ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

LÊ VĂN HÀO

NGHIÊN CỨU XÂY DỰNG HỆ THỐNG TÌM KIẾM VIDEO DỰA TRÊN NỘI DUNG

Ngành: Công nghệ thông tin
Chuyên ngành: Hệ thống thông tin
Mã số: 60.48.01.04

TÓM TẮT LUẬN VĂN THẠC SĨ CÔNG NGHỆ THÔNG TIN

Hà Nội - 2016

MỤC LỤC

BẢNG CHỮ CÁI VIẾT TẮT

STT	Từ viết tắt	Ý nghĩa
1	ASR	Automatic Speech Recognition – Nhận dạng tiếng nói tự động
2	FPS	Frame Per Second – Số khung hình trên một giây
3	FTP	File Transfer Protocol – Giao thức truyền tệp tin
4	GNU	General Public License – Giấy phép công cộng
5	OCR	Optical Character Recognition – Nhận dạng kí tự quang học
6	PDF	Portable Document Format – Định dạng tài liệu di động.
7	NDD	Near Duplicate Detection – Phát hiện gần trùng lặp
8	TIFF	Tagged Image File Format – Định dạng tệp tin trên máy tính để lưu trữ các hình ảnh.
9	UTF-8	Unicode Transformation Format - Định dạng chuyển đổi Unicode.

DANH MỤC CÁC BẢNG BIỂU

DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ

MỞ ĐẦU

Cùng với sự phát triển của công nghệ thông tin, tốc độ internet đang cải thiện đáng kể. Số lượng video bài giảng, diễn thuyết… phục vụ học tập cho mọi lứa tuổi đang được tải lên và chia sẻ trên internet nhanh chóng. Mỗi ngày, hàng triệu video như vậy trên thế giới được đăng tải lên các ứng dụng internet như Youtube, Facebook, Yahoo. Đối với lượng video đang tăng trưởng từng ngày này, cơ chế tổ chức lưu trữ phục vụ cho việc tra cứu, tìm kiếm là một thách thức.

Giáo dục trực tuyến hay E-Learning không còn là khái niệm mới lạ và đang phát triển mạnh mẽ. Số lượng video bài giảng, diễn thuyết cũng vì thế ngày càng được tăng trưởng. Nhu cầu tìm kiếm của người học càng yêu cầu khắt khe hơn: cả về độ chính xác và thời gian tìm kiếm. Tuy nhiên, các chức năng tìm kiếm bài giảng cho của các hệ thống hiện tại thông thường chỉ cho phép người dùng tìm kiếm với tên bài giảng, tên học phần, hoặc tên giảng viên... Các chức năng này thường cho kết quả có độ chính xác không cao, và các kết quả trả về có nhiều nội dung không liên quan đến mục đích tìm kiếm thực sự của người dùng. Do đó, cần có một hệ thống mà có thể "hiểu" được nội dung của từng video bài giảng để phục vụ cho việc tìm kiếm của người dùng.

Những công cụ tìm kiếm phổ biến hiện nay - như Google, Yahoo, Bing…, là những hệ thống tìm kiếm dựa trên "từ khóa", và tìm kiếm trên dữ liệu văn bản (text). Chính vì thế, nếu video không có bất kỳ siêu dữ liệu (metadata) ví dụ như ngày, tác giả, từ khóa, hoặc mô tả thì không thể tìm kiếm được bằng cách sử dụng các công cụ nêu trên. Siêu dữ liệu thường được thêm bằng tay, quá trình này sẽ rất tốn thời gian. Hơn nữa, ngay cả khi một đoạn video có thể được tìm thấy bằng siêu dữ liệu của nó, công cụ tìm kiếm thông thường không có khả năng tìm kiếm một đoạn bài giảng, slide cụ thể trong video mà người dùng quan tâm.

Mục tiêu chính của của Luận văn là tập trung *nghiên cứu xây dựng một hệ thống tìm kiếm các bài giảng, thuyết trình, trình diễn bằng slide dưới dạng video*. Hệ thống sẽ cho phép người dùng chỉ cần nhập vào một phần nội dung của bài

giảng, kết quả trả về sẽ là những video bài giảng có liên quan đến chuỗi truy vấn. Ngoài ra, với giải pháp này cũng cho phép các hệ thống tìm kiếm có thể truy vấn dữ liệu video mà không cần có siêu dữ liệu. Xuất phát từ quan điểm nêu trên, ngoài phần mở đầu và kết luận, luận văn được chia làm 4 chương được tóm tắt như sau:

- Chương 1: Giới thiệu về công cụ tìm kiếm trên mạng internet, các khái niệm và kiến trúc của công cụ tìm kiếm. Các vấn đề cần giải quyết trong luận văn và ý nghĩa khoa học, thực tiễn của luận văn.

- Chương 2: Trình bày về các bài toán cần giải quyết trong khuôn khổ tìm kiếm video bài giảng dạng slide. Một số khái niệm, mô hình các bài toán con cần giải quyết. Các phương pháp tiếp cận để giải quyết vấn đề.

- Chương 3: Là chương quan trọng nhất của Luận văn. Nội dung chính của chương này là tập trung trình bày giải pháp thực hiện của tác giả, các kĩ thuật áp dụng để trích xuất văn bản, xử lý văn bản và đánh chỉ mục tìm kiếm cho video bài giảng.

- Chương 4: Là phần trình bày các kết quả thực nghiệm và đánh giá. Ở mỗi bài toán tác giả đều có những thực nghiệm để kiểm chứng và đánh giá về độ chính xác.

Tác giả xin bày tỏ lòng biết ơn chân thành tới PGS.TS. Nguyễn Trí Thành, thầy đã luôn ân cần, chỉ bảo, động viên, giúp đỡ tác giả trong suốt quá trình thực hiện Luận văn. Tác giả xin chân thành cảm ơn gia đình, bạn bè, đồng nghiệp đã luôn tin tưởng, động viên và giúp đỡ về nhiều mặt trong thời gian qua. Tác giả xin chân thành cảm ơn các thầy, cô giáo trong khoa Công nghệ Thông tin và Truyền thông, trường Đại học Hồng Đức đã động viên và tạo điều kiện giúp đỡ tác giả hoàn thành tốt nhất luận văn này.

CHƯƠNG 1: GIỚI THIỆU

1.1. Giới thiệu về công cụ tìm kiếm

Nếu bạn đã từng truy cập địa chỉ www.google.com.vn, nhập nội dung cần tra cứu và bấm vào "tìm với google". Một danh sách kết quả liên quan đến nội dung tìm kiếm được liệt kê trên màn hình cho phép người dùng lựa chọn các nội dung phù hợp với yêu cầu. Những công cụ cho phép người dùng tìm kiếm các thông tin trên mạng như Google, Bing, Yahoo… như vậy gọi là các công cụ tìm kiếm (web search engine).

Thuật ngữ "web search engine" được định nghĩa: "Một công cụ tìm kiếm là các ứng dụng thực tế của các kĩ thuật truy hồi thông tin trên miền dữ liệu văn bản qui mô lớn"[5].

1.2. Lịch sử phát triển của công cụ tìm kiếm

Năm 1990, Archie là công cụ tìm kiếm đầu tiên được phát triển bởi Alan Emtage, Bill Heelan and J. Peter Deutsch, hai sinh viên chuyên ngành khoa học máy tính của trường McGill University tại Montreal (Canada).

Năm 1991, một công cụ tương tự Archie là Gopher của tác giả Mark McCahill tại University of Minnesota, có chức năng tìm kiếm theo tên tệp tin và tiêu đề được lưu trữ trong hệ thống Gopher đã lập chỉ mục.

Năm 1993, đánh dấu những bước tiến mới về công cụ tìm kiếm như World Wide Web Wanderer bởi Matthew Gray, đây được xem là một web robot đầu tiên đo lường được dung lượng của trang web.

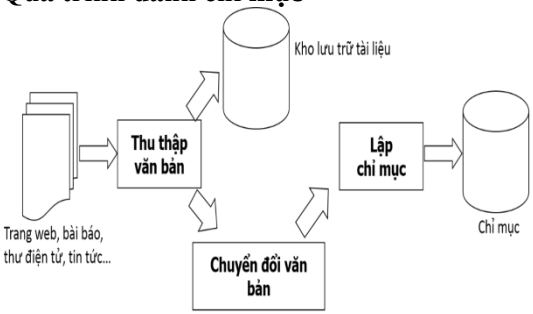
Năm 1994, với sự ra đời của WebCrawler công cụ tìm kiếm đầu tiên chỉ mục toàn trang web và cho phép người dùng tìm kiếm và thu thập với bất kỳ từ nào một cách tự động.

Năm 1995, công cụ tìm kiếm yahoo được tạo bởi David Filo và Jerry Yang. Sử dụng danh bạ web thay vì đánh chỉ mục toàn văn bản.

Năm 1996-nay, với sự phát triển mạnh mẽ của internet các công cụ tìm kiếm phát triển mạnh mẽ hơn, tối ưu hơn nhiều so với các công cụ trước đây. Năm 1998, Google được phát triển bởi Larry và Sergey đưa ra khái niệm về PageRank đánh dấu sự phát triển vượt bậc và hiện đang là công cụ tìm kiếm có thị phần lớn nhất hiện nay.

1.3. Kiến trúc của công cụ tìm kiếm

1.3.1. Quá trình đánh chỉ mục



Hình 1.1. Quá trình đánh chỉ mục

Các thành phần chính bao gồm việc thu thập văn bản, chuyển đổi văn bản và tạo chỉ mục.

1.3.2. Quá trình truy vấn

Phần còn lại của công cụ tìm kiếm là quá trình truy vấn. Quá trình truy vấn thông thường bao gồm ba thành phần chính là tương tác người dùng, xếp hạng và đánh giá.

1.4. Công cụ tìm kiếm video trên mạng internet

Công cụ tìm kiếm video cho phép người dùng thuận tiện trong quá trình tìm kiếm phục vụ nhiều mục đích khác nhau như giải trí, giáo trục và truyền thông. Nếu chia theo lĩnh vực thì video cũng rất đa dạng như video truyền hình, video quảng cáo, video bài giảng, học thuyết… Trong nội dung của luận văn này tác giả chỉ đề cập đến thể loại video bài giảng dạng slide, cách tiếp cận và hướng giải quyết để xây dựng một công cụ tìm kiếm video bài giảng dạng slide. Việc tìm kiếm yêu cầu bằng cách duyệt qua tập các video kết quả.

1.5. Tổng quan của đề tài và các vấn đề cần giải quyết

1.5.1. Tổng quan đề tài

Trong đề tài này, tác giả hướng tới xây dựng một hệ thống tìm kiếm các video bài giảng, thuyết trình, trình diễn bằng silde dưới dạng video… Cho phép tìm thấy những video bằng văn bản xuất hiện trong đó. Với giải pháp này, đơn giản bằng cách nhập từ khóa tìm kiếm, người dùng có thể tìm kiếm các video bài giảng và những cảnh trong đó mà thuật ngữ xuất

hiện. Giải pháp này cũng cho phép người dùng tìm kiếm các video không cần có siêu dữ liệu.

1.5.2. Các vấn đề cần giải quyết

Vấn đề cần giải quyết ở trong đề tài này là giải pháp xử lý video đầu vào. Phân tích và đánh chỉ mục cho video. Đầu tiên, các đoạn video tĩnh trong một thời gian nhất định được xác định là các slide và trích xuất từ video. Tiếp theo, các dữ liệu văn bản chứa trong hình ảnh của slide được trích xuất bằng cách sử dụng kĩ thuật nhận dạng kí tự quang học. Các văn bản trích xuất sẽ được xử lý trùng lặp, sửa lỗi chính tả và được đánh chỉ mục tương ứng với video gốc lưu trữ trong cơ sở dữ liệu.

1.6. Ý nghĩa khoa học và thực tiễn của đề tài nghiên cứu

1.6.1. Ý nghĩa khoa học

- Đề tài đã tổng quát được các phương pháp khoa học để giải quyết vấn đề lập chỉ mục video bài giảng, phục vụ quá trình truy hồi thông tin.
- Đề tài cung cấp các cơ sở khoa học, định hướng cho các nghiên cứu về xử lý lập chỉ mục cho video bài giảng.
- Từ kết quả nghiên cứu của đề tài, góp phần làm cơ sở thực tiễn và lý luận để phát triển hệ thống tìm kiếm video dựa trên nội dung.

1.6.2. Ý nghĩa thực tiễn

- Hướng tiếp cận mới cho các máy tìm kiếm, truy hồi thông tin video dựa trên nội dung.
- Góp phần nâng cao chất lượng của các máy tìm kiếm. Kết quả trả về có độ liên quan cao hơn so với phương pháp tìm kiếm dựa trên từ khóa hiện nay.

1.7. Kết luận

Trong chương này, luận văn đã giới thiệu khái quát một số khái niệm, lịch sử và kiến trúc của một công cụ tìm kiếm nói chung. Ngoài ra, luận văn cũng giới thiệu tổng quan các vấn đề cần giải quyết của đề tài. Ý nghĩa khoa học và ý nghĩa thực tiễn của đề tài nghiên cứu xây dựng hệ thống tìm kiếm video dựa trên nội dung.

CHƯƠNG 2: BÀI TOÁN TÌM KIẾM VIDEO BÀI GIẢNG DỰA TRÊN NỘI DUNG

2.1. Phát biểu bài toán

Tác giả sẽ xây dựng công cụ tìm kiếm cho phép nhận nội dung truy vấn là chuỗi văn bản và kết quả trả về là các video bài giảng mà nội dung có liên quan đến chuỗi văn bản người dùng truy vấn. Bài toán được chia thành hai bài toán con như sau:

Bài toán 1: Xử lý video đầu vào, trích xuất văn bản từ video.

Đầu vào:

- Tập videos bài giảng dạng slide.

Đầu ra:

- Văn bản trích xuất nội dung từ video đầu vào.

Bài toán 2: Lập chỉ mục và tìm kiếm video dựa trên nội dung bài giảng.

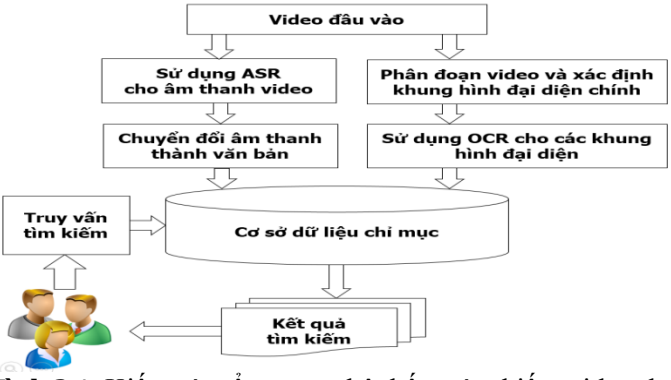
Đầu vào:

- Truy vấn từ người dùng.

Đầu ra:

- Danh sách kết quả videos có nội dung liên quan đến truy vấn.

Kiến trúc chung của một hệ thống tìm kiếm video dựa vào nội dung được miêu tả trong hình 2.1.



Hình 2.1. Kiến trúc tổng quan hệ thống tìm kiếm video dựa trên nội dung

Hình 2.1 cho thấy quá trình lập chỉ mục cho video được trải qua ba bước là phân đoạn video, trích xuất nội dung từ video và lập chỉ mục cho video.

2.2. Các nghiên cứu về tìm kiếm video dựa trên nội dung

Liška et al và cộng sự đã đề xuất một hệ thống tự động cho việc lập chỉ mục video bài giảng [8].

Hunter et al đề xuất một hệ thống lập chỉ mục cho các bài thuyết trình đa phương tiện[7].

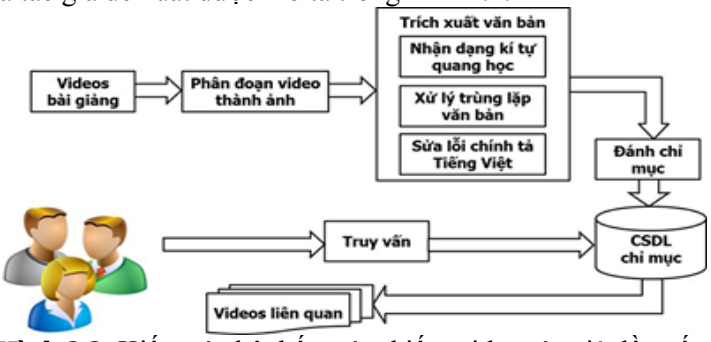
Yang et al sử dụng công cụ nhận dạng giọng nói tự động ASR để trích xuất nội dung video thành văn bản[8].

Lienhart et al đề xuất một phương pháp phát hiện văn bản trong video và hình ảnh[8].

2.3. Hướng nghiên cứu của tác giả

Dựa vào các phương pháp tiếp cận nghiên cứu đã nêu trong phần 2.2, tác giả lựa chọn phương pháp tiếp cận để trích xuất văn bản từ video bằng công nghệ OCR thay vì sử dụng ASR.

Kiến trúc của công cụ tìm kiếm video dựa vào nội dung mà tác giả đề xuất được mô tả trong hình 2.2.



Hình 2.2. Kiến trúc hệ thống tìm kiếm video tác giả đề xuất

2.4. Bài toán phân đoạn video thành ảnh

2.4.1. Khái niệm

Về mặt bản chất thì video mà chúng ta thấy trên tivi, máy tính, điện thoại… được cấu thành từ những ảnh tĩnh. Những ảnh này sau đó được sắp xếp liên tiếp nhau và cùng trình diễn trong một đơn vị thời gian đủ nhỏ để làm cho mắt của chúng ta cảm nhận rằng các đối tượng này đang chuyển

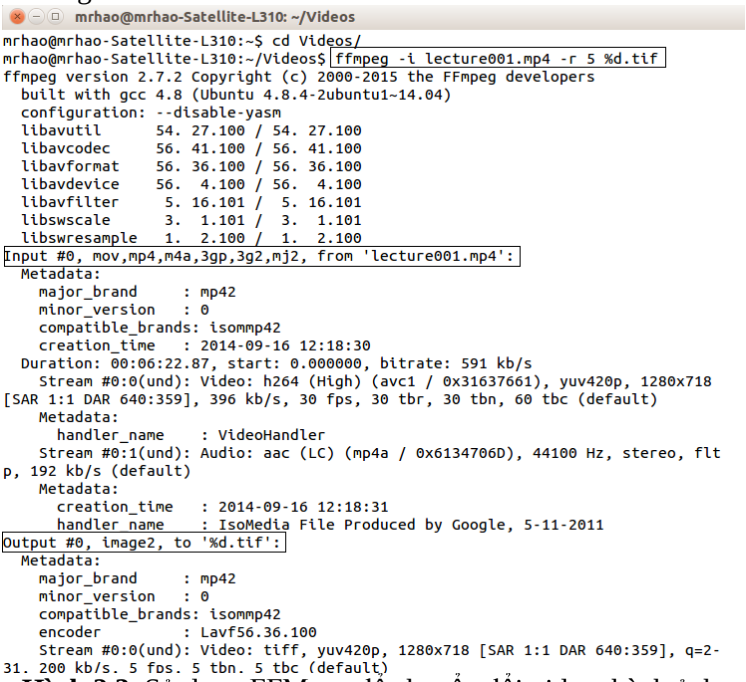
động. Thông thường thì các video được quay ở khoảng 24-30 hình mỗi giây. Mỗi hình này được gọi là một frame. Số frame trên một giây được đo bằng một số nguyên được kí hiệu FPS.

2.4.2. Phương pháp tiếp cận

FFMpeg là một thư viện có rất nhiều tiện ích cho việc xử lý video. Tính năng nổi bật nhất có lẽ là khả năng encode/decode nhiều video định dạng khác nhau, giúp chuyển đổi qua lại nhiều định dạng video. Ngoài ra, chúng ta cũng có thể dùng FFMpeg để chia cắt một đoạn video, chụp lại các frame và xuất ra dạng hình ảnh.

Công cụ mã nguồn mở Ffmpeg được tác giả quan tâm bởi ba lý do chính:

- Hỗ trợ nhiều định dạng video khác nhau, ví dụ .mp4, avi, flv…
- Điều chỉnh được FPS.
- Mã nguồn mở.



Hình 2.3. Sử dụng FFMpeg để chuyển đổi video thành ảnh

2.5. Bài toán trích xuất văn bản

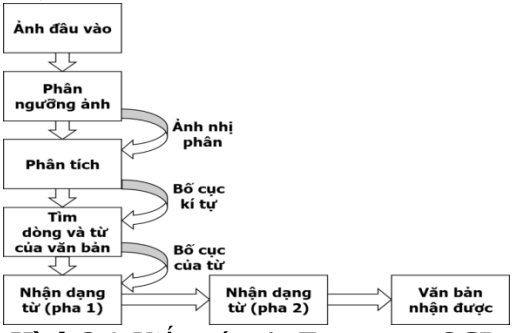
2.5.1. Bài toán nhận dạng kí tự quang học

2.5.1.1. Khái niệm OCR

OCR là công nghệ cho phép chuyển đổi các loại tài liệu khác nhau, ví dụ như các tài liệu giấy, ảnh chụp hoặc các tập tin PDF bằng một máy ảnh kỹ thuật số thành dữ liệu văn bản có thể chỉnh sửa và tìm kiếm. Những hình ảnh này có thể là các chữ viết tay hoặc đánh máy. Đây là một kỹ thuật phổ biến của việc số hóa các văn bản in để có thể tìm kiếm bằng điện tử, lưu trữ gọn gàng, hiển thị trên mạng.

2.5.1.2. Phương pháp tiếp cận

Tác giả sử dụng Tesseract- OCR để thực hiện trích xuất nội dung văn bản từ ảnh. Tesseract là một công cụ nhận diện kí tự quang học mã nguồn mở và hiện nay được phát triển bởi Google[8]. Hình 2.4 mô tả các bước mà công cụ Tesseract-OCR thực hiện.



Hình 2.4. Kiến trúc của Tesseract – OCR

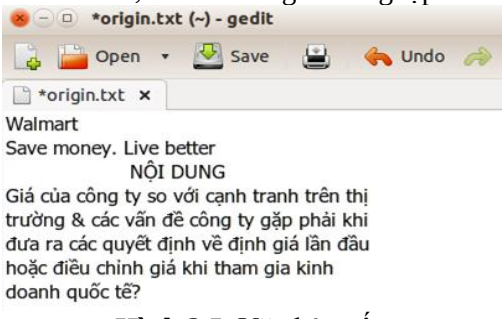
2.5.2. Bài toán xử lý trùng lặp văn bản

2.5.2.1. Khái niệm

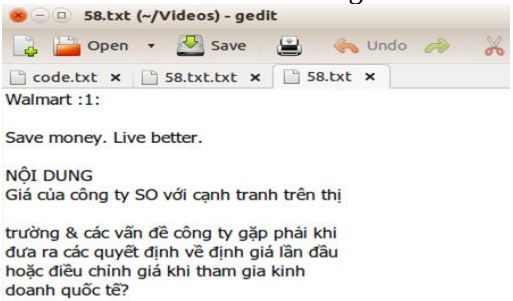
Khi hai văn bản mà nội dung đều giống hệt nhau thì chúng được coi là các văn bản trùng lặp hay gọi là bản sao của nhau. Trong nhiều trường hợp, hai tài liệu mà không phải giống nhau hoàn toàn vẫn có thể chứa cùng một nội dung thì được gọi là các văn bản gần trùng lặp.

Với đặc thù là các văn bản được trích xuất từ các khung hình video bài giảng liên tiếp theo nhau thời gian. Chính vì thế tập hợp văn bản thu được tồn tại cả hai loại đó là trùng lặp và

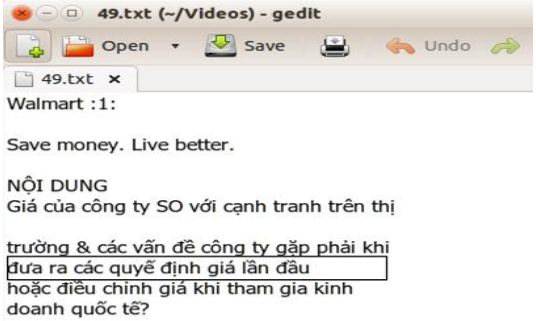
gần trùng lặp văn bản. Hình 2.6 là ví dụ về nội dung văn bản trùng lặp với hình 2.5, hình 2.7 là gần trùng lặp của hình 2.5.



Hình 2.5. Văn bản gốc



Hình 2.6. Văn bản trùng lặp của văn bản trong hình 2.5

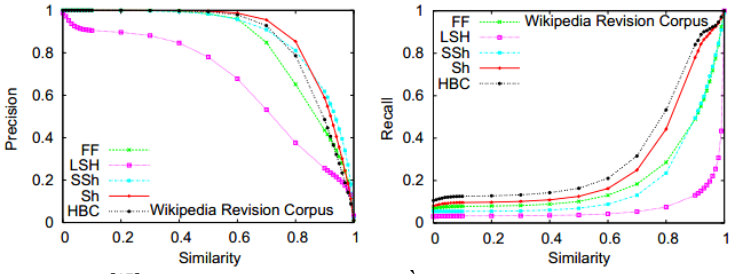


Hình 2.7. Văn bản gần trùng lặp của văn bản trong hình 2.5.

2.5.2.2. Phương pháp tiếp cận

Dựa trên các kết quả nghiên cứu [2], [6], [9], [13],[15] thì phương pháp shingling cho kết quả độ chính xác cao và phù hợp với kiểu dữ liệu đầu vào như tập dữ liệu của tác giả. Chính vì thế, trong luận văn này, tác giả lựa chọn và cài đặt

thuật toán phát hiện trùng lặp văn bản dựa vào kĩ thuật Shingling của Broder và cộng sự. Hình 3.12 bảng kết quả độ chính xác và độ hồi tưởng của các kĩ thuật tìm trùng lặp văn bản theo nghiên cứu [15].



Hình 2.8 [15]**.** Độ chính xác và độ hồi tưởng của độ đo tương tự cho phương pháp fuzzy-fingerprinting (FF), localitysensitive hashing (LSH), supershingling (SSh), shingling (Sh), and hashed breakpoint chunking (HBC).

2.5.3. Bài toán sửa lỗi chính tả văn bản

2.5.3.1. Khái niệm

Những lỗi chính tả phát sinh là do quá trình nhận dạng OCR phát sinh các lỗi chính tả cho từ nhận diện được. Bài toán này gồm ba bước chính là tiền xử lý tập văn bản đầu vào, phát hiện lỗi chính tả và sửa lỗi chính tả.

Lỗi chính tả được chia làm hai loại là non-word và real-word.

Lỗi non-word được hiểu là những từ lỗi không tìm thấy trong từ điển.

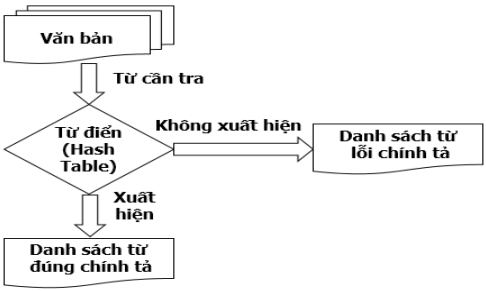
Lỗi real-word là những từ lỗi có trong từ điển nhưng không đúng trong ngữ cảnh của câu.

2.5.3.2. Phương pháp tiếp cận

Đối với vấn đề phát hiện lỗi chính tả thì thường có hai phương pháp tiếp cận chính [17].

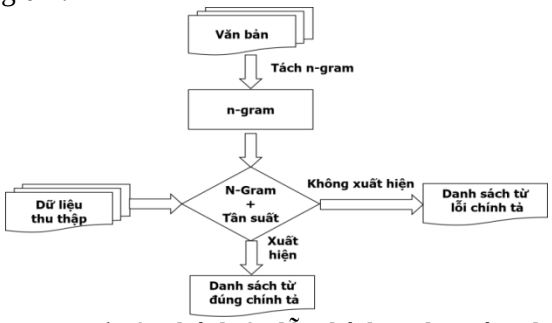
Kĩ thuật tra cứu dùng từ điển: Kĩ thuật đơn giản là kiểm tra sự hiện diện từng từ của văn bản đầu vào. Nếu từ đó có trong từ điển thì từ đó được coi là từ đúng chính tả, ngược lại thì từ đó được coi là lỗi chính tả. Kĩ thuật phổ biến nhất và nhanh chóng để phát hiện từ bị lỗi chính tả. Từ điển được xây dựng bằng cách sử dụng bảng băm để cải thiện tốc độ tra cứu.

Hình 2.9 mô tả quá trình kiểm tra lỗi chính tả bằng kĩ thuật dùng từ điển.



Hình 2.9. Kĩ thuật phát hiện lỗi chính tả dựa vào tra cứu từ điển

Kĩ thuật phân tích N-gram: N-gram là một chuỗi con gồm n từ, thường thì là hai, ba hoặc năm từ. Kĩ thuật này thực hiện bằng cách chia văn bản đầu vào thành n-gram tương ứng, đối với mỗi n-gram đầu vào, tìm kiếm trong bảng thống kê n-gram tính trước. Kết hợp thêm tần suất xuất hiện của n-gram trong bảng thống kê để kiểm tra sự tồn tại hoặc mức độ phổ biến của n-gram đầu vào nhằm xác định lỗi chính tả. Hình 2.10 mô tả quá trình kiểm tra lỗi chính tả bằng kĩ thuật sử dụng N-gram.



Hình 2.10. Kĩ thuật phát hiện lỗi chính tả dựa vào phân tích N-gram

2.6. Bài toán đánh chỉ mục và tìm kiếm

2.6.1. Khái niệm

Lập chỉ mục tài liệu là công việc sắp xếp tài liệu nhằm đáp ứng nhanh chóng yêu cầu tìm kiếm thông tin của người sử

dụng. Quá trình lập chỉ mục được hiểu là giai đoạn phân tích tập văn bản đã xử lý và thu được để xác định các chỉ mục biểu diễn nội dung của tệp văn bản này.

2.6.2. Phương pháp tiếp cận

Có nhiều công cụ để thực hiện lập chỉ mục cho tài liệu như Apache Sorl, Lucence, Sphinx. Nhưng đối với bài toán đánh chỉ mục tài liệu tác giả sử dụng công cụ Elasticsearch.

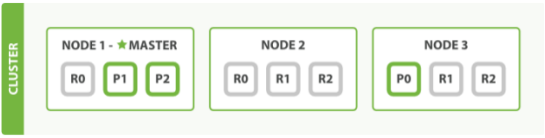
17 systems in ranking, April 2016

Rank Apr 2016	Rank Mar 2016	Rank Apr 2015	DBMS	Database Model	Score Apr 2016	Score Mar 2016	Score Apr 2015
1.	1.	2.	Elasticsearch	Search engine	82.58	+2.41	+17.92
2.	2.	1.	Solr	Search engine	66.02	-3.35	-15.98
3.	3.	3.	Splunk	Search engine	42.35	-1.38	+4.32
4.	4.	4.	MarkLogic	Multi-model	9.12	-0.25	-1.09
5.	5.	5.	Sphinx	Search engine	8.48	-0.24	-1.10
6.	6.	7.	Google Search Appliance	Search engine	3.57	-0.21	-0.12
7.	7.	8.	Amazon CloudSearch	Search engine	2.36	-0.03	+0.58
8.	8.	9.	Microsoft Azure Search	Search engine	1.25	-0.03	+0.53
9.	9.	10.	Xapian	Search engine	0.48	-0.02	-0.07
10.	10.	11.	Indica	Search engine	0.32	-0.02	+0.03
11.	12.	13.	Crate.IO	Multi-model	0.16	+0.02	+0.01
12.	11.	14.	SearchBlox	Search engine	0.16	-0.01	+0.07
13.	13.	16.	Srch²	Search engine	0.05	+0.02	+0.05
14.	14.	15.	DBSight	Search engine	0.02	-0.00	-0.03
15.	15.	16.	Exorbyte	Search engine	0.01	+0.01	+0.01
16.	15.	12.	Compass	Search engine	0.00	±0.00	-0.21
16.	15.		searchxml	Multi-model	0.00	±0.00	

Hình 2.11. Thứ hạng của 17 công cụ tìm kiếm. Nguồn http://db-engines.com

2.6.3. Kiến trúc của Elasticsearch

- Cluster
- Node
- Index
- Type
- Document
- Shard & Replicas



Hình 2.12. Kiến trúc cluster-node-shard của Elasticsearch

2.7. Kết luận

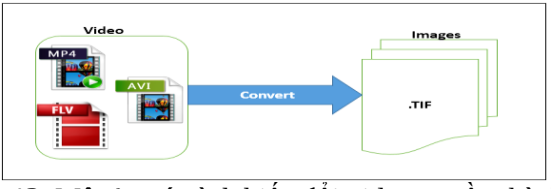
Kết thúc chương này, tác giả đã trình bày khái quát các bài toán cần giải quyết trong nội dung luận văn này. Các phương pháp tiếp cận để giải quyết vấn đề. Tiếp theo, chương ba tác giả xin trình bày chi tiết về các giải pháp kĩ thuật tiến hành của tác giả để thực hiện các bài toán đã nêu trong chương hai.

CHƯƠNG 3: KĨ THUẬT ĐỂ GIẢI QUYẾT CÁC BÀI TOÁN TRONG KHUÔN KHỔ LUẬN VĂN

3.1. Bài toán phân đoạn video thành định dạnh ảnh

3.1.1. Phát biểu bài toán

Hình 3.1 mô tả quá trình biến đổi video bài giảng thành tập ảnh.



Hình 3.13. Mô tả quá trình biến đổi video nguồn thành dạng ảnh

3.1.2. Giải pháp thực hiện

Sau khi cài đặt phần mềm Ffmpeg, sử dụng dòng lệnh "*ffmpeg -i lecture001.mp4 -r 1 %d.tif*" trong đó:

- i là video đầu vào với đường dẫn của tệp tin video. Trong ví dụ này video được định dạng là .mp4 với tên tệp tin là lecture001.

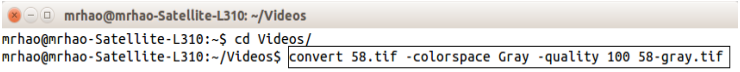
- r là số khung hình trên giây.

- %d.tif là định dạng tên tệp tin hình ảnh để lưu với tên là số nguyên và định dạng là .tif. Ví dụ 1.tif, 2.tif, 3.tif…

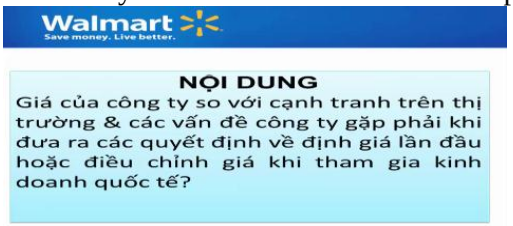
- Sử dụng số FPS là 1 (một khung hình một giây).

3.2. Bài toán trích xuất văn bản

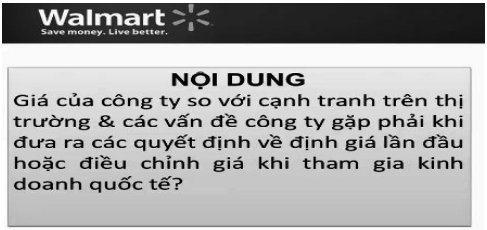
3.2.1. Bài toán nhận dạng kí tự quang học bằng công cụ Tesseract-OCR



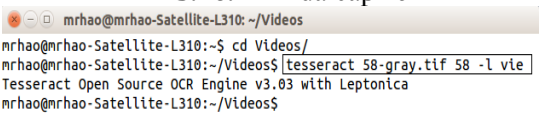
Hình 3.14. Chuyển đổi ảnh màu thành ảnh đa cấp xám



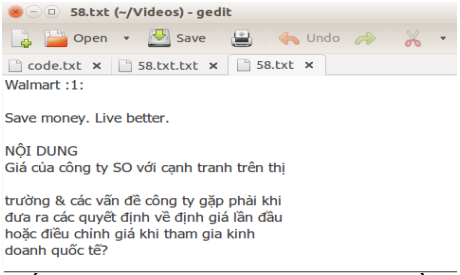
Hình 3.15. Ảnh màu



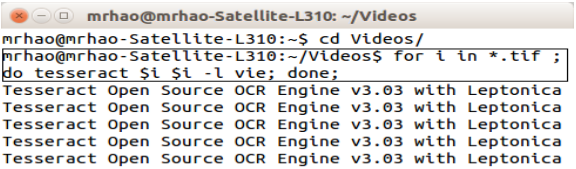
Hình 3.16. Ảnh đa cấp xám



Hình 3.17. Quá trình OCR ảnh trong hình 3.4 bằng Tesseract-OCR



Hình 3.18. Kết quả sau khi hoàn thành OCR bằng Tesseract-OCR



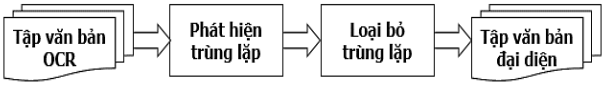
Hình 3.19. Thực hiện OCR tất cả ảnh trong thư mục bằng Tesseract-OCR

3.2.2. Bài toán xử lý trùng lặp văn bản bằng kĩ thuật Shingling

3.2.2.1. Phát biểu bài toán

Mục tiêu của quá trình này sẽ là phát hiện và loại bỏ những tệp văn bản có nội dung gần trùng nhau (các tệp được

trích xuất từ một slide). Quá trình này trải qua hai bước được trình bày trong hình.



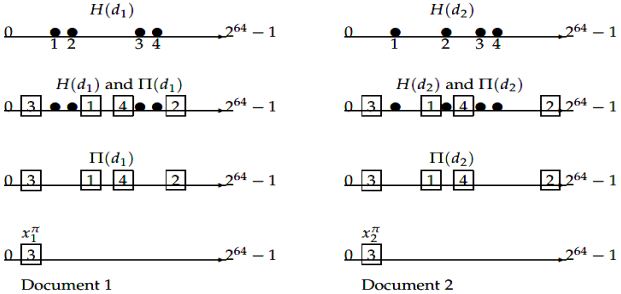
Hình 3.20. Quá trình xử lý trùng lặp văn bản

3.2.2.2. Giải thuật Shingling

Gọi tập $S(d_j)$ là tập shingles của tài liệu d_j. Sự tương đồng của hai tài liệu được đo bằng cách sử dụng hệ số Jaccard giữa các vector shingles. Giả sử với hai tập d_1 và d_2 thì hệ số Jaccard được tính theo công thức hình 3.9.

$$J(S(d_1), S(d_2)) = \frac{|S(d_1) \cap S(d_2)|}{|S(d_1) \cup S(d_2)|}$$

Hình 3.21. Hệ số Jaccard của tài liệu d_1 và d_2



Hình 3.22[4]. Bốn quá trình tính toán shingle của hai tài liệu.

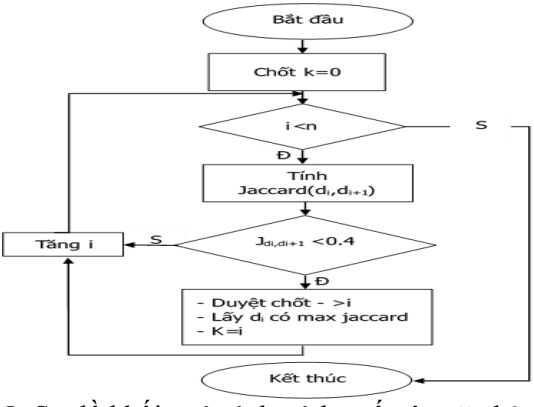
3.2.2.3. Kĩ thuật tiến hành

Dựa trên các cơ sở của phương pháp shingling, tác giả đã xác định và kết luận được hai tệp văn bản bất kỳ có phải là gần trùng lặp nhau hay không, căn cứ vào một giá trị ngưỡng của độ đo Jaccard trong hình 3.13. Bài toán tiếp theo trong nội dung này là xác định được tệp các văn bản đại diện cho video bài giảng. Bài toán được mô tả dưới dạng mã giả như sau:

Đầu vào: Cho tập D là tập tất cả văn bản được trích xuất OCR từ video, giá trị d_1, d_2,… d_n là các văn bản được thuộc tập D.

Đầu ra: Tập D’ là tập văn bản đại diện cho tập D.

Giải thuật

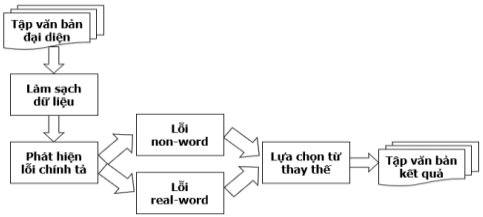


Hình 3.23. Sơ đồ khối quá trình trích xuất tập văn bản đại diện

3.2.3. Bài toán sửa lỗi chính tả văn bản tiếng Việt

3.2.3.1. Phát biểu bài toán

Hình 3.12 mô tả các bước để thực hiện phát hiện và sửa lỗi chính tả văn bản.



Hình 3.24. Quá trình phát hiện và sửa lỗi chính tả văn bản

3.2.3.2. Làm sạch dữ liệu trước khi sửa lỗi chính tả

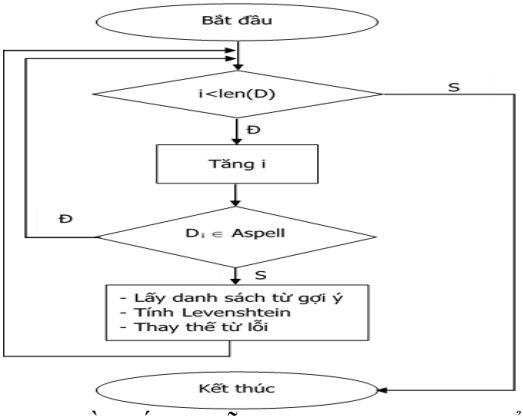
- Bước 1: Loại bỏ các kí tự khoảng trắng thừa ở đầu, giữa, và cuối câu. Ví dụ "bài giảng " sẽ được thay bằng "bài giảng".

- Bước 2: Bỏ qua các chuỗi là địa chỉ email, địa chỉ website.

- Bước 3: Loại bỏ các kí tự đặc biệt, các dấu chấm, kí tự số, ngày tháng…

3.2.3.3. Kĩ thuật sửa lỗi chính tả dạng non-word

Trong luận văn này tác giả sẽ sử dụng công cụ mã nguồn mở Aspell để cài đặt chương trình sửa lỗi chính tả đối với dạng lỗi non-word.



Hình 3.25. Sơ đồ khối sửa lỗi chính tả sử dụng từ điển Aspell

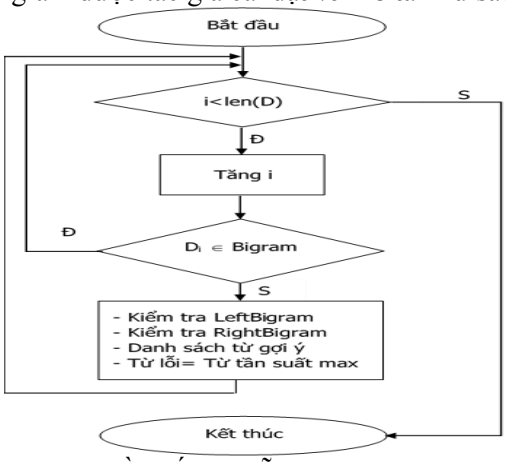
3.2.3.4. Kĩ thuật sửa lỗi chính tả dạng real-word

Dưới đây là mô tả về kĩ thuật kiểm tra và sửa lỗi chính tả dùng bigram.

Bảng 3.1. Kết quả Bigram tập dữ liệu

	Kích thước tệp tin trước khi tách Bigram	**Số Bigram tách được**	**Kích thước sau khi tách Bigram**
Bigram	66 MB	4.836.571	82 MB

Thuật toán phát hiện và sửa lỗi chính tả văn bản dựa vào kĩ thuật N-gram được tác giả cài đặt và mô tả như sau:

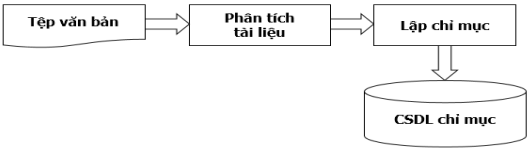


Hình 3.26. Sơ đồ khối sửa lỗi chính tả sử dụng Bigram

3.3. Bài toán đánh chỉ mục và tìm kiếm

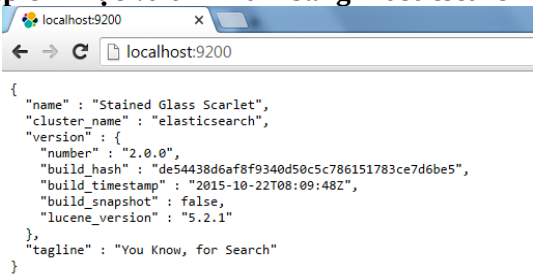
3.3.1. Phát biểu bài toán

Hình 3.15 mô tả các bước để lập chỉ mục tài liệu.

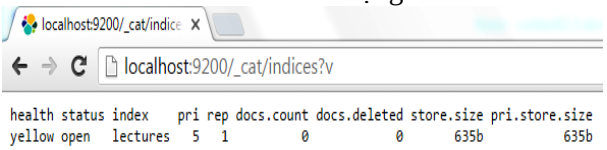


Hình 3.27. Mô tả quá trình lập chỉ mục tài liệu

3.3.2. Lập chỉ mục và tìm kiếm bằng Elasticsearch



Hình 3.28. Kiểm tra khởi động Elasticsearch



Hình 3.29. Danh sách các chỉ mục hiện có. Tên chỉ mục là lectures, số tài liệu docs.count hiện tại có giá trị bằng 0 (do chưa tạo tài liệu cho chỉ mục này).

```
curl -XPUT 'localhost:9200/lectures/external/1?pretty' -d '
{
  "url": "uploads/video001.mp4",
  "content":"ngày nay khi công nghệ thông tin..."
}'
```

Hình 3.30. Tạo type và document cho chỉ mục.

```
curl -XPOST 'localhost:9200/lectures/external/?pretty' -d '
{
  "url": "uploads/video001.mp4",
  "content":"ngày nay khi công nghệ thông tin..."
}'
```

Hình 3.31. Tạo type và document bằng lệnh POST. Id của document được Elasticsearch gán tự động.

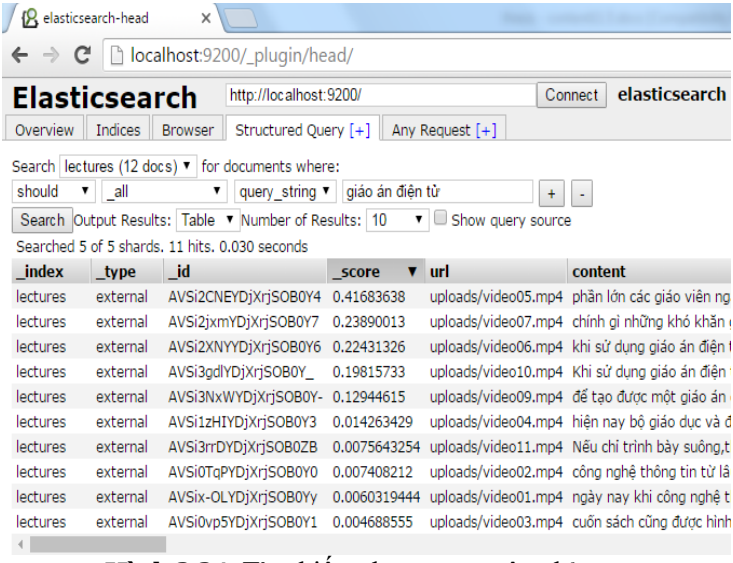
```
curl -XPUT 'localhost:9200/lectures/external/1?pretty' -d '
{
  "url": "uploads/video001.mp4",
  "content":"nội dung thay đổi..."
}'
```

Hình 3.32. Cập nhật lại document cho chỉ mục với id đã tồn tại.

Hoặc có thể sử dụng lệnh UPDATE trực tiếp được mô tả trong hình 3.21.

```
curl -XPOST 'localhost:9200/lectures/external/1/_update?pretty' -d '
{
  "doc":{
        "url": "uploads/video001.mp4",
        "content": "nội dung thay đổi..."
        }
}'
```

Hình 3.33. Thực hiện cập nhật lại document bằng câu lệnh UPDATE



Hình 3.34. Tìm kiếm document trên chỉ mục

CHƯƠNG 4: KẾT QUẢ THỰC NGHIỆM, ĐÁNH GIÁ VÀ KẾT LUẬN

4.1. Công cụ, môi trường thực nghiệm

Để phục vụ cho quá trình thực nghiệm, tác giả sử dụng cấu hình phần cứng và các công cụ phần mềm thể hiện trong hai bảng 4.1 và bảng 4.2 như sau:

Bảng 4.1. Thông số phần cứng

STT	Thành phần	Thông số kĩ thuật
1	CPU	Intel ® Pentium ® Dual core T3200 2.00GHz
2	RAM	DDR II - 3GB
3	Hệ điều hành	Ubuntu 14.04 LTS
4	Bộ nhớ ngoài	150 GB

Bảng 4.2. Danh sách công cụ phần mềm

STT	Tên công cụ	Chức năng	Nguồn tải
1	Sublime Text 3	Trình soạn thảo và bẫy lỗi chương trình.	https://www.sublimetext.com
2	PHP 5.0	Ngôn ngữ lập trình dùng thực nghiệm.	http://php.net/downloads.php
3	FFMpeg	Công cụ xử lý video.	https://ffmpeg.org/download.html
4	Imagemagick	Công cụ chuyển đổi ảnh màu thành ảnh đa cấp xám.	http://www.imagemagick.org/script/binary-releases.php
5	Tesseract -OCR	Công cụ nhận dạng kí tự quang học.	https://github.com/tesseract-ocr
6	Aspell	Công cụ kiểm tra lỗi chính tả.	http://aspell.net/
7	Pspell	Thư viện lập trình sửa lỗi chính tả trên nguôn ngữ PHP.	http://php.net/manual/en/intro.pspell.php
8	Vietnamese Dictionary	Từ điển từ vựng của Tiếng Việt.	https://github.com/1ec5/hunspell-vi/tree/master/dictionaries
9	Teleport Pro	Công cụ hỗ trợ tải dữ liệu trên mạng.	http://www.tenmax.com/teleport/pro/download.htm
10	Elasticsearch	Công cụ hỗ trợ đánh chỉ mục và tìm kiếm tài liệu.	https://www.elastic.co/

4.2. Kết quả thực nghiệm, đánh giá

Bảng 4.3. Kết quả thực hiện trích xuất khung hình từ video

STT	Định dạng	Kích thước (MB)	Thời gian (phút:giây)	Số khung hình thu được	Kích thước (MB)
1	mp4	23,8	6:22	382	404,6
2	mp4	48,1	6:38	398	450,7
3	mp4	32,1	3:07	187	174,8
4	mp4	137,6	28:27	1707	1740,8
5	mp4	19,6	2:35	155	139,4

Bảng 4.4. Kết quả thực hiện Tesseract-OCR đối với tập khung hình thu được

STT	Số lượng	Kích thước tập kết quả (KB)	Độ chính xác (%)	Độ hồi tưởng (%)	Độ F1 (%)
1	382	136,3	71,2	81,8	76,13
2	398	100,5	71,1	82,0	76,16
3	187	33,7	76,4	67,0	71,39
4	1707	529,1	66,4	76,2	70,96
5	155	45,0	77,5	66,3	71,46
Trung bình			**72,52**	**74,66**	**73,22**

Bảng 4.5. Kết quả thực hiện NDD với kĩ thuật Shingling

STT	Tập đầu vào	Số văn bản đại diện thu được	Số slide thực tế	Số văn bản đại diện đúng	Độ chính xác (%)	Độ hồi tưởng (%)	Độ F1 (%)
1	382	14	22	12	85,7	54,5	66,63
2	398	24	25	22	91,6	88,0	89,76
3	187	42	35	34	80,1	97,1	87,78
4	1707	14	18	13	92,8	72,2	81,21
5	155	21	24	18	85,7	75,0	79,99
Trung bình					**87,18**	**77,36**	**81,07**

Bảng 4.6. Kết quả quá trình phát hiện lỗi chính tả dùng Aspell kết hợp Bi-gram

STT	Tập đầu vào (số từ)	Tổng số lỗi thực tế	Số lỗi phát hiện được	Số lỗi phát hiện đúng	Độ chính xác (%)	Độ hồi tưởng (%)	Độ F1 (%)
1	946	77	71	66	92,9	85,7	89,15
2	1365	121	112	96	85,7	79,3	82,38
3	2482	43	33	18	54,54	41,8	47,33
4	786	96	91	85	93,4	88,54	90,91
5	1520	31	26	22	84,6	70,9	77,15
Trung bình					**82,23**	**73,25**	**77,38**

Bảng 4.7. Kết quả quá trình sửa lỗi chính tả

STT	Số lỗi phát hiện	Số lỗi sữa	Số lỗi sửa đúng	Độ chính xác (%)	Độ hồi tưởng (%)	Độ F1 (%)
1	71	69	49	71,0	69,0	69,99
2	112	102	62	65,8	55,4	57,97
3	33	16	9	56,3	27,3	36,77
4	91	84	43	51,2	50,5	49,17
5	26	28	18	64,3	69,2	66,66
Trung bình				**60,72**	**53,64**	**56,11**

Như đã trình bày ở mục 3.4 về khó khăn khi sửa lỗi chính tả Tiếng Việt. Vì vậy trong luận văn này, tác giả đã cố gắng để nhằm cải thiện chất lượng của quá trình sửa lỗi. Độ chính xác trung bình xấp xỉ khoảng 60,72%.

4.3. Kết luận

4.3.1. Kết quả đạt được

Trong luận văn này, tác giả hướng tới mục đích là tìm hiểu và nghiên cứu phương pháp để xây dựng một hệ thống tra cứu video dựa trên nội dung. Video tác giả quan tâm là các video bài giảng dạng silde. Nội dung của truy vấn sẽ là các từ hoặc các cụm từ có liên quan đến nội dung văn bản bên trong các video bài giảng.

Qua bốn chương, luận văn đã trình bày về các khái niệm liên quan đến công cụ tìm kiếm. Các phương pháp tiếp cận, kĩ thuật áp dụng để giải quyết các bài toán về xây dựng công cụ tìm kiếm video. Ứng dụng các phương pháp, kĩ thuật để thực nghiệm xây dựng một hệ thống tìm kiếm video bài giảng dựa trên nội dung.

Các đóng góp chính của luận văn:

- Hệ thống lại kiến thức, khái niệm liên quan và kiến trúc của công cụ tìm kiếm.
- Trình bày mô hình các bài toán cần xử lý trong quá trình xây dựng công cụ tìm kiếm video.
- Phân tích các phương pháp tiếp cận để giải quyết các bài toán và lựa chọn kĩ thuật để thực nghiệm.
- Xây dựng thử nghiệm ứng dụng tìm kiếm video bài giảng dạng slide dựa trên nội dung.

4.3.2. Định hướng phát triển

Với những kết quả đạt được trong luận văn này, tác giả hy vọng trong tương lai sẽ:

- Thử nghiệm với dữ liệu đa dạng hơn và lớn hơn. Thu thập và xử lý được với nhiều định dạng video.
- Nghiên cứu các phương pháp, kĩ thuật để nâng cao chất lượng chương trình sửa lỗi chính tả Tiếng Việt.
- Cải tiến và nghiên cứu để nâng cao chất lượng, giảm thời gian xử lý video đầu vào.

TÀI LIỆU THAM KHẢO

1. Andrei Z. Broder. (2000), "Identifying and Filtering Near-Duplicate Documents", *11th Annual Symposium on Combinatorial Pattern Matching* ,Springer-Verlag London, pp.1-10.

2. Bassma S. Alsulami. (2012), "Near Duplicate Document Detection Survey", *International Journal of Computer Science & Communication Networks*, pp. 147-151.

3. Chirag Patel, Atul Patel, Dharmendra Patel. (2012), "Optical Character Recognition by Open Source OCR Tool Tesseract: A Case Study", *International Journal of Computer Applications*, Volume 55 –No.10, pp. 50-56.

4. Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze. (2009), *Introduction to Information Retrieval*, Cambridge University Press, Cambridge University.

5. David C. Gibbon. (2012), *Introduction to Video Search Engines*, Springer Verlag Berlin Heidelberg, Spinger.

6. Gurmeet Singh Manku, Arvind Jain, Anish Das Sarma. (2007), "Detecting Near Duplicates for Web Crawling", *16th International Conference on World Wide Web*, pp. 141-150.

7. Haojin Yang, Maria Siebert, Patrick Lühne, Harald Sack, Christoph Meinel. (2011), "Automatic Lecture Video Indexing Using Video OCR Technology", *2011 IEEE International Symposium on*, pp. 111 – 116.

8. Haojin Yang. (2011), "Lecture Video Indexing and Analysis Using Video OCR Technology", *7th International Conference IEEE Dijon France*, pp. 54-61.

9. Hannaneh Hajishirzi, Wen-tau Yih, Aleksander Kolcz. (2010), "Adaptive Near-Duplicate Detection via Similarity Learning", *ACM SIGIR conference on Research and development in information retrieval*, pp. 419-426.

10. Nguyen Thi Xuan Huong, Tran-Thai Dang, The-Tung Nguyen, Anh-Cuong Le. (2015), "Using Large N-gram for Vietnamese Spell Checking", *Advances in Intelligent Systems and Computing*, pp. 617-627.

11. Kukich, Karen. (1992), "Techniques for Automatically Correcting Words in Text", *24th ACM Computing Surveys*, pp. 377–439.
12. Kurt Hornik, Duncan Murdoch. (2011), "Watch Your Spelling", *The R Journal Vol. 3*, pp. 22-28.
13. Kyle Williams, C. Lee Giles. (2013), "Near Duplicate Detection in an Academic Digital Library" , *2013 ACM Symposium on Document Engineering*, pp. 91-94.
14. Martin Røst Halvorsen. (2007), *Content-based lecture video indexing*, Master's Thesis, Department of Computer Science and Media Technology Gjøvik University College.
15. Martin Potthast, Benno Stein. (2008), "New Issues in Near-duplicate Detection", *31th Conf. of the German Classification Society*, pp. 601-609.
16. Pratip Samanta, Bidyut B. Chaudhuri. (2013), "A simple real-word error detection and correction using local word bigram and trigram", *Association for Computational Linguistics and Chinese Language Processing*, pp. 211-220.
17. Ritika Mishra, Navjot Kaur. (2013), "A Survey of Spelling Error Detection and Correction Techniques", *International Journal of Computer Trends and Technology*, pp. 372-374.
18. Radu Gheorghe, Matthew Lee Hinman, Roy Russo. (2016), *Elasticsearch in Action*, Manning Publications Co, Shelter Island.
19. Smith, R. (2007), *An Overview of the Tesseract OCR Engine*, In proceedings of Document analysis and Recognition. IEEE Ninth International Conference.
20. Suzan Verberne. (2002), *Context-sensitive spellchecking based on word trigram probabilities*, Master thesis Taal, Spraak & Informatica University of Nijmegen.
21. Youssef Bassil, Mohammad Alwani. (2012), "Context-sensitive Spelling Correction Using Google Web 1T 5-Gram Information", *Computer and Information Science*, Vol. 5, No. 3, May 2012, pp. 37-48.