tạo trên tập dữ liệu được xây dựng tập dữ liệu gồm có 4.800 câu tiếng Việt. Bộ dữ liệu này được các nhà ngôn ngữ học thuộc trung tâm từ điển học Việt Nam (Vietlex) xây dựng thủ công bằng tay. Với phương pháp này, theo bài báo mà các tác giả đã công bố thì độ chính xác đạt được 95% [10].

Ý tưởng của phương pháp là xây dựng mô hình xác suất ước lượng lớp *b* xảy ra trong ngữ cảnh *c, p(b,c)*.

$$p(b,c) = \pi \prod_{j=1}^{k} \alpha_j^{f_j(b,c)},$$

Trong đó: $b \in$ {no, yes}, αj là những tham số chưa biết của mô hình và mỗi αj ứng với một đặc trưng mô hình *fj*, π là một hằng số.

Gọi $\mathcal{B}$ = {no,yes} là tập khả năng của các lớp, $\mathcal{C}$ là tập khả năng về các ngữ cảnh. Khi đó các đặc trưng *fj* là hàm nhị phân $f_j = \mathcal{B} \times \mathcal{C} \rightarrow \{0,1\}$. Các hàm này dùng để mã hóa thông tin ngữ cảnh. Xác suất để biết ranh giới câu trong ngữ cảnh *c* được cho bởi *p(yes,c)*. αj được chọn để cực đại hàm likelihood của tập dữ liệu mẫu.

Mô hình sử dụng luật quyết định đơn giản để xác định khả năng ranh

giới câu. Ranh giới hiện tại là khả năng ranh giới câu nếu và chỉ nếu *p(yes,c) >0.5*, trong đó:

$$p(\text{yes}\,|c) = \frac{p(\text{yes}, c)}{p(c)} = \frac{p(\text{yes}, c)}{p(\text{yes}, c) + p(\text{no}, c)}$$

và *c* là ngữ cảnh có chứa khả năng là ranh giới câu.

Một phần quan trọng của phoịơng pháp là lựa chọn các đặc troịng *fj*. Các đặc troịng của mô hình Maximum Entropy có thể mã hóa bất kỳ thông tin nào có ích cho việc xác định các ranh giới câu. Các khả năng ranh giới câu đoịợc xác định bằng cách quét văn bản theo các chuỗi ký tự đoịợc ngăn cách bởi kí tự trắng mà trong đó có chứa một trong các ký hiệu ".", "!" hoặc "?" [4].

1.2. THUẬT TOÁN TÌM KIẾM VÀ SO KHỚP MẪU

Giải thuật so sánh chuỗi là quá trình tìm kiếm tất cả các lần xuất hiện của một chuỗi mẫu (pattern) trong một chuỗi khác. Quá trình so sánh chuỗi nhoị thế này là hoạt động diễn ra rất thoịờng xuyên trong các choịơng trình chỉnh sửa văn bản, các trình duyệt web, các máy tìm kiếm, … Các giải thuật này còn đoịợc sử dụng trong việc tìm các mẫu trong chuỗi ADN.

Cho *T[1..n]* là một chuỗi bao gồm n ký tự, trong đó các *T[i], 1<=i<=n* là từng ký tự ở trong chuỗi. Cho *P[1..m]* là chuỗi mẫu bao gồm m ký tự, *m<=n*. Ta giả sử rằng *P* và *T* chỉ chứa các ký tự có trong tập hữu hạn *S*. Ví dụ *S={0, 1}* hoặc *S={a, b, c…, z}*. Vấn đề đặt ra là tìm xem *P* có xuất hiện trong *T* hay không. Hay nói cách khác là tìm số nguyên *s (0<s<n)* sao cho *T[s+1..s+m] = P[1..m]*. Khi đó, ta nói *P* xuất hiện trong *T* với độ dịch chuyển *s*. Nếu *P* thực sự xuất hiện trong *T* với độ dịch chuyển *s*, ta gọi *s* là độ dịch chuyển hợp lệ, ngoịợc lại ta gọi *s* là độ dịch chuyển không hợp lệ.

Cho chuỗi *T[1..n],* một chuỗi con của *T* đoịợc định nghĩa là *T[i..j]* với *1<=i, j<=n*. Chuỗi con này chứa các ký tự từ chỉ số *i* đến chỉ số *j* của mảng

các ký tự trong *T*. Lọu ý rằng *T* cũng chính là một chuỗi con của *T* với *i=1, j=n.*

Một chuỗi con thực sự của chuỗi *T[1..n]* là chuỗi *T[i..j]* với *i<j* và *(i>0 hoặc j<n)*. Trong trọờng hợp *i > j* thì *T[i..j]* là một chuỗi rỗng. Tiền tố của một chuỗi *T[1..n]* là chuỗi *T[1..i]* với *1<=i<=n*. Hậu tố của một chuỗi *T[1..n]* là chuỗi *T[j..n]* với *1<=j<=n*

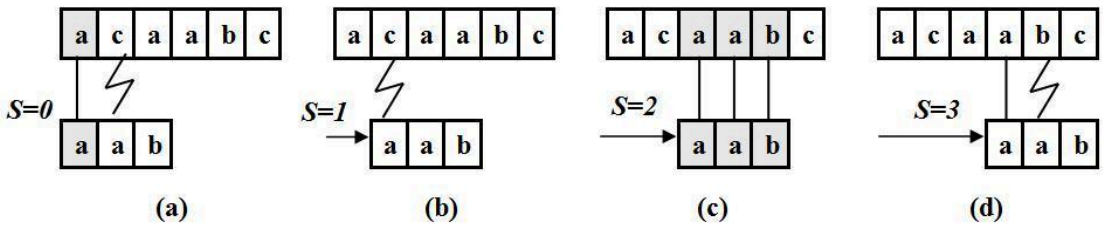
Chúng tôi tìm hiểu về 3 giải thuật cơ bản nhất trong so sánh chuỗi đó là: Naïve, Rabin - Karp, Knutt - Morris - Pratt.

1.2.1. Naïve

Đây là giải thuật cơ bản và đơn giản nhất, sử dụng nguyên lý vét cạn. Giải thuật này kiểm tra tất cả các khả năng của chuỗi mẫu *P[1..m]* nằm trong chuỗi *T[1..n]* bằng cách duyệt từ đầu tới cuối chuỗi *T*.

NAIVE-STRING-MATCHER(T, P)

```
1. n = T.length
2. m = P.length
3. for s = 0 to n-m do
4. j = 1
5. while (j <= m AND T[s+j] == P[j]) do
6. j = j+1
7. if (j > m)
8. "Tìm thấy mẫu với độ dịch chuyển s"
```



Hình 1.1. Minh họa giải thuật Naïve-String-Matcher

Phân tích: vòng lặp *while* bên trong chạy tối đa *m* lần, vòng lặp *for* bên ngoài chạy tối đa *n-m+1* lần. Do vậy, thời gian chạy của giải thuật này là *T(n) = O((n-m+1)*m) = O(n*m)*. Rõ ràng, giải thuật này không hiệu quả vì nó bỏ qua mọi thông tin hữu ích có đoợc trong quá trình so sánh chuỗi tại từng giá trị của *s*. Giải thuật Knuth - Morris - Pratt đoợc trình bày trong các phần sau tỏ ra tốt hơn nhiều so với Naïve vì tận dụng các thông tin hữu ích khi tìm kiếm.

1.2.2. Thuật toán Rabin - Karp

Thuật toán này do Rabin và Karp đề xuất [13]. Thuật toán tiêu tốn *O(m)* để tiền xử lý các dữ liệu nhập và thời gian chạy chậm nhất của nó là *O((n-m+1)m)*. Mặc dù vậy trung bình các troợng hợp đều tiêu tốn thời gian ít hơn.

Ta nhận thấy rằng mỗi chuỗi *S* cấu tạo từ *S* đều có thể số hóa thành 1 số đoợc . Ví dụ *S = {0,1,2..,9}*, *S = "1234"* thì ta sẽ có *digit(S) = 1,234*. Gọi *p* là giá trị số hóa của *P*, hay nói cách khác *p* là giá trị thập phân toợng ứng của *P*. Gọi t_s là giá trị thập phân toợng ứng của *T[s+1,...,s+m], s<n-m+1*. Ta nhận thấy rằng $t_S = p$ khi và chỉ khi *P = T[s+1,...,s+m]*.

Mặt khác, ta có thể tính *p* và t_0 theo 2 công thức :

$p = P[m]*10^0 + P[m-1]*10^1 + ...+ P[1]*10^{m-1}$.

$t_0 = T[1]*10^{m-1} + T[2]*10^{m-2} + ...+ T[m]*10^0$.

Ta nhận thấy rằng qua hai công thức trên ta sẽ phải tiêu tốn *O(m)* cho mỗi công thức.

Sau khi tính t_0 , việc tính các $t_1, t_2, ...t_{n-m-1}$ trở nên đơn giản hơn với và chỉ

tiêu tốn *O(1)* cho mỗi t_i mà thôi. Ta tính các $t_1, t_2, \ldots . t_{n-m-1}$ lần lượt theo công thức sau:

$$t_i = 10*(t_{i-1} - 10^{m-1}*T[i]) + T[i+m].$$

Sau khi tính đoợc các giá trị của p và t_i, bài toán so sánh chuỗi trở nên đơn giản vô cùng khi đoợc qui về bài toán "tìm một số trong một mảng số các số nguyên" - nghĩa là tìm sự xuất hiện của p trong t_i . Hay nói cách khác, bài toán tìm chuỗi qui về bài toán tìm i với $i \hat{I} [0,n-m-1]$ sao cho $p = t_i$.

Vì vậy để tính đoợc tất cả các giá trị p và t_i , hay nói cách khác là tìm đoợc chuỗi P trong T, ta chỉ cần tiêu tốn thời gian $O(m) + O(n-m-1)$ mà thôi. Và điều này cũng cho ta một kết quả khá tốt với thuật toán Rabin - Karp với độ phức tạp là $O(m)$ cho tiền xử lí và $O(n-m-1)$ để so sánh chuỗi. Thế nhoợng vấn đề sẽ phát sinh khi ta cài đặt nó lên bộ nhớ máy tính nơi mà p và t_i bị giới hạn trong kiểu (long) chỉ có 16 chữ số ↔ $Max(m) = 16$.

**** Cải tiến Rabin - Karp***

Để giải quyết troờng hợp $m>16$, p và t_i voợt quá các kiểu dữ liệu của máy tính ta sử dụng mảng băm (hash). Điều này có nghĩa là thay vì tính p và t_i ta sẽ tính p' và t_i' mà ở đó $p' = p \bmod q$, $t_i' = t_i \bmod q$, với q là một số nguyên tố lớn nằm trong khoảng mà máy tính có thể biểu diễn đoợc nhoị là long, integer,…

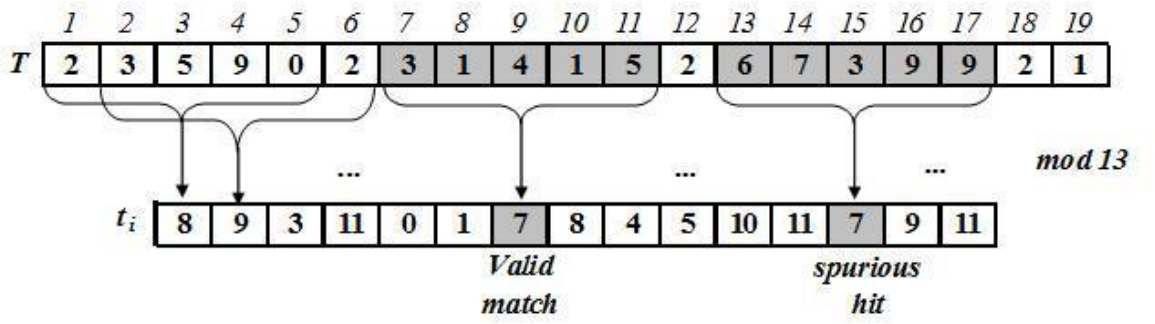
Lúc này, việc so sánh chuỗi sẽ qui về việc so sánh các số t'_i và p'. Nếu nhoị $t'_i \neq p'$ thì điều này đồng nghĩa với việc $P \neq T[i+1,i+m]$.

Chú ý rằng:

- $p' = t'_i$ ***không*** *đồng nghĩa với việc P <> T[i+1,i+m] (hình vẽ)*

- *Số q càng lớn thì xác xuất trường hợp $p' = t'_i$ và P <> T[i+1,i+m] càng thấp.*

- *Khi $p' = t_i'$ để kết luận ta cần phải kiểm tra lại việc P và T[i+1,i+m] có bằng nhau hay không.*



Hình 1.2. Minh họa giải thuật Rabin - Karp

**** Tổng quát cho hệ cơ số d***

Cũng phải chú ý một điều khác là không phải lúc nào *S* cũng là tập hợp các chữ số trong cơ số 10. Xét trọờng hợp *S* là tập hợp các chữ số của hệ cơ số *d*. Lúc này t'_i cần đọợc tính lại nhọ sau :

$t'_i = (d(t'_{i-1} - T[i]h) + T[i+m]) \bmod q$. Với $h = d^{m-1} \bmod q$

Trong hệ cơ số d, p và t_0 cũng đọợc tính lại :

$p = P[m] * d^0 + P[m-1]*d^1 + ...+ P[1]*d^{m-1}$

$t_0 = T[1]*d^{m-1} + T[2]*d^{m-2} + ... + T[m]*d^0$

Sau khi tổng quát hóa ta viết lại thuật toán Rabin - Karp nhọ sau :

```
RABIN-KARP-MATCHER(T, P, d, q)
```

```
1.  n ← length[T ]
2.  m ← length[P]
3.  h ← dm-1 mod q
4.  p' ← 0
5.  t' 0 ← 0
6.  for i ← 1 to m // Tiền xử lý
```

```
7.  p' ← (dp' + P[i ]) mod q
8.  t'0 ← (dt'0 + T [i ]) mod q
9.  for s ← 0 to n – m // so sánh
```

```
10. if p' == t's
11. if P[1 . .m] == T [s + 1 . . s + m]
12. print "Mẫu xuất hiện với độ dịch chuyển" s
13. if (s < n – m)
14. t's+1 ← (d(t's – T [s + 1]h) + T [s + m + 1]) mod q
```

Phân tích:

- Quá trình tiền xử lý tiêu tốn *Q(m)* thời gian với một vòng lặp *for i =1 to n.*

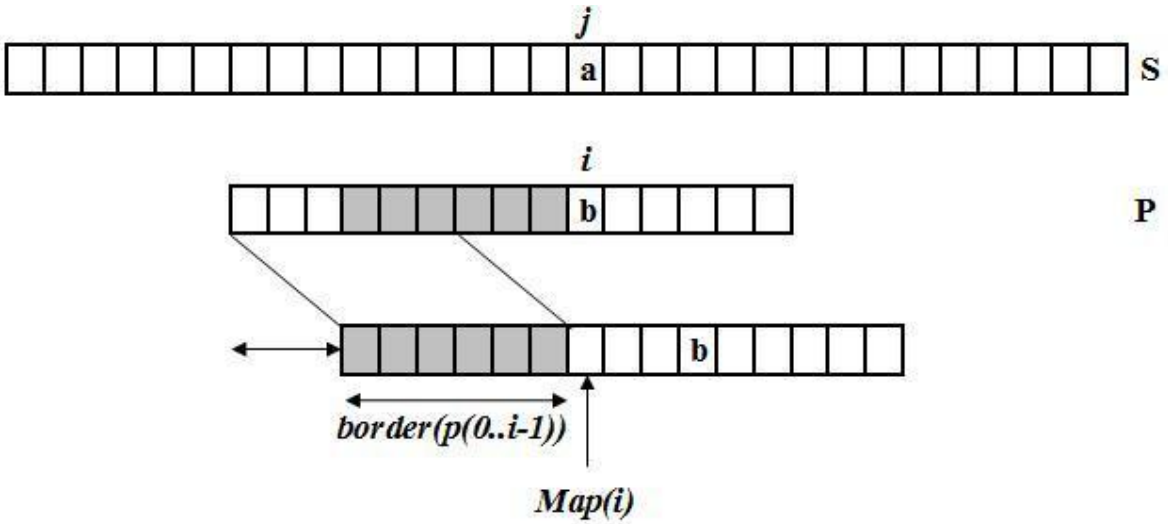
- Quá trình so sánh trong trọờng hợp tốt nhất là *p' != t'i* với mọi *i* thì việc so sánh tiêu tốn *O(n-m).* Nhọng bù lại trong trọờng hợp xấu nhất khi *p' == t'I* thì việc so sánh phải thực hiện thêm lệnh kiểm tra *P[1..m]* và *T[i+1,i+m],* điều này chỉ tiêu tốn *O(m)* thời gian chạy.

Tóm lại: độ phức tạp của Rabin - Karp là *O((n-m+1)m).*

1.2.3. Thuật toán Knuth - Morris - Pratt

Giải thuật có độ phức tạp tuyến tính này đọợc Knuth, Morris và Pratt phát hiện ra nhờ việc phân tích chặt chẽ giải thuật Naïve. Giả sử ta muốn tìm chuỗi mẫu *P[1..m]* trong *T[1..n],* đến một lúc nào đó thì ta sẽ có *P[i] != T[j].*

Nếu dùng thuật toán Naïve thì ta dịch *P* sang phải một vị trí. Nhọng vì ta đã so sánh đến *T[j]* nên ta tìm cách dịch *P* đi càng xa càng tốt. Cách tốt nhất là dịch *P* sang phải một đoạn sao cho tiền tố của *P[1...i]* xếp trùng với một đoạn hậu tố của *T[1..j].* Khi đó, chỉ cần so sánh *T[j]* và *P[k]* (với *P[1..k]* là tiền tố của *P* trùng với hậu tố của *T[1..j]*) mà không cần phải làm lại từ đầu. Ta gọi chuỗi vừa là tiền tố, vừa là hậu tố của chuỗi *x* là biên của *x.*



Hình 1.3. Cách xác định biên trong giải thuật Knuth – Morris - Pratt

Nếu gọi *p[i]* là biên có độ dài lớn nhất của chuỗi *P[1..i]* thì khi đó tại vị trí *P[i]* và *T[j]* khác nhau, ta sẽ dịch *P* sang phải một đoạn *I - p[i]*. Trong trường hợp tốt nhất *p[i] = 0* thì ta sẽ dịch chuyển *P* sang phải một đoạn m. Giá trị các *p[i]* sẽ được tính toán trước. Hình bên dưới liệt kê tất cả các giá trị *p[i]* trong chuỗi mẫu *P=abababababca* cho trước.

i	1	2	3	4	5	6	7	8	9	10
$P[i]$	a	b	a	b	a	b	a	b	c	a
$\pi[i]$	0	0	1	2	3	4	5	6	0	1

Hình 1.4. Giai đoạn tiền xử lý trong giải thuật Knuth – Morris - Pratt

*** Cách xây dựng mảng p:**

Định lý: Nếu *r, s* là biên của chuỗi *x* mà $|r| < |s|$ thì *r* là biên của *s*.

Định nghĩa: Cho *x* là một chuỗi và *c* là một ký tự. Biên *r* của *x* có thể được mở rộng thành *rc* nếu nhưng *rc* là biên của *xc*.

Trong quá trình tiền xử lý chuỗi *P*, mỗi *p[i] (với 1 <= i <= m)* lọu lại độ dài của biên rộng nhất của *P[1..i]*. Vì chuỗi rỗng không có biên nên ta gán: *p[0] = -1*. Giả sử các giá trị *p[0], ..., p[i]* đã biết, giá trị *p[i+1]* sẽ đọợc tính

bằng cách kiểm tra xem biên của chuỗi *P[1..i]* có thể đợợc mở rộng bằng ký tự *P[i+1]* hay không. Ta sử dụng biến k lợu trữ các *p[i]*. Nếu *P[i+1] = P[k]* thì ta gán *p[i+1] = k+1*, ngợợc lại ta xét *k = p[k]* và quay lại các bợớc so sánh *P[i+1]* với *P[k]* ở trên.

Giải thuật so khớp chuỗi KMP-Matcher đợợc trình bày trong đoạn mã giả sau đây. Giải thuật này gọi tới giải thuật tiền xử lý Compute-Prefix-Function để tính *p*.

KMP-MATCHER(T, P)

```
1.    n ← length[T ]
2.    m ← length[P]
3.    π ← COMPUTE-PREFIX-FUNCTION(P)
4.    q ← 0 //Số lượng ký tự trùng nhau
5.    for i ← 1 to n //Duyệt chuỗi T từ trái qua phải
6.    do while q > 0 and P[q + 1] ≠ T [i ]
7.    do q ← π[q] //Ký tự không trùng nhau
8.    if P[q + 1] = T [i ]
9.    then q ← q + 1 //Ký tự trùng nhau
10.   if q = m //Nếu đã kiểm tra toàn bộ chuỗi P
11.   then print "Mẫu xuất hiện với độ dịch chuyển" i - m
12.   q ← π[q] //Tìm ký tự trùng nhau tiếp theo
```

COMPUTE-PREFIX-FUNCTION(P)

```
1.    m ← length[P]
2.    π[1] ← 0
```

```
3.    k ← 0
4.    for q ← 2 to m do
5.    while k > 0 and P[k + 1] ≠ P[q]
```

```
6.    do k ← π[k]
7.    if P[k + 1] = P[q]
8.    then k ← k + 1
9.    π[q] ← k
10.   return π
```

Đánh giá: độ phức tạp của giải thuật tiền xử lý Compute-Prefix-Function là *O(m)* bởi vì vòng lặp *while* bên trong sẽ không bao giờ thực hiện quá *m* lần. Tương tự, giải thuật tìm kiếm KMP-Matcher cũng chỉ có độ phức tạp là *O(n)*.

Bởi vì *m* <= *n* nên độ phức tạp cuối cùng của giải thuật Knuth – Morris - Pratt là *O(n)*.

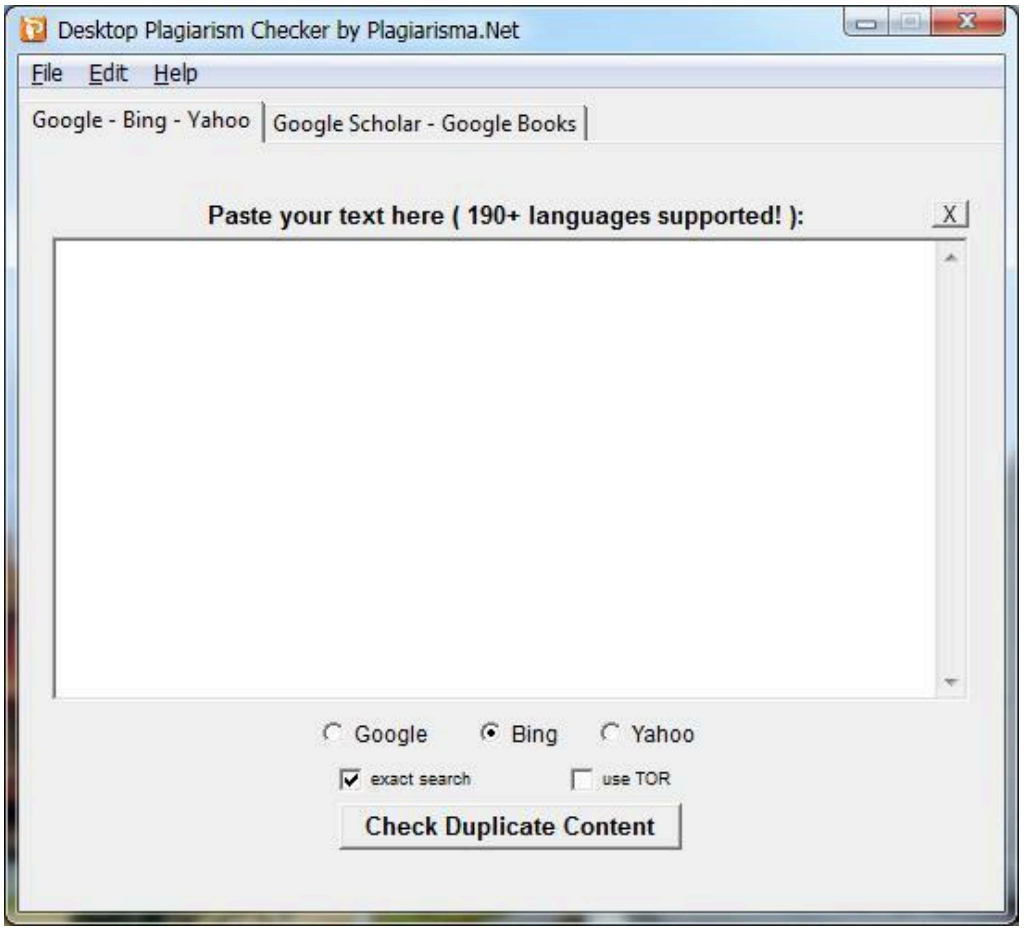
1.3. HỆ THỐNG PHẦN MỀM PLAGIARISM CHECKER SOFTWARE

1.3.1. Giới thiệu

Plagiarism Checker Software là một sản phẩm của Plagiarisma.Net (trang chủ tại https://plagiarisma.net). Với phần mềm này, chúng ta có thể kiểm tra được những tài liệu của mình có trùng lặp hoặc sao chép từ các tài liệu khác được đăng tải trên các trang mạng hay không. Phần mềm miễn phí này cung cấp tìm kiếm trên các công cụ tìm kiếm phổ biến như Google, Bing, Yahoo!, … Phần mềm cũng sẽ hữu ích cho các blogger, những người có thể muốn kiểm tra nếu bài viết của họ đã được sao chép hoặc ăn cắp ý tưởng của người khác.

1.3.2. Cách sử dụng

Giao diện của Plagiarism Checker Software như sau:



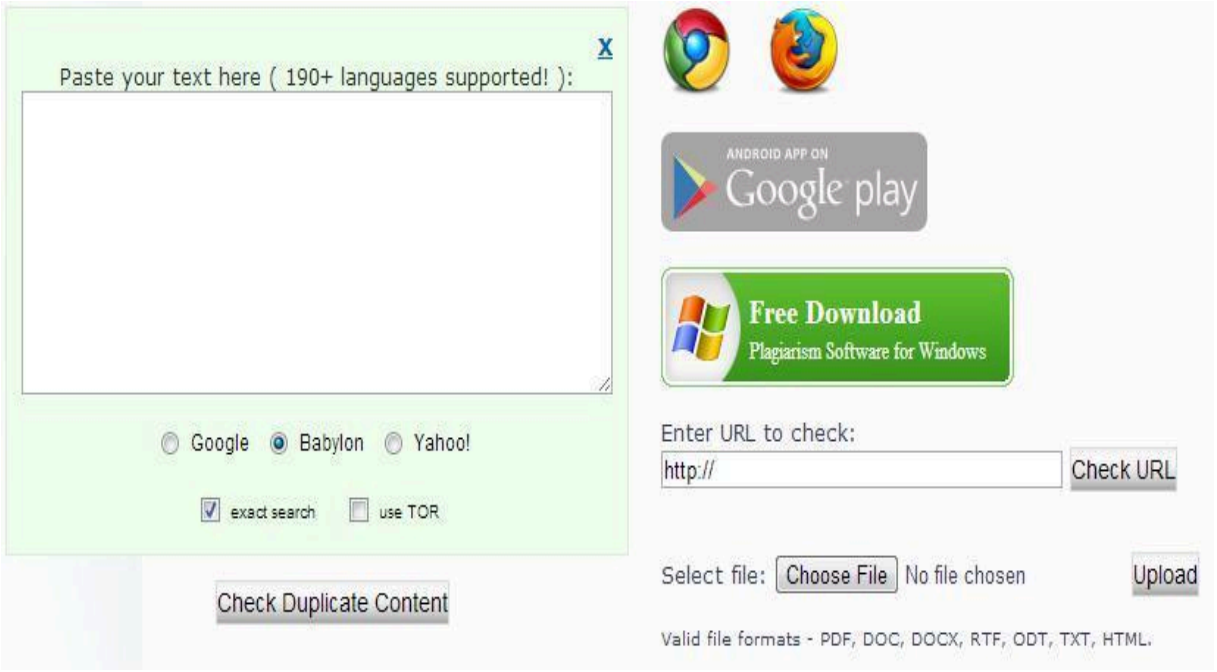
Hình 1.5. Giao diện của Plagiarism Checker Software

Để thực hiện kiểm tra chỉ cần dán nội dung đợợc kiểm tra vào một trong hai thẻ của chợơng trình và chọn bất kỳ một trong các công cụ tìm kiếm nói trên để tìm kiếm nội dung trùng lặp trên Internet.

Ngoài ra, chúng ta cũng có thể kiểm tra trực tiếp trên trang http://plagiarisma.net/. Ở đây, nó hỗ trợ ngợời dùng có thể tải một tập tin văn bản (doc, txt, htm, pdf, odt, rtf, ...) vào chợơng trình để tìm kiếm các nội dung

của tập tin thay vì click vào tab 

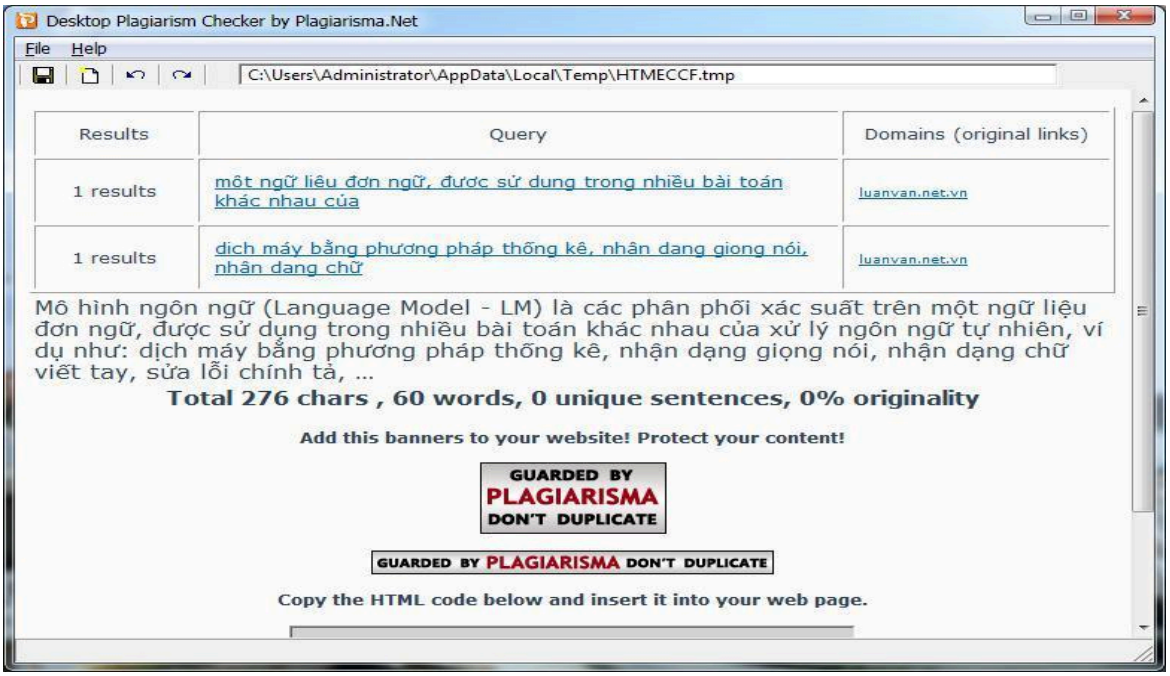
(kiểm tra nội dung trùng lặp) để thực hiện một tìm kiếm nhanh chóng cho bất kỳ nội dung trùng lặp.



Hình 1.6. Giao diện web của Plagiarism Checker Software

Mỗi câu đoợc phân tích kỹ loỡng và một khi tìm kiếm đoợc kết quả hoàn thành đoợc hiển thị trong một cửa sổ choơng trình mới.

Ví dụ:



Hình 1.7. Kết quả so khớp với Plagiarism Checker Software

1.3.3. Ưu điểm

Plagiarism Checker Software có một số ưu điểm:

- Tránh được hiện tượng đạo văn và phát hiện trùng lặp nội dung.

- Hỗ trợ nhiều ngôn ngữ.

- Kiểm tra sự độc đáo của nội dung.

- Tương thích với hệ điều hành Windows.

1.3.4. Nhược điểm

Bên cạnh những ưu điểm đã nêu thì Software còn có một số nhược điểm sau đây:

- Chỉ có các tên miền gốc được hiển thị trong cửa sổ kết quả kiểm tra.

- Chương trình chỉ có thể thực hiện kiểm tra trực tuyến cho nội dung trùng lặp.

1.4. TỔNG KẾT CHƯƠNG

Chương 1 đã tập trung nghiên cứu sâu để làm rõ lý thuyết về đặc điểm câu tiếng Việt, thuật toán tách câu, các thuật toán tìm kiếm và so khớp mẫu, một số ứng dụng tương tự tạo tiền nền tảng để phân tích thiết kế hệ thống ứng dụng.

CHƯƠNG 2
PHÂN TÍCH HỆ THỐNG ỨNG DỤNG

Chương 2 được dành để phân tích hiện trạng đào tạo tại Trường Đại học Quảng Bình, trình bày mô hình phát triển và các giải pháp xây dựng ứng dụng. Giải pháp được đề xuất như sau: xây dựng mô hình đặc trưng cho các văn bản trong tập dữ liệu đầu vào (tập các khóa luận tốt nghiệp) dựa trên công cụ tách câu tiếng Việt `vnSentDetector`, ứng dụng thuật toán tìm kiếm và so khớp mẫu Knuth – Morris - Pratt đã được đề xuất ở Chương 1 là phần cốt lõi để xây dựng ứng dụng.

2.1. HOẠT ĐỘNG ĐÀO TẠO TẠI TRƯỜNG ĐẠI HỌC QUẢNG BÌNH

2.1.1. Phân tích hiện trạng đào tạo ở Trường Đại học Quảng Bình

Trường Đại học Quảng Bình được thành lập theo Quyết định số 237/QĐ-TTg ngày 24/10/2006 của Thủ tướng Chính phủ trên cơ sở Trường CĐSP Quảng Bình mà tiền thân là Trường Trung cấp Sư phạm Quảng Bình được thành lập từ năm 1959. Đây là trường đại học duy nhất của tỉnh Quảng Bình, đào tạo đa ngành, đa cấp, đa lĩnh vực. Với hơn nửa thế kỷ xây dựng và phát triển, Trường Đại học Quảng Bình đã trải qua nhiều giai đoạn với nhiều thăng trầm khác nhau.

Ở trường Đại học Quảng Bình có 2 hệ đào tạo chính là: - Hệ Đại học: Thời gian đào tạo 4 – 4,5 năm

- Hệ Cao đẳng: Thời gian đào tạo 3 năm

Sau khi tốt nghiệp hệ cao đẳng sinh viên có thể học lên cao hơn với các hệ đào tạo liên thông và liên kết với các cơ sở đào tạo trong nước.

Hầu hết sinh viên theo học các khoa trong Trường Đại học Quảng Bình phải làm báo cáo thực tập cuối khóa và khóa luận tốt nghiệp trước khi ra

trọờng. Với yêu cầu các khóa luận năm sau phải khác các khóa luận của các năm trọớc đó do vậy số lọợng KLTN cũng tọơng đọơng với số sinh viên đạt điểm làm KLTN. Đây là con số tọơng đối lớn yêu cầu GVHD phải cập nhật dữ liệu từ các KLTN của các năm trọớc để đối chiếu và gợi ý đề tài cho sinh viên không bị trùng lặp về nội dung. Tuy nhiên, rất khó kiểm soát đọợc hiện tọợng trùng ý tọởng, nội dung giữa các KLTN nếu cứ tiến hành kiểm tra bằng phọơng pháp thủ công. Bởi vậy, nó đòi hỏi sự nổ lực cố gắng của đội ngũ giảng viên và nhà trọờng nhằm khơi dậy sự say mê sáng tạo trong nghiên cứu khoa học của sinh viên. Nghiên cứu và xây dựng thành công ứng dụng kiểm tra nội dung giữa các tài liệu (cụ thể là các KLTN, báo cáo thực tập tốt nghiệp, …) sẽ phần nào nâng cao ý thức tìm tòi nghiên cứu của sinh viên nhà trọờng.

2.1.2. Quá trình làm khóa luận tốt nghiệp của sinh viên *a. Giảng viên hướng dẫn giao đề tài cho sinh viên*

Giảng viên họớng dẫn định họớng cho sinh viên lựa chọn lĩnh vực mà sinh viên muốn nghiên cứu. Sau khi thống nhất đọợc phọơng án giữa GVHD và sinh viên GVHD sẽ lập danh sách mục các đề tài, sinh viên thực hiện gửi Bộ môn duyệt chuyển phòng Đào tạo ký ban hành cho tất cả sinh viên biết để thực hiện.

b. Sinh viên thực hiện đề tài

Sinh viên tiến hành thực hiện đề tài theo trình tự các bọớc sau:

Bọớc 1: Làm đề cọơng sơ bộ

Bọớc 2: Nghiên cứu, phân tích

Bọớc 3: Hoàn thành báo cáo KLTN

Bọớc 4: Nộp KLTN lên Bộ môn

c. Đánh giá khóa luận tốt nghiệp

Trưởng bộ môn phân công giảng viên phản biện, thành lập hội đồng đánh giá KLTN gồm những người có chuyên môn sâu về lĩnh vực mà sinh viên nghiên cứu, ấn định ngày tiến hành đánh giá. Sinh viên thực hiện theo lịch đã phân công và tiến hành bảo vệ KLTN của mình trước hội đồng.

2.1.3. Quy trình kiểm tra thủ công khóa luận tốt nghiệp

Thông thường để kiểm tra khóa luận tốt nghiệp giáo vụ khoa thường thực hiện theo các cách sau đây:

Cách thứ nhất:

Bước 1: Xếp khóa luận riêng cho từng khối ngành.

Bước 2: Chuẩn bị nguồn khóa luận cũ đã lưu trữ trước đây.

Bước 3: Lần lượt dò tên của KLTN mới và KLTN cũ.

Bước 4: Tiến hành lặp lại cho đến khi hết số KLTN mới cần kiểm tra.

Trong quá trình kiểm tra nếu thấy tên các KLTN có sự trùng lặp thì tiến hành kiểm tra nội dung bên trong.

Kết luận: Với cách kiểm tra này thì tốn nhiều thời gian, công sức và hiệu quả không cao.

Cách thứ hai:

Bước 1: Xếp khóa luận riêng cho từng khối ngành.

Bước 2: Nhập tên tất cả các KLTN cũ vào bảng tính Microsoft Excel.

Bước 3: Nhập tên các KLTN mới tiếp theo sau các KLTN cũ.

Bước 4: Sử dụng chức năng sắp xếp trong Microsoft Excel (Data/Sort) để sắp xếp toàn bộ dữ liệu đã nhập. Lúc này, các KLTN cũ và mới đan xen nhau.

Kết thúc sắp xếp giáo vụ khoa sẽ đoıa ra đánh giá, kết luận từ đó đi đến kiểm tra nội dung nếu các KLTN thuộc nhóm có tên gần nhau nhất.

Kết luận: Với cách kiểm tra này thì hao phí điện năng, tốn thời gian, công sức và hiệu quả cũng không cao.

2.2. PHÂN TÍCH NHU CẦU

Chỉ cần so sánh hai văn bản với nhau đã là rất khó nên việc so sánh một văn bản với nhiều văn bản khác càng khó khăn hơn gấp nhiều lần. Một KLTN có quy định ít nhất cũng phải từ 50 trang văn bản trở lên với nội dung trong cùng chuyên ngành nên việc trùng lặp nội dung là không thể tránh khỏi. Với trách nhiệm của một ngoời GVHD họ sẽ đọc hết từng trang KLTN rồi so sánh tỉ mỉ mới đoa ra kết luận có sao chép từ các nguồn khác hay không. Đây là một công việc không dễ thực hiện.

Với việc kiểm tra thủ công nhoị trên có những nhoợc điểm sau đây:

- Sự tốn kém về mặt thời gian: rất cao
- Độ phức tạp khi thực hiện: rất cao
- Độ chính xác: không cao
- Sự tốn kém về mặt nhân lực: rất lớn

Cứ mỗi năm học kết thúc, số loợng KLTN lại tăng dần lên. Không thể dám chắc đoợc rằng các KLTN không sao chép từ các troờng khác cùng chuyên ngành, từ các tỉnh thành khác trên cả noớc, từ các Website mua bán trực tuyến hoặc các đề tài noớc ngoài thực hiện đoợc dịch sang tiếng Việt.

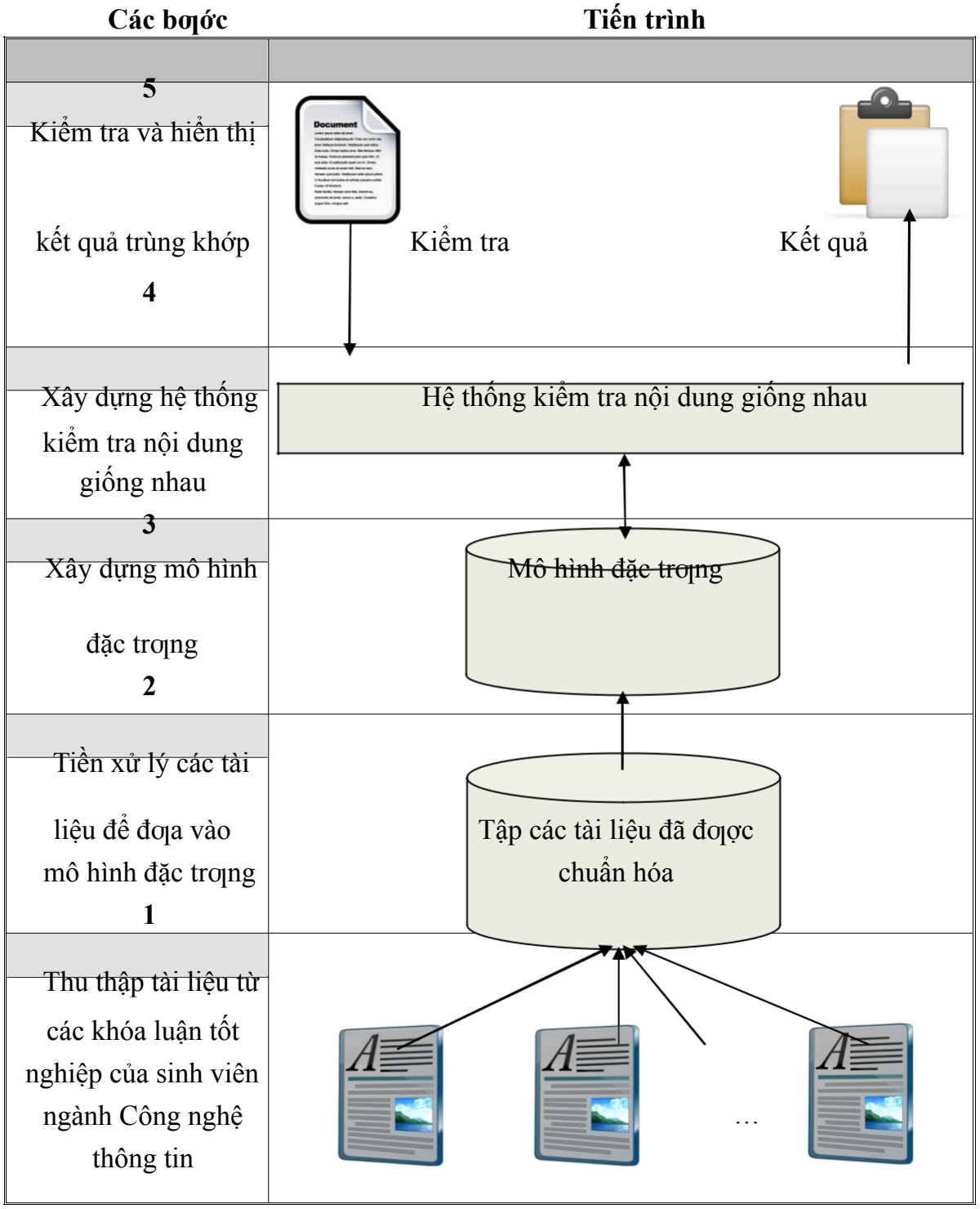
Do đó, cần có một choơng trình kiểm tra sự sao chép nhanh chóng, khoa học và có độ chính xác cao.

2.3. GIỚI THIỆU HỆ THỐNG

Nhoị đã đề cập ở phần troịớc, với sự phát triển của công nghệ thông tin đặc biệt là mạng Internet thì việc phát tán, sao chép nội dung ý toịởng đã diễn

ra rất phổ biến. Việc sử dụng một số phần mềm nhơ Plagiarism Checker Software để kiểm tra nội dung các tài liệu cũng có một vài hiệu quả. Tuy vậy, nó còn hạn chế ở việc tìm ra đơợc những tên miền gốc chứa nội dung tài liệu cần kiểm tra và phải dựa vào một số công cụ tìm kiếm phổ biến nhơ Google, Bing, Yahoo!, … Do đó, mục tiêu của đề tài nghiên cứu này là xây dựng đơợc một ứng dụng nhằm kiểm tra nội dung trùng nhau giữa các tài liệu ngay trên máy tính cá nhân và phát triển tích hợp lên Internet để mở rộng phạm vi tìm kiếm. Mặt khác, kết quả tìm kiếm sẽ đơợc hiển thị chi tiết hơn cho thấy mức độ giống nhau giữa các tài liệu cần kiểm tra thay vì hiển thị tên miền gốc nhơ phần mềm đã giới thiệu.

2.4. MÔ HÌNH TỔNG QUÁT HỆ THỐNG



Hình 2.1. Mô hình tổng quát hệ thống

Kiến trúc tổng thể của hệ thống bao gồm những thành phần sau:

- *Bộ sưu tập tài liệu:* Soạu tập các nguồn tài liệu từ các khóa luận tốt nghiệp của sinh viên ngành Công nghệ thông tin, Khoa Kỹ thuật – Công nghệ, Trợờng Đại học Quảng Bình.

- *Tiền x l :* Là hoạt động nhằm chuẩn hóa dữ liệu đầu vào theo quy định đầu ra phù hợp với CSDL yêu cầu. Những hoạt động này có thể: chuyển đổi định dạng phông chữ, loại bỏ các thành phần không cần thiết (hình ảnh, biểu đồ, bảng biểu,...), chuyển đổi cấu trúc, kiểm tra tính đúng đắn của dữ liệu,… Ở bợớc này trong đề tài thì việc xử lý bằng phợơng pháp thủ công, chuẩn hóa dữ liệu trợớc khi đợa vào kho. Việc chuẩn hóa dữ liệu là việc chuyển đổi định dạng dữ liệu thành định dạng tợơng thích với mục đích của hệ thống.

- *Xây dựng mô hình đặc trưng (với đơn vị là câu)*: Sử dụng công cụ tách câu vnSentDetector để tách câu từ tập dữ liệu đầu vào (tập các KLTN) và thống kê tập các câu trùng nhau.

- *Xây dựng hệ thống kiểm tra nội dung giống nhau:* Xây dựng ứng dụng nhằm phát hiện nội dung giống nhau giữa tài liệu cần kiểm tra và tập tài liệu

đã đợợc chuẩn hóa trong mô hình đặc trợng.

- *Kiểm tra và hiển thị kết quả trùng khớp:* Là thành phần sau cùng của hệ thống. Nó giúp cho ngợời dùng kiểm tra xem tài liệu của mình có trùng nội dung với những tài liệu khác trong CSDL hay ko từ đó có những điều chỉnh hợp lý phù hợp với mục đích sử dụng.

2.5. THUẬT TOÁN SỬ DỤNG

2.5.1. Giai đoạn xây dựng tập dữ liệu

Mục đích: Tạo mô hình đặc trợng cho tập các KLTN bao gồm:

- Thống kê tổng số câu đoợc đã đoợc xây dựng trong tập CSDL.
- Nội dung các câu.

- Tần số xuất hiện của nó trong bộ sưu tập các KLTN.

Đầu vào: bộ sưu tập các KLTN

Đầu ra: mô hình đặc trưng cho từng KLTN trong bộ sưu tập các KLTN

Xử lý:

- **Bước 1:** Sưu tầm các tài liệu chủ yếu là các khóa luận của sinh viên ngành Công nghệ thông tin – Trường Đại học Quảng Bình.

- **Bước 2:** Tiền xử lý.

Ở giai đoạn này thực hiện các công việc như:

* Loại bỏ các nội dung không cần thiết từ tập tài liệu đã sưu tầm ở bước 1.

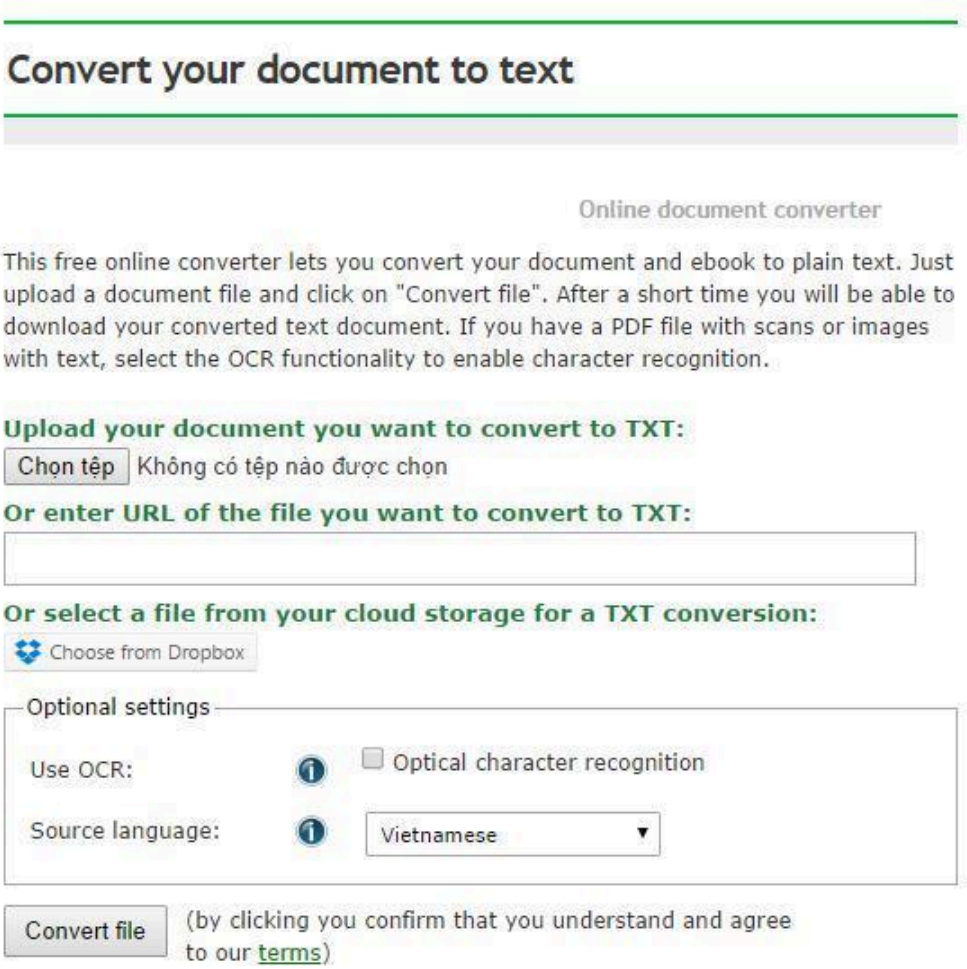
* Chuyển từ định dạng tệp văn bản *.doc sang tệp văn bản dạng *.txt bằng công cụ trên Website http://www.online-convert.com.

Cách chuyển đổi định dạng tệp văn bản:

Sau khi truy cập vào Website, chọn menu **Document converter/Convert to TXT.**

Hình 2.2. Menu Document converter

Ở phần nội dung, lựa chọn tệp cần chuyển đổi ở nút **Chọn tệp**, lựa chọn ngôn ngữ là **Vietnamese** và thực hiện chuyển đổi bằng việc nhấn nút **Convert file.** Sau đó, chương trình sẽ xuất hiện hộp thoại cho phép chọn nơi lưu tệp đã chuyển đổi thành công.

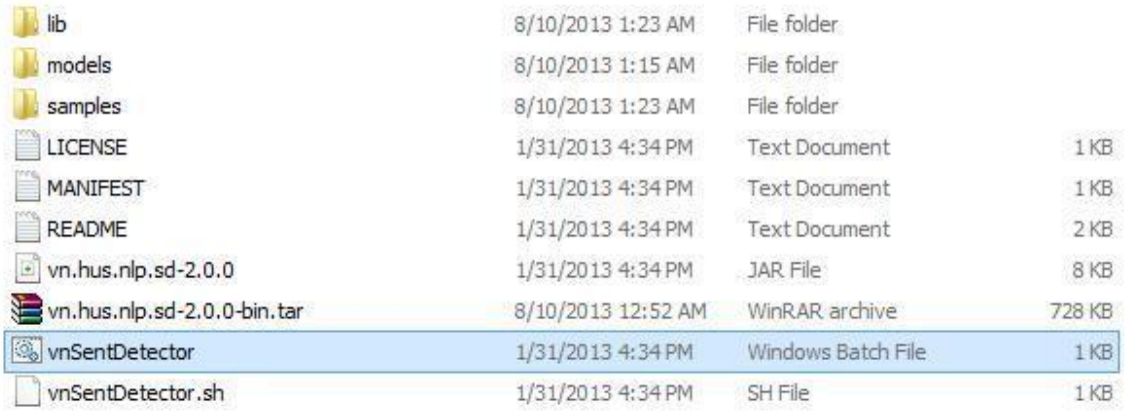


Hình 2.3. Giao diện website chuyển đổi tệp

Ưu điểm của ứng dụng chuyển đổi tệp này là hỗ trợ ngôn ngữ tiếng Việt nên quá trình chuyển đổi tệp không ảnh hưởng tới nội dung của tệp.

* Tách câu sử dụng công cụ của hai tác giả Lê Hồng Phương và Hồ Tường Vinh được xây dựng dựa trên mô hình xác suất với Maximum Entropy bằng ngôn ngữ Java.

Cấu trúc của công cụ `vnSentDetector` như sau:



Hình 2.4. Cấu trúc của công cụ tách câu vnSentDetector

Cách sử dụng công cụ `vnSentDetector`:

Trong hệ điều hành Unix/Linux, sử dụng file "vnSentDetector.sh" để chạy chương trình còn trong hệ điều hành Microsoft Windows sử dụng file `"vnSentDetector.bat"`.

Chương trình này là một công cụ tách câu của văn bản tiếng Việt, nó không có giao diện đồ họa người dùng (GUI). Để có kết quả tách câu cần cung cấp hai đối số cho chương trình:

- Một tệp văn bản cần tách câu sau tùy chọn `-i` (một tập tin mã hóa UTF-8).

Một tệp văn bản có chứa kết quả của chương trình sau đây tùy chọn `-o`.

Để thực thi chương trình cần sử dụng của sổ cmd: Run/cmd và nhập cấu trúc lệnh để tách câu vào cửa sổ lệnh đó.

Ví dụ: `vnSentDetector.sh -i samples/test0.txt -o samples/test0.sd.txt`

Ở ví dụ trên thì tệp văn bản đầu vào là: `test0.txt`, tệp văn bản kết quả đầu ra là `test0.sd.text` đã đợợc tách thành các câu và mỗi câu đợợc ghi trên 1 dòng trong tệp văn bản.

- **Bợớc 3:** Tạo mô hình đặc trợng bộ sợu tập các KLTN.

Thực hiện chuyển tất cả các câu đã đoợc tách ra bằng công cụ vnSententDetector trong tệp văn bản (*.txt) vào mảng 1 chiều. Sau đó duyệt tất cả các phần tử của mảng cần xây dựng tập dữ liệu nếu có phần tử trùng nhau thì tăng biến đếm lên 1 đơn vị và lặp cho đến khi hết phần tử cuối cùng trong mảng.

Giải thuật tổ g quát hoq sau:

```
BEGIN
Ti ề n xử lý
Đưa vào 1 KLTN cần để xây dựng tập CSDL(dạng File
text) Dem:= 0
n:= số phần tử a[i]
m:= số phần b[j] (mảng đã được xây dựng trong tập dữ
liệu)
a[i]:=KLTN
For i:=1 to n do
For j:=1 to m do
If a[i]=a[j] then dem:dem+1;
END.
```

2.5.2. Giai đoạn so khớp

Mục đích: liệt kê những câu đoợc sao chép từ các KLTN và đánh giá mức độ giống giữa các KLTN.

Đầu vào: KLTN cần đánh giá.

Đầu ra:

- Hiển thị những câu giống nhau từ các KLTN.

- Mức độ giống nhau của KLTN cần đánh giá với KLTN đã đọợc xây dựng trong tập dữ liệu.

Xử lý:

- **Bước 1:** Chuẩn bị tài liệu cần đánh giá.
- **Bước 2:** Tiền xử lý (Thực hiện tương tự bước 2 ở giai đoạn xây dựng tập dữ liệu).
- **Bước 3:** Xây dựng đặc trưng cho KLTN cần đánh giá.
- **Bước 4:** Đánh giá nội dung giống nhau của các KLTN.

Đánh giá các câu của KLTN này với các câu của tài liệu có trong cơ sở dữ liệu đã được xây dựng ở giai đoạn xây dựng tập dữ liệu. Kết thúc giai đoạn này chúng tôi đưa ra kết quả đánh giá với 2 tiêu chí cơ bản là: những câu của KLTN cần đánh giá đã sao chép từ KLTN nào? Mức độ giống nhau cao của các câu trong KLTN này với những KLTN nào?

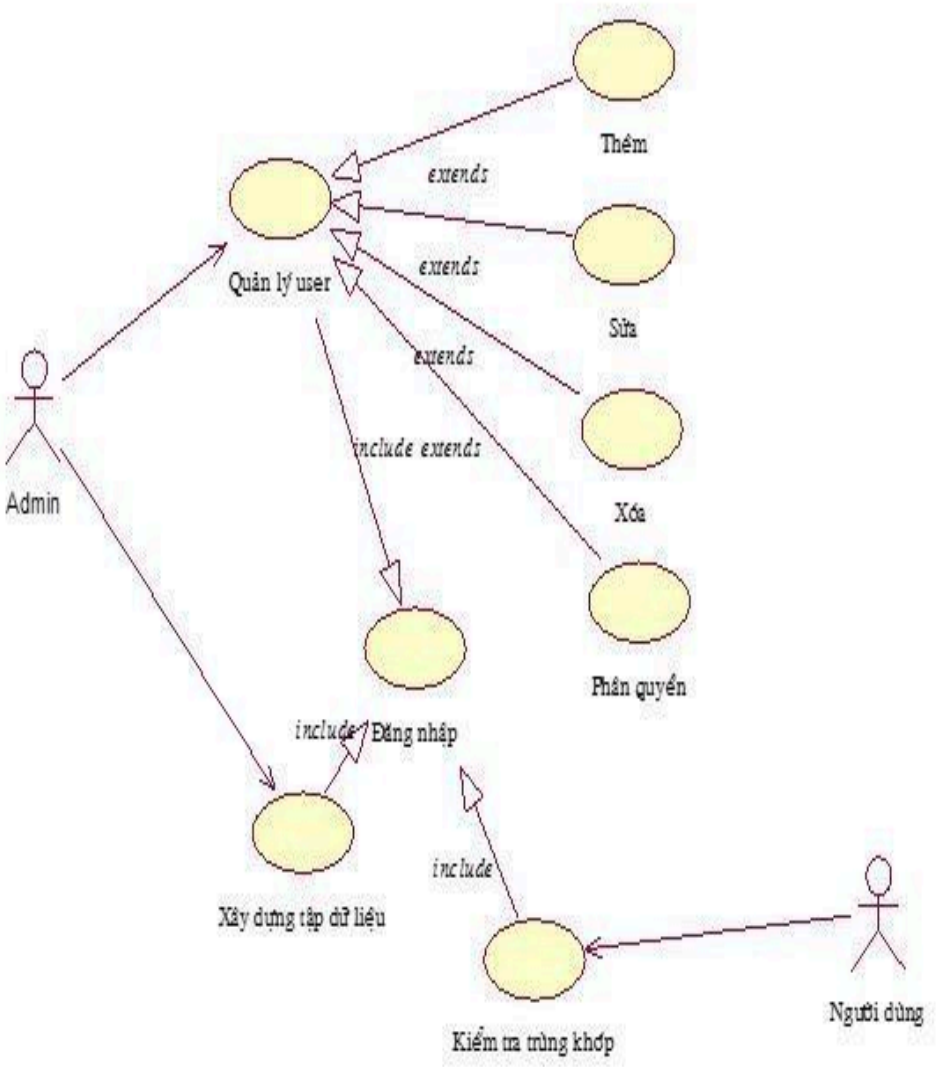
Giải thuật tổng quát như sau:

```
BEGIN
Tiền xử lý
Đưa vào 1 KLTN cần được xây dựng tập dữ liệu (dạng File text)
n:= số phần tử a[i]
m:= số phần b[j] (mảng đã được xây dựng trong tập dữ liệu)
a[i]:=KLTN
 For i:=1 to n do
For j:=1 to m do
If a[i]=a[j] then
Thông báo câu trùng từ
Else
```

Sử dụng thuật toán Knuth-Morris-Pratt để xác định câu gần giống.

END.

2.6. THIẾT KẾ MÔ HÌNH



Hình 2.5. Mô hình use case tổng quát

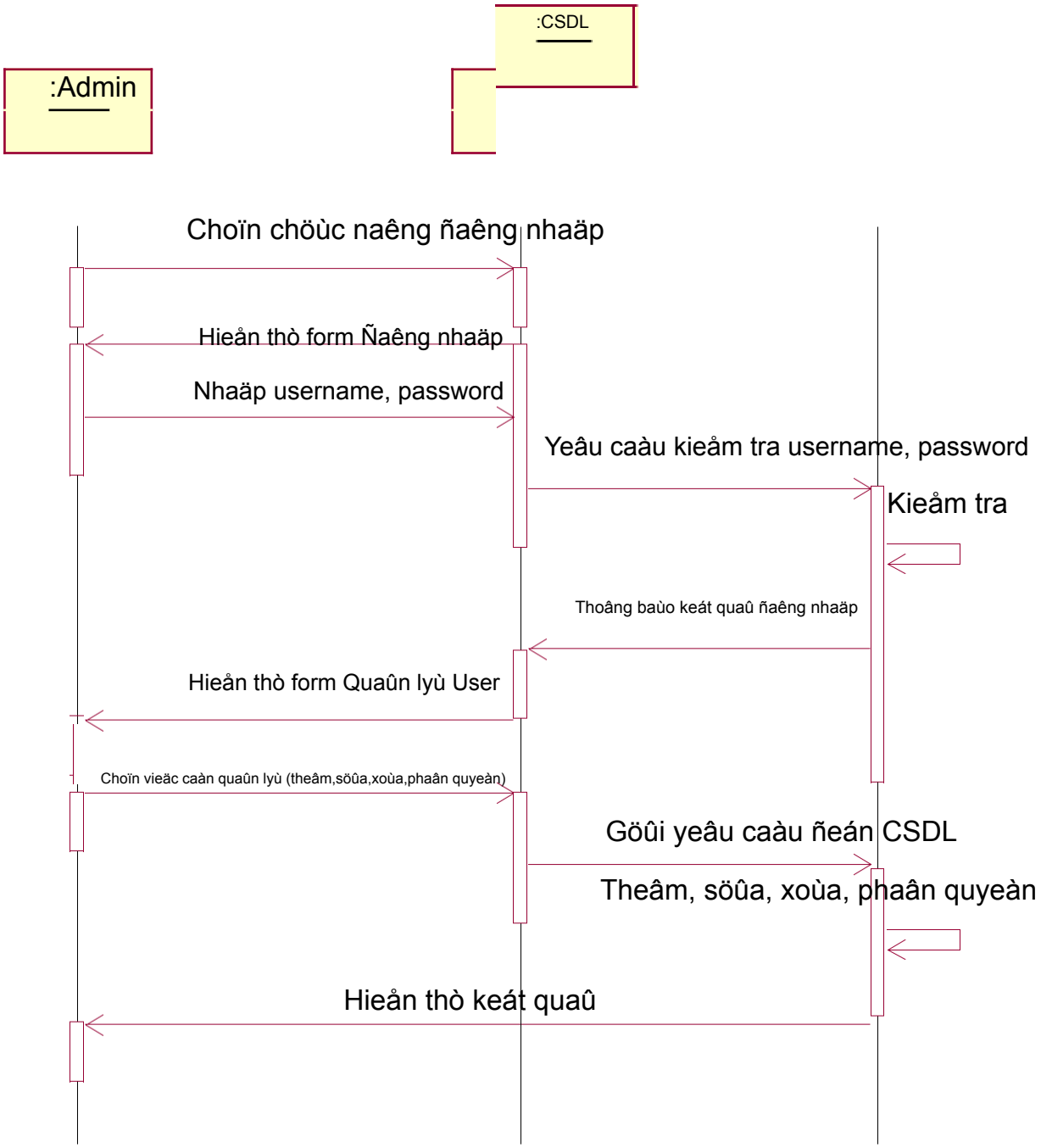
2.6.1. Chức ă g Quản lý User

a. ***Kịch bản "Quản lý User"***

Bảng 2.1. Kịch bản "Quản lý User"

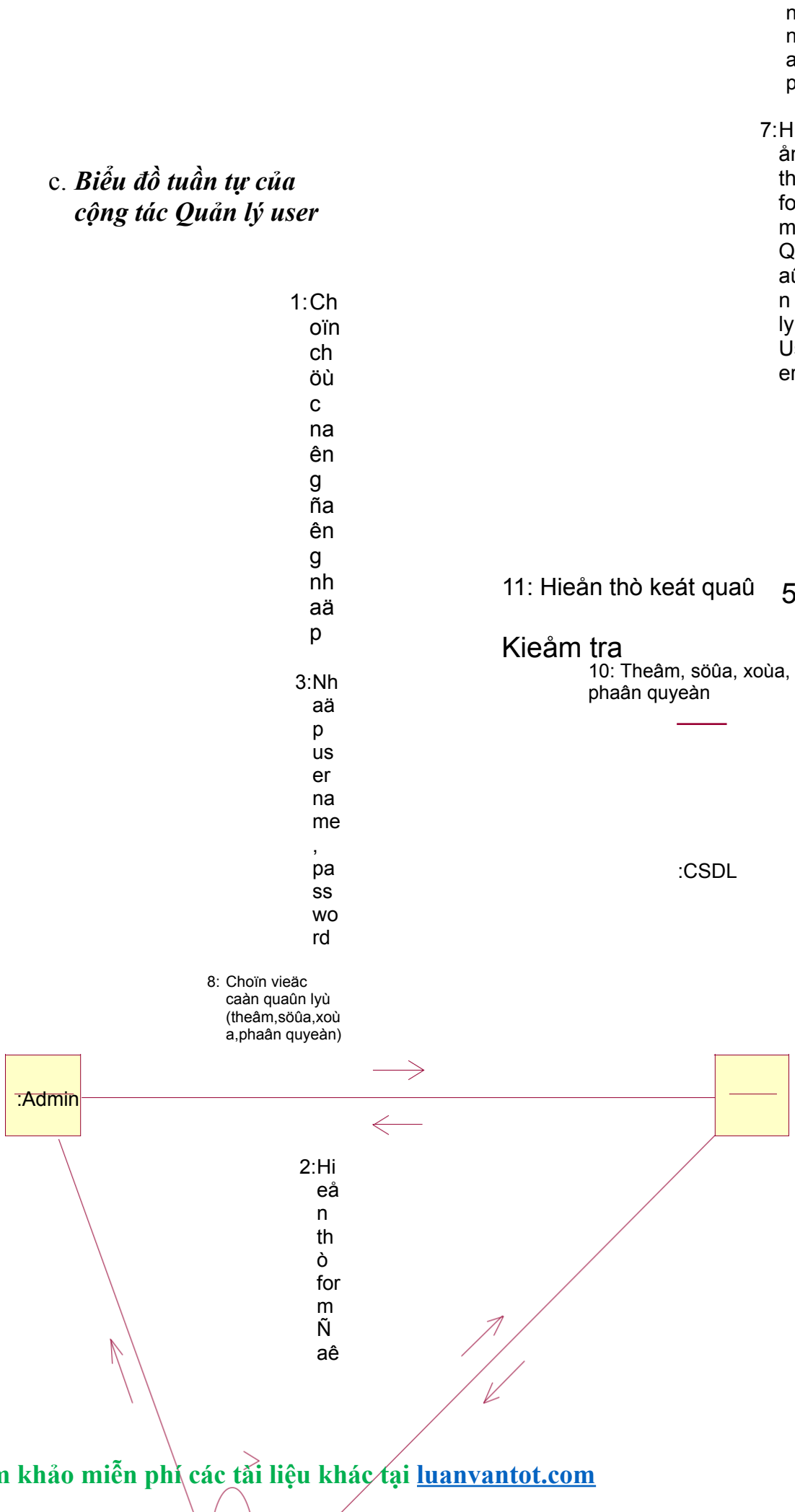
Hành động của tác nhân	*Hành động của hệ thống*
1. Chọn chức năng đăng nhập.	
	2. Hiển thị Form đăng nhập .
3. Nhập thông tin username, password.	
4. Yêu cầu kiểm tra thông tin username và password.	
	5. Kiểm tra và thông báo kết quả đăng nhập.
	6. Hiển thị form Quản lý User
7. Chọn việc cần quản lý (Thêm, Sửa, Xóa, Phân quyền).	
8. Gửi yếu cầu đến CSDL	
	9. Thực hiện Thêm, Sửa, Xóa, Phân quyền và hiển thị kết quả.

b. Biểu đồ tuần tự của chức năng Quản lý user



Hình 2.6. Biểu đồ tuần tự của chức năng Quản lý user

c. ***Biểu đồ tuần tự của cộng tác Quản lý user***



:Form

Hình 2.7. Biểu đồ tuần tự của cộng tác Quản lý user

2.6.2. Chức ă g xây dựng tập dữ liệu

a. Kịch bản "xây dựng tập dữ liệu"

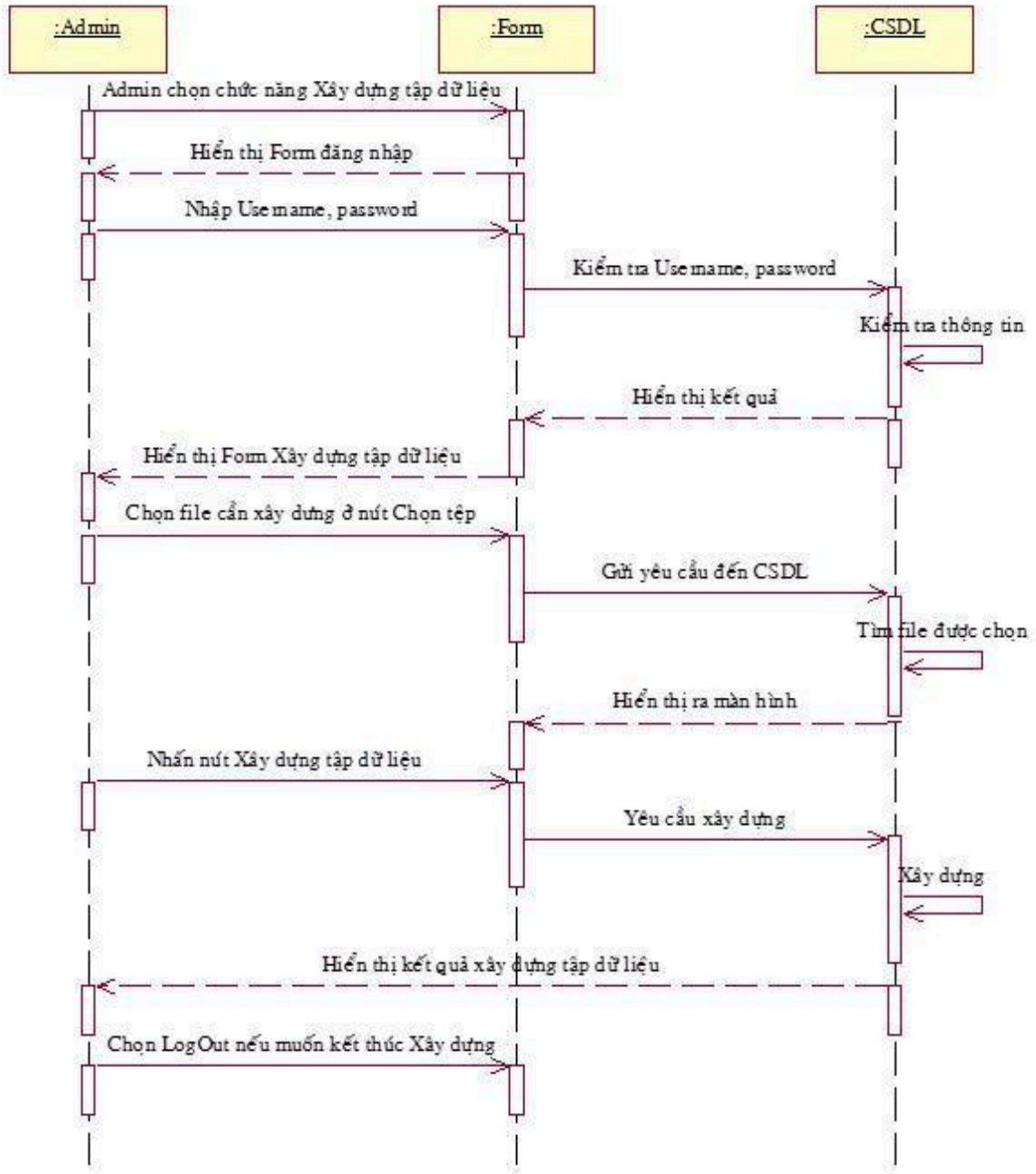
Bảng 2.2. Kịch bản "xây dựng tập dữ liệu"

Hành động của tác nhân	*Hành động của hệ thống*
1. Admin chọn chức năng xây dựng tập dữ liệu.	
	2. Hiển thị Form Đăng nhập .
3. Nhập username, password.	
4.Yêu cầu kiểm tra username, password.	
	5. Kiểm tra thông tin và hiển thị kết quả.
	6. Hiển thị Form xây dựng tập dữ liệu.
7. Chọn tệp cần xây dựng tập dữ liệu ở nút Chọn tệp.	
8. Gửi yêu cầu đến CSDL.	
	9. Tìm tệp đọợc chọn và hiển thị ra màn hình.
10. Nhấn nút xây dựng tập dữ liệu dữ liệu.	
11. Yêu cầu xây dựng tập dữ liệu.	
	12. Xây dựng tập dữ liệu và hiển thị kết quả xây dựng tập dữ liệu.

13. Chọn Logout nếu muốn kết thúc xây dựng tập dữ liệu.

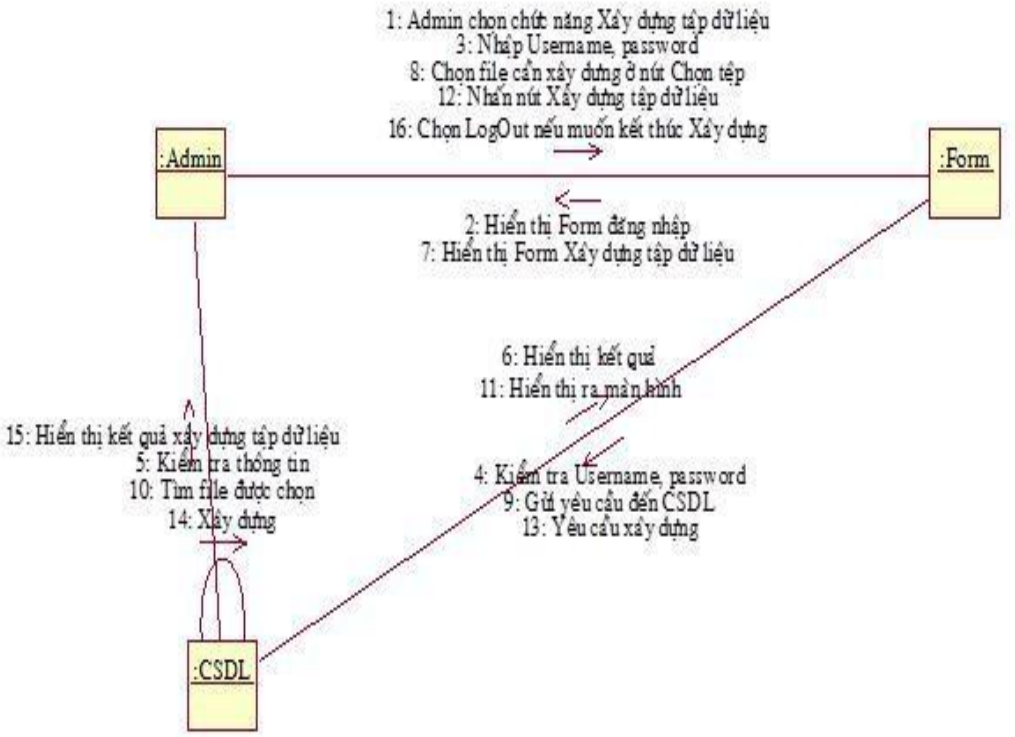
---

b. Biểu đồ tuần tự của chức năng xây dựng tập dữ liệu



Hình 2.8. Biểu đồ tuần tự của chức năng xây dựng tập dữ liệu

c. Biểu đồ cộng tác của chức năng xây dựng tập dữ liệu



Hình 2.9. Biểu đồ cộng tác của chức năng xây dựng tập dữ liệu

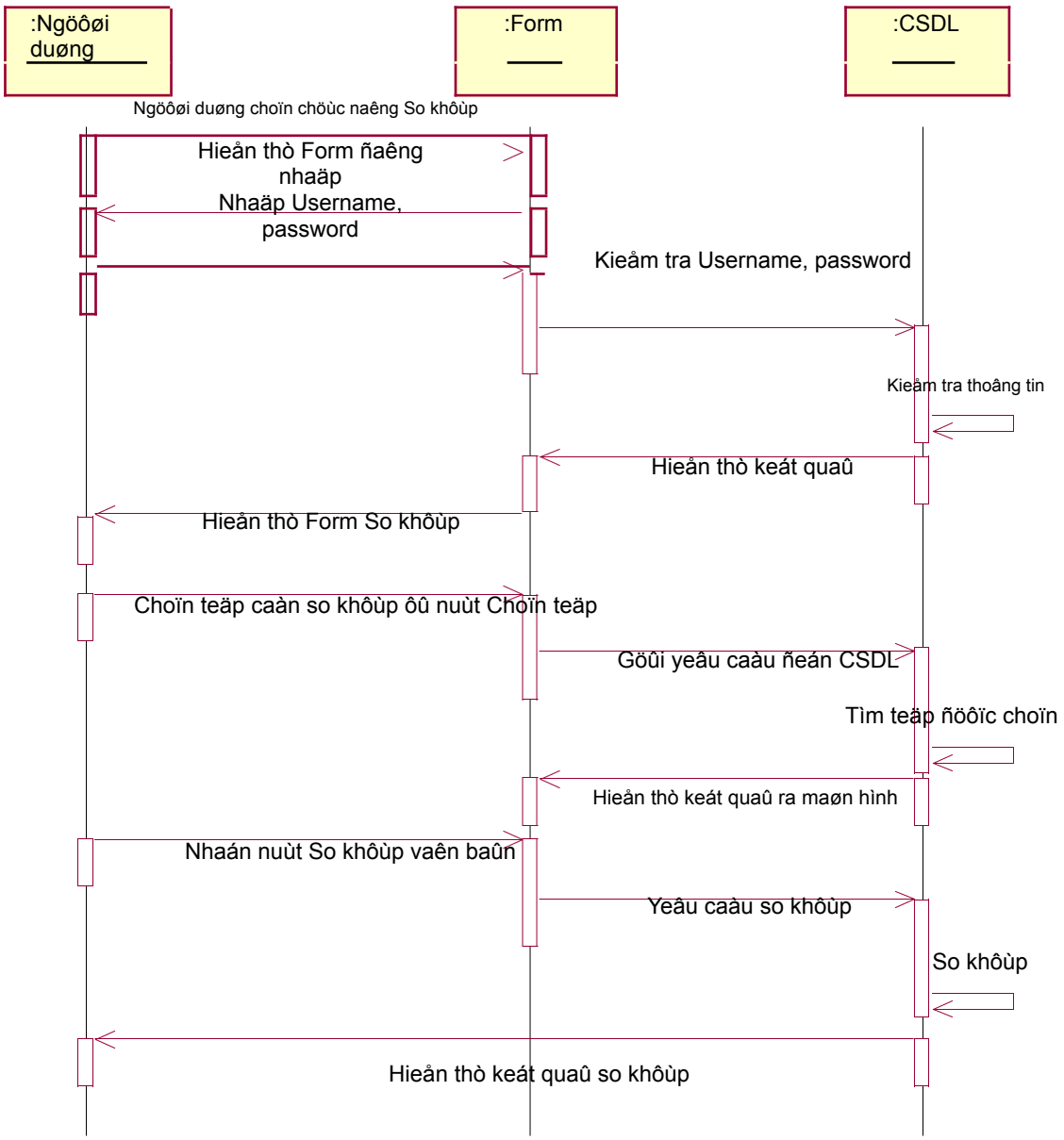
2.6.3. Chức ă g so khớp

a. Kịch bản "so khớp"

Bảng 2.3. Kịch bản "so khớp"

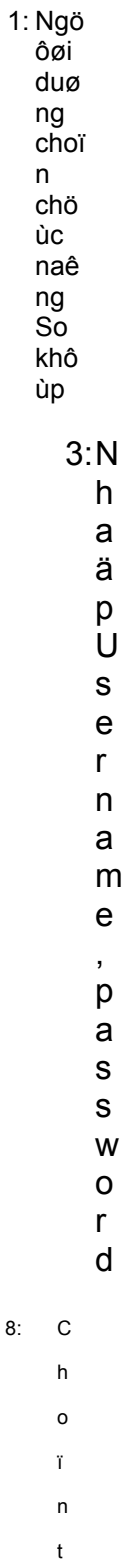
Hành động của tác nhân	*Hành động của hệ thống*
1. Ngọời dùng chọn chức năng So khớp.	
	2. Hiển thị Form Đăng nhập .
3. Nhập username, password.	
4.Yêu cầu kiểm tra username, password.	
	5. Kiểm tra thông tin và hiển thị kết quả.
	6. Hiển thị Form So khớp.
7. Chọn tệp cần so khớp ở nút Chọn tệp.	
8. Gửi yêu cầu đến CSDL.	
	9. Tìm tệp đọợc chọn và hiển thị ra màn hình.
10. Nhấn nút So khớp văn bản.	
11. Yêu cầu so khớp.	
	12. So khớp và hiển thị kết quả so khớp.

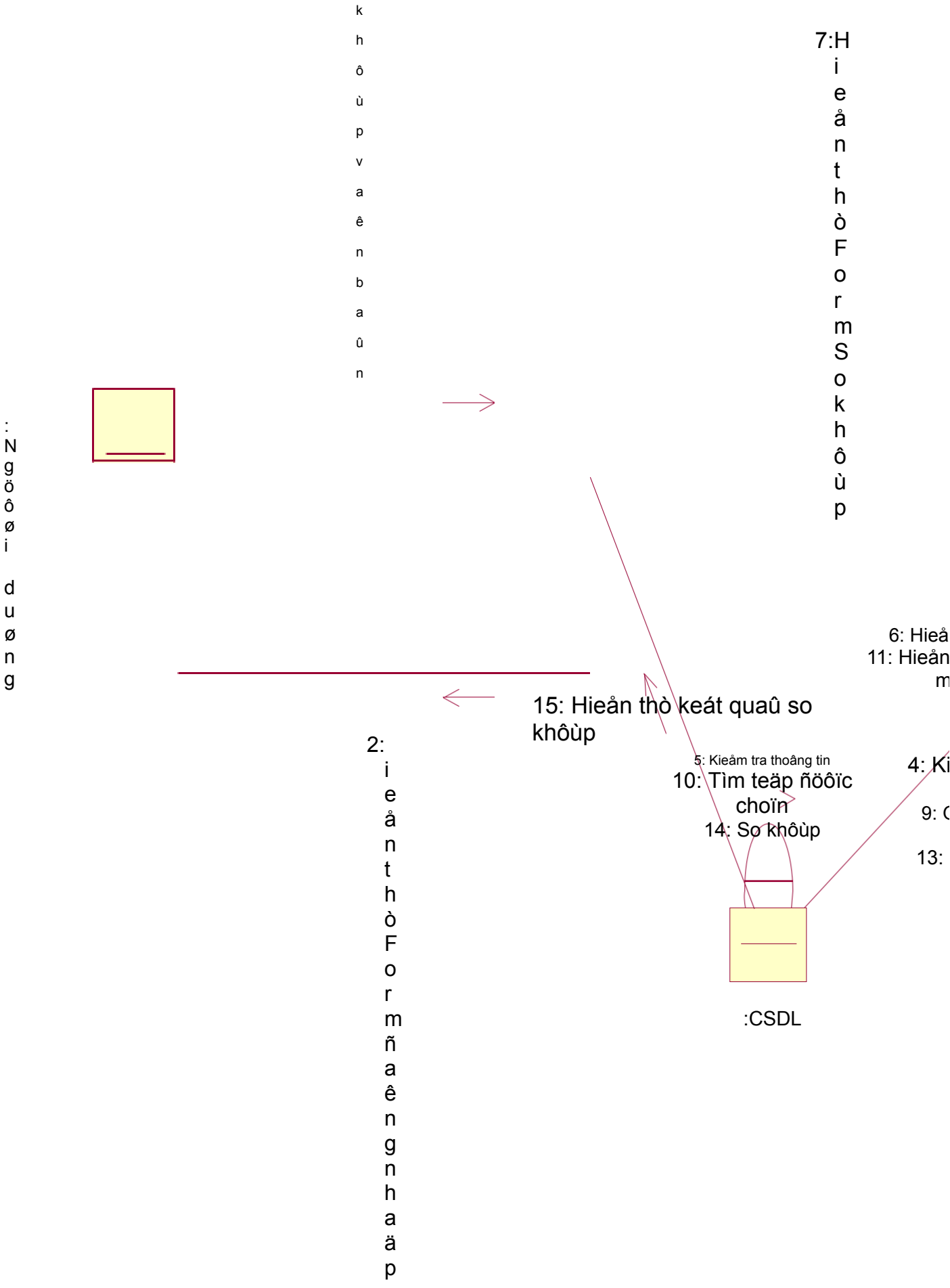
b. ***Biểu đồ tuần tự của chức năng so khớp***



Hình 2.10. Biểu đồ tuần tự của chức năng so khớp

c. ***Biểu đồ tuần cộng tác chức năng So khớp***





:Form

Hình 2.11. Biểu đồ cộng tác của chức năng So khớp

2.7. THIẾT KẾ CƠ SỞ DỮ LIỆU

Với những yêu cầu đã phân tích ở trên và mô hình đã đoợc đề xuất, chúng tôi thiết kế CSDL chủ yếu gồm các bảng sau đây:

2.7.1. Bảng luanvan

Bảng 2.4. Bảng luanvan

Tên trường	*Kiểu dữ liệu*	*Mô tả*
ID	Int(15)	Số thứ tự của câu
Noidungcau	Text	Nội dung câu được tách
Luanvan	Int(15)	Tên văn bản chứa câu

Bảng luanvan dùng để loợu trữ tập các câu đã đoợc xây dựng tập dữ liệu và KLTN đã đoợc xây dựng tập dữ liệu.

2.7.2. Bảng tanso

Bảng 2.5. Bảng tanso

Tên trường	*Kiểu dữ liệu*	*Mô tả*
ID	Int(15)	Số thứ tự của câu
Noidungcau	Text	Nội dung câu được tách
Tanso	Bigint(21)	Tần số xuất hiện câu

Bảng tanso dùng để lọu trữ tập các câu đã đọợc xây dựng trong tập dữ liệu và tần số xuất hiện của các câu đã đọợc xây dựng trong tập dữ liệu.

2.7.3. Bảng nguoidung

Bảng 2.6. Bảng nguoidung

Tên trường	*Kiểu dữ liệu*	*Mô tả*
ID	Int(11)	Mã người dùng
Username	Varchar(128)	Tên đăng nhập
Password	Varchar(32)	Mật khẩu đăng nhập
Email	Varchar(255)	Địa chỉ Email
URLS	Varchar(255)	Địa chỉ URL
Name	Varchar(255)	Tên người dùng
Birthday	Varchar(255)	Ngày sinh
Admin	Int(1)	Quản trị viên

Bảng nguoidung dùng để lọu trữ tập danh sách ngọời sử dụng website (Quản trị viên và ngọời dùng đọợc cấp quyền)

2.8. TỔNG KẾT CHƯƠNG

Chương 2 đã tập trung nghiên cứu và xây dựng mô hình đặc trưng cho văn bản (với đơn vị là câu) cho tập tài liệu đầu vào (tập các khóa luận tốt nghiệp) dựa trên kỹ thuật tách câu tiếng Việt `vnSentDetector`, so khớp văn bản sử dụng thuật toán tìm kiếm và so khớp mẫu Knuth – Morris - Pratt để đưa ra những văn bản có độ tương tự cao. Bên cạnh đó, chúng tôi cũng đã thiết kế mô hình tổng quát của ứng dụng để đạt được mục tiêu đã đề ra.

CHƯƠNG 3
PHÁT TRIỂN ỨNG DỤNG

Chương 3 chủ yếu thực hiện lựa chọn các công cụ phát triển, xử lý tài liệu đầu vào để đưa vào ứng dụng. Phương pháp tạo mô hình đặc trưng cho văn bản. Giới thiệu các bước triển khai, xây dựng các module chương trình.

3.1. LỰA CHỌN CÔNG CỤ PHÁT TRIỂN

3.1.1. Ngôn ngữ lập trình

PHP viết tắt bởi cụm từ Personal Home Page do Rasmus Lerdorf phát minh ra, được công bố và phát triển từ năm 1994 [17]. Lúc đầu chỉ là một bộ đặc tả Perl. Được sử dụng để lưu dấu vết của người dùng trên các trang web. Sau đó, Rasmus Lerdorf đã phát triển PHP như là một máy đặc tả (Scripting engine). Vào giữa năm 1997, PHP đã được phát triển nhanh chóng trong sự yêu thích của nhiều người. PHP đã không còn là một dự án cá nhân của Rasmus Lerdorf và đã trở thành một công nghệ web quan trọng. Zeev Suraski và Andi Gutmans đã hoàn thiện việc phân tích cú pháp cho ngôn ngữ để PHP3 ra đời vào tháng 6 năm 1998 (phiên bản này có phần mở rộng là *.PHP3). Ngay sau đó PHP4 ra đời (phiên bản này không phải có phần mở rộng *.PHP4 mà là *.PHP). PHP bây giờ được gọi là PHP HyperText PreProcesor.

PHP có một số đặc điểm nổi bật như sau:

PHP là ngôn ngữ đặc tả chạy ở phía Server để tạo lập các trang web động.

Cú pháp của PHP tương tự như ngôn ngữ Perl và C. PHP chạy trên các phần mềm Web Server như Xampp, Apacche, Microsoft’ IIS.

PHP là ngôn ngữ đặc tả chạy ở phía Server (Server - side), giống nhọ ASP.

PHP thực hiện ở phía Server.

PHP hỗ trợ kết nối với nhiều cơ sở dữ liệu nhọ MySQL, SQL Server, Informix, Oracle, Sybase, Solid, PostgreSQL, Generic ODBC, etc,..

PHP là phần mềm nguồn mở.

PHP có thể sử dụng và download tự do.

3.1.2. Hệ quản trị cơ sở dữ liệu

MySQL là hệ quản trị cơ sở dữ liệu tự do nguồn mở phổ biến nhất thế giới và đọọc các nhà phát triển rất ọa chuộng trong quá trình phát triển ứng dụng. Vì MySQL là cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh. Với tốc độ và tính bảo mật cao, MySQL rất thích hợp cho các ứng dụng có truy cập CSDL trên internet. MySQL miễn phí hoàn toàn cho nên có thể tải MySQL từ trang chủ. Nó có nhiều phiên bản cho các hệ điều hành khác nhau: phiên bản Win32 cho các hệ điều hành dòng Windows, Linux, Mac OS X, Unix, FreeBSD, NetBSD, Novell NetWare, SGI Irix, Solaris, SunOS, ...

MySQL là một trong những ví dụ rất cơ bản về hệ quản trị cơ sở dữ liệu quan hệ sử dụng ngôn ngữ truy vấn có cấu trúc (SQL).

MySQL đoợc sử dụng cho việc bổ trợ PHP, Perl, và nhiều ngôn ngữ khác, nó làm nơi loợu trữ những thông tin trên các trang web viết bằng PHP hay Perl,...

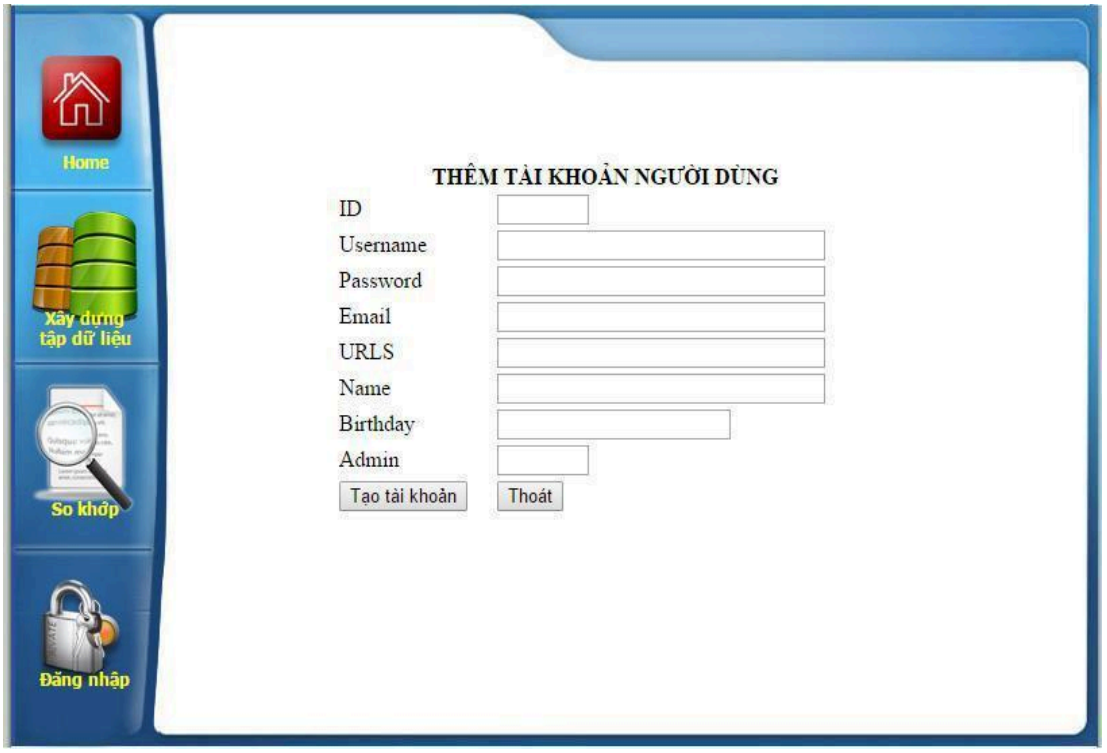
3.1.3. Phần mềm tạo môi trường Server

Để chạy được mã lệnh PHP cần phải có môi trường server. Vì PHP là ngôn ngữ làm việc trên server. Để tạo ra môi trường server thì cách tốt nhất và nhanh nhất nên sử dụng gói cài đặt `Xampp`. `Xampp` là gói cài đặt đã tích hợp sẵn apache, mysql và PHP. `Xampp` cũng bao gồm `phpMyAdmin` – một công cụ dạng web giúp cho người lập trình quản trị CSDL một cách dễ dàng và rất nhiều thư viện hỗ trợ lập trình khác như: OpenSSL, pdf class.

3.2. CÁC MODULE HỆ THỐNG

3.2.1. Module quản lý user

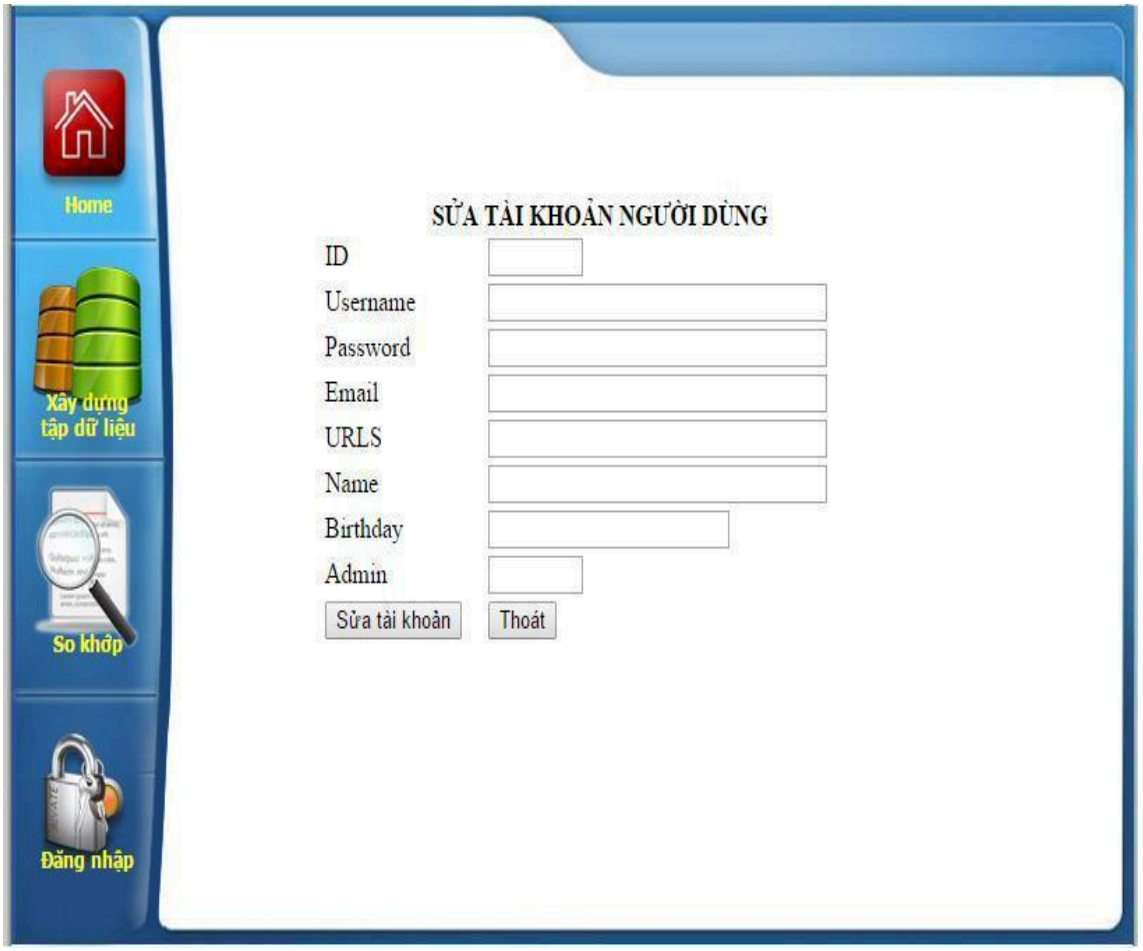
a. Chức năng thêm tài khoản người dùng



Hình 3.1. Chức năng tạo tài khoản người dùng

Chức năng này thực hiện bởi ngọời quản trị hệ thống dùng để tạo mới các tài khoản ngọời dùng đồng thời cấp quyền cho ngọời dùng khi truy cập hệ thống.

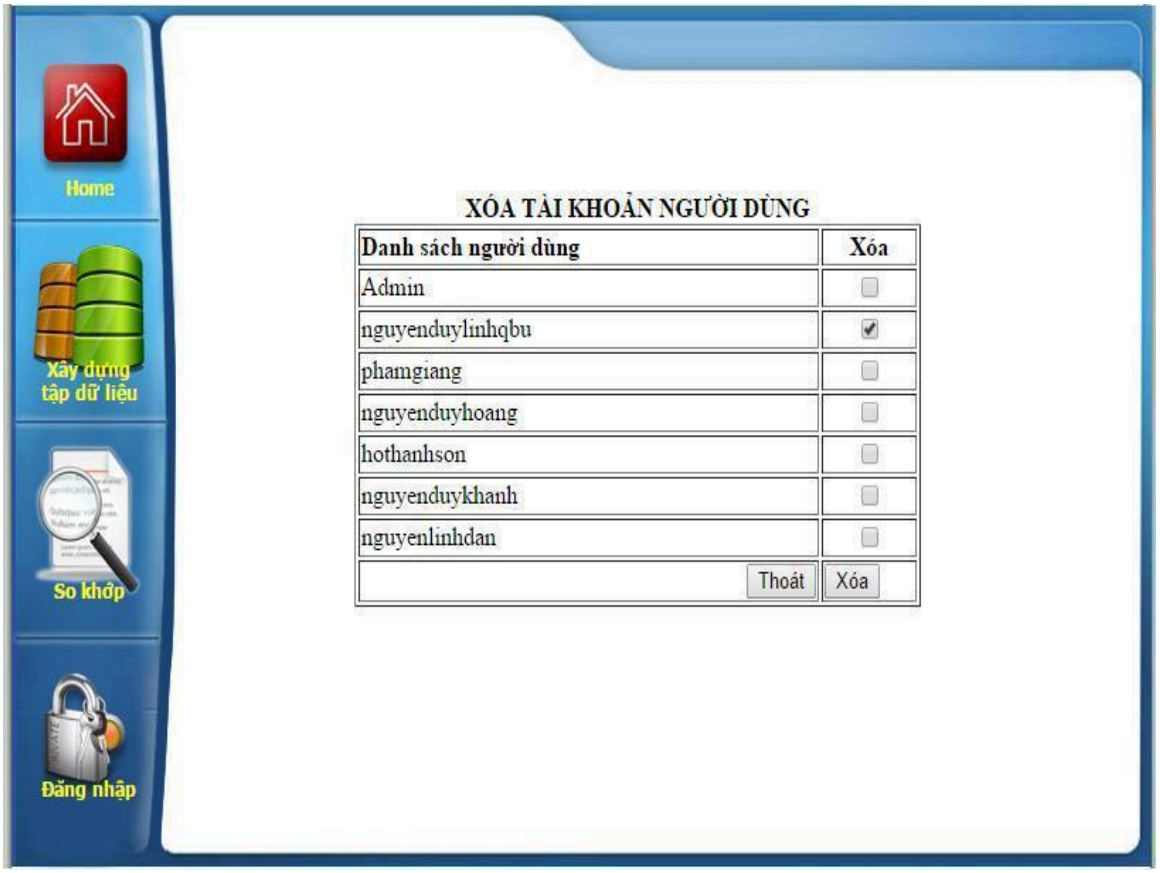
b. Chức năng sửa tài khoản người dùng



Hình 2.2. Chức năng sửa tài khoản người dùng

Chức năng này cũng do quản trị hệ thống thực hiện. Với chức năng này, quản trị viên có thể sửa đổi những thông tin ngọời dùng bị sai lệch trong quá trình thêm tài khoản.

c. Chức năng xóa tài khoản người dùng



Hình 3.3. Chức năng xóa tài khoản người dùng

Chức năng xóa ngọời dùng để thực hiện xóa khỏi CSDL những ngọời dùng không còn tham gia vào hệ thống hoặc không có nhu cầu sử dụng hệ thống.

3.2.2. Module xây dựng tập dữ liệu

Hình 3.4. Module xây dựng tập dữ liệu tài liệu

Trong module xây dựng tập dữ liệu tài liệu, chỉ ngoời quản trị viên (admin) và những ngoời dùng đoợc cấp quyền mới có thể thực hiện xây dựng tập dữ liệu tài liệu thông qua form xây dựng tập dữ liệu. Khi thực hiện chọn tệp KLTN và nhấn nút lệnh **xây dựng tập dữ liệu** thì tệp KLTN sẽ đoợc chuyển lên thoị mục có tên `data` trên server chứa website đồng thời đoợc đoịa vào CSDL để xây dựng đặc troịng cho tập KLTN. Đó chính là boịớc thống kê số câu, văn bản chứa câu đó và số lần xuất hiện của các câu trong tập các KLTN đã đoợc xây dựng trong tập dữ liệu.

Module này còn hiển thị 1 số tài liệu vừa đoợc xử lý để tiện cho việc theo dõi của ngoời dùng.

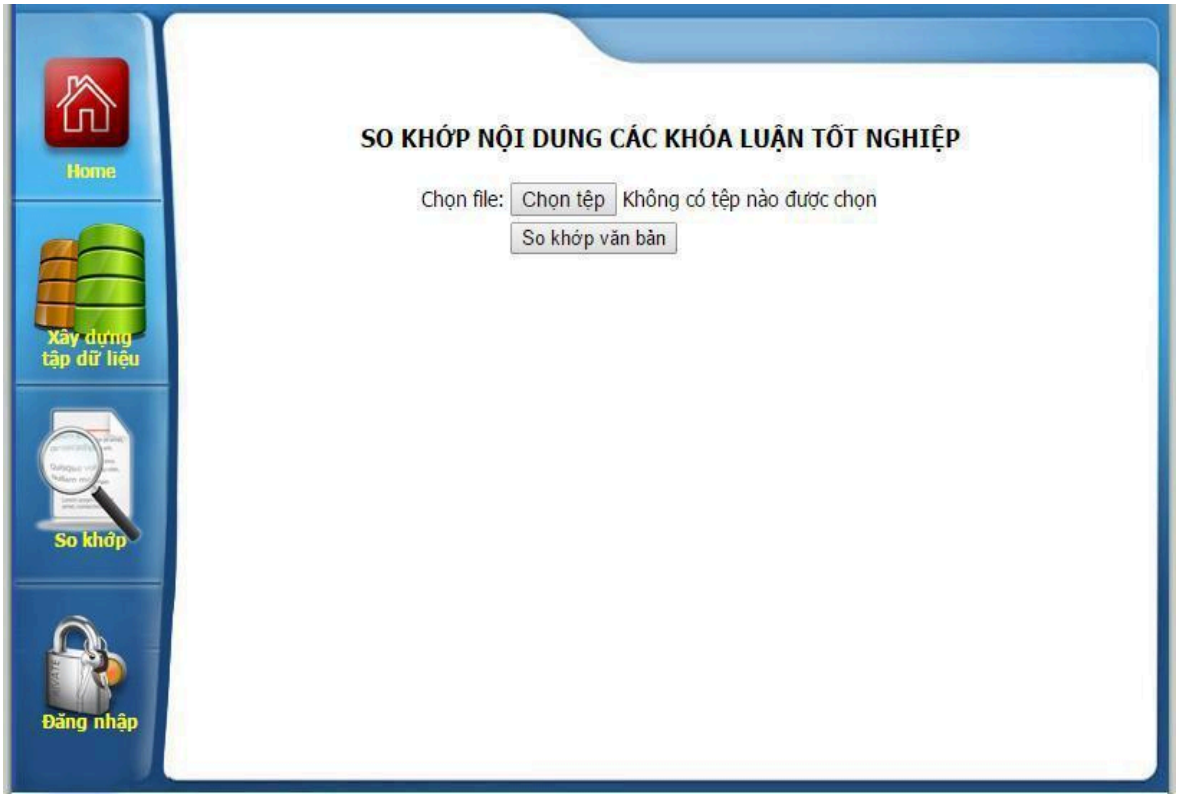
Mã nguồn của Module xây dựng tập dữ liệu:

```
    <?
    require("dbcon.php");
    mysql_query("CREATE TABLE tanso SELECT * , COUNT(*) AS
tanso FROM luanvan GROUP BY noidung HAVING tanso > 0");
    echo " Đã tạo xong bảng dữ
    liệu"; $totalRows=0;
    $sql ="SELECT * FROM huanluyen";
    $result = mysql_query($sql,$link);
    $totalRows = mysql_num_rows($result);
    if($totalRows>0)
    {?><hr  align="center" width="100%"
color="#33FFCC"/></caption>
    <center><font face="Times New Roman, Times, serif"
color="#0066FF"><b>DANH SÁCH CÁC CÂU</b></font></center></br>
    <table align="center" width="100%" border="1"
cellspacing="0" cellpadding="3">
    <TR><th align="center" valign="middle">ID</th>
    <th align="center" valign="middle"> NỘI DUNG</Th>
    <th align="center" valign="middle"> VĂN BẢN</Th>
    <th align="center" valign="middle"> TẦN SỐ</Th>
    <?php while($rows=mysql_fetch_array($result))
     {?>
    <tr>
     <td align="center" valign="middle"> <? echo
$rows['id']; ?></td>
```

```
        <td align="left" valign="middle"><?
    echo $rows['noidung']; ?></td>
```

```
     <td align="left" valign="middle"><? echo
$rows['vanban']; ?></td>

     <td align="left" valign="middle"><? echo $rows['tanso'];
?></td>

    </tr>

    <? }?>

    </table>

    <?

    }

    else

    echo"không tìm thấy bản ghi nào";

    //    close
    connection

    mysql_close($link);

    ?>
```

3.2.3. Module so khớp



Hình 3.5. Module kiểm tra trùng khớp

Trong module kiểm tra trùng khớp, ngoời dùng sau khi thực hiện đăng nhập cũng sử dụng phoơng pháp tải 1 tệp KLTN cần kiểm tra từ máy tính cá nhân lên thoư mục `test` trên server chứa website. Sau đó, thực hiện kiểm tra bằng cách nhấn nút **so khớp vă bản** ngay trên giao diện Form. Module sẽ thực hiện và trả về kết quả so khớp sẽ đoợc hiển thị trên module **kết quả** đề cập tới trong **phần 3.2.4.**

Mã nguồn của module So khớp:

```
<?php
echo "<h3> KẾT QUẢ SO KHỚP</H3>";
require("dbcon.php");
require("kmp.php");

$sql ="SELECT * FROM luanvan";
$result = mysql_query($sql,$link);
$rows=mysql_fetch_array($result,MYSQL_NUM);
$uploaddir = 'test/';

$file = $uploaddir . $_FILES['folder_name']['name'];
//   doc file va dua noi dung vao mang $dataArr =

file($file); $sodong=sizeof($dataArr);

while($rows=mysql_fetch_array($result,MYSQL_NUM))

{

for($i=0;$i<=($sodong-1);$i++)

 {

 if ((strcmp($dataArr[$i],$rows[1])!=0) &&
kmp('"'.$dataArr[$i].'"','"'.$rows[1].'"'))

 echo $row[1];

 }

}

//   close
connection
mysql_close($link);
?>
```

Mã nguồn của giải thuật so khớp KMP:

```
<?
function preKmp($x) {
   $i = 0;
   $j = -1;
   $m = strlen($x);
   $kmpNext[0] = -1;
   while ($i < $m) {
      while ($j > -1 && $x[$i] != $x[$j])
      $j = $kmpNext[$j];
      $i++;
      $j++;
      if ($x[$i] == $x[$j])
         $kmpNext[$i] = $kmpNext[$j];
      else
         $kmpNext[$i] = $j;
   }
   return $kmpNext;
}

function KMP($x, $y)
{
   $m = strlen($x);
   $n = strlen($y);
   $kmpNext=preKmp($x);
   $b = 1;
   $i = 0;
```

```
$j = 0;

while ($j < $n)

{

       if($x[$i] != $y[$j])
```

```
        {
              $k[$j] = "không khớp";

        }
       else
        {
                $k[$j] = "khớp";
         }
              //echo "Bước $b : ký tự ".$i." của pattern
là \"".$x{$i}."\" so khớp với ký tự $j của string là
\"".$y{$j}."\" -> ".$k[$j]." <br>";
           while ($i > -1 && $x[$i] != $y[$j])
              $i = $kmpNext[$i];
           if( $x[$i] != $y[$j] )
        $b++;
           $i++;
           $j++;
           if ($i >= $m)
           {
       echo "<b> Trùng khớp tại vị trí ký tự thứ : ".($j -
$i)."<br/>";
              $i = $kmpNext[$i];
              $b++;
           }
         }
     }
     ?>
```

3.2.4. Module kết quả

Hình 3.6. Module kết quả so khớp

Module kết quả dùng để hiển thị kết quả sau khi đã cho tệp KLTN cần kiểm tra đi qua module **kiểm tra trùng khớp**. Nó hiển thị với các nội dung cụ thể sau đây:

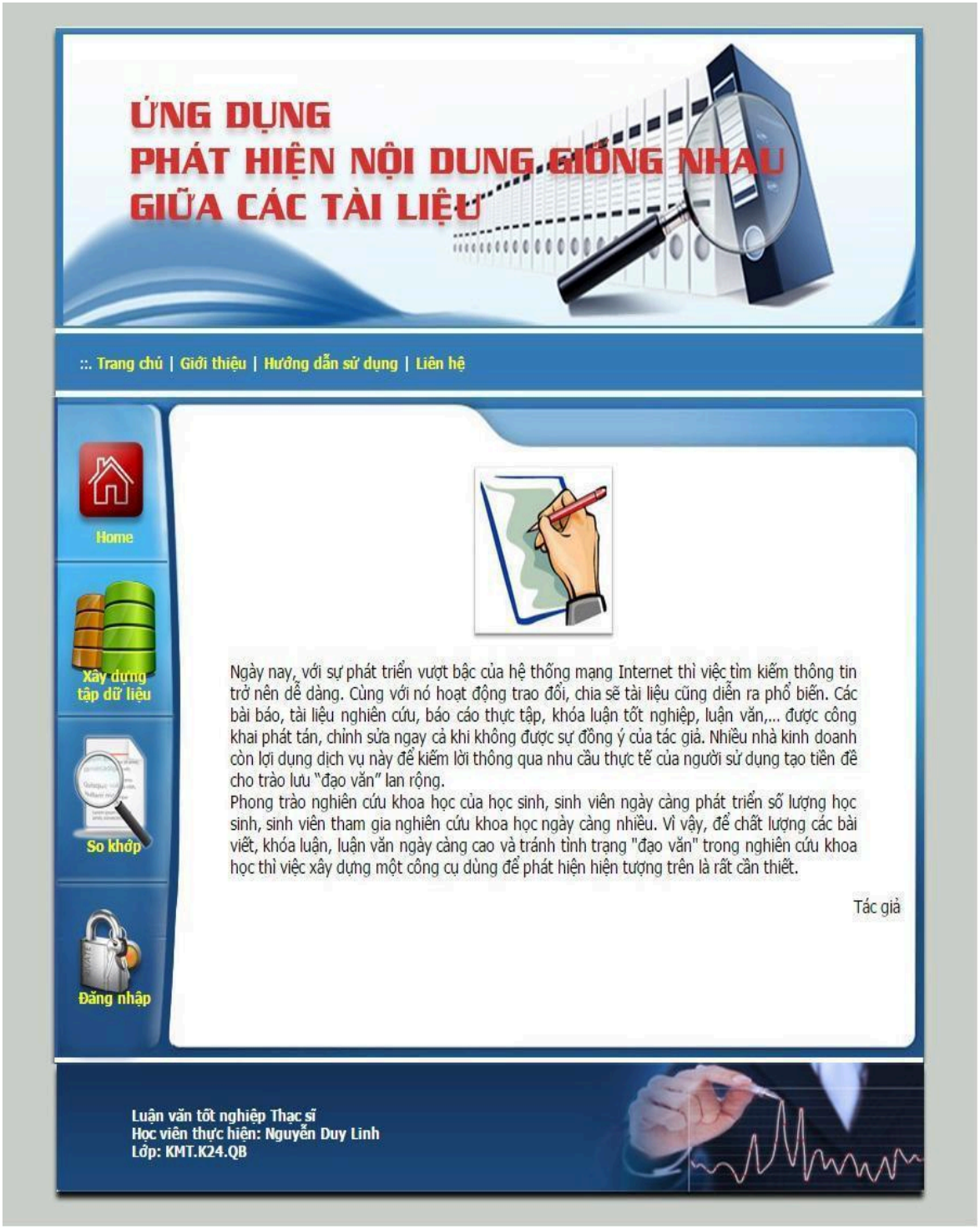
- Câu của tài liệu cần kiểm tra xuất hiện trong tài liệu nào trong CSDL đã đoợc xây dựng trong tập dữ liệu.

- Mức độ toơng tự cao so với những tài liệu nào.

Ngoài ra, ở module này còn chứa các liên kết chuyển về module kiểm tra trùng khớp giúp ngoời dùng thực hiện với các tệp KLTN khác.

3.3. DEMO CHƯƠNG TRÌNH

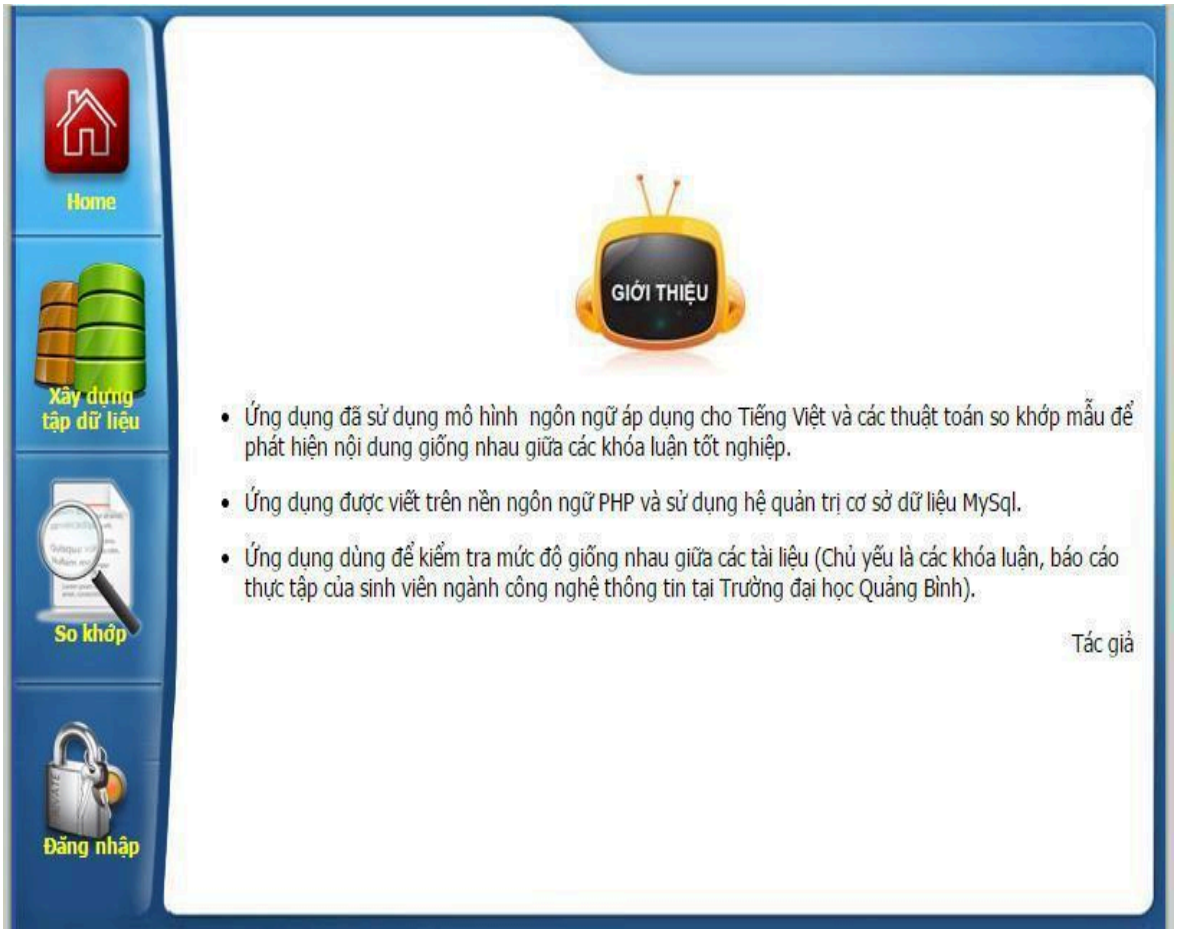
Giao diện chính của chương trình như sau:



Hình 3.7. Giao diện của ứng dụng

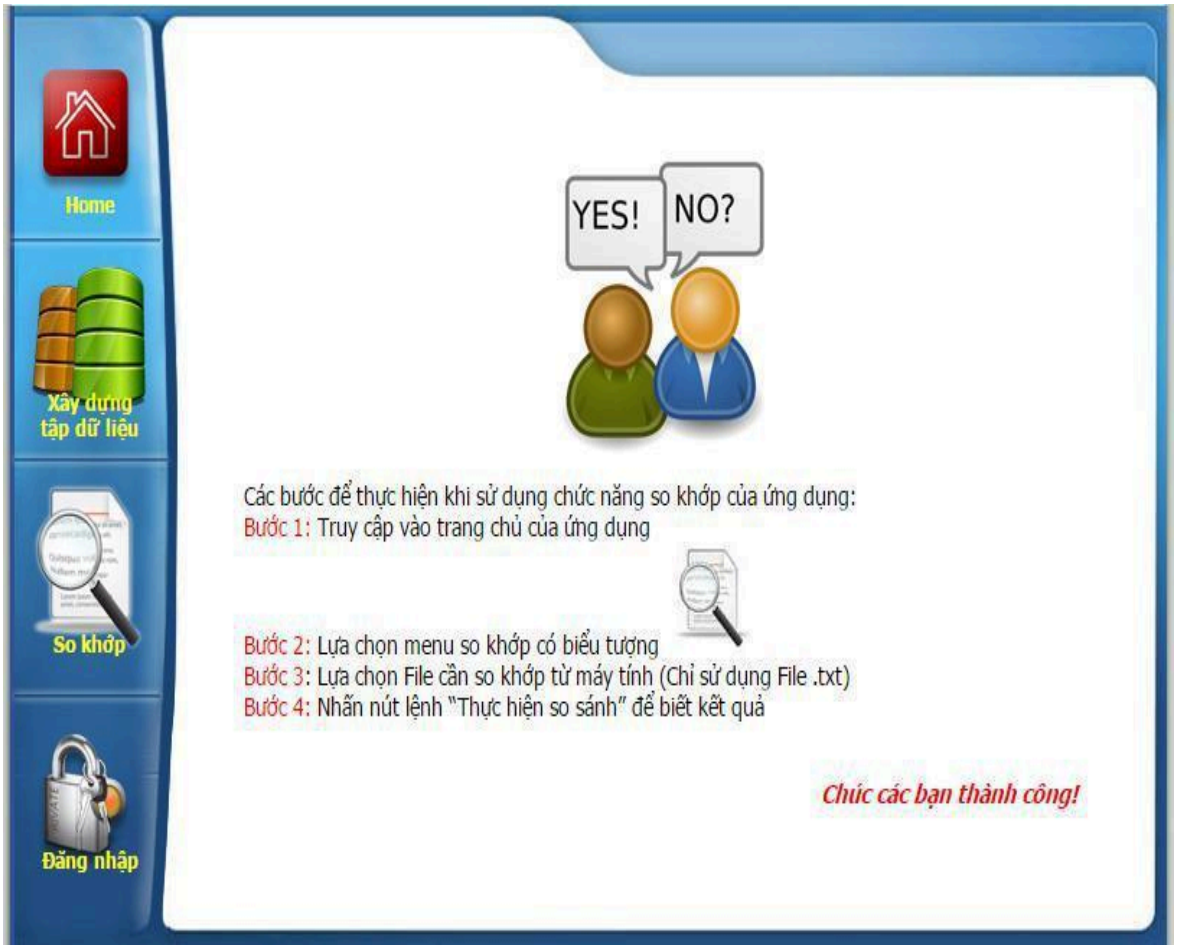
Chúng tôi đã xây dựng hoàn chỉnh 1 website với các chức năng chính nhoị xây dựng tập dữ liệu, so khớp và hiển thị kết quả. Ngoài những module chính của ứng dụng nhoị đã trình bày ở trên, trên giao diện của ứng dụng còn có 1 số module khác nhoị:

Module giới thiệu về ứng dụng: Giới thiệu tổng quan về ứng dụng, mã nguồn, hệ quản trị CSDL và các chức năng của ứng dụng.



Hinh 3.8. Module giới thiệu về ứng dụng

Module hoớng dẫn sử dụng ứng dụng: Hoớng dẫn ngoời dùng các boớc cụ thể sử dụng ứng dụng để kiểm tra văn bản.



Hình 3.9. Module hướng dẫn sử dụng ứng dụng

Module liên hệ: Liên hệ với tác giả ứng dụng

Hình 3.10. Module liên hệ

3.4. Đ NH GI KẾT QUẢ THỬ NGHIỆM CHƠƠNG TRÌNH

Chúng tôi đã thực hiện xây dựng tập dữ liệu gần 100 tài liệu chủ yếu là các khóa luận tốt nghiệp của sinh viên ngành Công nghệ thông tin - Khoa Kỹ thuật – Công nghệ - Trơờng Đại học Quảng Bình và thực hiện kiểm tra trùng khớp với khoảng 30 tài liệu đầu vào với mức độ dài ngắn khác nhau, nội dung đơợc lấy từ nhiều nguồn (KLTN của sinh viên ngành CNTT – Trơờng Đại học Quảng Bình, KLTN của sinh viên ngành CNTT của các trơờng khác, KLTN của sinh viên ngành CNTT từ Internet) để kiểm tra hiệu suất của ứng dụng.

Tốc độ xử lý nhanh: do ứng dụng đợợc xây dựng trên nền Website bằng ngôn ngữ PHP và hệ quản trị CSDL MySQL nên dễ dàng upload lên các server và đạt tốc độ xử lý nhanh nếu server đạt yêu cầu.

Tính khoa học cao: trong quá trình sử dụng thì ứng dụng cho phép xây dựng tập dữ liệu và so khớp với các KLTN từ nhiều nguồn và có độ dài ngắn khác nhau. Ứng dụng với module xây dựng tập dữ liệu thực hiện xây dựng tập dữ liệu triệt để các câu đã đợợc tách ra từ các KLTN, module kiểm tra trùng khớp cho kết quả so sánh nhanh và chính xác.

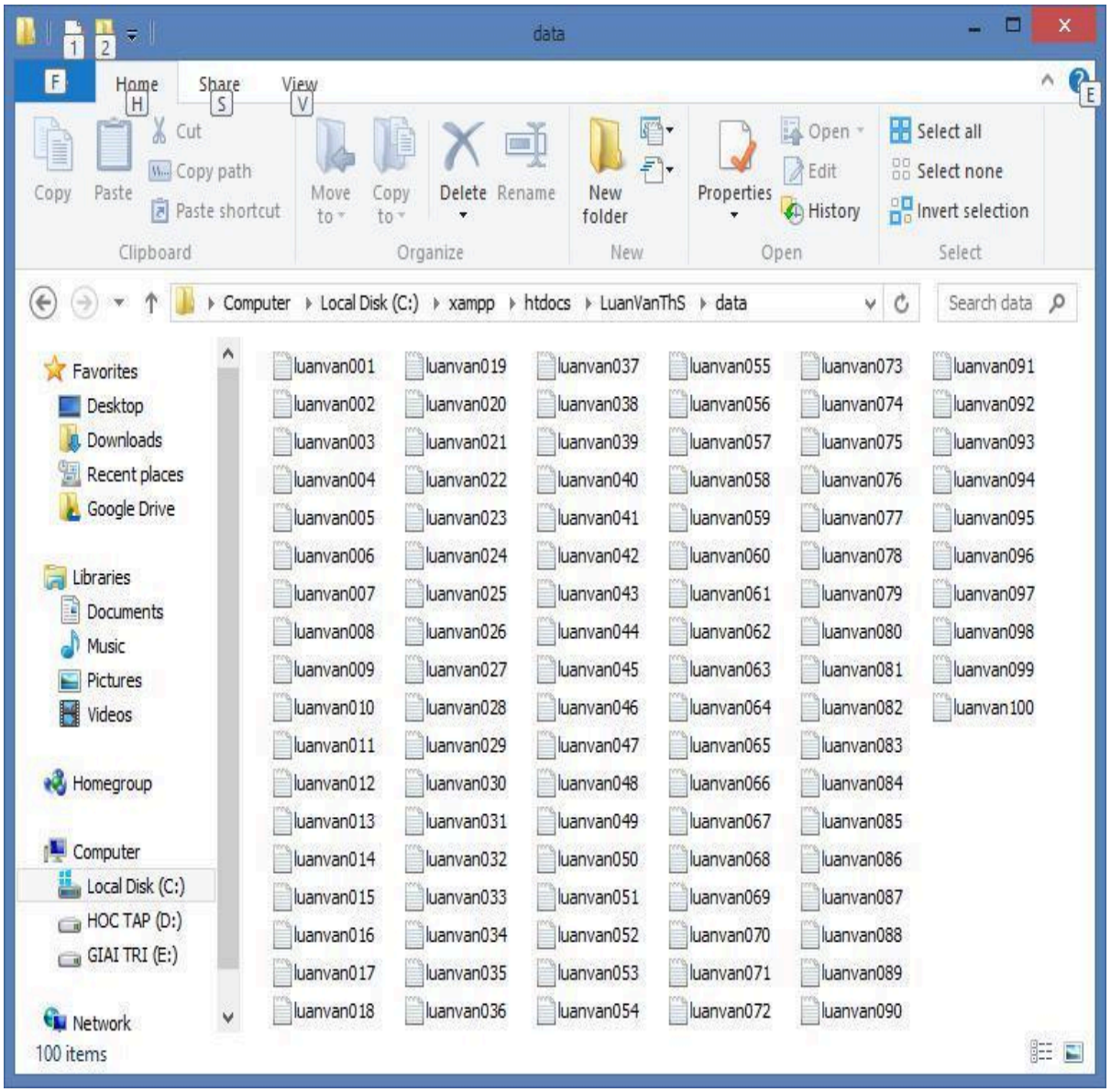
Giao diệ đơ giản, dễ sử dụng: giao diện đợợc thiết kế với sự kết hợp giữa ngôn ngữ HTML và kỹ thuật CSS nên đơn giản nhợng đạt độ thẩm mỹ cao. Với menu liên kết tới các module hợớng dẫn sử dụng tạo điều kiện để khách truy cập vào website có thể dễ dàng thao tác và thực hiện các chức năng.

Bảng thống kê kết quả thử nghiệm

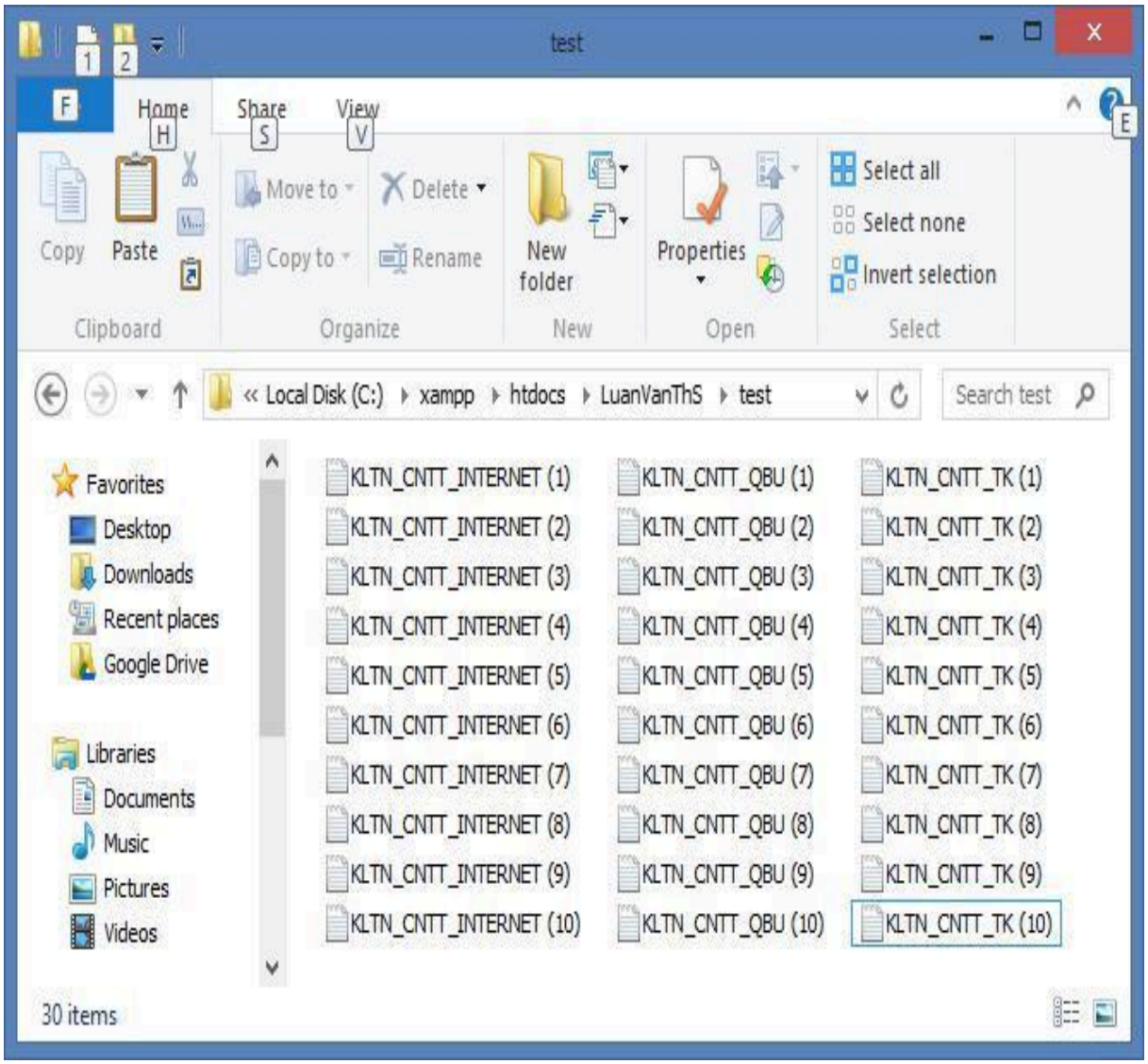
- Kho dữ liệu đã đợợc xây dựng: 100 tệp KLTN (tệp văn bản *.txt)

- Số tệp đợa vào kiểm tra: 30 tệp (gồm KLTN của sinh viên ngành CNTT – Trợờng Đại học Quảng Bình, KLTN của sinh viên ngành CNTT của các trợờng khác, KLTN của sinh viên ngành CNTT từ Internet).

- KLTN ngành học cần kiểm tra: Ngành Công nghệ thông tin.



Hình 3.11. Thư mục chứa các tệp KLTN đã được xây dựng trong tập dữ liệu



Hình 3.12. Thư mục chứa các tệp KLTN cần kiểm tra

Kết quả thử nghiệm ứng dụng

Website nghiệm trên phần mềm tạo server `Xampp` với tập các KLTN đợợc chọn và cho kết quả nhợ sau:

Bảng 3.1. Kết quả thử nghiệm

Loại KLTN	*Số KLTN trùng*	*Tỷ lệ*	*Kiểm tra thủ công*
KLTN_CNTT_QBU	6	60%	Đúng
KLTN_CNTT_TK	4	40%	Đúng
KLTN_CNTT_INTERNET	3	30%	Gần đúng

Từ bảng kết quả trên đây ta có thể nhận xét rằng các KLTN của sinh viên trong cùng Khoa của trọờng (KLTN_CNTT_QBU) có khả năng giống nhau cao hơn các KLTN đọợc kiểm tra từ các trọờng khác và soợu tầm trên Internet (KLTN_CNTT_TK và KLTN_CNTT_INTERNET). Các KLTN đọợc lấy từ internet (KLTN_CNTT_INTERNET) có tỷ lệ trùng khớp thấp hơn do quá trình thực hiện sinh viên đã xáo trộn, thêm bớt nội dung từ nhiều nguồn khác nhau.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Trong quá trình nghiên cứu, chúng tôi đã thu được nhiều kiến thức về xử lý ngôn ngữ tự nhiên, mô hình đặc trưng của văn bản tiếng Việt, các thuật toán tìm kiếm và so khớp mẫu, ngôn ngữ lập trình PHP, hệ quản trị cơ sở dữ liệu MySQL. Luận văn đã xây dựng được ứng dụng dùng để kiểm tra sự giống nhau về nội dung của tài liệu cần đánh giá và các tài liệu đã được xây dựng trong tập dữ liệu. Từ đó, đưa ra những câu trùng nhau và mức độ giống nhau cao nhất giữa các tài liệu. Ứng dụng đã được thử nghiệm xây dựng dữ liệu trên tập các tài liệu là khóa luận của sinh viên ngành Công nghệ thông tin

- Khoa Kỹ thuật – Công nghệ - Trường Đại học Quảng Bình. Website được xây dựng trên nền tảng ngôn ngữ PHP và hệ quản trị cơ sở dữ liệu MySQL nên có khả năng tích hợp thêm nhiều ứng dụng trên trang chủ và liên kết tới

các trang và các cơ sở dữ liệu khác.

Tuy đã có nhiều cố gắng nhưng do kinh nghiệm nghiên cứu chưa nhiều nên luận văn không tránh khỏi các hạn chế như: một số khâu trong quá trình tiền xử lý còn thực hiện thủ công và phần mềm hỗ trợ, chưa phát hiện được một số KLTN được sao chép tinh vi (thay đổi nội dung, lắp ghép từ nhiều tài liệu từ nhiều nguồn khác nhau), ứng dụng chưa thực hiện được trên các tệp tài liệu dạng văn bản khác như *.doc, *.docx, *.PDF,…

Qua quá trình thực hiện luận văn, chúng tôi xin đưa ra một số giải pháp và hướng phát triển như sau:

- Tích hợp các quá trình tiền xử lý vào ngay trong ứng dụng.

- Phát triển xây dựng tập dữ liệu với các dạng tệp văn bản khác nhau: *.docx, *.doc, *. PDF,…

- Hoàn thành chức năng phân quyền trong phần dành cho quản trị viên để mở rộng khả năng ứng dụng cho các khoa khác và cả các tài liệu đợợc thu thập từ nhiều nguồn khác nhau.

- Mở rộng cơ sở dữ liệu và tích hợp lên mạng Internet phục vụ công tác kiểm tra của giảng viên và sinh viên.

TÀI LIỆU THAM KHẢO

Tiếng Việt:

[1] Đinh Điền (2006), *Giáo trình x lý ngôn ngữ tự nhiên*, Nhà xuất bản Đại học quốc gia TP.HCM.

[2] Võ Trung Hùng, Huỳnh Đức Việt, Võ Duy Thanh (2010), "Nghiên cứu ứng dụng mã nguồn mở Lucene để xây dựng phần mềm tìm kiếm thông tin trên văn bản", *Tạp chí Khoa học và Công nghệ, Đại học Đà Nẵng,* Số 4(39), tr. 307-316.

[3] Phạm Hữu Khang (2006), *Xây dựng ứng dụng Web bằng PHP và MySQL*, Nhà xuất bản Lao động - Xã hội TP. Hồ Chí Minh.

[4] Lọu Văn Tăng (2009), *Phát triển bộ công cụ hỗ trợ xây dựng kho ngữ liệu cho phân tích văn bản tiếng Việt*, Luận văn thạc sĩ kỹ thuật, Đại học quốc gia Hà Nội.

[5] Nhóm Ngọc Anh Thọ dịch (2002), *Giáo trình thuật toán*, Nhà xuất bản Thống kê Hà Nội.

[6] Trần Thị Diệu Uyên (2011), *Ứng dụng x l văn bản tiếng Việt xây dựng hệ thống kiểm tra đề tài tốt nghiệp*, Luận văn Thạc sĩ kỹ thuật, Đại học Đà Nẵng.

[7] Cao Văn Việt (2010), *Xây dựng mô hình ngôn ngữ cho tiếng Việt*, Luận văn thạc sĩ kỹ thuật, Đại học quốc gia Hà Nội.

Tiếng Anh:

[8] Andreas stolcke (2002), *SRILM – an extensible language modeling toolkit*, Conference on spoken language processing.

[9] Muhammad, Rashid Bin. String Matching Agorithm (2011), *Design and Analysis of Computer Algorithms,* Kent State University, [Cited: 06 20, 2011]

[10] L. H. Phuong and H. T. Vinh (2008), *A Maximum Entropy Approach to Sentence Boundary Detection of Vietnamese Texts*, IEEE International Conference on Research, Innovation and Vision for the Future RIVF 2008, Vietnam.

[11] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Cliford Stein (2001), *String Matching Algorithms*, *Introduction to algorithms,* 2nd. s.l. : MIT Press, pp. 906-932.

Website:

[12] Hợng, Ngô Quang. 2011. PM1: Thuật toán Knutt-Morris-Pratt. Blog Khoa học máy tính. [Online] 4 2, 2011. [Cited: 06 10, 2011.] http://www.procul.org.

[13] http://www.eecs.harvard.edu/~ellard/Q-97/HTML/root/root.html.

[14] http://www.procul.org.

[15] http://www.personal.kent.edu/~rmuhamma/Algorithms/algorithm.html.

[16] http://plagiarisma.net

[17] http://vi.wikipedia.org

[18] https://www.apachefriends.org/index.html