Lời cám ơn!

Không có sự thành công nào mà không gắn liền với những sự hỗ trợ, giúp đỡ dù ít hay nhiều, dù trực tiếp hay gián tiếp của người khác. Được sự phân công của khoa Khoa Học Máy Tính trường Đại Học Công Nghệ Thông Tin, và sự đồng ý của Thầy giáo hướng dẫn Th.s Nguyễn Trọng Chỉnh nhóm đã thực hiện đề tài “Xây Dựng website hỏi đáp thông tin bóng đá”. Để hoàn thành tốt đề tài này, trước tiên, chúng em xin chân thành cảm ơn các thầy cô và ban giám hiệu trường ĐH Công Nghệ Thông Tin nói chung và các thầy cô trong khoa nói riêng đã tạo mọi điều kiện giúp đỡ chúng em trong quá trình học tập và nghiên cứu.

Kế đến, chúng em xin chân thành gửi lời cảm ơn đến thầy Nguyễn Trọng Chỉnh, là giáo viên hướng dẫn và cũng là người đã nhiệt tình chỉ bảo cho chúng em trong suốt quá trình hoàn thành đề tài.

Cuối cùng xin gửi lời cảm ơn đặc biệt đến gia đình luôn là một chỗ dựa vững chắc cũng như luôn tạo mọi điều kiện tốt nhất cho chúng em được học tập.

Tp Hồ Chí Minh, 10/06/2016

Nhóm sinh viên thực hiện

Bùi Đức Lực

Nguyễn Hải Đăng

Lời nói đầu.

Với nhu cầu trao đổi thông tin của con người ngày càng cao, thông tin tràn ngập trên mọi phương tiện truyền thông, đặc biệt là sự phát triển rộng rãi của mạng toàn cầu Internet, hằng ngày con người phải xử lý một lượng thông tin khổng lồ. Những thắc mắc của người dùng dưới dạng truy vấn sẽ được tìm kiếm và trả về một cách ngắn gọn, súc tích, chính xác nhất những gì mà họ mong muốn. Đó chính là mục tiêu của hệ thống hỏi-đáp tự động. Rất nhiều hệ thống hỏi đáp thông tin qua mạng ra đời nhằm đáp ứng nhu cầu này.

Qua báo cáo này, Nhóm sẽ trình bày những gì đã tìm hiểu được về một Hệ Thống Hỏi Đáp Tự Động. Từ đó xây dựng một Hệ Thống Hỏi Đáp Tự Động cụ thể.

Báo Cáo được trình bày thành năm chương, nội dung được trình bày sơ bộ như  
dưới dây:

Chương 1: Trình bày bài toán, mục tiêu, giới hạn

Chương 2: Giới thiệu về hệ thống hỏi đáp, các dạng hệ thống hỏi đáp, các thư viện sử dụng phương pháp đánh giá, Giới thiệu sơ lược về tìm kiếm thông tin, Bing API, ứng dụng web.

Chương 3: Phương pháp trả lời câu hỏi về bóng đá: Trình bày quá trình xác định dạng câu hỏi, phân tích câu hỏi, rút trích câu trả lời, xếp hạng câu trả lời, trả kết quả

Chương 4: Phân tích, Thiết kế và cài đặt hệ thống. Trình bày các chức năng của hệ thống, mô hình hệ thống với chức năng của từng thành phần trong hệ thống, các lớp được cài đặt để thực hiện các chức năng của từng tah2nh phần. Giới thiệu phần mềm đã làm được.

Chương 5: Đánh giá, kết luận và hướng phát triển. Đánh giá kết quả trả lời theo bộ số liệu, nêu kết luận và hướng phát triển.

Chương 1: Giới Thiệu Bài Toán, Mục Tiêu, Giới Hạn

Chương 2: Giới thiệu về hệ thống hỏi đáp, các thư viện, các kỹ thuật sử dụng

1. **Tổng Quan về Hệ Thống Hỏi Đáp Tự Động:**

Hệ thống hỏi-đáp tự động (Question Answering-QA) là một hệ thống được xây dựng để thực hiện việc tìm kiếm câu trả lời cho một câu hỏi của người dùng. Hệ thống hỏi-đáp tự động liên quan đến 3 lĩnh vực lớn là xử lý ngôn ngữ tự nhiên (Natural Language Processing), tìm kiếm thông tin (Information Retrieval) và rút trích thông tin (Information Extraction).

Hệ thống hỏi đáp tự động có thể coi như một lựa chọn thứ hai bên cạnh hệ thống trích chọn thông tin khi người dùng muốn tìm kiếm thông tin họ cần. Hệ thống trích chọn thông tin nhận đầu vào là các từ khóa và trả về tập các tài liệu liên quan (có chứa các từ khóa đó). Kết quả mà hệ thống trích chọn thông tin (máy tìm kiếm) trả lại cho người dùng là rất lớn, có thể lên đến hàng nghìn trang web mà phần nhiều không chứa thông tin người dùng mong muốn. Trong khi đó, hệ thống hỏi đáp nhận đầu vào là câu hỏi dưới dạng ngôn ngữ tự nhiên của người dùng, trả lại các đoạn văn bản ngắn (các snippet) chứa câu trả lời trực tiếp cho câu hỏi. Nghiên cứu về hệ thống hỏi đáp tự động hiện đang thu hút sự quan tâm của rất nhiều các nhà nghiên cứu từ các trường đại học, các viện nghiên cứu và cả các doanh nghiệp lớn trong ngành công nghệ thông tin, có ý nghĩa khoa học lẫn ý nghĩa thực tế. Rất nhiều các hội nghị thường niên về khai phá dữ liệu, trích chọn thông tin dành một chủ đề riêng cho các nghiên cứu về hệ thống hỏi đáp như TREC1 , CLEF2 … Bài toán xây dựng hệ thống hỏi đáp là một bài toán khó thuộc lĩnh vực xử lý ngôn ngữ tự nhiên. Chúng ta biết rằng ngôn ngữ tự nhiên vốn nhập nhằng, đa nghĩa, việc xác định được ngữ nghĩa của câu hỏi cũng như phát hiện ra câu trả lời là một thách thức không nhỏ. Không những vậy, giữa câu hỏi và câu trả lời còn tồn tại các quan hệ “ngầm” hay phụ thuộc vào ngữ cảnh.

Ví dụ câu hỏi “Ai là tác giả Nhật ký trong tù ?” Câu trả lời: “ Hồ Chí Minh viết Nhật ký trong tù trong thời gian Người bị chính quyền Tưởng Giới Thạch bắt giam và giải đi khắp các nhà giam ở Quảng Tây, Trung Quốc.”

“Nhật ký trong tù, dịch tiếng Hán- Ngục trung nhật ký là một tập thơ của Hồ Chí Minh,…”. Hay câu hỏi “Mozart sinh năm nào ?” và câu trả lời “….Mozart (1751 – 1791)…”. Để tìm được câu trả lời trên cho câu hỏi, hệ thống cần có cơ chế để biết được rằng “tác giả của một tập thơ là người viết tập thơ đó” hoặc cần học được các mẫu thường gặp của câu trả lời (các mẫu về ngày tháng năm sinh, về thời gian, địa chỉ….) tương ứng với từng loại câu hỏi. Các hệ thống Q&A trên thế giới hiện nay sử dụng rất nhiều các công cụ xử lý ngôn ngữ như: Bộ gán nhãn từ loại (POS Tagger), bộ nhận dạng tên thực thể (Named Entity Recognizer), bộ phân tích ngữ pháp (Parser)… và các tài nguyên ngôn ngữ như Wordnet, ontology để phân tích câu hỏi và trích xuất câu trả lời.

1. **Lịch Sử Phát Triển:**

Nghiên cứu về QA đã bắt đầu từ những năm 60 của thế kỷ trước. Tuy nhiên, những khó khăn trong việc xử lý ngôn ngữ tự nhiên đã giới hạn việc nghiên cứu QA vào việc nghiên cứu các hệ chuyên gia trả lời câu hỏi trong các lĩnh vực nhỏ hẹp. Vào những năm gần đây, nghiên cứu về hệ QA tổng quát đã được đẩy mạnh cùng với sự xuất hiện của tiểu ban QA (QA Track) ở hội nghị TREC ( Text Retrieval Conference), sự thảo luận về QA ở diễn đàn CLEF (Cross Language Evaluation Forum).

Thiết kế một hệ thống hỏi-đáp không phải là khái niệm mới. Một số hệ thống đầu tiên đã được ra đời từ những năm 1960. Một ví dụ của một hệ thống như vậy là hệ thống BASEBALL được phát triển năm 1961 do nhóm tác giả Green, Chomsky, và Laughery . Hệ thống này được thiết kế để cung cấp các thông tin về các số liệu thống kê của liên đoàn bóng chày Mỹ. Một hệ thống khác tương tự như BASEBALL đã được phát triển bởi Woods (1973) và được đặt tên LUNAR. LUNAR có thể trả lời các câu hỏi liên quan đến các mẫu đá trở về từ tàu thăm dò mặt trăng Apollo. Hệ thống này dịch các câu hỏi thành một hay nhiều truy vấn cơ sở dữ liệu. Hệ thống TEAM được phát triển bởi Grosz (1983) đã có những đặc điểm cơ bản như một chuỗi các biểu diễn ngữ nghĩa và một lược đồ phiên dịch làm cho nó tốt hơn 2 hệ thống trước. Điểm tương đồng giữa cả ba hệ thống là tất cả đều sử dụng cơ sở dữ liệu để lưu trữ cơ sở tri thức. Việc thiết kế của cơ sở dữ liệu này và các dữ liệu có cấu trúc được xây dựng một cách thủ công bởi tất cả các chuyên gia trong các lĩnh vực tương ứng.

Xử lý ngôn ngữ tự nhiên trong quá trình thực hiện đã nhận được rất nhiều sự quan tâm trong những năm 1970. Một trong những hệ thống như thế là Margie (Schank, Goldman, Riesbeck, & Rieger, 1975). Hệ thống này đã có thể xử lý các tài liệu mà sau này có thể trả lời các câu hỏi cơ bản. Điều này được thực hiện bằng cách phân tích và tổ chức tài liệu của thông tin trong một mô hình tương tự như bộ nhớ của con người. Điều này đã được cải thiện hơn bởi Lenhert, Dyer, Johnson, Yang, và Harley (1983) trong hệ thống Boris. Boris và Margie là các hệ thống gần giống nhất với hệ thống hỏi-đáp lĩnh vực rộng (open-domain) hoạt động ngày nay.

Hệ thống hỏi-đáp nhận được rất nhiều mối quan tâm của cộng đồng các nhà nghiên cứu. Điều này bắt nguồn từ việc một lượng tài liệu lớn có gắn chỉ mục sẵn có trên Internet, cùng với các thành tựu trong lĩnh vực rút trích thông tin và sự gia tăng nhu cầu thương mại cho các sản phẩm QA trên toàn cầu. Một lợi thế nữa, dễ dàng cho việc phát triển các hệ thống QA phức tạp hơn, là các nguồn tài nguyên như là WordNet (Fellbaum, 1998) và OpenCyc (Curtis, Matthews, & Baxter, 2005) . WordNet cung cấp sự truy cập tới các thông tin ngữ nghĩa và các mối liên hệ ngữ nghĩa giữa các cụm từ. OpenCyC cung cấp sự truy cập tới thông tin ontology và tri thức thông dụng. Cả 2 hệ thống này đều được cung cấp miễn phí. Hội thảo TREC (Text Retrieval and Evaluation Conference) về lĩnh vực QA năm 1999 đã đóng góp một phần to lớn cho sự phát triển của lĩnh vực QA với 20 thành viên tham gia. Sự kiện này đã thúc đẩy các nhà nghiên cứu chia sẻ kinh nghiệm và so sánh kết quả của họ sử dụng một độ đo chung.

Năm 2000, Carbonell và các đồng nghiệp trong bài báo The Vision Statement to Guide Research in Question Answering and Text Summarization đã đưa ra các tư tưởng chung cho việc nghiên cứu Q&A. Theo đó một hệ thống hỏi đáp được người dùng đánh giá là hữu ích nếu đáp ứng được các tiêu chuẩn:

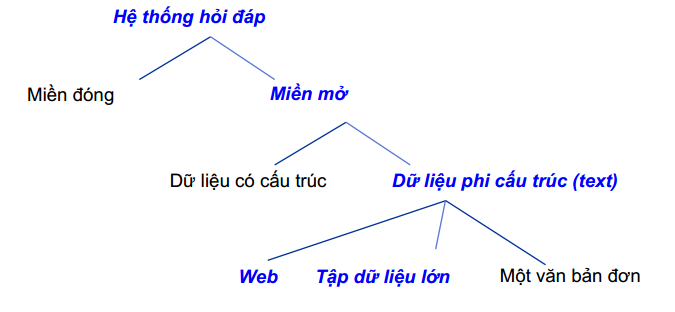
* Tính hợp lý về thời gian (Timeliness): Câu trả lời phải được đưa ra trong thời gian ngắn, ngay cả khi có hàng ngàn người dùng cùng truy nhập hệ thống một lúc. Các nguồn dữ liệu mới cần phải được tích hợp vào hệ thống ngay khi chúng sẵn sàng để có thế cung cấp cho người dùng câu trả lời cho những câu hỏi về các sự kiện có tính thời sự.
* Tính chính xác: Tính chính xác của hệ thống hỏi đáp tự động là cực kì quan trọng bởi việc đưa ra câu trả lời sai còn tai hại hơn nhiều là không đưa ra câu trả lời. Nghiên cứu về Q&A cần tập trung vào việc đánh giá tính đúng đắn của câu trả lời đưa ra, bao gồm cả phương thức để phát hiện các trường hợp mà dữ liệu hiện thời không chứa câu trả lời cho câu hỏi. Các thông tin mâu thuẫn trong dữ liệu cũng cần được tìm ra và các thông tin này cần được xử lý theo một cách phù hợp, nhất quán. Để đạt được sự chính xác, hệ thống Q&A cần được tích hợp các nguồn tri thức (world knowledge ) và cơ chế “bắt chước” việc suy luận thông thường (việc bắt chước có thể hiểu như là một quá trình học).
* Tính khả dụng: Hệ thống Q&A cần đáp ứng được các yêu cầu cụ thể của một người dùng. Các ontology trên từng miền cụ thể và ontology trên miền mở cần được tích hợp trong hệ thống. Hệ thống Q&A cần có khả năng khai phá câu trả lời 5 từ bất kì dạng dữ liệu gì (văn bản, web, cơ sở dữ liệu, …) và đưa ra câu trả lời dưới định dạng mà người dùng mong muốn, cho phép người dùng miêu tả ngữ cảnh của câu hỏi và cung cấp các thông tin giải thích, trích dẫn nguồn cho câu trả lời.
* Tính hoàn chỉnh: Câu trả lời hoàn chỉnh cho câu hỏi của người dùng là điều mà các hệ thống Q&A hướng tới. Trong nhiều trường hợp (câu hỏi về danh sách, nguyên nhân, cách thức…), các phần của câu trả lời nằm rải rác trong một văn bản, thậm chí trong nhiều văn bản. Vì vậy cần phải hợp nhất các phần này dựa trên các thông tin liên kết để tạo ra câu trả lời hoàn chỉnh.
* Tính thích hợp của câu trả lời: Trong ngôn ngữ tự nhiên, câu hỏi đưa ra luôn gắn với ngữ cảnh nào đó và câu trả lời cũng nằm trong một ngữ cảnh nhất định. Câu trả lời mà hệ thống Q&A đưa ra phải phù hợp ngữ cảnh với câu hỏi. Một hệ thống Q&A có khả năng giao tiếp (interactive Q&A) là cần thiết trong nhiều trường hợp bởi chuỗi các câu hỏi liên quan đến một vấn đề sẽ giúp làm sáng tỏ thông tin mà người dùng đang hỏi. Việc đánh giá một hệ thống Q&A cần hướng người dùng bởi ý kiến người dùng là đánh giá tốt nhất cho tính thích hợp của câu trả lời. Các tiêu chuẩn trên được đặt ra với mong muốn xây dựng được một hệ thống Q&A hoàn chỉnh. Tuy nhiên, không phải hệ thống nào cũng có khả năng thông minh và hoàn thiện như thế. Các nghiên cứu về Q&A hiện nay đang tập trung vào xây dựng hệ thống hỏi đáp có tính chính xác cao và có khả năng sử dụng nguồn dữ liệu web khổng lồ trên Internet.

Rất nhiều phương pháp tiếp cận khác nhau được sử dụng trong các hệ thống hỏi-đáp gần đây như phương pháp thống kê, phương pháp dựa trên luật, và các phương pháp tổng hợp. Hầu hết các hệ thống hỏi-đáp ngày càng trở nên phức tạp, thường sử dụng rất nhiều các mô-đun khác nhau, chẳng hạn như tìm kiếm thông tin, phân tích cú pháp câu, phân tích loại câu hỏi, phân tích ngữ nghĩa, và thậm chí dùng phương pháp suy luận (Moldovan et al., 2002) để đánh giá, và xếp hạng câu trả lời. Với sự gia tăng tính phức tạp của hệ thống hỏi-đáp, sẽ rất khó khăn để xác định hiệu quả của hệ thống đến từ hiệu quả của giai đoạn nào.

1. **Phân Loại Hệ Thống Hỏi Đáp:**

Có nhiều cách phân loại hệ thống hỏi đáp dựa trên các tiêu chí khác nhau như: Phân loại theo miền ứng dụng, theo khả năng trả lời câu hỏi, theo cách tiếp cận giải quyết bài toán…

1. **Phân loại theo miền ứng dụng (domain):**

* Hệ thống hỏi đáp miền mở (open domain Question answering): Hệ thống trả lời bất kỳ câu hỏi nào được đưa vào. Khó khăn cho hệ thống miền mở đó chính là việc xây dựng các tri thức nên cho việc trả lời cũng như phân tích câu hỏi, các phương pháp hiện nay thường sử dụng một số các ontology khái quát hay các mạng tri thức như: wikipedia, bách khoa từ điển... Tuy nhiên, dữ liệu cho việc trích rút câu trả lời là phong phú, dễ thu thập.
* Hệ thống hỏi đáp miền đóng (close domain Question answering): Hệ thống tập trung vào trả lời các câu hỏi liên quan đến một miền cụ thể (giáo dục, y tế, thể thao...). Xây dựng hệ thống hỏi đáp miền đóng được coi là bài toán dễ hơn so với xây dựng hệ thống hỏi đáp miền mở vì có thể sử dụng các tri thức miền (thường là ontology của miền cụ thể).

Các nghiên cứu hiện nay về Q&A đang tập trung vào xây dựng hệ thống hỏi đáp trên miền mở, sử dụng nguồn dữ liệu phi cấu trúc (kho văn bản lớn hay dữ liệu web) để tìm câu trả lời. Các nghiên cứu mới và cải tiến những phương pháp cũ để có thể áp dụng cho nguồn dữ liệu web vốn đa dạng, nhiều “nhiễu” và trùng lặp đang rất được quan tâm.

1. **Phân loại theo khả năng trả lời câu hỏi:**

Hệ thống có khả năng trả lời các câu hỏi liên quan đến sự vật, hiện tượng,... dựa trên việc trích ra câu trả lời có sẵn trong tập tài liệu. Câu trả lời là các chuỗi ký tự trong một tài liệu. Kỹ thuật chính được sử dụng là xử lý chuỗi và từ khóa.

Hệ thống có cơ chế lập luận đơn giản: Trích xuất các câu trả lời có sẵn trong tập tài liệu sau đó sử dụng các suy luận để tìm mối liên kết giữa câu trả lời và câu hỏi. Hệ thống sử dụng các nguồn tri thức như ontology về từng miền cụ thể và ontology chung.

Hệ thống trả lời các câu hỏi yêu cầu khả năng tổng hợp: Các phần của câu trả lời được trích rút từ nhiều tài liệu sau đó được tổng hợp lại thành câu trả lời hoàn chỉnh. Câu hỏi thường là về danh sách, về cách thức, nguyên nhân...

Hệ thống có khả năng giao tiếp với người dùng: Trả lời chuỗi các câu hỏi của người dùng về cùng một vấn đề. Ví dụ các câu hỏi của người dùng như: “Giáo sư A sinh năm nào? Ở đâu? Ông ấy đang công tác ở đâu?”.

Hệ thống có khả năng lập luận tương tự: Có thể trả lời các câu hỏi có tính chất suy đoán, câu trả lời ẩn trong tập tài liệu. Hệ thống cần trích ra các luận chứng và sử dụng lập luận tương tự để tìm ra câu trả lời.

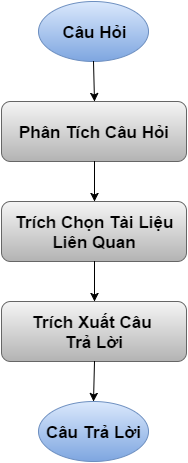
1. **Phân loại theo hướng tiếp cận:**

* Hướng tiếp cận nông (shalow): Nhiều phương pháp sử dụng trong Q&A dùng các kĩ thuật dựa trên từ khóa để định vị các câu, đọan văn có khả năng chứa câu trả lời từ các văn bản được trích chọn về. Sau đó giữ lại các câu, đoạn văn có chứa chuỗi ký tự cùng loại với loại câu trả lời mong muốn (ví dụ các câu hỏi về tên người, địa danh, số lượng…).
* Hướng tiếp cận sâu (deep): Trong những trường hợp khi mà hướng tiếp cận bề mặt không thể tìm ra câu trả lời, những quá trình xử lý về ngữ pháp, ngữ nghĩa và ngữ cảnh là cần thiết để trích xuất hoặc tạo ra câu trả lời. Các kĩ thuật thường dùng như nhận dạng thực thể (named-entity recognition), trích xuất mối quan hệ, loại bỏ nhập nhằng ngữ nghĩa,… Hệ thống thường sử dụng các nguồn tri thức như Wordnet, ontology để làm giàu thêm khả năng lập luận thông qua các định nghĩa và mối liên hệ ngữ nghĩa. Các hệ thống hỏi đáp dựa theo mô hình ngôn ngữ thống kê cũng đang ngày càng phổ biến.

1. **Các Bước Chung của Hệ Thống Hỏi Đáp:**

Một hệ thống hỏi đáp tự động thường gồm 3 bước chung sau:

* Bước 1: Phân tích câu hỏi: Bước phân tích câu hỏi tạo truy vấn cho bước trích chọn tài liệu liên quan và tìm ra những thông tin hữu ích cho bước trích xuất câu trả lời.
* Bước 2: Trích chọn tài liệu liên quan: Bước này sử dụng câu truy vấn được tạo ra ở bước phân tích câu hỏi để tìm các tài liệu liên quan đến câu hỏi.
* Bước3: Trích xuất câu trả lời: Bước này phân tích tập tài liệu trả về từ bước 2 và sử dụng các thông tin hữu ích do bước phân tích câu hỏi cung cấp để đưa ra câu trả lời chính xác nhất.



Các hệ thống hỏi đáp tự động hiện nay có kiến trúc rất đa dạng, tuy nhiên chúng đều bao gồm ba phần cơ bản như trên. Sự khác nhau chính giữa các hệ thống là ở quá trình xử lý trong từng bước, đặc biệt là ở cách tiếp cận trong việc xác định câu trả lời.

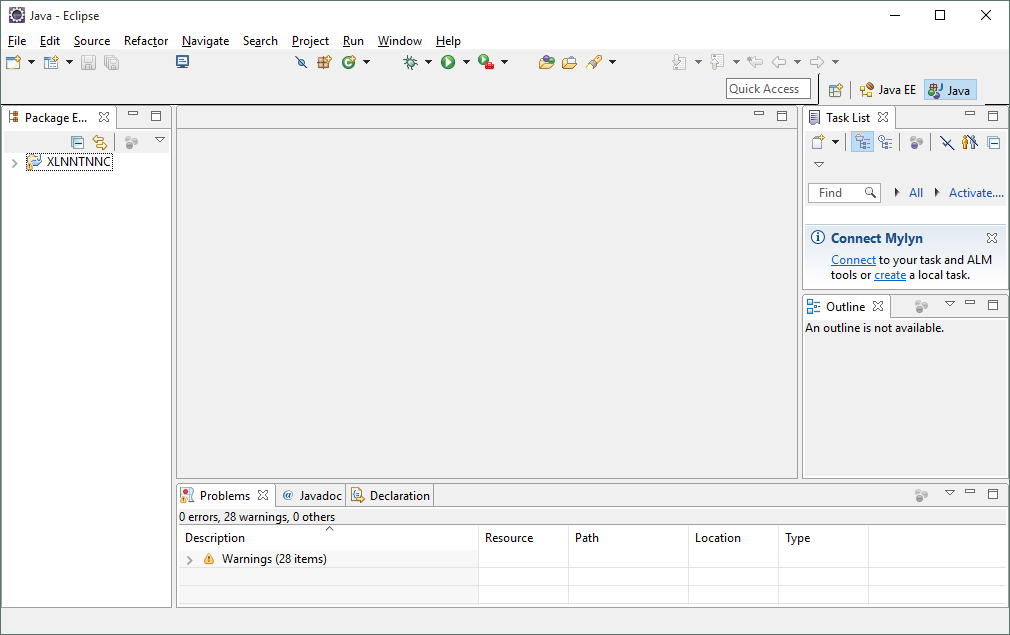
Cách tiếp cận theo trích chọn thông tin thuần túy (pure IR) là: chia nhỏ một tài liệu trong tập dữ liệu thành chuỗi các tài liệu con, trích chọn các tài liệu con có độ tương đồng lớn nhất với câu truy vấn (do bước phân tích câu hỏi tạo ra) và trả lại chúng cho người dùng.Thách thức lớn nhất ở đây là làm sao chia nhỏ được tài liệu thành các phần với kích cỡ tương ứng với kích cỡ của câu trả lời mà vẫn đủ lớn để có thể đánh chỉ mục được (nếu chia quá nhỏ thì số lượng tài liệu để đánh chỉ mục sẽ rất lớn, gây gánh nặng cho hệ thống trích chọn thông tin).

Cách tiếp cận theo xử lý ngôn ngữ tự nhiên (pure NLP) là: so khớp giữa biểu diễn ngữ pháp và (hoặc) biểu diễn ngữ nghĩa của câu hỏi với dạng biểu diễn ngữ pháp, ngữ nghĩa của các câu trong các tài liệu liên quan trả về. Khó khăn của cách tiếp cận này là hệ thống phải thực hiện việc phân tích ngữ pháp, ngữ nghĩa và so khớp đủ nhanh để đưa ra câu trả lời trong thời gian chấp nhận được, bởi số lượng các tài liệu cần xử lý là rất lớn trong khi các bước phân tích trên lại phức tạp và tốn nhiều thời gian.

Sự khác nhau trong cách trích xuất câu trả lời dẫn đến việc phân tích câu hỏi cũng trở nên đa dạng. Trong hướng tiếp cận theo trích xuất thông tin thuần túy, phân tích câu hỏi chỉ cần làm tốt việc tạo truy vấn, trong khi với hướng tiếp cận theo xử lý ngôn ngữ tự nhiên, câu hỏi cần được phân tích ngữ pháp, ngữ nghĩa một cách chính xác. Các hệ thống hiện nay thường là sự kết hợp giữa hai hướng tiếp cận, sử dụng hệ thống trích chọn thông tin để thu hẹp không gian tìm kiếm câu trả lời, đồng thời phân tích câu hỏi để tìm ra các thông tin về ngữ pháp, ngữ nghĩa nhằm tìm ra câu trả lời chính xác nhất. Kết quả của bước phân tích câu hỏi là đầu vào cho cả hai bước trích chọn tài liệu liên quan và trích xuất câu trả lời. Bước phân tích câu hỏi có ý nghĩa rất quan trọng, bởi nó ảnh hưởng đến hoạt động của các bước sau và do đó quyết định đến hiệu quả của toàn hệ thống. Chương 2 trình bày chi tiết các nội dung liên quan đến phân tích câu hỏi.

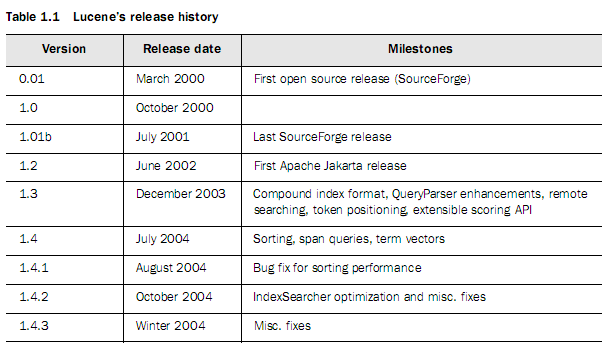
**Chương 3 Giới Thiệu Tổng Quan Công Cụ Sử Dụng:**

1. **Eclipse:**

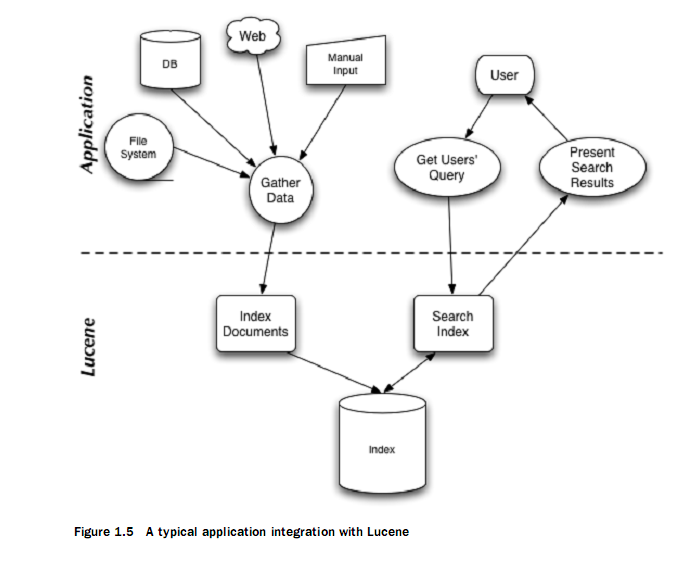
* Eclipse là một nền tảng phát triển mã nguồn mở dựa trên Java có thể mở rộng được. Eclipse bao gồm một tập hợp các dịch vụ để xây dựng các ứng dụng từ các thành phần plug-in. Eclipse được tạo thành bởi một tập hợp các plug-in chuẩn JDT (Java Development Tools).
* Dự án Eclipse được IBM bắt đầu vào tháng 11 năm 2001. Các dự án của Eclipse tập trung vào việc xây dựng nền tảng mã nguồn mở bao gồm các framework có thể mở rộng, các tool và thư viện để xây dựng, triển khai và quản lý các phần mềm.
* Eclipse là một tổ chức phi lợi nhuận, là nơi phát triển cộng đồng mã nguồn mở và hệ thống các sản phẩm, dịch vụ. Những người phát triển plug-in cho Eclipse hoặc những người sử dụng Eclipse như là nền tảng cho các ứng dụng phát triển phần mềm cần tuân thủ giấy phép EPL.
* Download, cài đặt Eclipse: [*http://www.eclipse.org/downloads/*](http://www.eclipse.org/downloads/)

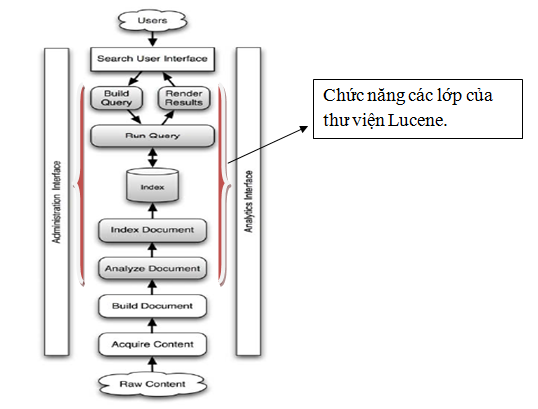
1. **Lucene:**

* Tác giả : được khởi xướng bởi Doug Cutting (<http://en.wikipedia.org/wiki/Doug_Cutting> ) và trở thành một trong những sản phẩm chất lượng cao của Apache Foundation tháng 9-2001.
* Source code : <http://www.manning.com/hatcher2>.
* Một vài ứng dụng search trên apache : <http://wiki.apache.org/jakarta-lucene/PoweredBy> .
* Lịch sử phát triển Lucene:



* Lucene là một thư viện cho việc tìm kiếm toàn văn, được phát triển bởi Dough Cutting (hiện đang làm việc cho Apache). Thư viện này cung cấp các hàm cơ bản hỗ trợ cho việc đánh chỉ mục và tìm kiếm. Để có thể sử dụng Lucene, bạn cần phải có sẵn dữ liệu. Dữ liệu có thể là tập hợp các tập tin dạng PDF, Word hay là các trang web HTML… Hoặc là dữ liệu lưu trong các hệ quản trị CSDL như MS SQL Server hay MySQL. Dùng Lucene, bạn có thể tiến hành đánh chỉ mục trên dữ liệu hiện có để sau này có thể thực hiện thao tác tìm kiếm toàn văn trên dữ liệu đó.
* Lucene là một thư viện tìm kiếm thông tin có khả năng xử lý và khả năng mở rộng ở mức cao, cho phép chúng ta có thể tích hợp vào các ứng dụng. Lucene là một dự án mã nguồn mở và nguyên thuỷ được phát triển bằng ngôn ngữ Java, ngày nay Lucene được phát triển bằng nhiều ngôn ngữ khác nhau như Delphi, Perl, C#, C++, Python, Ruby và PHP…
* Thành phần chức năng chính của Lucene bao gồm hai phần: Thành phần tạo chỉ mục và thành phần tìm kiếm. Đây là hai thành phần quan trọng cho một hệ thống tìm kiếm.
* Từ thư viện Lucene này, có nhiều kịch bản sử dụng sau :
* Dùng Lucene tích hợp vào ứng dụng hiện có. Muốn phát triển một semantic video search engine, trong đó đã có dữ liệu văn bản là các transcript và muốn có một công cụ hỗ trợ cho việc tìm kiếm dựa trên văn bản. Bằng cách này, người dùng có thể gõ vào từ khóa President Bush để tìm các video transcript có nói về President Bush. Sử dụng các hàm trong thư viện Lucene liên quan đến việc đánh chỉ mục và tìm kiếm, ta có thể thực hiện thao tác này khá dễ dàng.
* Xây dựng một search engine riêng. Lúc này cần phải có một web crawler chuyên đi thu thập các trang web trên Internet, để đem về đánh chỉ mục và cho phép tìm kiếm. Ứng dụng kiểu này có thể thấy tương tự ở trang www.baomoi.com. Trong ứng dụng này, web crawler sẽ được dùng để đi thu thập các tin từ các website (chủ yếu là tin tức, ví dụ vnexpress, tuoitre.com.vn, nld.com.vn, etc), sau đó tiến hành phân loại, lập chỉ mục để hỗ trợ tìm kiếm. Trong trường hợp cần web crawler, Nutch là một phần mềm mã nguồn mở có thể dùng , cũng do chính tác giả của Lucene là Dough Cutting phát triển. Để có thể tiến hành thu thập và lưu trữ hàng triệu trang web một cách có hiệu quả ở nhiều máy khác nhau, Hadoop sử dụng công nghệ của GoogleFS có thể được tích hợp cùng. Bên cạnh đó Solr, một phần mềm mã nguồn mở dùng cho xây dựng các search server, cung cấp giao diện bằng web với người sử dụng cũng được xây dựng dựa trên thư viện Lucene.
* Lucene ban đầu được viết hoàn toàn bằng Java. Sau đó được port qua các ngôn ngữ khác ví dụ như C, C++ ( CLucene), .NET (Lucene.NET ), Perl (Plucene), Ruby ( Ferret) và đặc biệt là PHP (Zend Framework ).

Sơ đồ thể hiện mối tương tác của một ứng dụng sử dụng Lucene.

* Những người mới tiếp xúc với Lucene thường nhầm Lucene như một chương trình tìm kiếm hoặc một web site search engine, điều này là hoàn toàn không đúng, Lucene là một thư viện nguồn mở hỗ trợ việc đánh chỉ mục và tìm kiếm toàn văn.
* Như ta thấy trong hình trên đó là các thành phần trong ứng dụng tìm kiếm cơ bản, ta cũng sẽ thấy rõ được chức năng các lớp trong thư viện Lucene hỗ trợ (phần màu trong ngoặc nhọn đỏ):
* Phân tích dữ liệu (dạng văn bản) để đánh chỉ mục: Analyze Document
* Đánh chỉ mục: Index Document và Index
* Thực hiện việc xây dựng các câu truy vấn và tìm kiếm trong chỉ mục: Build Query, Render results, RunQuery.
* Lucene cho phép thêm khả năng tìm kiếm cơ bản vào ứng dụng, Lucene có thể đánh chỉ mục (index) và tìm kiếm bất cứ dữ liệu gì (các loại dữ liệu này cần chuyển về dữ liệu dạng văn bản (text)).
* Thư viện này cung cấp các lớp với các hàm cơ bản hỗ trợ cho việc đánh chỉ mục và tìm kiếm. Lucene chỉ là một phần lõi của một search engine, nó không bao gồm web spider hay bộ phân tích định dạng dữ liệu (phân tích các định dạng dữ liệu khác nhau thành dữ liệu dạng văn bản (text)).
* Lucene không quan tâm đến nguồn dữ liệu, định dạng dữ liệu, thậm chí ngôn ngữ của nó, chỉ cần bạn có thể chuyển đổi nó sang text. Nghĩa là bạn có thể đánh chỉ mục và tìm kiếm các dữ liệu từ servers từ xa hoặc là local, các tập tin văn bản, MS word, html, pdf … (Wikipedia).

1. **VnTagger:**

* VnTagger là chương trình gán nhãn từ loại tự động tiếng Việt mã nguồn mở, có độ chính xác cao.
* VnTagger thực hiện tách câu, tách từ và gán nhãn từ loại (danh từ, động từ, tính từ...) tự động cho văn bản tiếng Việt.

1. **VnTokenizer:**

* VnTokenizer là công cụ tách từ tiếng Việt được nhóm tác giả Nguyễn Thị Minh Huyền, Vũ Xuân Lương và Lê Hồng Phương phát triển dựa trên phương pháp so khớp tối đa (Maximum Matching) với tập dữ liệu sử dụng là bảng âm tiết tiếng Việt và từ điển từ vựng tiếng Việt.
* Công cụ được xây dựng bằng ngôn ngữ Java, mã nguồn mở. Có thể đễ dàng sửa đổi nâng cấp và tích hợp vào các hệ thống phân tích văn bản tiếng Việt khác.
* Quy trình thực hiện tách từ theo phương pháp khớp tối đa:
* Quy trình tách từ
  + Đầu vào của công cụ tách từ vnTokenizer là một câu hoặc một văn bản được lưu dưới dạng tệp.
  + Đầu ra là một chuỗi các đơn vị từ được tách.
* Các đơn vị từ bao gồm các từ trong từ điển cũng như các chuỗi số, chuỗi kí từ nước ngoài, các hình vị ràng buộc (gồm các phụ tố), các dấu câu và các chuỗi kí tự hỗn tạp khác trong văn bản (ISO, 2008). Các đơn vị từ không chỉ bao gồm các từ có trong từ điển, mà cả các từ mới hoặc các từ được sinh tự do theo một quy tắc nào đó (như phương thức thêm phụ tố hay phương thức láy) hoặc các chuỗi kí hiệu không được liệt kê trong từ điển.
* Công cụ sử dụng tập dữ liệu đi kèm là tập từ điển từ vựng tiếng Việt, danh sách các đơn vị từ mới bổ sung, được biểu diễn bằng ôtômat tối tiểu hữu hạn trạng thái, tệp chứa các biểu thức chính quy cho phép lọc các đơn vị từ đặc biệt (xâu dạng số, ngày tháng,…), và các tệp chứa các thống kê unigram và bigram trên kho văn bản tách từ mẫu.
* Với các đơn vị từ đã có trong từ điển, khi thực hiện tách từ cũng được xử lý hiện tượng nhập nhằng bằng cách kết hợp với các thống kê unigram và bigram. Chẳng hạn trong tiếng Việt thường gặp các trường hợp nhập nhằng như:
* Xâu AB vừa có thể hiểu là 1 đơn vị từ, vừa có thể là chuỗi 2 đơn vị từ A-B.
* Xâu ABC có thể tách thành 2 đơn vị AB-C hoặc A-BC.
* Đánh giá kết quả:

Kết quả đánh giá của công cụ được cho là ổn định đối với nhiều loại văn bản/ văn phong khác nhau. Độ chính xác trung bình đạt được là khoảng 94%.