ĐẠI HỌC QUỐC GIA HÀ NỘI

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA TOÁN – CƠ – TIN HỌC

**Phùng Văn Thành**

**XÂY DỰNG MỘT HỆ THỐNG XỬ LÝ NGÔN NGỮ TỰ NHIÊN CHO VĂN BẢN TIẾNG VIỆT**

Khóa luận tốt nghiệp đại học hệ chính quy

Ngành: Máy tính và khoa học thông tin

(Chương trình đào tạo: chuẩn)

**Hà Nội – 2018**

ĐẠI HỌC QUỐC GIA HÀ NỘI

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA TOÁN – CƠ – TIN HỌC

**Phùng Văn Thành**

**XÂY DỰNG MỘT HỆ THỐNG XỬ LÝ NGÔN NGỮ TỰ NHIÊN CHO VĂN BẢN TIẾNG VIỆT**

Khóa luận tốt nghiệp đại học hệ chính quy

Ngành: Máy tính và khoa học thông tin

(Chương trình đào tạo: chuẩn)

**Cán bộ hướng dẫn: Nguyễn Việt Hùng**

**Hà Nội – 2018**

# Lời cảm ơn

Trước hết, cho em xin phép được gửi lời cảm ơn sâu sắc nhất tới Thạc sỹ Nguyễn Việt Hùng – người trực tiếp hướng dẫn khóa luận đã luôn dành nhiều thời gian, công sức, tận tình giúp đỡ và tạo điều kiện thuận lợi nhất để hướng dẫn và chỉ bảo em trong suốt quá trình thực hiện và hoàn thành khóa luận tốt nghiệp này một cách tốt nhất có thể.

Cũng cho phép em được gửi lời cảm ơn chân thành nhất tới các thầy cô trong bộ môn tin học nói riêng và Khoa Toán – Cơ – Tin học trường Đại học Khoa học Tự Nhiên nói chung đã dạy dỗ, giúp đỡ và cho em một môi trường có thể thoải mái sáng tạo và phát triển bản thân trong suốt quá trình học tập tại trường.

Xin được gửi lời cảm ơn chân thành tới các anh chị, bạn bè, gia đình… những người đã luôn sát cánh bên tôi trong suốt những năm học qua.

Mặc dù đã cố gắng hết sức nhưng trong đề tài khóa luận này không tránh khỏi những thiếu sót. Em rất mong quý thầy cô, bạn bè, gia đình, cộng đồng, những người quan tâm đến đề tài của em có thể đưa ra những góp ý để đề tài của em được hoàn thiện và đưa vào thực tế hơn.

Em xin chân thành cảm ơn!

Hà Nội, ngày 21 tháng 5 năm 2018

Sinh viên

Phùng Văn Thành

# Mục lục

[Lời cảm ơn 3](#_Toc514682927)

[Mục lục 4](#_Toc514682928)

[Mở đầu 6](#_Toc514682929)

[1 Đặt vấn đề 8](#_Toc514682930)

[1.1 Xử lý ngôn ngữ tự nhiên 8](#_Toc514682931)

[1.2 Các bước xử lý 8](#_Toc514682932)

[1.3 Các bài toán và ứng dụng 8](#_Toc514682933)

[1.4 Thực trạng hiện nay và hướng giải quyết 10](#_Toc514682934)

[1.5 Các bài toán cơ bản trong xử lý tiếng Việt 10](#_Toc514682935)

[2 Chương 2. Cơ sở lý thuyết 11](#_Toc514682936)

[2.1 Bài toán tách từ 11](#_Toc514682937)

[2.2 Bài toán gán nhãn từ loại 13](#_Toc514682938)

[2.3 Bài toán phân tích cú pháp 17](#_Toc514682939)

[2.4 Bài toán phân tích tình cảm 21](#_Toc514682940)

[3 Chương 3. Phân tích – thiết kế hệ thống 23](#_Toc514682941)

[3.1 Kịch bản 23](#_Toc514682942)

[3.2 Mô hình use case của hệ thống 24](#_Toc514682943)

[3.3 Mô tả thực thể 27](#_Toc514682944)

[3.3.1 Người dùng 27](#_Toc514682945)

[3.3.2 Danh mục tài liệu 27](#_Toc514682946)

[3.3.3 Tài liệu 27](#_Toc514682947)

[3.3.4 File 28](#_Toc514682948)

[3.3.5 Câu 28](#_Toc514682949)

[3.3.6 Đề xuất chỉnh sửa 28](#_Toc514682950)

[3.3.7 Nhãn từ loại 28](#_Toc514682951)

[3.4 Mối quan hệ giữa các thực thể 29](#_Toc514682952)

[3.5 Xử lý mối quan hệ 30](#_Toc514682953)

[3.6 Mô hình ER 31](#_Toc514682954)

[3.7 Sơ đồ lớp 32](#_Toc514682955)

[3.8 Công nghệ sử dụng 32](#_Toc514682956)

[4 Chương 4. Kết quả 35](#_Toc514682957)

[4.1 Các kết quả đã làm được: 35](#_Toc514682958)

[4.2 Hạn chế: 36](#_Toc514682959)

[4.3 Kết quả chi tiết 36](#_Toc514682960)

[5 Tài liệu tham khảo 48](#_Toc514682961)

# Mở đầu

Trong cuộc cách mạng công nghiệp lần thứ 4 hiện nay, các nghiên cứu đều đang có xu hướng về việc xử lý các vấn đề liên quan đến internet vạn vật (Internet of thing - IoT), xử lý ảnh (Image processing - IP), học máy (Machine learning - ML), trí tuệ nhân tạo (Artificial Intelligence - AI)… Và trong số đó không thể không nhắc đến việc xử lý ngôn ngữ tự nhiên. Ở Việt Nam hiện nay, vấn đề này luôn được các nhà nghiên cứu cũng như số lượng lớn người quan tâm và tìm hiểu. Mặc dù đã có những mô hình xử lý các bài toán ngôn ngữ tự nhiên từ cơ bản cho đến nâng cao đã thành công phần lớn đối với tiếng Anh. Tuy nhiên, việc áp dụng các mô hình đó vào việc xử lý ngôn ngữ tự nhiên ở Việt Nam hiện vẫn đang còn gặp nhiều khó khăn do sự khác nhau về mặt cấu trúc ngữ pháp và sự đa dạng của tiếng Việt. Và điều đặc biệt hơn cả là việc thiếu hụt về dữ liệu cũng như một hệ thống tích hợp các công cụ xử lý ngôn ngữ tự nhiên được công bố miễn phí cho cộng đồng. Mục đích của khóa luận này nhằm xây dựng nên một trang web trên đó tích hợp các công cụ cũng như các hệ thống xử lý để phục vụ cho việc xử lý ngôn ngữ tự nhiên cho văn bản tiếng Việt.

Mục tiêu chính được đề ra đó là xây dựng một trang web cho phép người dùng có quyền đăng tải tài liệu lên, sử dụng các bộ công cụ được thiết kế thuận tiện cho việc chỉnh sửa và quản trị được các kho ngữ liệu của riêng mình. Ngoài ra các người dùng khác khi muốn tải tài liệu về thì phải đăng ký tài khoản trên hệ thống và phải gửi đơn xin phép đến người dùng đăng tải, khi đã được cấp phép người dùng đó sẽ được tải tài liệu về. Hệ thống còn tích hợp thêm cả các bộ công cụ về thu thập dữ liệu từ nhiều nguồn khác nhau trên các website như wikipedia, foody.vn, facebook,… Mục đích của việc thu thập dữ liệu như ở trên đã nói đó là việc phục vụ cho cộng đồng có thể chỉnh sửa và tải dữ liệu về. Ngoài ra còn là đầu vào cho các bài toán xử lý ngôn ngữ tự nhiên như bài toán tách từ, gán nhãn từ loại, phân tích cú pháp, phân tích tình cảm, xác định thực thể tên… Hệ thống không chỉ thu thập các kho dữ liệu trên các trang web mà còn chia sẻ một lượng lớn các kho ngữ liệu sẵn có như: kho ngữ liệu hai triệu âm tiết đã tách từ, mười nghìn câu được gán nhãn từ loại, mười nghìn câu ngữ pháp. Hơn thế nữa, website còn tích hợp cả công cụ bán tự động cho phép xử lý các bài toán tách từ, gán nhãn từ loại. Về cơ bản sẽ đáp ứng được nhu cầu cho nhiều người làm việc cùng một lúc do việc xử lý được gọi từ một máy chủ độc lập. Trang web được xây dựng đầy đủ các phần từ việc quản trị các bộ ngữ liệu cho đến việc hiển thị giao diện ở phía người dùng có thể tương tác được.

Về mặt cấu trúc, báo cáo được chia thành 4 chương:

**Chương 1: Đặt vấn đề**

Chương này trình bày khái niệm cơ bản về xử lý ngôn ngữ tự nhiên, ngoài ra còn giới thiệu khái quát về ý tưởng cũng như ý nghĩa của một số bài toán xử lý ngôn ngữ tự nhiên cơ bản. Tiếp đến đề cập đến thực trạng các khó khăn hiện nay của việc nghiên cứu xử lý ngôn ngữ tự nhiên bằng tiếng Việt. Từ đó vạch định ra những công việc cần phải làm để giải quyết những vấn đề còn tồn đọng.

**Chương 2: Cơ sở lý thuyết**

Chương này sẽ trình bày về hướng tiếp cận cũng như các cách thức để giải quyết các bài toán. Với mỗi bài toán sẽ tập trung đi vào mục tiêu đạt được và trình bày khái quát cơ sở lý thuyết cho việc giải quyết bài toán đó.

**Chương 3: Phân tích – Thiết kế hệ thống**

Dựa vào yêu cầu bài toán đặt ra, chương này sẽ tập trung vào việc phân tích, thiết kế hệ thống và từ đó đưa ra cái nhìn tổng quan về cả hệ thống. Cấu trúc các thành phần của hệ thống, cách thức hoạt đông, ý nghĩa cũng như thao tác người dùng thực hiện. Bên cạnh đó, cũng trình bày rõ quá trình triển khai các công nghệ áp dụng vào hệ thống.

**Chương 4: Kết quả**

Nêu lên những nội dung đã đạt được tính đến thời điểm hiện tại, ưu điểm, nhược điểm, cách khắc phục. Ngoài ra đề xuất hướng phát triển để cải thiện hiệu năng cũng như tính năng của hệ thống.

# Đặt vấn đề

## Xử lý ngôn ngữ tự nhiên

Xử lý ngôn ngữ tự nhiên (NLP) là một nhánh của Trí tuệ nhân tạo, tập trung vào việc nghiên cứu sự tương tác giữa máy tính và ngôn ngữ tự nhiên của con người. Mục tiêu của lĩnh vực này là giúp máy tính hiểu và thực hiện hiệu quả những nhiệm vụ liên quan đến ngôn ngữ của con người như: tương tác giữa người và máy, cải thiện hiệu quả giao tiếp giữa con người với con người, hoặc đơn giản là nâng cao hiệu quả xử lý văn bản và lời nói[[1]](#footnote-2).

## Các bước xử lý

Theo wikipedia, các bước xử lý ngôn ngữ tự nhiên, các bài toán và ứng dụng sẽ được trình bày dưới đây:

**Phân tích hình thái** - Trong bước này từng từ sẽ được phân tích và các ký tự không phải chữ (như các dấu câu) sẽ được tách ra khỏi các từ. Trong tiếng Anh và nhiều ngôn ngữ khác, các từ được phân tách với nhau bằng dấu cách. Tuy nhiên trong tiếng Việt, dấu cách được dùng để phân tách các tiếng (âm tiết) chứ không phải từ. Cùng với các ngôn ngữ như tiếng Trung, tiếng Hàn, tiếng Nhật phân tách từ trong tiếng Việt là một công việc không hề đơn giản.

**Phân tích cú pháp** - Dãy các từ sẽ được biến đổi thành các cấu trúc thể hiện sự liên kết giữa các từ này. Sẽ có những dãy từ bị loại do vi phạm các luật văn phạm.

**Phân tích ngữ nghĩa** - Thêm ngữ nghĩa vào các cấu trúc được tạo ra bởi bộ phân tích cú pháp.

**Tích hợp văn bản** - Ngữ nghĩa của một câu riêng biệt có thể phụ thuộc vào những câu đứng trước, đồng thời nó cũng có thể ảnh hưởng đến các câu phía sau.

**Phân tích thực nghĩa** - Cấu trúc thể hiện điều được phát ngôn sẽ được thông dịch lại để xác định nó thật sự có nghĩa là gì.

Tuy nhiên, ranh giới giữa 5 bước xử lý này cũng rất mong manh. Chúng có thể được tiến hành từng bước một, hoặc tiến hành cùng lúc - tùy thuộc vào giải thuật và ngữ cảnh cụ thể.

## Các bài toán và ứng dụng

**Nhận dạng chữ viết**: Có hai kiểu nhận dạng, thứ nhất là nhận dạng chữ in, ví dụ nhận dạng chữ trên sách giáo khoa rồi chuyển nó thành dạng văn bản điện tử như dưới định dạng doc của Microsoft Word chẳng hạn. Phức tạp hơn là nhận dạng chữ viết tay, có khó khăn bởi vì chữ viết tay không có khuôn dạng rõ ràng và thay đổi từ người này sang người khác. Với chương trình nhận dạng chữ viết in có thể chuyển hàng ngàn đầu sách trong thư viện thành văn bản điện tử trong thời gian ngắn. Nhận dạng chữ viết của con người có ứng dụng trong khoa học hình sự và bảo mật thông tin (nhận dạng chữ ký điện tử).

**Nhận dạng tiếng nói:** Nhận dạng tiếng nói rồi chuyển chúng thành văn bản tương ứng. Giúp thao tác của con người trên các thiết bị nhanh hơn và đơn giản hơn, chẳng hạn thay vì gõ một tài liệu nào đó bạn đọc nó lên và trình soạn thảo sẽ tự ghi nó ra. Đây cũng là bước đầu tiên cần phải thực hiện trong ước mơ thực hiện giao tiếp giữa con người với robot. Nhận dạng tiếng nói có khả năng trợ giúp người khiếm thị rất nhiều.

**Tổng hợp tiếng nói:** Từ một văn bản tự động tổng hợp thành tiếng nói. Thay vì phải tự đọc một cuốn sách hay nội dung một trang web, nó tự động đọc cho chúng ta. Giống như nhận dạng tiếng nói, tổng hợp tiếng nói là sự trợ giúp tốt cho người khiếm thị, nhưng ngược lại nó là bước cuối cùng trong giao tiếp giữa robot với người.

**Dịch từ động** (machine translate): Như tên gọi đây là chương trình dịch tự động từ ngôn ngữ này sang ngôn ngữ khác. Một phần mềm điển hình về tiếng Việt của chương trình này là Evtrans của Softex, dịch tự động từ tiếng Anh sang tiếng Việt và ngược lại, phần mềm từng được trang web vdict.com mua bản quyền, đây cũng là trang đầu tiên đưa ứng dụng này lên mạng. Tháng 10 năm 2008 có hai công ty tham gia vào lĩnh vực này cho ngôn ngữ tiếng Việt là công ty Lạc Việt (công ty phát hành từ điển Lạc Việt) và Google, một thời gian sau đó Xalo\_vn cũng đưa ra dịch vụ tương tự.

**Tìm kiếm thông tin** (information retrieval): Đặt câu hỏi và chương trình tự tìm ra nội dung phù hợp nhất. Thông tin ngày càng đầy lên theo cấp số nhân, đặc biệt với sự trợ giúp của internet việc tiếp cận thông tin trở lên dễ dàng hơn bao giờ hết. Việc khó khăn lúc này là tìm đúng nhất thông tin mình cần giữa bề bộn tri thức và đặc biệt thông tin đó phải đáng tin cậy. Các máy tìm kiếm dựa trên giao diện web như Google hay Yahoo hiện nay chỉ phân tích nội dung rất đơn giản dựa trên tần suất của từ khoá và thứ hạng của trang và một số tiêu chí đánh giá khác để đưa ra kết luận, kết quả là rất nhiều tìm kiếm không nhận được câu trả lời phù hợp, thậm chí bị dẫn tới một liên kết không liên quan gì do thủ thuật đánh lừa của các trang web nhằm giới thiệu sản phẩm (có tên tiếng Anh là SEO viết tắt của từ search engine optimization). Thực tế cho đến bây giờ chưa có máy tìm kiếm nào hiểu được ngôn ngữ tự nhiên của con người trừ trang www.ask.com được đánh giá là "hiểu" được những câu hỏi có cấu trúc ở dạng đơn giản nhất. Mới đây cộng đồng mạng đang xôn xao về trang Wolfram Alpha, được hứa hẹn là có khả năng hiểu ngôn ngữ tự nhiên của con người và đưa ra câu trả lời chính xác. Lĩnh vực này hứa hẹn tạo ra bước nhảy trong cách thức tiếp nhận tri thức của cả cộng đồng.

**Tóm tắt văn bản:** Từ một văn bản dài tóm tắt thành một văn bản ngắn hơn theo mong muốn nhưng vẫn chứa những nội dung thiết yếu nhất.

**Khai phá dữ liệu** (data mining) và phát hiện tri thức: Từ rất nhiều tài liệu khác nhau phát hiện ra tri thức mới. Thực tế để làm được điều này rất khó, nó gần như là mô phỏng quá trình học tập, khám phá khoa học của con người, đây là lĩnh vực đang trong giai đoạn đầu phát triển. Ở mức độ đơn giản khi kết hợp với máy tìm kiếm nó cho phép đặt câu hỏi để từ đó công cụ tự tìm ra câu trả lời dựa trên các thông tin trên web mặc cho việc trước đó có câu trả lời lưu trên web hay không (giống như trang Yahoo! hỏi và đáp, nơi chuyên đặt các câu hỏi để người khác trả lời), nói một cách nôm na là nó đã biết xử lý dữ liệu để trả lời câu hỏi của người sử dụng, thay vì máy móc đáp trả những gì chỉ có sẵn trong bộ nhớ.

## Thực trạng hiện nay và hướng giải quyết

Về thực trạng tình hình nghiên cứu xử lý ngôn ngữ tự nhiên tiếng Việt hiện nay ở Việt Nam còn gặp 1 số khó khăn như: không có dữ liệu đủ lớn, thiếu những nghiên cứu nền tảng, ngoài ra không thể không nhắc đến việc khác nhau về cấu trúc, ngữ pháp, ngữ nghĩa giữa tiếng Việt và tiếng Anh nói riêng cũng như các ngôn ngữ khác nói chung do vậy dẫn đến sự nhập nhằng về mặt ngữ nghĩa, ngữ cảnh mà nó xuất hiện. Lớn nhất phải kể đến sự khó khăn về nhân sự, những người nắm giữ những kiến thức về xử lý ngôn ngữ tự nhiên tiếng Việt quả thực không nhiều, và cũng không có được 1 mạng lưới liên kết, trao đổi và hỗ trợ một cách hiệu quả.

Với những khó khăn kể ra ở trên, khóa luận này góp một phần nhỏ nào đấy trong việc xây dựng một hệ thống tích hợp các công cụ xử lý ngôn ngữ tự nhiên cho tiếng Việt. Ở đấy chia sẻ một số kho ngữ liệu sẵn có và một hệ thống các công cụ giúp cho người dùng có thể làm việc cùng nhau qua mạng, thu thập dữ liệu và xử lý dữ liệu với các thao tác đơn giản, dễ sử dụng trên giao diện web.

Tuy nhiên hệ thống vẫn còn 1 số giới hạn như: chỉ hỗ trợ 1 số công cụ về việc tách từ, gán nhãn từ loại, phân tích cú pháp, phân tích tình cảm. Do khả năng thực thi còn giới hạn cũng như thời gian khá gấp nên còn nhiều sai sót không tránh khỏi.

## Các bài toán cơ bản trong xử lý tiếng Việt

**Bài toán Tách từ:** Tách từ là một quá trình xử lý nhằm mục đích xác định ranh giới của các từ trong câu văn, cũng có thể hiểu đơn giản rằng tách từ là quá trình xác định các từ đơn, từ ghép… có trong câu. Bài toán này nằm trong bước xử lý phân tích cú pháp trong quá trình xử lý ngôn ngữ tự nhiên. Đối với xử lý ngôn ngữ, để có thể xác định cấu trúc ngữ pháp của câu, xác định từ loại của một từ trong câu, yêu cầu nhất thiết đặt ra là phải xác định được đâu là từ trong câu. Vấn đề này tưởng chừng đơn giản với con người nhưng đối với máy tính, đây là bài toán rất khó giải quyết.

**Bài toán Gán nhãn từ loại:** Một trong các vấn đề nền tảng của phân tích ngôn ngữ là việc phân loại các từ thành các lớp từ loại dựa theo thực tiễn hoạt động ngôn ngữ. Mỗi từ loại tương ứng với một hình thái và một vai trò ngữ pháp nhất định. Các bộ chú thích từ loại có thể thay đổi tuỳ theo quan niệm về đơn vị từ vựng và thông tin ngôn ngữ cần khai thác trong các ứng dụng cụ thể. Mỗi từ trong một ngôn ngữ nói chung có thể gắn với nhiều từ loại, và việc giải thích đúng nghĩa một từ phụ thuộc vào việc nó được xác định đúng từ loại hay không. Công việc gán nhãn từ loại cho một văn bản là xác định từ loại của mỗi từ trong phạm vi văn bản đó.[1]

**Bài toán phân tích cú pháp:** Việc phân tích cú pháp câu có thể chia làm hai mức chính. Mức thứ nhất là tách từ và xác định thông tin từ loại. Mức thứ hai là sinh cấu trúc cú pháp cho câu dựa trên các từ và từ loại do bước trước cung cấp.

**Bài toán phân tích tình cảm:** Phân tích tình cảm nhằm mục đích xác định thái độ của người nói, nhà văn, hoặc chủ đề khác liên quan đến một số chủ đề hoặc phân cực ngữ cảnh tổng thể hoặc phản ứng cảm xúc với tài liệu, tương tác hoặc sự kiện. Thái độ có thể là một phán quyết hoặc đánh giá, trạng thái cảm xúc (có nghĩa là trạng thái cảm xúc của tác giả hoặc diễn giả), hoặc giao tiếp cảm xúc dự định (có nghĩa là, tác động tình cảm của tác giả hoặc người đối thoại). Bài toán này nằm trong bước phân tích ngữ nghĩa của bài toán xử lý ngôn ngữ tự nhiên.

**Bài toán phân tích cú pháp phụ thuộc:** Cú pháp phụ thuộc là cấu trúc cú pháp chứa các mục từ vựng nối với nhau bởi các quan hệ nhị phân không đối xứng gọi là sự phụ thuộc [6]. Quan hệ phụ thuộc này có thể được đặt tên để làm rõ liên hệ giữa hai mục từ.

# Chương 2. Cơ sở lý thuyết

## Bài toán tách từ

Tách từ là một quá trình xử lý nhằm mục đích xác định ranh giới của các từ trong câu văn, cũng có thể hiểu đơn giản rằng tách từ là quá trình xác định các từ đơn, từ ghép… có trong câu. Đối với xử lý ngôn ngữ, để có thể xác định cấu trúc ngữ pháp của câu, xác định từ loại của một từ trong câu, yêu cầu nhất thiết đặt ra là phải xác định được đâu là từ trong câu. Vấn đề này tưởng chừng đơn giản với con người nhưng đối với máy tính, đây là bài toán rất khó giải quyết. Chính vì lý do đó tách từ được xem là bước xử lý quan trọng đối với các hệ thống Xử Lý Ngôn Ngữ Tự Nhiên, đặc biệt là đối với các ngôn ngữ thuộc vùng Đông Á theo loại hình ngôn ngữ đơn lập, ví dụ: tiếng Trung Quốc, tiếng Nhật, tiếng Thái, và tiếng Việt. Với các ngôn ngữ thuộc loại hình này, ranh giới từ không chỉ đơn giản là những khoảng trắng như trong các ngôn ngữ thuộc loại hình hòa kết như tiếng Anh…, mà có sự liên hệ chặt chẽ giữa các tiếng với nhau, một từ có thể cấu tạo bởi một hoặc nhiều tiếng. Vì vậy đối với các ngôn ngữ thuộc vùng Đông Á, vấn đề của bài toán tách từ là khử được sự nhập nhằng trong ranh giới từ.

Để hiểu đơn giản hơn về bài toán, ta có thể lấy 1 ví dụ đơn giản sau:

* Đầu vào:

*Tôi đã mua quyển sách mà thầy giáo giới thiệu.*

* Đầu ra:

*Tôi đã mua quyển sách mà thầy\_giáo giới\_thiệu.*

Như chúng ta đã biết, văn bản tiếng Việt đặt dấu cách giữa các âm tiết chứ không phải giữa các từ. Một từ có thể có một, hai hoặc nhiều âm tiết nên có nhiều cách phân chia các âm tiết thành các từ, gây ra nhập nhằng. Việc phân giải nhập nhằng này gọi là bài toán tách từ. Tiêu chí quan trọng nhất trong bài toán tách từ đương nhiên là độ chính xác. Hiện tại người ta đã đạt được độ chính xác lên đến 97% tính theo từ[[2]](#footnote-3). Tuy nhiên nếu tính theo câu (số câu được tách hoàn toàn đúng/tổng số câu) thì độ chính xác chỉ khoảng 50%[[3]](#footnote-4). Đây là vấn đề nghiêm trọng đối với các bước xử lý sau như phân tích ngữ pháp, ngữ nghĩa vì một từ bị tách sai có ảnh hưởng toàn bộ đến cách phân tích cả câu.

Ngoài ra tiêu chí độ chính xác tách từ mới cũng quan trọng với các ứng dụng thực tế. Tiếng Việt là một sinh ngữ – nó luôn luôn biến đổi. Các từ mới thuần Việt cũng như vay mượn được tạo ra hàng ngày. Nếu một ứng dụng không xử lý được những từ này thì hiệu năng của nó sẽ giảm dần theo thời gian.

Hiện tại bài toán tách từ có 3 phương pháp tiếp cận chính như sau: 3

* Tiếp cận dựa vào từ điển cố định.
* Tiếp cận dựa vào thống kê thuần túy.
* Tiếp cận dựa trên cả hai phương pháp trên.

Các phương pháp được sử dụng rất đa dạng và phong phú:3

* Ghép cực đại: Đặt các từ vào câu sao cho phủ hết được câu đó, thỏa mãn một số heuristic nhất định. Phương pháp này có ưu điểm là rất nhanh nhưng có rất nhiều hạn chế, ví dụ như độ chính xác thấp, không xử lý được những từ không có trong từ điển.
* Luật: Xây dựng tập luật bằng tay hoặc tự động để phân biệt các cách kết hợp được phép và không được phép.
* Đồ thị hóa: Xây dựng một đồ thị biểu diễn câu và giải bài toán tìm đường đi ngắn nhất trên đồ thị.
* Gán nhãn: Coi như bài toán gán nhãn chuỗi.
* Dùng mô hình ngôn ngữ: Cho trước một số cách tách từ của toàn bộ câu, một mô hình ngôn ngữ có thể đánh giá được cách nào có khả năng cao hơn.

Ngoài ra còn có thể kết hợp những phương pháp trên, ở đây em sẽ trình bày thuật toán phổ biến nhất đang được sử dụng đó là trọng số cực tiểu được quy về bài toán đồ thị như sau:[[4]](#footnote-5)

* Tạo ra 2 đỉnh ảo là start, và end (đầu và cuối câu).
* Lần lượt so sánh các đoạn với độ dài bất kì với 1 từ điển ngôn ngữ có sẵn.
* Các đoạn có xuất hiện trong từ điển sẽ tạo thành 1 đỉnh mới trên đồ thị.
* Trọng số giữa 2 đỉnh (2 đoạn phải liên tiếp nhau trong câu) được tính theo công thức f(i,j) với i,j là 2 từ.
* Tìm đường đi từ đỉnh start đến đỉnh end có trọng số nhỏ nhất trên đồ thị đó.

Trong bước 4, công thức f(i,j) thường được tính theo giá trị uni - gram(khả năng xuất hiện của 1 từ) và bi - gram(khả năng 2 từ xuất hiện liên tiếp nhau). Ngoài ra còn có thể có thêm 1 số yếu tố khác như từ loại, khả năng liên kết từ loại,... cũng được sử dụng trong hàm f.

Trước đây, các giá trị này (trừ uni - gram và bi - gram được lấy từ cách thống kê corpus) thường được đánh giá bằng tay (do người thực hiện). Nhưng với sự phát triển của lý thuyết máy học như Markov ẩn, CRFs,... các giá trị này thường được tính 1 cách tự động.

Trong bước 5, thuật toán tìm đường đi từ đỉnh start đến đỉnh end thường sử dụng thuật toán Viterbi với độ phức tạp thuật toán O(n) với n là độ dài câu cần tách.

Ngoài ra, có 1 vấn đề nảy sinh trong khi tách từ là việc xuất hiện các từ mời (các từ không được định nghĩa trong từ điển).

Đây là 1 vấn đề không thể bỏ qua khi ngôn ngữ là luôn luôn thay đổi và sinh ra các từ mới, trong khi từ điển (dành cho xử lý ngôn ngữ tự nhiên) không thể cập nhật hết được.

Các nghiên cứu đối với tiếng Nhật và tiếng Trung đã xử lý vấn đề này khá tốt. Với sự gần gũi về ngữ pháp và đặc điểm ngôn ngữ, ta có thể ứng dụng những nghiên cứu đó đối với tiếng Việt.

Dưới đây là một số công cụ tách từ có kết quả tốt cho tiếng Việt:

* vnTokenizer của tác giả Lê Hồng Phương được viết trên ngôn ngữ Java. Công cụ này sử dụng kết hợp từ điển và ngram, trong đó mô hình ngram được huấn luyện sử dụng treebank tiếng Việt (70,000 câu đã được tách từ) với độ chính xác trên 97%2. Ngoài ra công cụ còn kết hợp các phương pháp máy hữu hạn trạng thái, phân tích dạng chính tắc, và ghép cực đại.
* DongDu được viết trên ngôn ngữ C++ với độ chính xác cũng rất cao (>98%)[[5]](#footnote-6)
* JvnSegmenter là một công cụ phân đoạn từ tiếng Việt nguồn mở và dựa trên Java. Mô hình phân đoạn trong công cụ này được đào tạo trên khoảng 8.000 câu[[6]](#footnote-7) văn bản tiếng Việt có nhãn sử dụng các trường ngẫu nhiên có điều kiện.

## Bài toán gán nhãn từ loại

Một trong các vấn đề nền tảng của phân tích ngôn ngữ là việc phân loại các từ thành các lớp từ loại dựa theo thực tiễn hoạt động ngôn ngữ. Mỗi từ loại tương ứng với một hình thái và một vai trò ngữ pháp nhất định. Các bộ chú thích từ loại có thể thay đổi tuỳ theo quan niệm về đơn vị từ vựng và thông tin ngôn ngữ cần khai thác trong các ứng dụng cụ thể. Mỗi từ trong một ngôn ngữ nói chung có thể gắn với nhiều từ loại, và việc giải thích đúng nghĩa một từ phụ thuộc vào việc nó được xác định đúng từ loại hay không. Công việc gán nhãn từ loại cho một văn bản là xác định từ loại của mỗi từ trong phạm vi văn bản đó[1]. Khi hệ thống văn bản đã được gán nhãn, hay nói cách khác là đã được chú thích từ loại thì nó sẽ được ứng dụng rộng rãi trong các hệ thống tìm kiếm thông tin, trong các ứng dụng tổng hợp tiếng nói, các hệ thống nhận dạng tiếng nói cũng như trong các hệ thống dịch máy.

Quá trình gán nhãn từ loại có thể chia làm 3 bước[1]:

* Phân tách xâu kí tự thành chuỗi các từ. Giai đoạn này có thể đơn giản hay phức tạp tuỳ theo ngôn ngữ và quan niệm về đơn vị từ vựng. Chẳng hạn đối với tiếng Anh hay tiếng Pháp, việc phân tách từ phần lớn là dựa vào các kí hiệu trắng. Tuy nhiên vẫn có những từ ghép hay những cụm từ công cụ gây tranh cãi về cách xử lí. Trong khi đó với tiếng Việt thì dấu trắng càng không phải là dấu hiệu để xác định ranh giới các đơn vị từ vựng do tần số xuất hiện từ ghép rất cao.
* Gán nhãn tiên nghiệm, tức là tìm cho mỗi từ tập tất cả các nhãn từ loại mà nó có thể có. Tập nhãn này có thể thu được từ cơ sở dữ liệu từ điển hoặc kho văn bản đã gán nhãn bằng tay. Đối với một từ mới chưa xuất hiện trong cơ sở ngữ liệu thì có thể dùng một nhãn ngầm định hoặc gắn cho nó tập tất cả các nhãn. Trong các ngôn ngữ biến đổi hình thái người ta cũng dựa vào hình thái từ để đoán nhận lớp từ loại tương ứng của từ đang xét.
* Quyết định kết quả gán nhãn, đó là giai đoạn loại bỏ nhập nhằng, tức là lựa chọn cho mỗi từ một nhãn phù hợp nhất với ngữ cảnh trong tập nhãn tiên nghiệm. Có nhiều phương pháp để thực hiện việc này, trong đó người ta phân biệt chủ yếu các phương pháp dựa vào quy tắc ngữ pháp mà đại diện nổi bật là phương pháp Brill ([5]) và các phương pháp xác suất. Ngoài ra còn có các hệ thống sử dụng mạng nơ - ron ([8]), các hệ thống lai sử dụng kết hợp tính toán xác suất và ràng buộc ngữ pháp, gán nhãn nhiều tầng.

Về mặt ngữ liệu, các phương pháp phân tích từ loại thông dụng hiện nay dùng một trong các loại tài nguyên ngôn ngữ sau:[1]

* Từ điển và các văn phạm loại bỏ nhập nhằng.
* Kho văn bản đã gán nhãn, có thể kèm theo các quy tắc ngữ pháp xây dựng bằng tay.
* Kho văn bản chưa gán nhãn, có kèm theo các thông tin ngôn ngữ như là tập từ loại và các thông tin mô tả quan hệ giữa từ loại và hậu tố.
* Kho văn bản chưa gán nhãn, với tập từ loại cũng được xây dựng tự động nhờ các tính toán thống kê. Trong trường hợp này khó có thể dự đoán trước về tập từ loại.

Các hướng tiếp cận giải bài toán: Chúng ta có hai hướng tiếp cận chính cho gán nhãn từ loại tự động:[6]

* Tiếp cận có hướng dẫn.
* Tiếp cận không hướng dẫn.

Bộ gán nhãn có hướng dẫn có đặc thù là dựa trên kho ngữ liệu đã được gán nhãn cho việc tạo ra các công cụ được sử dụng cho quá trình gán nhãn. Ví dụ như là Từ điển bộ gán nhãn, các tần suất từ/nhãn, các xác suất chuỗi nhãn, tập các luật.

Các mô hình không hướng dẫn không yêu cầu kho ngữ liệu đã gán nhãn nhưng lại sử dụng các thuật toán tính toán phức tạp để tự động xây dựng các nhóm từ (nghĩa là xây dựng các tập nhãn) và dựa trên các nhóm từ này để tính toán các thông tin xác suất cần thiết cho các bộ gán nhãn thống kê hoặc để xây dựng các luật ngữ cảnh cần thiết cho các hệ thống dựa trên luật.[4]

Gán nhãn dựa trên luật

* Gán nhãn dựa trên luật sử dụng từ điển để tìm các từ loại có thể cho các từ, sử dụng các luật làm thành một nghĩa.
* Các tiếp cận gán nhãn dựa trên luật sử dụng thông tin ngữ cảnh để gán các nhãn cho các từ chưa biết hoặc các từ nhập nhằng.

Gán nhãn thống kê

Bộ gán nhãn thống kê đơn giản nhất giải quyết nhập nhằng các từ chỉ đặt cơ sở vào xác suất mà một từ xuất hiện với một nhãn đặc biệt. Nói cách khác, nhãn được gặp thường xuyên nhất trong tập huấn luyện là nhãn được gán cho một thể hiện không rõ ràng của từ đó. Vấn đề với tiếp cận này là trong khi nó có thể mang lại một nhãn hợp lệ cho một từ đưa ra, lại cũng có thể mang lại chuỗi không hợp lệ các nhãn.

Các từ chưa biết

Có vài giải pháp tiềm năng cho vấn đề này: Một trong những giải pháp đó là sử dụng thông tin hình thái. Trong trường hợp này, bộ gán nhãn tính toán xác suất mà một hậu tố trên một từ chưa biết xuất hiện với một nhãn đặc biệt. Nếu một mô hình Markov ẩn đang được sử dụng, xác suất mà một từ chứa đựng hậu tố mà xuất hiện với một nhãn đặc biệt trong chuỗi đã cho được tính toán. Một giải pháp khác là gán một tập các nhãn mặc định (các lớp mở đặc biệt: Danh từ, tính từ, trạng từ, động từ..) cho các từ chưa biết và để giải quyết nhập nhằng sử dụng các xác suất mà các nhãn đó xuất hiện tại cuối n-gram trong câu hỏi.

Bài toán gán nhãn từ loại tiếng Việt

Qua khảo sát các nghiên cứu gần đây của tiếng Việt cho bài toán gán nhãn từ loại, có thể thấy có hai dạng tập nhãn từ loại thường được sử dụng cho các công cụ gán nhãn từ loại tiếng Việt:[4]

* Dạng thứ nhất, xuất phát từ tập gồm 8 nhãn từ loại tiếng Việt thông dụng được các− nhà nghiên cứu ngôn ngữ học công nhận nhiều nhất (bao gồm: danh từ, động từ, tính từ, đại từ, phụ từ, kết từ, trợ từ, cảm từ) để xây dựng tập nhãn “mịn” hơn bằng cách phân nhỏ mỗi từ loại trên thành các tiểu từ loại.
* Dạng thứ hai, tập nhãn tiếng Việt được xây dựng thông qua việc xây dựng kho ngữ liệu song ngữ Anh - Việt mà trong đó các câu tiếng Việt đã được gán nhãn từ loại chính xác nhờ kết quả liên kết từ Anh-Việt và phép chiếu từ loại từ Anh sang Việt.

Dưới đây là ví dụ đơn giản về bài toán gán nhãn từ loại và một số cơ sở toán học để giải quyết bài toán:

* Đầu vào:

*Đi tìm tàu phải không?*

* Đầu ra:

*Đi/V tìm/V tàu/N phải/V không/R ?/?*

Định lý Bayes

Cho phép tính xác suất xảy ra của một sự kiện ngẫu nhiên A khi biết sự kiện liên quan B đã xảy ra. Xác suất này được ký hiệu là P(A|B), và đọc là “xác suất của A nếu có B”. Đại lượng này được gọi xác suất có điều kiện hay xác suất hậu nghiệm vì nó được rút ra từ giá trị được cho của B hoặc phụ thuộc vào giá trị đó.

Thuật toán cực đại hóa kỳ vọng (EM)

Thuật toán EM (Expectation Maximization) nhằm tìm ra sự ước lượng về khả năng lớn nhất của các tham số trong mô hình xác suất (các mô hình phụ thuộc vào các biến ẩn chưa được quan sát), nó được xem như thuật toán dựa trên mô hình.

Mô hình Markov ẩn

Mô hình Markov ẩn (tiếng Anh là Hidden Markov Model - HMM) là mô hình thống kê trong đó hệ thống được mô hình hóa được cho là một quá trình Markov với các tham số không biết trước và nhiệm vụ là xác định các tham số ẩn từ các tham số quan sát được, dựa trên sự thừa nhận này

Dưới đây là một số công cụ có kết quả tốt cho việc gán nhãn từ loại tiếng Việt:

* vnTagger được phát triển bởi tác giả Lê Hồng Phương sử dụng phương pháp tiếp cận entropy tối đa với độ chính xác khoảng 94%[[7]](#footnote-8) dựa trên bộ dữ liệu test (Vietnam treebank).
* vnQTag là công cụ gán nhãn từ loại cho văn bản tiếng Việt đã được tách từ được phát triển bởi nhóm tác giả Nguyễn Thị Minh Huyền, Vũ Xuân Lương và Lê Hồng Phương dựa theo phương pháp xác suất.[[8]](#footnote-9)
* JvnTagger – Phan Xuân Hiếu được xây dựng theo phương pháp học máy thống kê Maximum Entropy và Conditional Random Fields.[[9]](#footnote-10)

Ở khóa luận này sử dụng bộ công cụ của tác giả Lê Hồng Phương do đạt được độ chính xác cao trên tập dữ liệu lớn và ngoài ra cũng có sẵn tài liệu hướng dẫn sử dụng cũng như được viết bằng ngôn ngữ Java rất phù hợp cho việc triển khai hệ thống.

Như vậy, có thể thấy rằng bài toán gán nhãn từ loại cho tiếng Việt đang ngày càng được quan tâm nghiên cứu. Tuy nhiên đây vẫn là hướng nghiên cứu đầy tiềm năng và cũng đầy thử thách, cùng với đó là việc các nghiên cứu đã có hầu hết vẫn còn mang tính cá thể, chưa có được sự đối chiếu so sánh khách quan, và sự thống nhất về bộ nhãn giữa các nhà ngôn ngữ, đồng thời cũng chưa xây dựng được bộ nhãn đủ lớn để bài toán gán nhãn tiếng Việt có thể đạt độ chính xác cao.

## Bài toán phân tích cú pháp

Phân tích cú pháp là bước xử lý quan trọng trong các bài toán hiểu ngôn ngữ tự nhiên. Nó cung cấp một nền tảng vững chắc cho việc xử lý văn bản thông minh như các hệ thống hỏi đáp, khai phá văn bản và dịch máy.

Việc phân tích cú pháp câu có thể chia làm hai mức chính:[2]

* Mức thứ nhất là tách từ và xác định thông tin từ loại.
* Mức thứ hai là sinh cấu trúc cú pháp cho câu dựa trên các từ và từ loại do bước trước cung cấp.

Do tiếng Việt là ngôn ngữ đơn âm tiết nên chúng ta thường gặp phải vấn đề nhập nhằng ở cả hai mức.

Để hiểu đơn giản hơn về ý tưởng của bài toán chúng ta có thể xem ví dụ dưới đây:

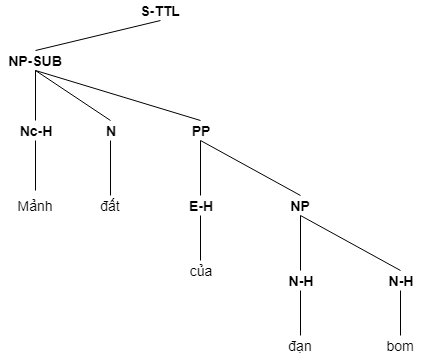
* Đầu vào:

*Mảnh đất của đạn bom*

* Đầu ra:

*(S-TTL (NP-SUB (Nc-H Mảnh) (N đất) (PP (E-H của) (NP (N-H đạn) (N-H bom))))*

Và với kết quả đầu ra này chúng ta có thể xây dựng được cây như sau (treebank)



Về nguyên tắc, các thông tin về từ có thể được chứa trong nhãn từ loại bao gồm: từ loại cơ sở (danh từ, động từ, v.v.), thông tin hình thái (số ít, số nhiều, thì, ngôi, v.v.), thông tin về phân loại con (ví dụ động từ đi với danh từ, động từ đi với mệnh đề, v.v.), thông tin ngữ nghĩa, hay một số thông tin cú pháp khác.

Thông qua việc tham khảo bài báo về xây dựng treebank tiếng Việt [3], dưới đây là bảng danh sách về tập nhãn từ loại, tập nhãn cú pháp:

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên** | **Chú thích** |
| 1 | N | Danh từ |
| 2 | Np | Danh từ riêng |
| 3 | Nc | Danh từ chỉ loại |
| 4 | Nu | Danh từ đơn vị |
| 5 | V | Động từ |
| 6 | A | Tính từ |
| 7 | P | Đại từ |
| 8 | L | Định từ |
| 9 | M | Số từ |
| 10 | R | Phụ từ |
| 11 | E | Giới từ |
| 12 | C | Liên từ |
| 13 | I | Thán từ |
| 14 | T | Trợ từ, tiểu từ, từ tình thái |
| 15 | U | Từ đơn lẻ |
| 16 | Y | Từ viết tắt |
| 17 | X | Các từ không phân loại được |

*Bảng 1. Tập nhãn từ loại*

Tập nhãn cú pháp mô tả các thành phần cú pháp cơ bản là cụm từ và mệnh đề. Nhãn thành phần cú pháp là thông tin cơ bản nhất trên cây cú pháp, nó tạo thành xương sống của câycú pháp. Tập nhãn của các ngôn ngữ khác nhau là khác nhau do 2 nguyên nhân sau:

* Thứ nhất: nguyên nhân cơ bản nhất là do sự khác biệt về ngôn ngữ.
* Thứ hai: do kỹ thuật thiết kế tập nhãn.

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên** | **Chú thích** |
|  | S | Câu trần thuật (khẳng định hoặc phủ định) |
|  | SQ | Câu hỏi |
|  | SBAR | Mệnh đề phụ (bổ nghĩa cho danh từ, động từ, và tính từ) |

*Bảng 2. Tập nhãn mệnh đề*

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên** | **Chú thích** |
|  | NP | Cụm danh từ |
|  | VP | Cụm động từ |
|  | AP | Cụm tính từ |
|  | RP | Cụm phụ từ |
|  | PP | Cụm giới từ |
|  | QP | Cụm từ chỉ số lượng |
|  | MDP | Cụm từ tình thái |
|  | WHNP | Cụm danh từ nghi vấn (ai, cái gì, con gì, v.v.) |
|  | WHAP | Cụm tính từ nghi vấn (lạnh thế nào, đẹp ra sao, v.v.) |
|  | WHRP | Cụm từ nghi vấn dùng khi hỏi về thời gian, nơi chốn, v.v. |
|  | WHPP | Cụm giới từ nghi vấn (với ai, bằng cách nào, v.v.) |

*Bảng 3. Tập nhãn cụm từ*

Nhãn chức năng của một thành phần cú pháp cho biết vai trò của nó trong thành phần cú pháp mức cao hơn. Nhãn chức năng cú pháp được gán cho các thành phần chính trong câu như chủ ngữ, vị ngữ, tân ngữ. Nhờ thông tin do nhãn chức năng cung cấp ta có thể xác định các loại quan hệ ngữ pháp cơ bản sau đây:

* Chủ - vị, đề - thuyết, phần chêm, bổ ngữ, sự kết hợp.

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên** | **Chú thích** |
| 1 | SUB | Nhãn chức năng chủ ngữ |
| 2 | DOB | Nhãn chức năng tân ngữ trực tiếp |
| 3 | IOB | Nhãn chức năng tân ngữ gián tiếp |
| 4 | TPC | Nhãn chức năng chủ đề |
| 5 | PRD | Nhãn chức năng vị ngữ không phải cụm động từ |
| 6 | LGS | Nhãn chức năng chủ ngữ logic của câu ở thể bị động |
| 7 | EXT | Nhãn chức năng bổ ngữ chỉ phạm vi hay tần suất của hành động |
| 8 | H | Nhãn phần tử trung tâm (của cụm từ hoặc mệnh đề) |
| 9 -12 | TC, CMD, EXC, SPL | Nhãn phân loại câu: đề - thuyết, mệnh lệnh, cảm thán, đặc biệt |
| 13 | TTL | Tít báo hay tiêu đề |
| 14 | VOC | Thành phần than gọi |

*Bảng 4. Nhãn chức năng cú pháp*

Ngoài ra nhãn chức năng cũng có thể tương ứng với một loại trạng ngữ nào đó như thời gian, nơi chốn, hay mục đích. Bảng 5 liệt kê các nhãn chức năng trạng ngữ.

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên** | **Chú thích** |
| 1 | TMP | Nhãn chức năng trạng ngữ chỉ thời gian |
| 2 | LOC | Nhãn chức năng trạng ngữ chỉ nơi chốn |
| 3 | DIR | Nhãn chức năng trạng ngữ chỉ hướng |
| 4 | MNR | Nhãn chức năng trạng ngữ chỉ cách thức |
| 5 | PRP | Nhãn chức năng trạng ngữ chỉ mục đích hay lý do |
| 6 | ADV | Nhãn chức năng trạng ngữ nói chung (dùng khi trạng ngữ không thuộc một trong các loại cụ thể trên) |

*Bảng 5. Nhãn chức năng trạng ngữ*

Nhãn thành phần rỗng:

* Đây là một loại thành phần khá đặc biệt.
* Nó chỉ ra sự tồn tại (được ngầm hiểu) của một thành phần cú pháp cho dù nó không xuất hiện ở vị trí đó.
* Thông thường thành phần rỗng được gán chỉ số của thành phần mà nó đại diện.

Vấn đề mức sinh cấu trúc cú pháp danh ngữ hiện vẫn còn đang nhập nhằng, có thể đúng trong trường hợp này nhưng lại không đúng trong các trường hợp khác.

Để giải quyết vấn đề này, chúng ta cần đưa thông tin cú pháp và ngữ nghĩa vào tập luật văn phạm bằng cách sử dụng văn phạm cấu trúc đoạn hướng trung tâm (Head - Driven Pharse Structure Grammar - HPSG) [2]. Văn phạm này cho phép biểu diễn các mối quan hệ giữa các từ, và làm tăng ràng buộc kết hợp.

HPSG tạm dịch là văn phạm cấu trúc đoạn hướng trung tâm, do Carl Pollard và Ivan Sag đưa ra với mục đích xây dựng một học thuyết khoa học về khả năng hiểu ngôn ngữ nói. HPSG có thể được nhìn nhận như một sự mở rộng của văn phạm phi ngữ cảnh (context free grammar – CFG) bằng việc thêm vào các thuộc tính trong cấu trúc mô tả từ và các ràng buộc trong các luật cú pháp. Khi đó quá trình phân tích cú pháp sẽ là sự kết hợp giữa luật cú pháp và những ràng buộc ngữ nghĩa.

HPSG có hai đặc điểm chính:

* HPSG sử dụng cấu trúc thuộc tính để biểu diễn các thông tin về từ. Cấu trúc này thường được mô tả dưới dạng một ma trận giá trị thuộc tính (attribute – value - matrix (AVM)), nhằm mô tả các đặc tính cụ thể của từ như các thông tin cú pháp và ngữ nghĩa.
* HPSG tích hợp các ràng buộc về cú pháp và ngữ nghĩa vào tập luật. Các ràng buộc này được dùng để kiểm soát các quan hệ cú pháp và ngữ nghĩa giữa các từ/ngữ trong câu.

Trên đây chỉ là trình bày những khái niệm cơ bản cũng như cách thức tiếp cận bài toán phân tích cú pháp. Ở khóa luận này em sẽ không đi sâu vào một giải thuật nào đấy mà chỉ trình bày cơ sở lý thuyết của bài toán.

## Bài toán phân tích tình cảm

Ngoài các các ứng dụng từ các bài toán nêu trên thì bài toán phát hiện được cảm xúc muốn truyền tải thông qua từ ngữ trong một văn bản cũng là một bài toán quan trọng trong xử lý ngữ nghĩa văn bản.

Phần nội dung dưới đây được trình bày lại dựa trên bài báo: “A SUPERVISED METHOD FOR ASPECT BASED SENTIMENT ANALYSIS”[9]

Phân tích cảm xúc được chia thành nhiều cấp độ trên cụm từ, câu văn, đoạn văn hoặc cả văn bản:

* Phân tích thái độ, đánh giá, thông qua phân loại cảm xúc tích cực, tiêu cực, trung tính.
* Phân tích và phát hiện cảm xúc phức tạp của người viết như buồn, vui, giận dữ.

Ngày nay, sự phát triển nhanh chóng của Internet mang lại nhiều cơ hội và thách thức cho các tổ chức cung cấp các sản phẩm hoặc dịch vụ khác nhau cho mọi người. Ngoài ra, khối lượng người dùng Internet đang phát triển nhanh chóng và có xu hướng tăng thêm trong tương lai gần. Thông thường mọi người tham khảo sản phẩm (máy tính xách tay, điện thoại,…) hoặc dịch vụ (khách sạn, nhà hàng,...) đánh giá trên Internet trước khi đưa ra quyết định. Bên cạnh đó, có nhiều cá nhân bị ảnh hưởng bởi người khác đánh giá.

Hơn nữa, việc biết thích và không thích của người dùng có thể là sự giúp đỡ rất nhiều trong việc phát triển sản phẩm mới. Vì lý do đó, đánh giá của người dùng đóng một vai trò quan trọng và trở thành một giá trị thông tin cho các công ty, nhà cung cấp quan tâm đến ý kiến của người dùng.

Trang web là một trong những nền tảng tuyệt vời cho người dùng ngay lập tức chia sẻ nhận xét hoặc trải nghiệm của họ về một số môn học. Mọi người có thể tự do bày tỏ ý kiến của họ về những gì họ muốn nói trên các trang web như bình luận, đánh giá trên các thiết bị điện tử. Do đó, số lượng việc xem xét người dùng ngày càng tăng.

Đối với các công ty thương mại điện tử, chăm sóc phản hồi của người dùng là điều cần thiết và họ thường có một nhóm để phân tích và đánh giá nhận xét của người dùng.

Tuy nhiên, với nguồn dữ liệu lớn hiện có thì việc phân tích thủ công là không khả thi. Phân tích tình cảm (Sentiment Analysis) là tiểu lĩnh vực xử lý ngôn ngữ tự nhiên nhằm mục đích trích xuất và phân tích thông tin chủ quan từ ý kiến, nhận xét

hoặc đánh giá được chia sẻ bởi con người. Do đó, phân tích tình cảm được nghiên cứu rất sớm trên thế giới và tiếng Việt, chủ đề nghiên cứu này đã trở thành xu hướng từ năm 2010.

Để phân tích sâu hơn cảm xúc phức tạp được thể hiện trong một văn bản, mô hình phân tích tình cảm dựa trên khía cạnh (Aspect-Based Sentiment Analysis) được phát triển. Mô hình này nhận đầu vào là một đánh giá sau đó phát hiện những khía cạnh chính mà đánh giá đề cập (ở đây có thể là thực thể hoặc thực thể và thuộc tính của nó tùy mức độ sâu của phân tích), từ đó phân loại thái cực của cảm xúc(tích cực, tiêu cực) hoặc cảm xúc cụ thể (buồn, vui, giận) của khía cạnh đó.

Theo những nghiên cứu trước cho tiếng Anh (Tog & Wang, 2014) và tiếng Viêt (Anh & Anh, 2018), mô hình SVM(Support Vector Machine) cho kết quả tốt nhất với những bài toán phân tích tình cảm. Mạng nơ-ron được phát triển, sử dụng rộng rãi trong thời gian gần đây để xử lý các bài toán liên quan đến trí tuệ nhân tạo nói chung.

Cách thức giải quyết:

* Trong tiếng Anh đã có nhiều nghiên cứu cho việc giải quyết vấn đề phân tích tình cảm trong các đánh giá. Các đặc trưng được họ lựa chọn thường là: ngram, postag, quan hệ phụ thuộc, từ điển… Và áp dụng chúng vào các mô hình. Hiện tại mô hình đang cho kết quả tốt nhất là mô hình SVM (Support Vector Machine).
* Trong tiếng Việt gần tương tự như tiếng Anh mô hình linear SVM (Support Vector Machine) cho kết quả cao nhất với các đặc trưng được lựa chọn là:
  + Trong phát hiện thực thể: ngram, postag, N, V, A.
  + Trong phát hiện phân cực: ngram, từ đặc biệt, thực thể, #hagtag, #POS, dấu!?.

# Chương 3. Phân tích – thiết kế hệ thống

## Kịch bản

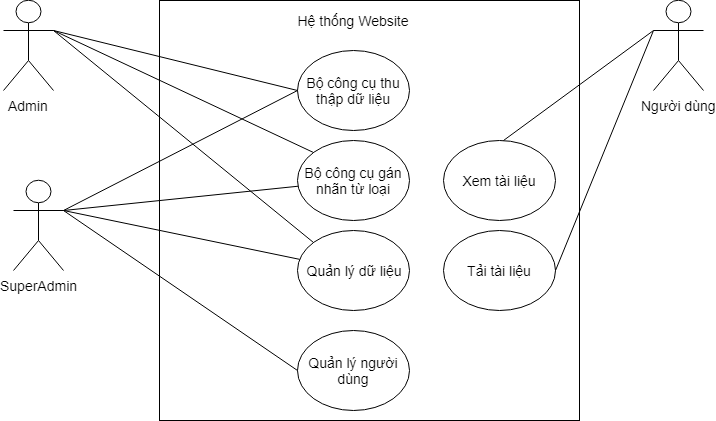
Với một chiếc máy tính có kết nối mạng là chúng ta có thể truy cập vào trang web và thao tác trên đó. Trang web được phân quyền thành 2 đối tượng sử dụng chính: quản trị viên và người dùng. Với mỗi một đối tượng sử dụng sẽ có vai trò cũng như cách thức và mục đích khác nhau.

Về phía người dùng: với một người dùng thông thường có thể truy cập vào trang web và xem thông tin về các kho ngữ liệu đã được cung cấp sẵn từ phía quản trị. Ngoài ra để có thể tải kho ngữ liệu về thì người dùng trước hết phải đăng ký cho mình một tài khoản trên hệ thống. Sau khi nhận được email kích hoạt tài khoản từ hệ thống, người dùng tiếp tục phải gửi đơn đăng ký với thông tin cá nhân như là email, số điện thoại và lý do muốn tải tài liệu thông qua giao diện form rất đơn giản và dễ sử dụng. Tiếp sau đó, người quản trị viên sẽ lọc thông tin và xác thực người dùng nếu đúng người dùng và mục đích sử dụng chính đáng thì khi đấy sẽ kích hoạt cho phép người dùng đó được tải tài liệu tương ứng về.

Về phía Admin: vai trò của người admin ở đây đơn giản là những người dùng cao cấp hơn và họ có thể đăng tải bộ kho ngữ liệu của mình lên. Mỗi một người quản trị cao cấp sẽ được phép xem toàn bộ tài liệu cũng như các bộ kho ngữ liệu của mình và của cả các thành viên quan trị cao cấp khác đăng tải lên. Tuy nhiên họ không được phép sửa, xóa các bộ ngữ liệu của các thành viên khác. Bên cạnh đó người dùng cao cấp còn được phép đánh giá và đề xuất chỉnh sửa tài liệu của những người dùng khác. Và đặc biệt hơn cả là họ được sử dụng các bộ công cụ thu thập dữ liệu từ các trang web trên mạng và các bộ công cụ chỉnh sửa với thao tác kéo thả rất thuận tiện và dễ sử dụng phù hợp với từng loại ngữ liệu khác nhau. Mục đích chính của người quản trị cao cấp ở đây là xử lý và thu thập dữ liệu dựa trên các bộ công cụ đã được xây dựng và tích hợp sẵn trên hệ thống, để từ đó phục vụ cho nhu cầu của các người dùng khác về việc xem và tải tài liệu phù hợp về để nghiên cứu cũng như là bước cơ bản phục vụ cho các bài toán xử lý ngôn ngữ tự nhiên phức tạp và nâng cao hơn.

Về phía SuperAdmin : có vai trò tương tự như admin nhưng có thêm quyền cao hơn đó là được phép quản trị tài khoản của các người dùng khác với các hành vi thêm, sửa, xóa, tìm kiếm.

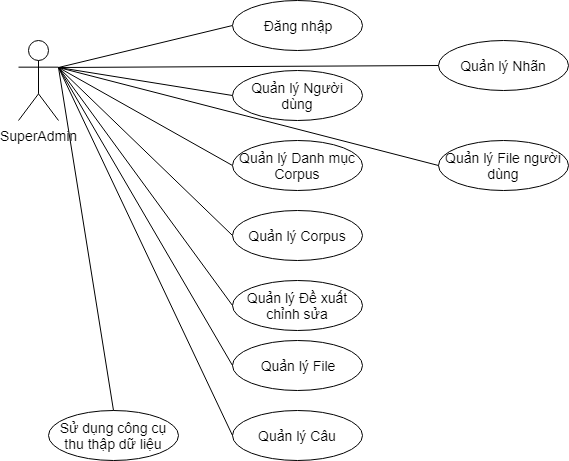
## Mô hình use case của hệ thống

****

*Mô hình use case mức đỉnh của hệ thống*

Trong mô hình use case mức đỉnh này có các thành phần: 3 actor chính sẽ tương tác với hệ thống là *Người dùng, Admin, SuperAdmin*; một use case tổng quát tương tác với tất cả các actor nói trên là *Hệ thống website.*

Các chức năng của hệ thống được thể hiện chi tiết qua các sơ đồ use case dưới đây:



*Mô hình use case mô tả các chức năng của actor SuperAdmin*

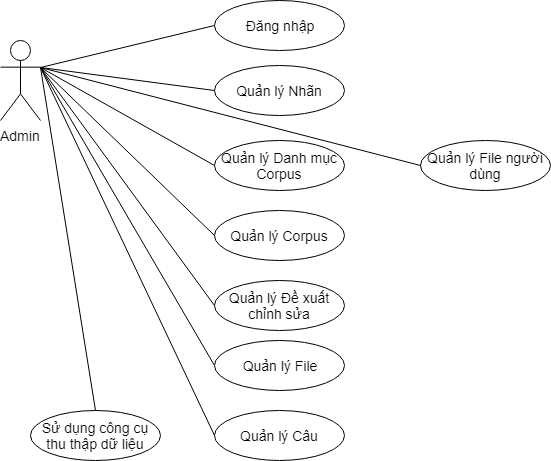
SuperAdmin là người có chức năng có quyền cao nhất sau khi đăng nhập có thể thêm, sửa, xóa thông tin của các tài khoản khác.

Ngoài ra còn có chức năng tương tự như admin về việc quản lý cũng như sử dụng công cụ thu thập dữ liệu. Đặc biệt ở phần quản lý câu sẽ có bộ công cụ chỉnh sửa phù hợp với từng loại tài liệu.

* Bộ công cụ chỉnh sửa kéo thả tách từ.
* Bộ công cụ chỉnh sửa kéo thả gán nhãn từ loại.
* Bộ công cụ chỉnh sửa kéo thả phân tích cú pháp theo cấu trúc cây.

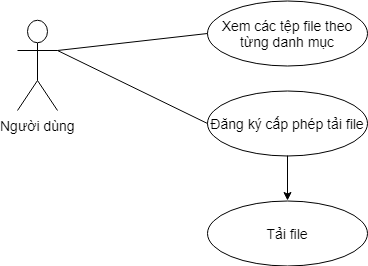
Quản lý File có thêm một số chức năng sau:

* Đánh giá bình luận (thích, không thích, trung tính).
* Tiền xử lý dữ liệu với bộ công cụ bán tự động cho việc tách từ.
* Đề xuất chỉnh sửa File.



*Mô hình use case mô tả các chức năng của actor Admin*

Với mỗi một module quản lý sẽ bao gồm các chức năng: thêm, sửa, xóa, tìm kiếm.



*Mô hình use case mô tả các chức năng của actor Người dùng*

## Mô tả thực thể

### Người dùng

|  |  |
| --- | --- |
| Tên thuộc tính | Diễn giải |
| group\_id | Nhóm người dùng  1 - Superadmin, 2 - Người dùng, 3 - Admin |
| password | Mật khẩu |
| name | Họ và tên |
| age | Tuổi |
| phone | Số điện thoại (dùng làm tên đăng nhập) |
| email | Email |
| address | Địa chỉ |
| sex | Giới tính 0 - Nam, 1 - Nữ, 2 - Không xác định |
| confirmed | Trạng thái 1 - Duyệt, 0 - Chưa duyệt, 2 - Khóa |
| confirmation\_code | Mã xác nhận kích hoạt tài khoản |

*Các thuộc tính của người dùng*

### Danh mục tài liệu

|  |  |
| --- | --- |
| Tên thuộc tính | Diễn giải |
| name | Tên danh mục tài liệu |
| description | Mô tả danh mục tài liệu |
| orderSort | Thứ tự hiển thị danh mục |

*Các thuộc tính của danh mục tài liệu*

### Tài liệu

|  |  |
| --- | --- |
| Tên thuộc tính | Diễn giải |
| name | Tên danh mục tài liệu |
| description | Mô tả danh mục tài liệu |
| type | Loại tài liệu |

*Các thuộc tính của tài liệu*

### File

|  |  |
| --- | --- |
| Tên thuộc tính | Diễn giải |
| name | Tên file |
| description | Mô tả file |
| summary | Tóm tắt file |
| content | Nội dung file |
| view | Lượt xem |
| link\_download | Liên kết tải file |
| file | Đường dẫn lưu file |
| point | Điểm đánh giá file |
| status | Trạng thái (hiển thị, ẩn) |
| like | Lượt thích |
| dislike | Lượt không thích |
| neutral | Lượt trung tính |

*Các thuộc tính của file*

### Câu

|  |  |
| --- | --- |
| Tên thuộc tính | Diễn giải |
| content | Nội dung câu |
| point | Điểm đánh giá câu |

*Các thuộc tính của câu*

### Đề xuất chỉnh sửa

|  |  |
| --- | --- |
| Tên thuộc tính | Diễn giải |
| content | Nội dung đề xuất chỉnh sửa |
| status | Trạng thái (Hiển thị, ẩn) |

*Các thuộc tính của đề xuất chỉnh sửa*

### Nhãn từ loại

|  |  |
| --- | --- |
| Tên thuộc tính | Diễn giải |
| name | Tên nhãn từ loại |
| description | Mô tả nhãn từ loại |
| type | Loại nhãn từ loại |

*Các thuộc tính của nhãn từ loại*

## Mối quan hệ giữa các thực thể

**ER.png**

*Mối quan hệ giữa Danh mục tài liệu và Tài liệu*

*ER (1).png*

*Mối quan hệ giữa File và Tài liệu*

*ER (3).png*

*ER (5).png*

*Mối quan hệ giữa Người dùng và File*

*ER (4).png*

*Mối quan hệ giữa File và Đề xuất chỉnh sửa*

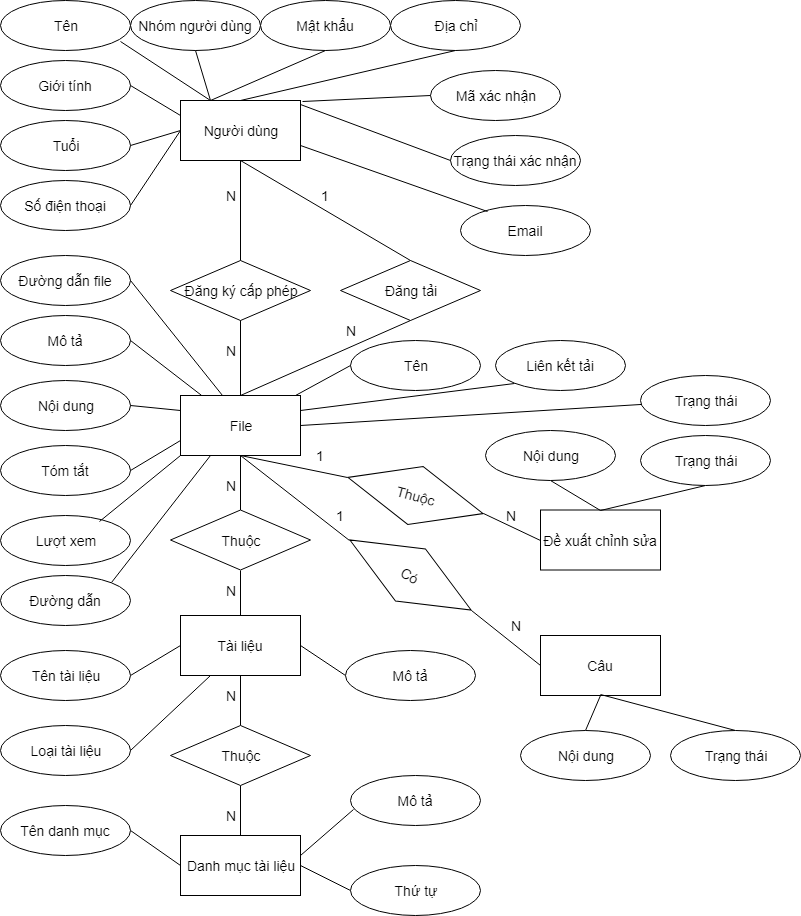
*Test.png*

*Mối quan hệ giữa File và Câu*

## Xử lý mối quan hệ

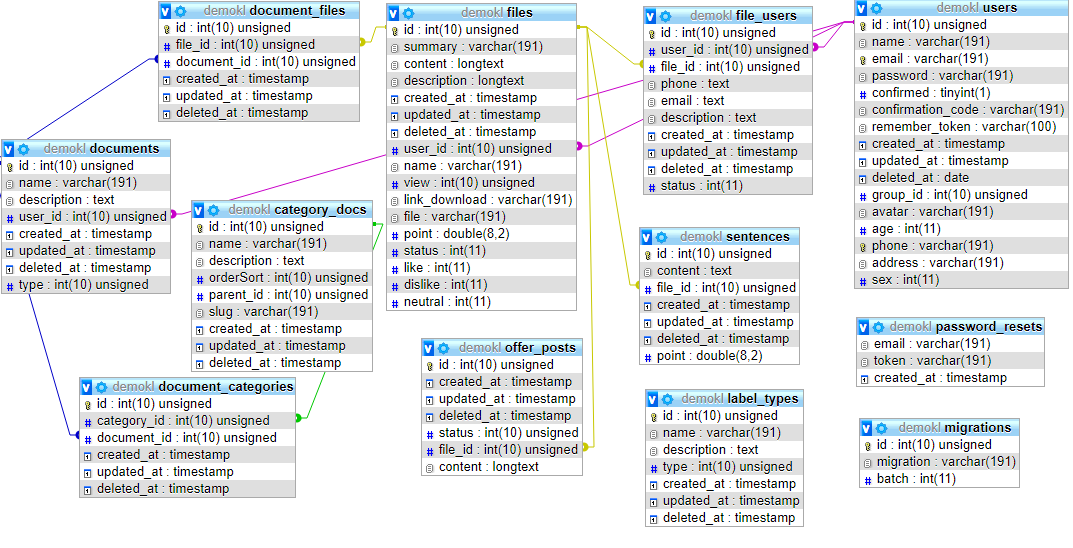
* Users (Id, group\_id, password, name, age, phone, email, address, sex, confirmed, confirmation\_code).
* Category\_docs (Id, name, description, orderSort).
* Documents (Id, name, description, type, user\_id).
* Document\_categories (Id, category\_id, document\_id).
* Files (Id, name, description, summary, content, view, link\_download, file, point, status, like, dislike, neutral).
* Document\_files (Id, file\_id, document\_id).
* File\_users (Id, user\_id, file\_id, phone, email, description, status).
* Label\_types (Id, name, description, type).
* Offer\_posts (Id, file\_id, status).
* Sentences (Id, file\_id, content, point).

## Mô hình ER

****

*Mô hình ER*

## Sơ đồ lớp

****

*Sơ đồ diagram thể hiện mối quan hệ trong cơ sở dữ liệu giữa các bảng với nhau*

## Công nghệ sử dụng

* **Bootstrap3[[10]](#footnote-11)**
* **Bootstrap** là một **framework** CSS được Twitter phát triển. Nó là một tập hợp các bộ chọn, thuộc tính và giá trị có sẵn để giúp web designer tránh việc lặp đi lặp lại trong quá trình tạo ra các class CSS và những đoạn mã HTML giống nhau trong dự án web của mình. Ngoài CSS ra, thì bootstrap còn hỗ trợ các function tiện ích được viết dựa trên JQuery(Carousel, Tooltip, Popovers,...)
* **PHP[[11]](#footnote-12)**
* **PHP** (PHP: Hypertext Preprocessor) là ngôn ngữ lập trình kịch bản (scripting language) mã nguồn mở được dùng phổ biến để ra tạo các ứng dụng web chạy trên máy chủ. Mã lệnh PHP có thể được nhúng vào trong trang HTML nhờ sử dụng cặp thẻ PHP.
* PHP khởi đầu như là một dự án mã nguồn mở nhỏ, nhưng theo đà phát triển, ngày càng nhiều người thấy rằng nó càng ngày càng hữu ích. PHP được phát triển từ một sản phẩm có tên là PHP/FI. PHP/FI do Rasmus Lerdorf tạo ra năm 1994, ban đầu được xem như là một tập con đơn giản của các mã kịch bản Perl để theo dõi tình hình truy cập đến bản sơ yếu lý lịch của ông trên mạng. Ông đã đặt tên cho bộ mã kịch bản này là "Personal Home Page Tools". Khi cần đến các chức năng rộng hơn, Rasmus đã viết ra một bộ thực thi bằng C lớn hơn để có thể truy vấn tới Database và giúp cho người sử dụng phát triển các ứng dụng web đơn giản. Rasmus đã quyết định công bố mã nguồn của PHP/FI cho mọi người xem, sử dụng cũng như sửa các lỗi có trong nó, đồng thời cải tiến mã nguồn.
* Nó được tích hợp với một số Database thông dụng như MySQL, PostgreSQL, Oracle, Sybase, Informix, và Microsoft SQL Server.
* PHP thực thi rất tuyệt vời, đặc biệt khi được biên dịch như là một Apache Module trên Unix side. MySQL Server, khi được khởi động, thực thi các truy vấn phức tạp với các tập hợp kết quả khổng lồ trong thời gian Record - setting.
* PHP hỗ trợ một số lượng rộng rãi các giao thức lớn như POP3, IMAP, và LDAP. PHP4 bổ sung sự hỗ trợ cho Java và các cấu trúc đối tượng phân phối (COM và CORBA).
* Cú pháp PHP là giống C.
* **PHP Framework – Laravel[[12]](#footnote-13)**
* **Laravel Framwork** là một PHP Framework lần đầu tiên được ra mắt cộng đồng vào tháng 6/2011. Ngay từ khi ra mắt, Laravel đã được chú ý đến bởi nhiều đặc điểm nổi trội so với đàn anh trước như Eloquent ORM, localization, models and relationships, routing, caching, sessions, views,... Chính tác giả Laravel - Taylor Otwell đã nói rằng Laravel được tạo ra để khắc phục những vấn đề còn vướng mắc và thiếu sót mà CodeIgniter còn gặp phải..
* Tuy ra đời khá trễ so với các Framework khác nhưng nó được rất nhiều lập trình viên đón nhận và có một cộng động lớn sử dụng.Vì vậy Laravel 5.x luôn thừa hưởng những đặc tính tốt nhất và tối ưu nhất từ các Framework trước, cũng như khắc phục được các nhược điểm mà các Framework khác mắc phải. Cho nên, chọn Laravel Framework để làm những project với quy mô vừa và nhỏ là một sự lựa chọn vô cùng hợp lý và sáng suốt.
* Ưu điểm của laravel là dễ dùng và mạnh mẽ. Đặc biệt tài liệu được viết rất công phu nhưng lại cô đọng và dễ hiểu.
* Sử dụng mô hình ORM rất đơn giản khi thao tác với DB
* Các lệnh tương tác với cơ sở dữ liệu cực kỳ ngắn gọn và thân thiện.
* Việc quản lý layout thật sự giản đơn với Blade Templating.
* Dễ dàng tích hợp các thư viện khác vào dự án, và được quản lý với Composer.
* Phần route rất mạnh!!!
* Tuy vậy Laravel vẫn có một nhược điểm và số lượng module dùng tương đối lớn, dẫn đến tốc độ không được tối ưu khi xây dựng những service nhỏ.
* Cơ sở dữ liệu **SQL[[13]](#footnote-14)**
* **SQL** (Structured Query Language - ngôn ngữ truy vấn mang tính cấu trúc) là một loại ngôn ngữ máy tính phổ biến để tạo, sửa, và lấy dữ liệu từ một hệ quản trị cơ sở dữ liệu quan hệ.
* **Lịch sử của SQL bắt đầu từ 1970 -**Dr. Edgar F. "Ted" Codd của IBM được biết đến như là cha đẻ của Relational Database (Cơ sở dữ liệu quan hệ). Ông miêu tả một mô hình quan hệ (Relational Model) cho các Database.
* **1974 -**SQL (Structured Query Language) xuất hiện.
* **1978 -**IBM tiếp tục phát triển ý tưởng của Codd và công bố một sản phẩm tên là System/R.
* **1986 -**IBM phát triển nguyên mẫu đầu tiên về Relation Database và được chuẩn hóa bởi ANSI. Relation Database đầu tiên được công bố là Relational Software và sau đó là Oracle.
* Theo đó SQL là một công cụ quản lý dữ liệu được sử dụng phổ biến ở nhiều lĩnh vực. Nếu bạn không có nhiều thời gian để nghiên cứu sâu về tin học mà lại có nhiều dữ liệu cần phải được phân tích, xử lý, thì hãy dùng SQL vì SQL đơn giản nhưng rất hiệu quả. Hầu hết tất cả các ngôn ngữ lập trình cấp cao đều có hỗ trợ SQL.
* Tất cả DBMS như MySQL, Oracle, MS Access, Sybase, Informix, Postgres và SQL Server sử dụng SQL như là ngôn ngữ cơ sở dữ liệu chuẩn.
* Quản lý mã nguồn **SourceTree**
  + **SourceTree** là một công cụ để quản lý phiên bản mã nguồn (Source version control) - SVC trong quá trình phát triển phần mềm.
  + Ưu điểm của SourceTree đó là các thao tác đều thực thi trên giao diện thân thiện, dễ sử dụng cho người mới bắt đầu mà không cần gõ các dòng lệnh trên command line.
  + Ngoài ra SourceTree nói riêng cũng như Git nói chung hơn các phiên bản quản lý mã nguồn trước đây như SVN(Subversion), Concurrent Versioning System(CVS) ở điểm Git có thêm một lớp “server” ngay trên máy của mình, để nếu có muốn thay đổi gì, thì cũng đồng bộ trên server trên máy mà không lo ảnh hưởng tới các thành viên khác trong team. Khi nào xong xuôi hết thì mới đẩy lên trên server chung kia để đồng bộ. Nó còn rất nhiều chức năng khác như là phân nhánh, gắn thẻ (tag), nhập code, giải quyết mâu thuẫn code,…
* **Javascript Framework – jQuery[[14]](#footnote-15)**
  + jQuery là một thư viện JavaScript đa trình duyệt, phát hành vào tháng 1 năm 2006 tại BarCamp NYC bởi John Resig. Nó làm cho những thao tác với HTML và xử lý sự kiện, hoạt ảnh và Ajax đỡ mất thời gian và đơn giản hơn nhiều với một API dễ sử dụng hoạt động trên vô số các trình duyệt.
  + Với sự kết hợp của tính linh hoạt và khả năng mở rộng, jQuery đã thay đổi cách hàng triệu người viết JavaScript và jQuery cũng là thư viện JavaScript phổ biến nhất được sử dụng ngày nay.
* **JsTree[[15]](#footnote-16)**
  + JsTree là một plugin của jQuery, nó hỗ trợ rất tốt cho việc xây dựng nên cây và tương tác kéo thả cũng như chỉnh sửa trên cây.
  + Nó hoàn toàn là mã nguồn mở, dễ mở rộng, thay đổi cũng như có thể cấu hình. Hơn nữa còn hỗ trợ cả các nguồn dữ liệu HTML và JSON và AJAX loading giúp cho việc xây dựng nên cây một cách linh hoạt, tùy biến và không bị đơn điệu.
* **Goutte[[16]](#footnote-17)**
  + Goutte là một thư viện thu thập dữ liệu và thu thập thông tin trên web cho PHP. Với cơ chế có thể đọc được cấu trúc của 1 trang web với các cách thức như DOM(Document Object Model - mô hình các đối tượng trong tài liệu HTML), Xpath(Xpath sử dụng path expressions để di chuyển hay truy xuất thuộc tính trong các node của XML document).
  + Goutte cung cấp một API tốt để thu thập thông tin các trang web và trích xuất dữ liệu từ các phản hồi HTML / XML.
* **HTML Unit[[17]](#footnote-18)**
  + HtmlUnit là một trình duyệt không có giao diện cho các chương trình Java. Nó mô hình hóa các tài liệu HTML và cung cấp một API cho phép bạn gọi các trang, điền vào các biểu mẫu, nhấp vào các liên kết... giống như bạn làm trong trình duyệt bình thường của bạn.
  + Nó có hỗ trợ JavaScript khá tốt (không ngừng cải tiến) và có thể làm việc ngay cả với các thư viện AJAX khá phức tạp, mô phỏng Chrome, Firefox hoặc Internet Explorer tùy thuộc vào cấu hình được sử dụng.
  + Nó thường được sử dụng cho mục đích thử nghiệm hành vi người dùng hoặc để lấy thông tin từ các trang web.
  + HtmlUnit ban đầu được viết bởi Mike Bowler của phần mềm Gargoyle và được phát hành theo giấy phép Apache 2. Kể từ đó, nó đã nhận được nhiều đóng góp từ các nhà phát triển khác và đang ngày càng hoàn thiện dần về tính năng phục vụ các nhu cầu khác nhau của người sử dụng.

# Chương 4. Kết quả

## Các kết quả đã làm được:

* Xây dựng công cụ thu thập dữ liệu.
* Import các kho ngữ liệu sẵn có phục vụ cho người dùng.
* Thu thập dữ liệu từ các trang web để tiến hành cho các bài toán.
* Xây dựng trang web:
  + Chia sẻ các kho ngữ liệu:
    - Đã có sẵn (Kho ngữ liệu 2 triệu âm tiết đã tách từ, 10000 câu được gán nhãn từ loại, 10000 cây cú pháp).
    - Thu thập dữ liệu từ foody, facebook, wikipedia.
  + Hỗ trợ người dùng gán nhãn dữ liệu.
  + Gán nhãn thủ công (Tách từ, gán nhãn từ loại, phân tích cú pháp).
  + Gán nhãn bán tự động (Tách từ sử dụng thư viện Vitk-tok-5.1, gán nhãn từ loại sử dụng thư viện Vitk-pos-5.1.jar).

## Hạn chế:

* Bộ công cụ thu thập dữ liệu còn hạn chế đối với một số trang web sử dụng cơ chế tải dữ liệu bằng Ajax (Asynchronous JavaScript and XML).
* Chưa tích hợp được thêm các công cụ xử lý ngôn ngữ tự nhiên nâng cao hơn.
* Giao diện vẫn còn đơn giản và chưa mang đến trải nghiệm tốt nhất cho người dùng.

Hướng phát triển:

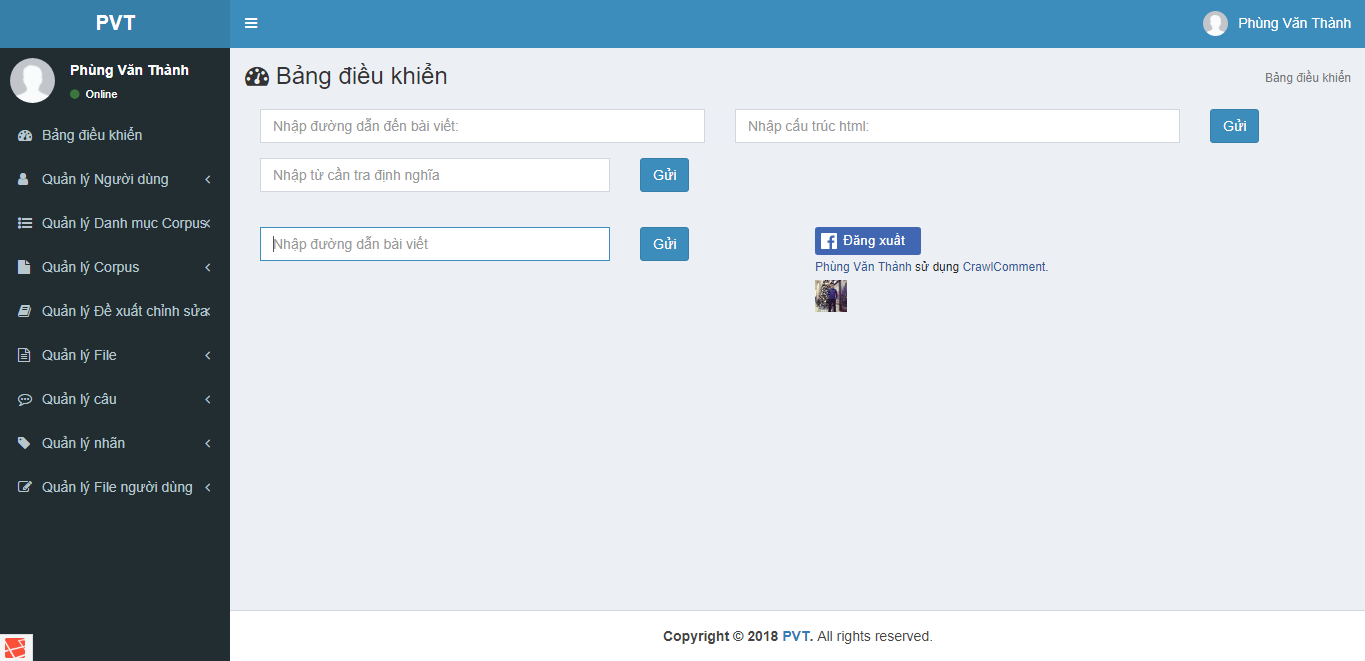
* Tích hợp thêm hệ thống thanh toán trực tuyến phục vụ cho nhu cầu trao đổi, giao dịch giữa các người dùng trên hệ thống.
* Tìm hiểu và tích hợp thêm một số bộ công cụ xử lý ngôn ngữ tự nhiên nâng cao hơn.
* Áp dụng các công nghệ mới để cải thiện hiệu suất cũng như trải nghiệm của người dùng.

## Kết quả chi tiết

Dưới đây là một số hình ảnh của trang web về kết quả đã đạt được.

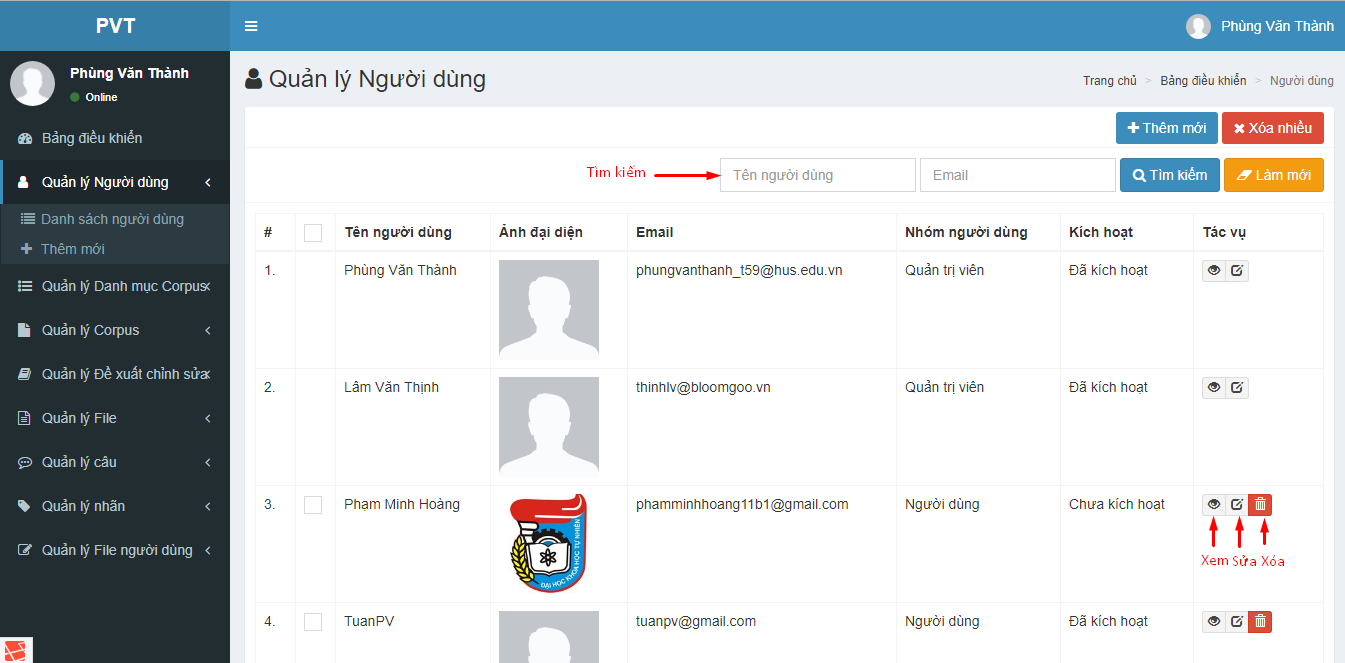
**Phần quản trị:**

* Bảng điều khiển: quản trị viên tại đây có thể sử dụng các bộ công cụ được tích hợp vào hệ thống phục vụ cho việc thu thập dữ liệu trên các trang web.

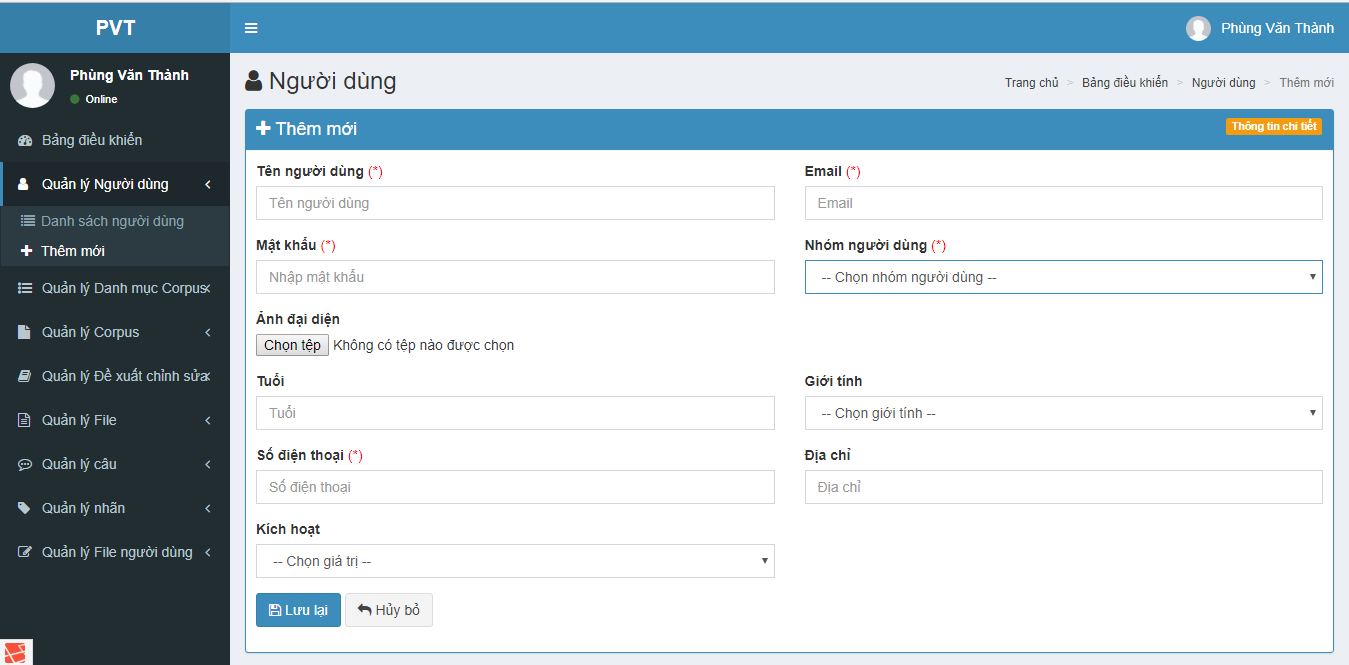


*Giao diện Bảng điều khiển*

* Tại đây sẽ có bộ 3 công cụ được xây dựng:
  + Bộ công cụ lấy dữ liệu từ 1 trang web bất kì với cấu trúc và đường dẫn do người dùng nhập vào.
  + Bộ công cụ lấy định nghĩa của một từ trả về từ wikipedia được xử lý thông qua hệ thống.
  + Bộ công cụ thu thập các bình luận của 1 bài viết từ fanpage trên facebook. Để sử dụng được trước hết quản trị viên cần phải đăng nhập vào facebook, sau đó nhập đường dẫn bài viết cần lấy bình luận và ấn nút gửi. Hệ thống sẽ trả dữ liệu về dưới dạng file text.
* Quản lý người dùng: quản trị cao cấp nhất sẽ có thêm phần quản lý người dùng bao gồm các hành vi như thêm, sửa, xóa, phân quyền người dùng. Ngoài ra còn có thể tìm kiếm theo các tiêu chí khác nữa

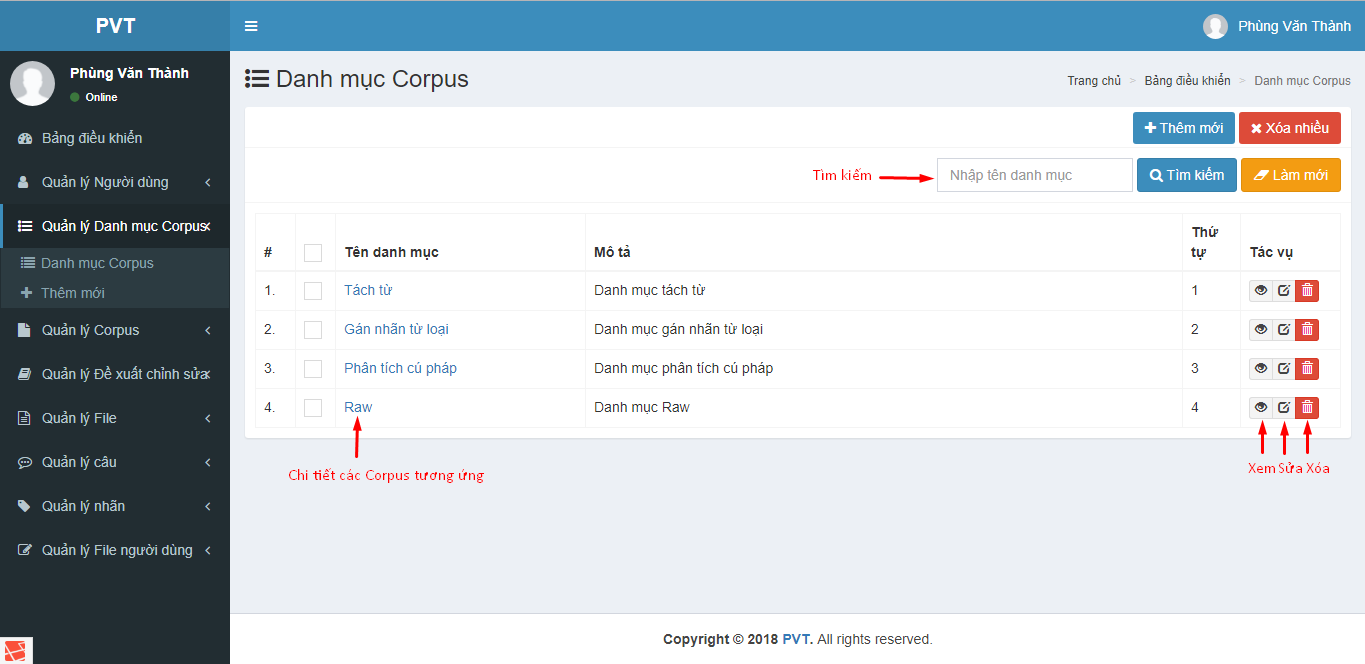


*Giao diện quản lý người dùng*



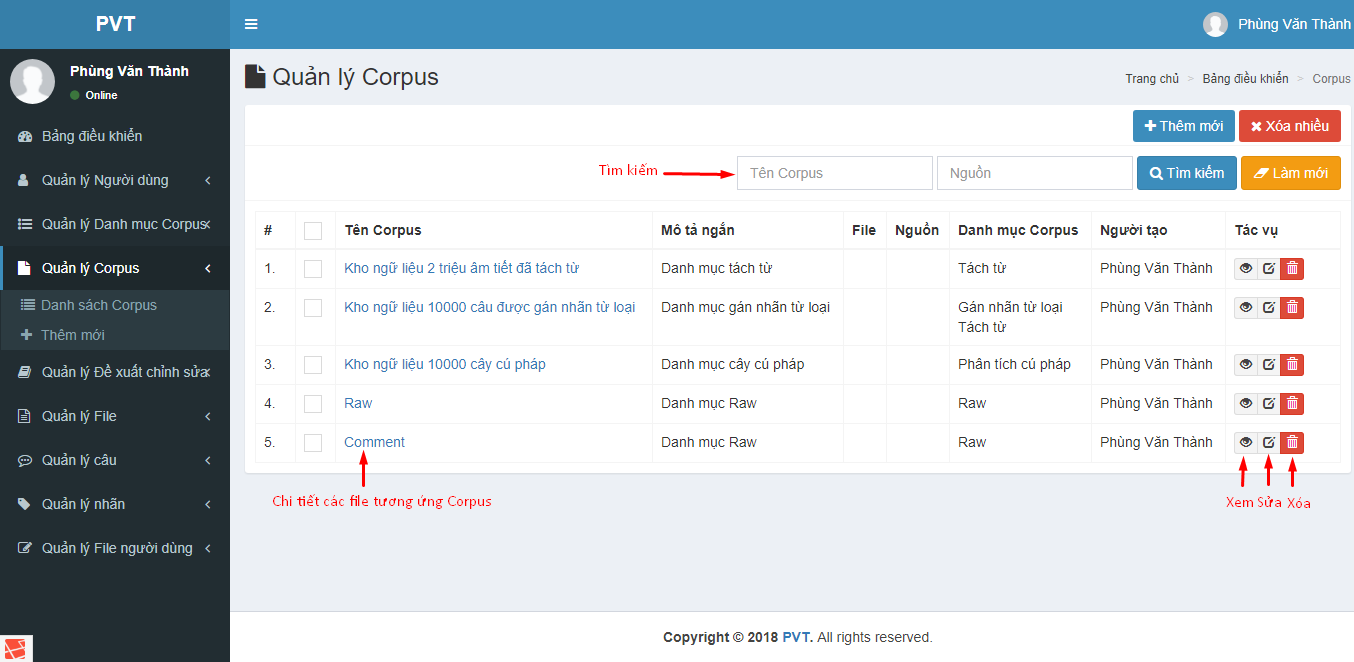
*Giao diện thêm mới người dùng*

* Quản lý Danh mục Corpus: cho phép người quản trị cao cấp có thể thêm, sửa, xóa, tìm kiếm đơn thuần các bộ ngữ liệu của mình.



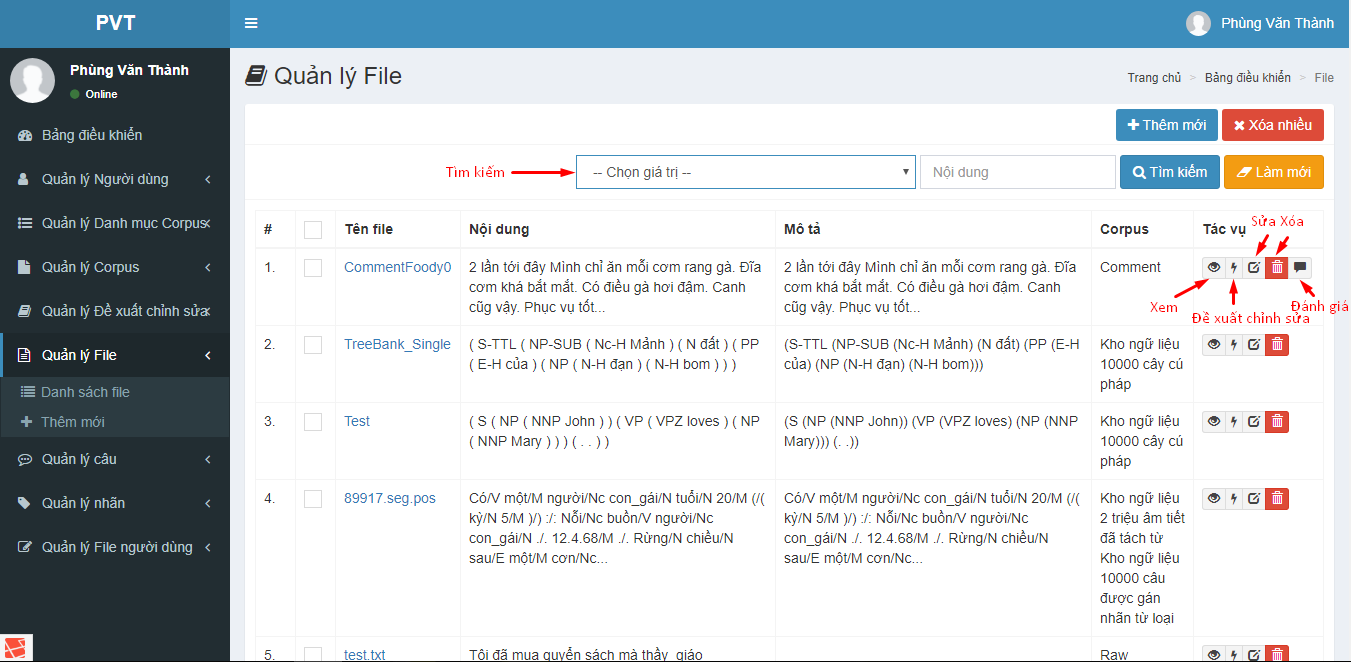
*Giao diện quản lý danh mục Corpus*

* Quản lý Corpus: quản trị cao cấp có thể thêm, sửa, xóa, cũng như tìm kiếm các danh mục corpus của mình.



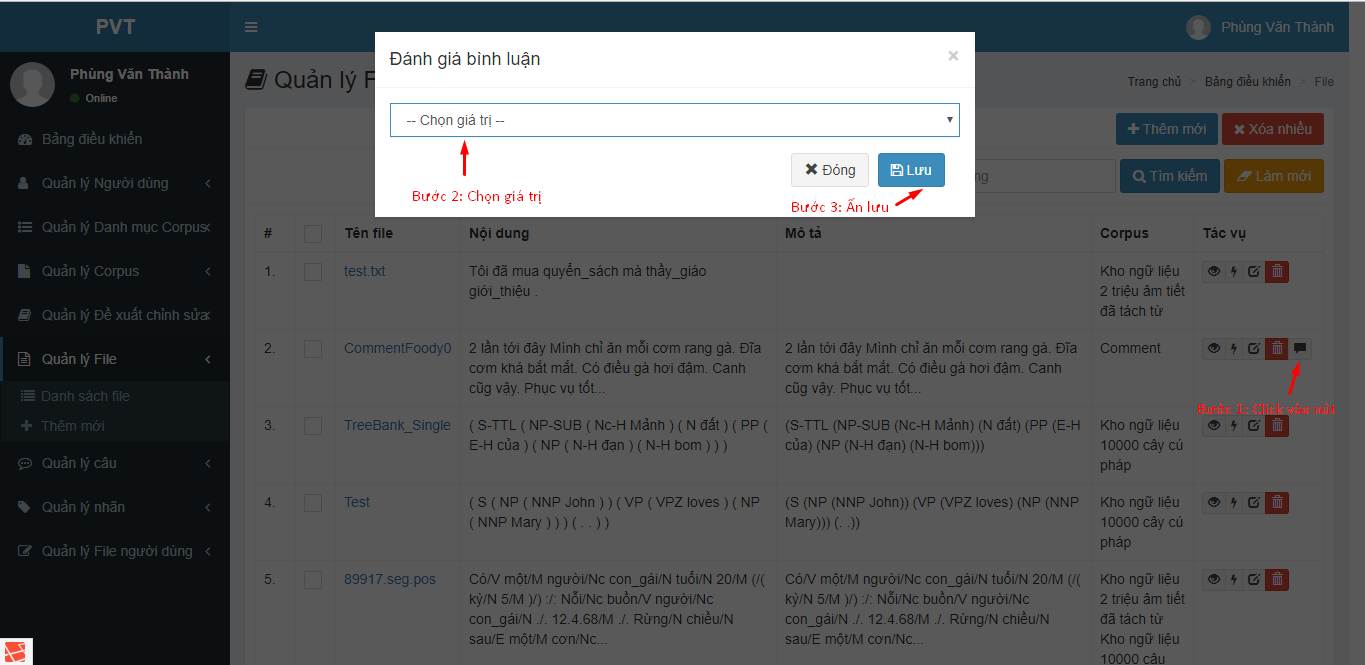
*Giao diện quản lý Corpus*

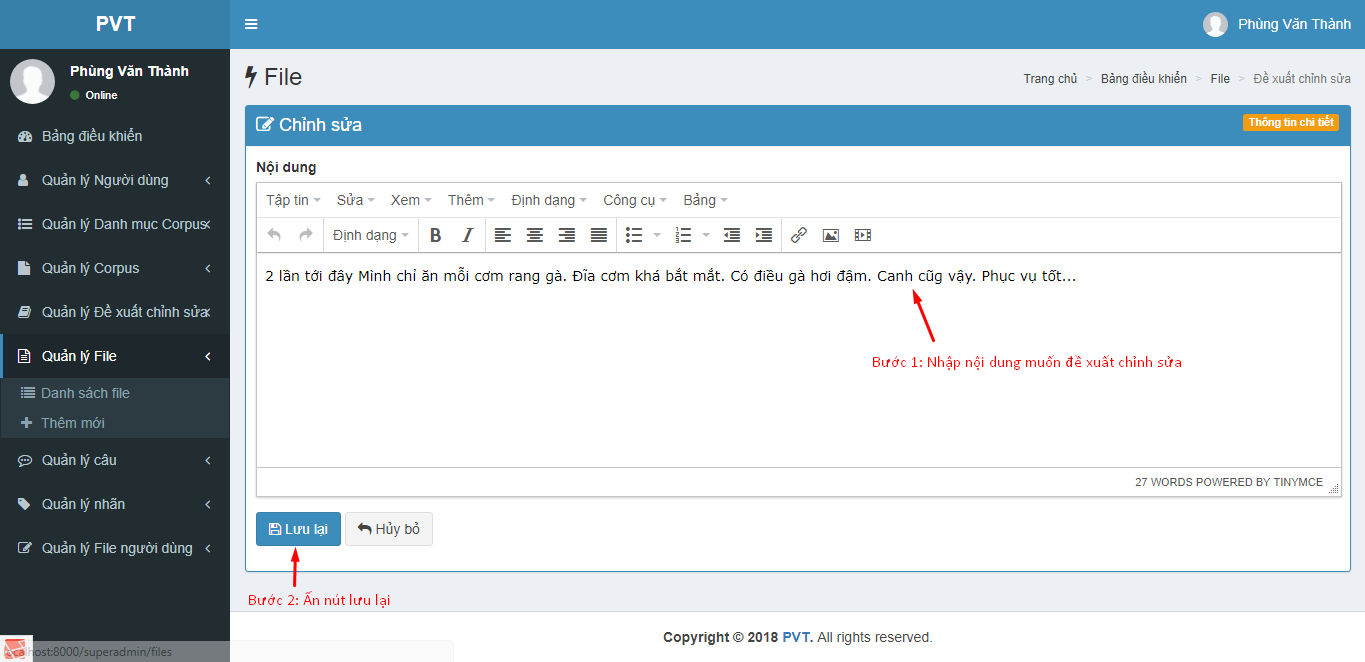
* Quản lý tệp các file: cho phép người quản trị có thể thêm, sửa, xóa, tìm kiếm tệp file trong các danh mục Corpus tương ứng.



*Giao diện quản lý File*

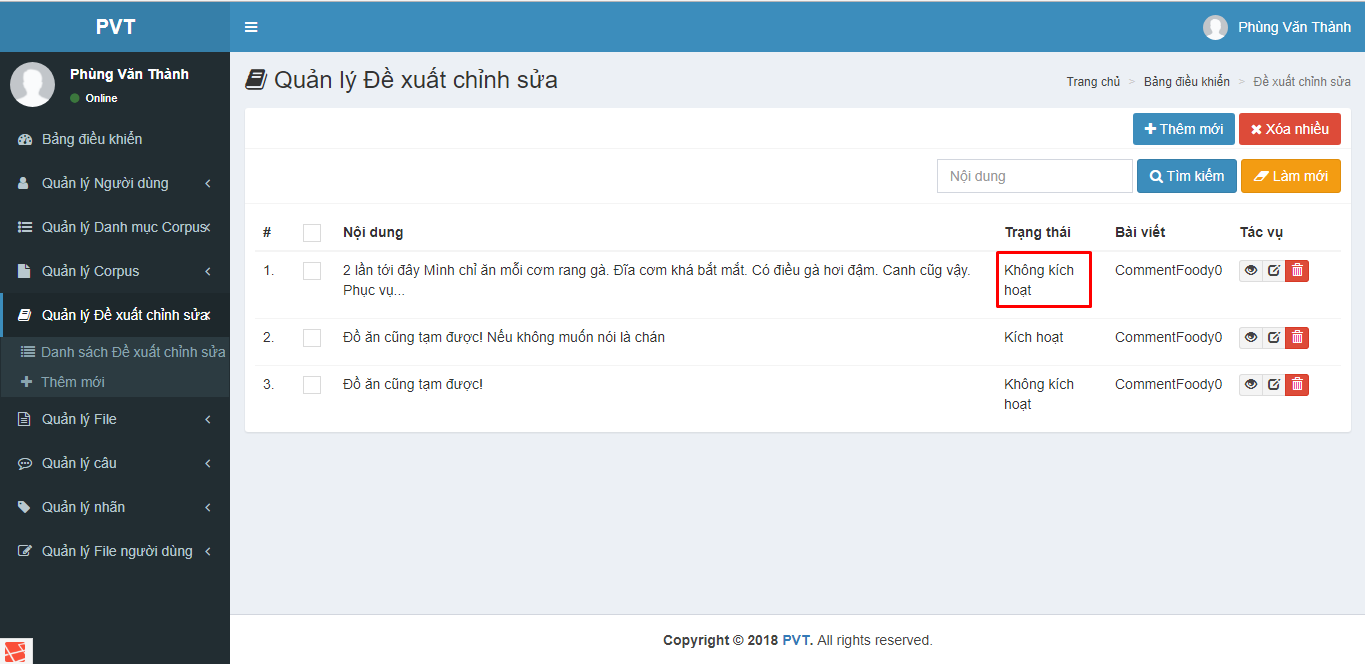
Ngoài ra còn cho phép người dùng có thể đánh giá bình luận là tốt, không tốt hay trung tính dựa trên loại Corpus là comment.

*Giao diện đánh giá bình luận*Hơn thế nữa người quản trị cấp cao còn có thể đề xuất chỉnh sửa bài đăng của các người quản trị khác.



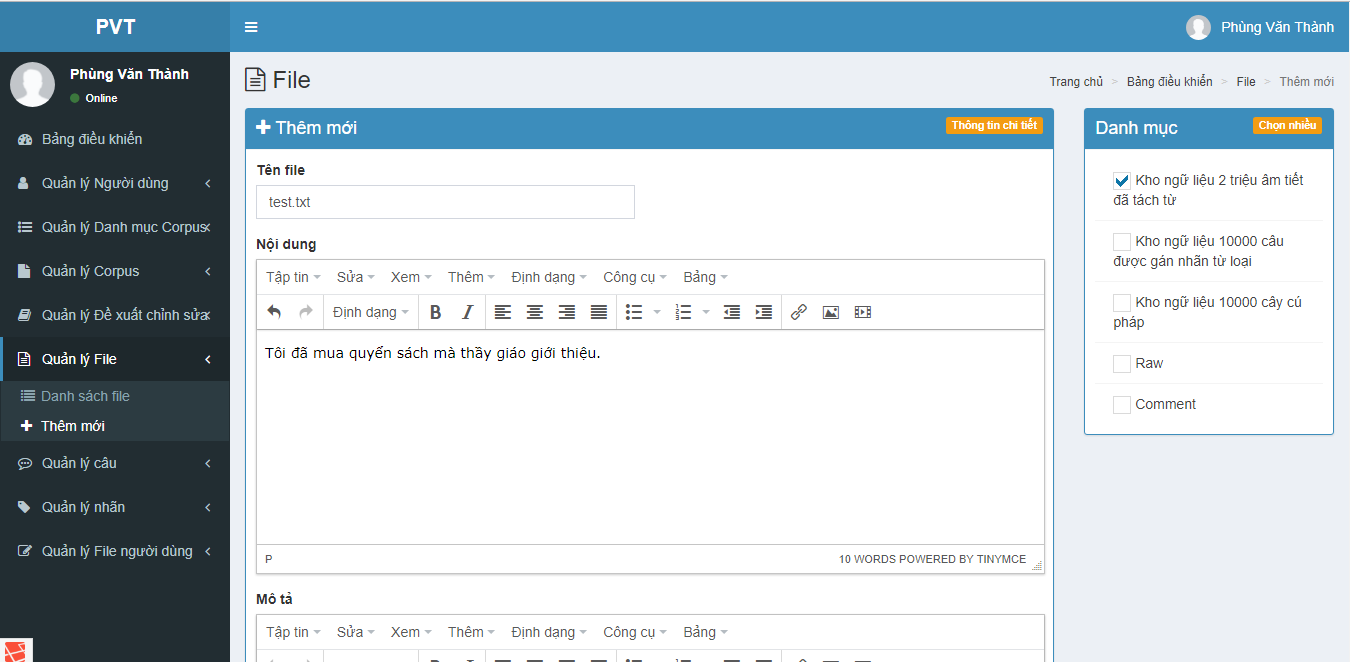
*Giao diện đề xuất chỉnh sửa bài viết*

Sau đó thông tin đề xuất chỉnh sửa sẽ được lưu lại ở phần quản lý đề xuất chỉnh sửa, bài viết sẽ được thay đổi nội dung khi người đăng bài kích hoạt.

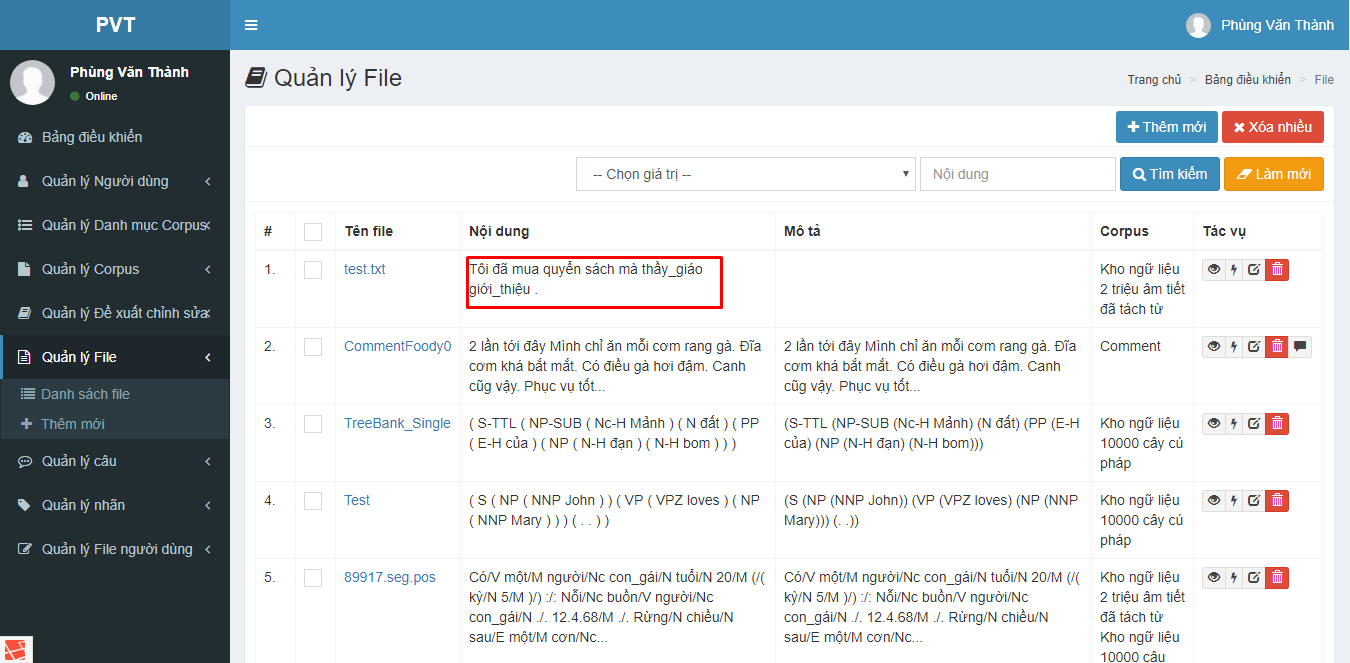


*Giao diện quản lý đề xuất chỉnh sửa*

Ở phần quản lý File có thêm việc chuẩn hóa dữ liệu đầu vào cho bài toán tách từ. Tức là người dùng nhập nội dung văn bản thông thường và chọn kho ngữ liệu đã tách từ, đầu ra sẽ là nội dung đã được xử lý tách từ thông qua bộ công cụ Tokenizer của thầy Lê Hồng Phương

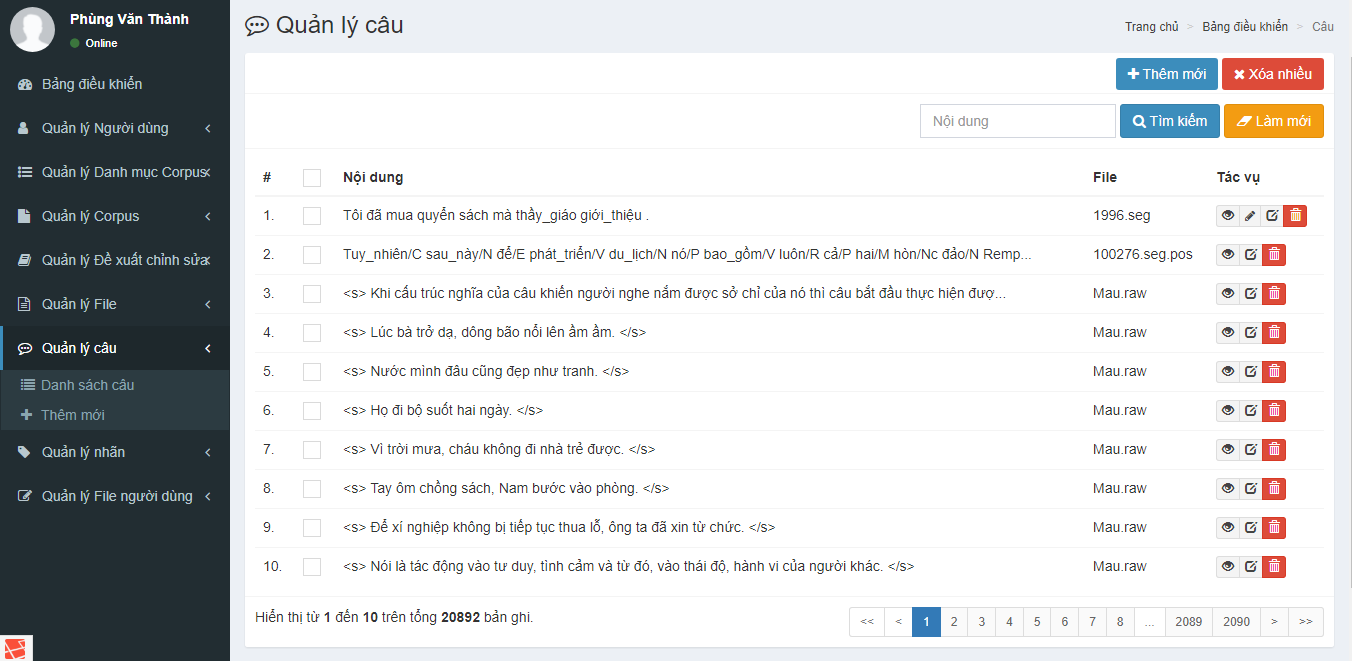


*Đầu vào*

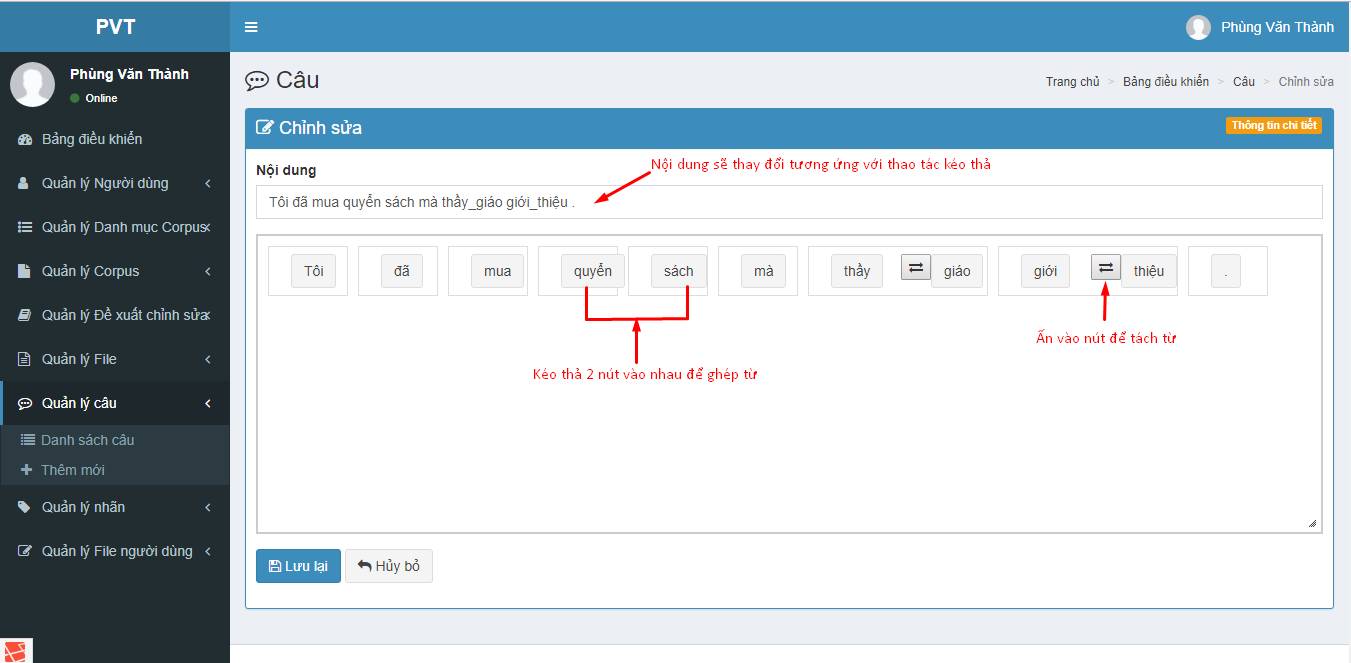
**

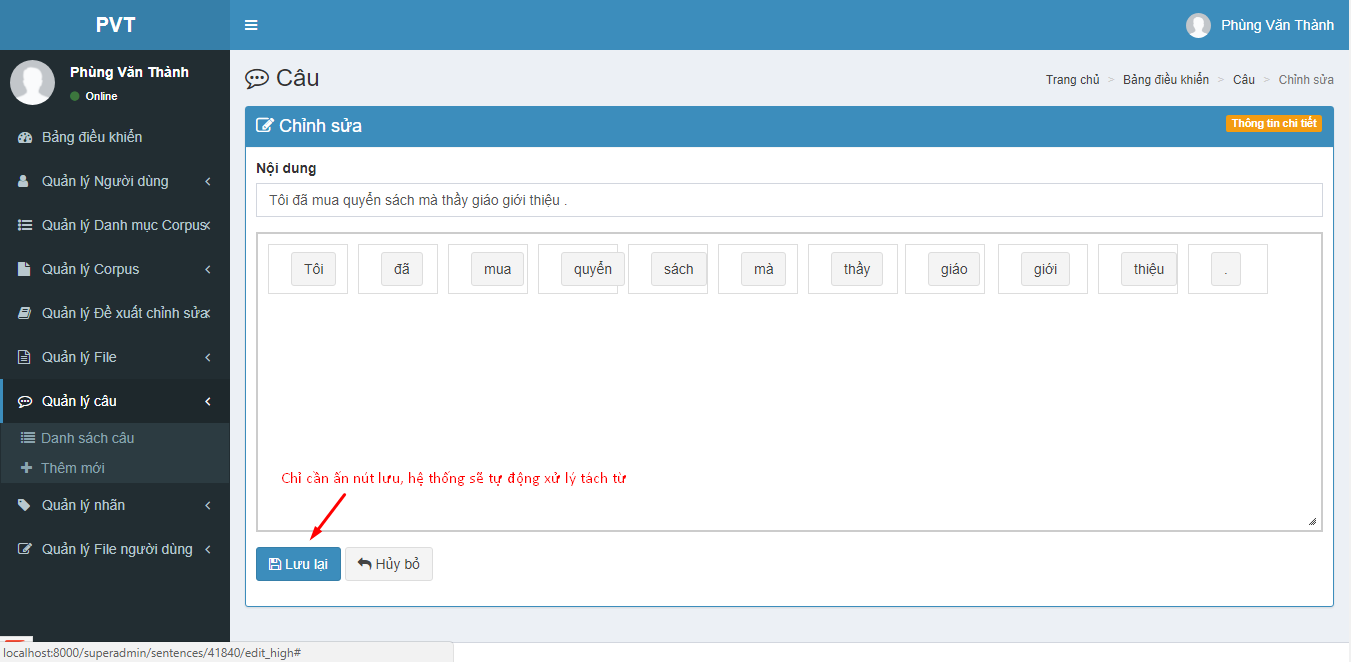
*Đầu ra*

* Quản lý câu: cho phép người quản trị có thể thêm, sửa, xóa, tìm kiếm các câu trong cùng 1 file tương ứng. Điểm đặc biệt ở phần này đó là có xây dựng các bộ công cụ chỉnh sửa phù hợp với từng loại Corpus như tách từ, gán nhãn từ loại, phân tích cú pháp. Tất cả đều được xây dựng với thao tác kéo thả đơn giản rất dễ sử dụng và thích hợp cho người mới bắt đầu.

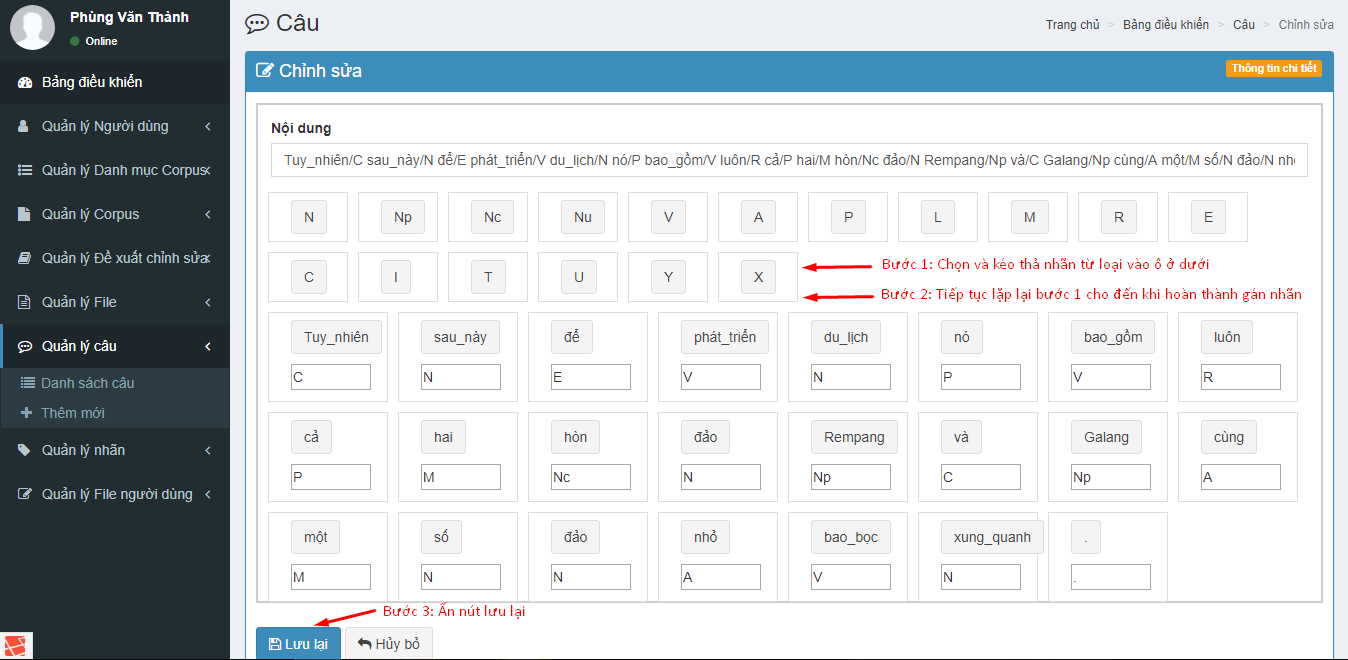


*Giao diện quản lý câu*

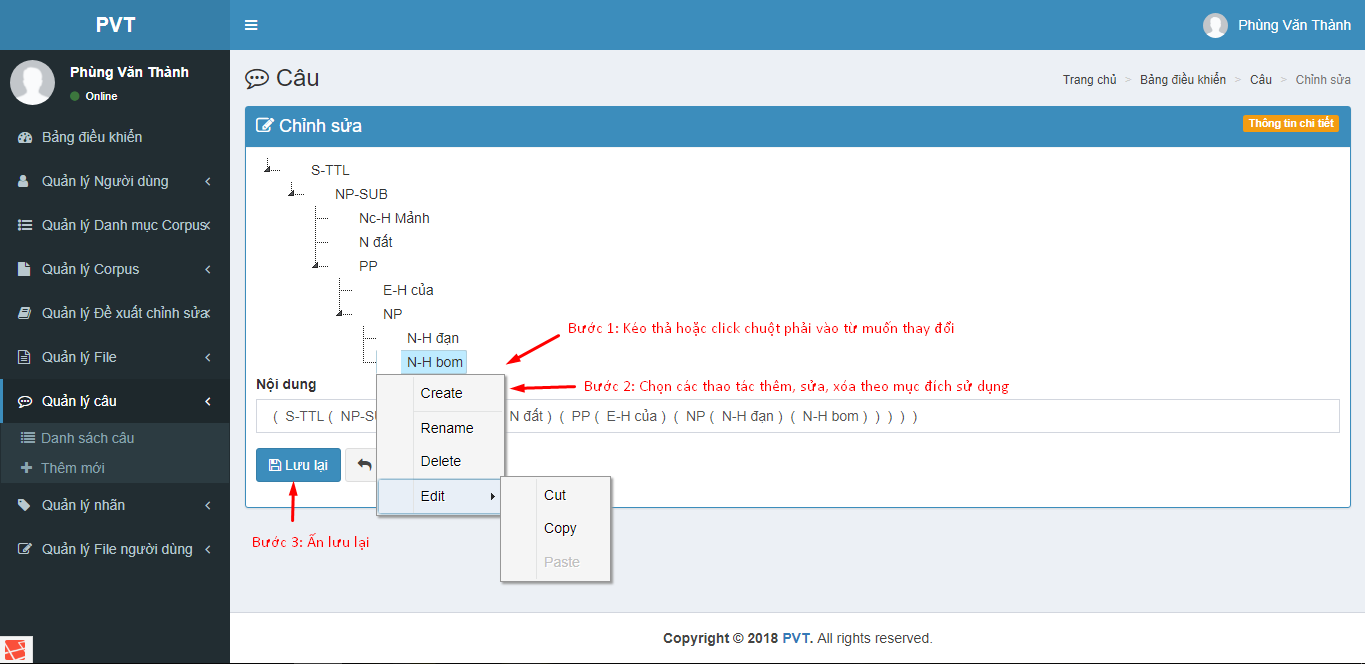
*Giao diện kéo thả tách từ thủ công*

**

*Giao diện tách từ bán tự động*

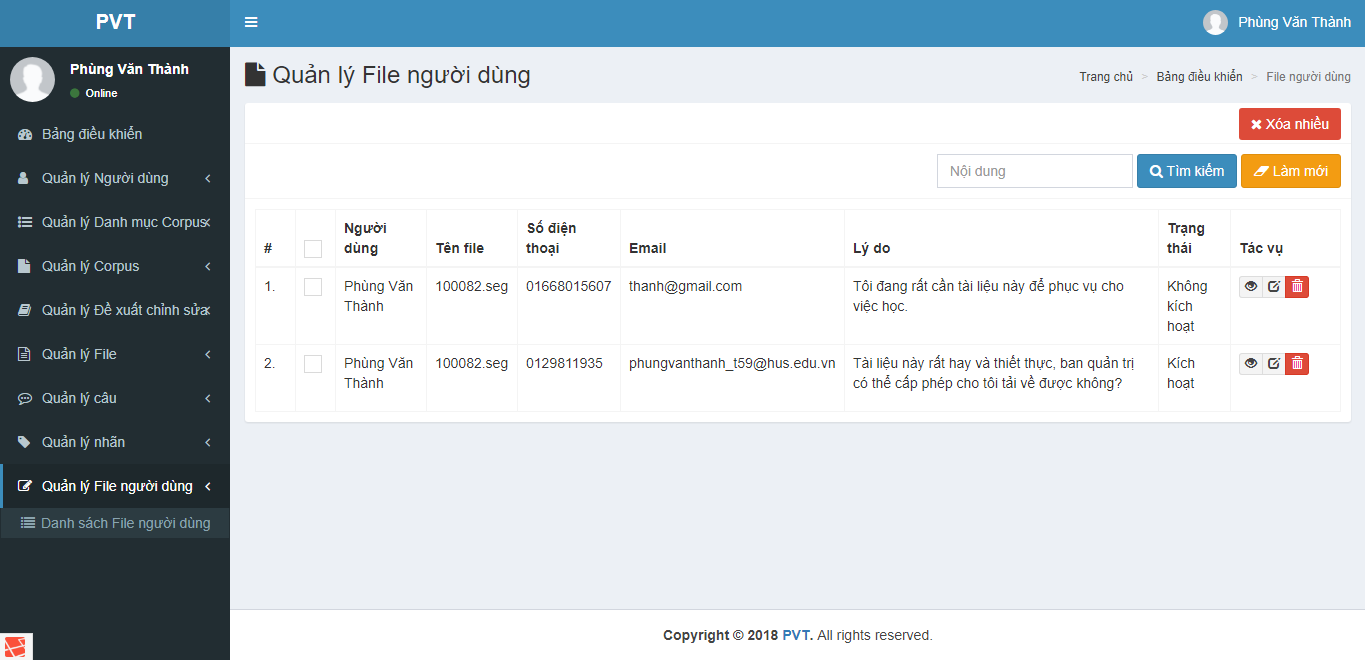
**

*Giao diện gán nhãn từ loại thủ công*

**

*Giao diện phân tích cây cú pháp thủ công*

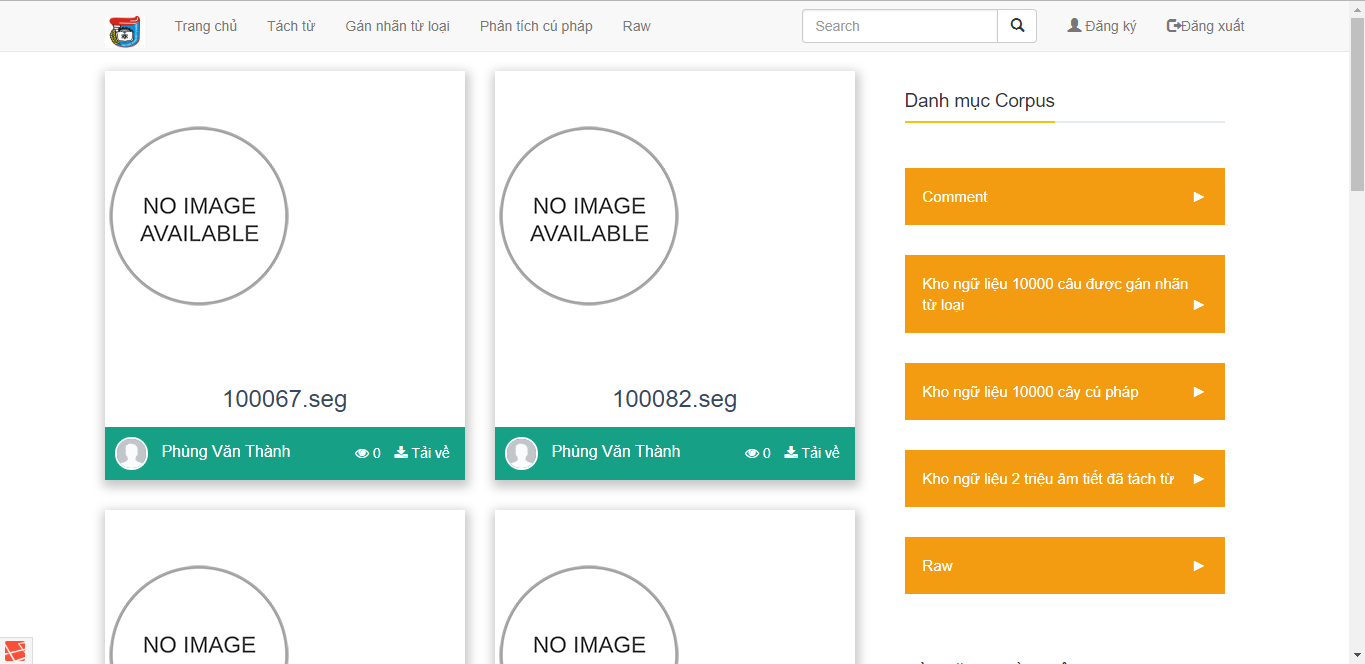
* Quản lý File người dùng: cho phép người đăng tải tài liệu biết được có những ai muốn tải tài liệu của mình về. Và từ đó có thể liên hệ lại thông qua thông tin cũng như lý do đã được cung cấp để có thể quyết định cho phép tải tài liệu hay không.



*Giao diện quản lý File người dùng*

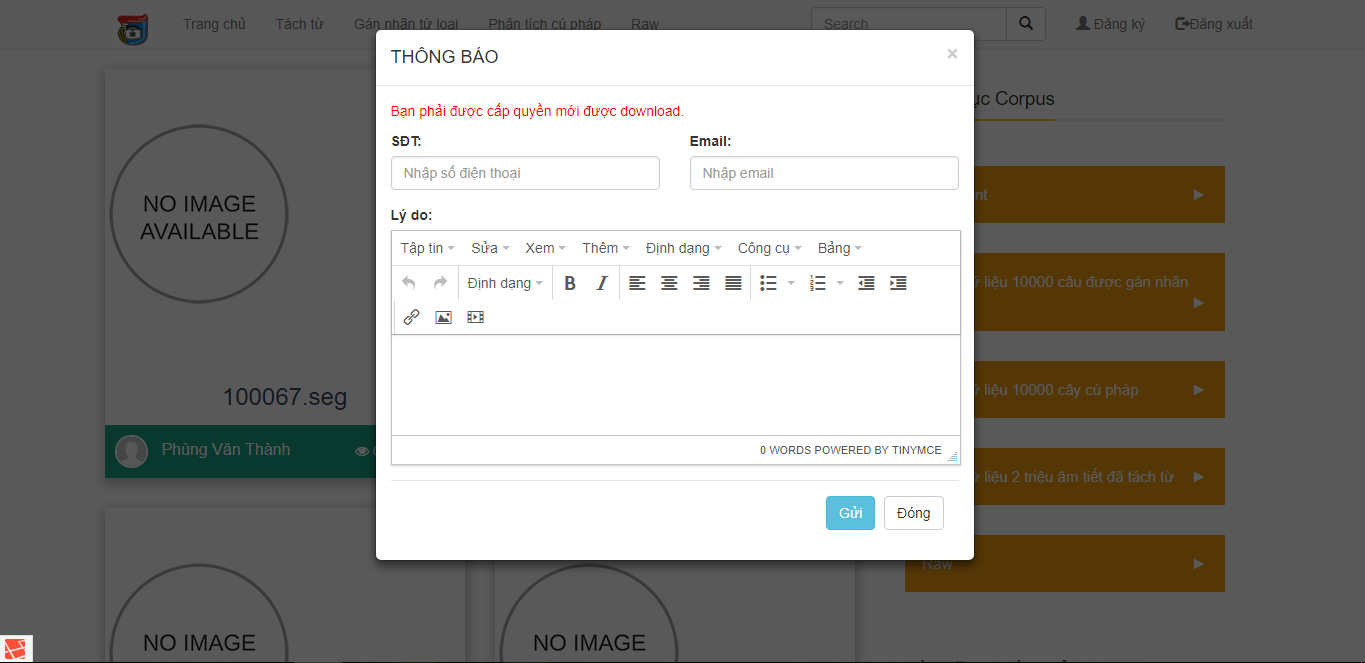
Phần frontend:

* Chức năng xem các tệp file hiển thị theo từng danh mục Corpus



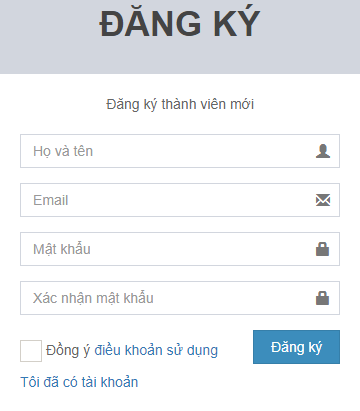
*Giao diện các tệp file theo từng danh mục*

* Chức năng gửi đơn đăng ký cấp phép tải tài liệu.

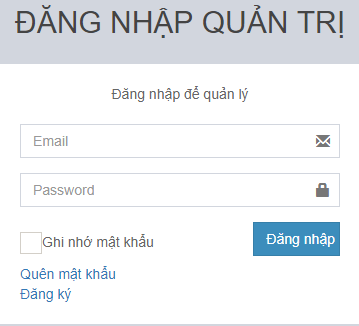


*Giao diện đăng ký cấp phép*

* Chức năng đăng nhập, đăng ký.

****

*Giao diện form Đăng ký*

**

*Giao diện form Đăng nhập*

# Tài liệu tham khảo

**Tiếng Việt**

1. Nguyễn Thị Minh Huyền, Vũ Xuân Lương, Lê Hồng Phương (2003), “Sử dụng bộ gán nhãn từ loại xác suất QTAG cho văn bản tiếng Việt”, *Báo cáo hội thảo ICT.rda*.
2. Đỗ Bá Lâm, Lê Thanh Hương, “Xây dựng hệ thống phân tích cú pháp tiếng Việt sử dụng văn phạm HPSG”, *Báo cáo hội thảo ICT.rda*.
3. Nguyễn Phương Thái, Vũ Xuân Lương, Nguyễn Thị Minh Huyền, “Xây dựng treebank tiếng Việt”, *Báo cáo hội thảo ICT.rda*.
4. Trần Thu Trang (2012),  *Nghiên cứu gán nhãn từ loại cho văn bản tiếng Việt bằng phương pháp học máy không có hướng dẫn*, Luận văn ThS. ngành: Bảo đảm toán học cho máy tính và hệ thống tính toán, Trường Đại học Khoa học Tự nhiên, Hà Nội.

**Tiếng Anh**

1. Brill E, "Transformation - Based Error - Driven Learning and Natural Language Processing: A Case Study in Part of Speech Tagging", *Computational Linguistics*, 21(4), December 199, pp.543-565.
2. Linda Van Guilder (1995), “Automated Part of Speech Tagging: A Brief Overview”.
3. Ryan McDonald and Joakim Nivre (2007), “Introduction to Data - Driven Dependency Parsing”.
4. Schmid H (1994), "Part - of - Speech Tagging with Neural networks", *International Conference on Computational Linguistics*, pp. 172 – 176.
5. Dang Van Thin, Nguyen Van Kiet, & Nguyen Luu Thuy Ngan (2018), “NLP@UIT at VLSP 2018: A supervised method for aspect-based sentiment analysis”, *Proceedings of the 5th International Workshop on Vietnamese Language and Speech Processing (VLSP 2018).*

1. https://vi.wikipedia.org/wiki/Xử\_lý\_ngôn\_ngữ\_tự\_nhiên [↑](#footnote-ref-2)
2. http://viet.jnlp.org/kien-thuc-co-ban-ve-xu-ly-ngon-ngu-tu-nhien/thuat-toan-tach-tu-tokenizer/thuat-toan-tach-tu [↑](#footnote-ref-3)
3. http://vlsp.wikia.com/wiki/Tách\_từ [↑](#footnote-ref-4)
4. http://viet.jnlp.org/kien-thuc-co-ban-ve-xu-ly-ngon-ngu-tu-nhien/thuat-toan-tach-tu-tokenizer/thuat-toan-tach-tu [↑](#footnote-ref-5)
5. http://viet.jnlp.org/dongdu [↑](#footnote-ref-6)
6. http://jvnsegmenter.sourceforge.net/ [↑](#footnote-ref-7)
7. http://mim.hus.vnu.edu.vn/dsl/tools/tagger [↑](#footnote-ref-8)
8. https://raweb.inria.fr/rapportsactivite/RA2003/led/id2642150.html [↑](#footnote-ref-9)
9. http://www.jaist.ac.jp/~bao/VLSP-text/ICTrda08/ICT08-VLSP-SP83.pdf [↑](#footnote-ref-10)
10. https://getbootstrap.com [↑](#footnote-ref-11)
11. https://vi.wikipedia.org/wiki/PHP [↑](#footnote-ref-12)
12. https://viblo.asia/p/gioi-thieu-ve-laravel-924lJrnzlPM [↑](#footnote-ref-13)
13. https://vi.wikipedia.org/wiki/SQL [↑](#footnote-ref-14)
14. https://vi.wikipedia.org/wiki/JQuery [↑](#footnote-ref-15)
15. https://www.jstree.com/ [↑](#footnote-ref-16)
16. https://github.com/FriendsOfPHP/Goutte [↑](#footnote-ref-17)
17. http://htmlunit.sourceforge.net/ [↑](#footnote-ref-18)