BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG

-------o0o-------

ISO 9001:2008

ĐỒ ÁN TỐT NGHIỆP

NGÀNH CÔNG NGHỆ THÔNG TIN

HẢI PHÒNG - 2016

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG

-------o0o-------

ÁP DỤNG KỸ THUẬT PHÂN TÍCH NGỮ NGHĨA TIỀM ẨN TRONG ĐỐI SÁNH VĂN BẢN

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ LIÊN THÔNG

Ngành Công nghệ thông tin

HẢI PHÒNG – 2016

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG

-------o0o-------

ÁP DỤNG KỸ THUẬT PHÂN TÍCH NGỮ NGHĨA TIỀM ẨN TRONG ĐỐI SÁNH VĂN BẢN

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ LIÊN THÔNG

Ngành Công nghệ thông tin

Sinh viên thực hiện: Nguyễn Minh Thành

Mã số sinh viên: 1513101003

Giáo viên hướng dẫn: Nguyễn Trịnh Đông

HẢI PHÒNG – 2016

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh phúc
-------o0o-------

NHIỆM VỤ TỐT NGHIỆP

SINH VIÊN: NGUYỄN MINH THÀNH

MÃ SỐ: 1513101003

LỚP: CTL901

NGÀNH: CÔNG NGHỆ THÔNG TIN

TÊN ĐỀ TÀI:

ÁP DỤNG KỸ THUẬT PHÂN TÍCH NGỮ NGHĨA TIỀM ẨN TRONG ĐỐI SÁNH VĂN BẢN

NHIỆM VỤ ĐỀ TÀI

NỘI DUNG VÀ YÊU CẦU CẦN GIẢI QUYẾT TRONG NHIỆM VỤ ĐỀ TÀI TỐT NGHIỆP

A. NỘI DUNG.

- Tìm hiểu các phương pháp phân cụm.
- Tìm hiểu một số phương pháp tạo các luật cơ bản và các giải thuật liên quan.
- Đề ra phương pháp xâp dựng hệ thống.
- Thử nghiệm với các công cụ để giải quyết bài toán.

B. KẾT QUẢ CẦN ĐẠT ĐƯỢC:

a. Lý thuyết

- Nắm được các phương pháp phân cụm dữ liệu.
- Nắm được phương pháp luật hóa các chi thước và các giải thuật liên quan.
- Áp dụng kiến thức trong xâp dựng phần mềm thử nghiệm.

b. Thực nghiệm (chương trình)

- Thử nghiệm với các chương trình mã nguồn mở.

C. CÁC YÊU CẦU VỚI SINH VIÊN

- Có tinh thần trách nhiệm đối với công việc.
- Biết ít nhất một ngôn ngữ lập trình.
- Khả năng đọc và tổng hợp dữ liệu.

CÁN BỘ HƯỚNG DẪN ĐỀ TÀI TỐT NGHIỆP

Người hướng dẫn thứ nhất :

Họ và tên : Nguyễn Trịnh Đông

Học hàm, học vị : Thạc Sĩ

Cơ quan công tác : Trường Đại Học Dân Lập Hải Phòng

Nội dung hướng dẫn :

- Tìm hiểu các phương pháp phân cụm.
- Tìm hiểu một số phương pháp tạo các luật cơ bản và các giải thuật liên quan.
- Đề ra phương pháp xâp dựng hệ thống.
- Thử nghiệm với các công cụ để gải quyết bài toán.

Người hướng dẫn thứ hai :

Họ và tên : …….....

Học hàm, học vị :……..……

Cơ quan công tác :

…….....

Nội dung hướng dẫn : …….....

Đề tài tốt nghiệp được giao ngày 03 tháng 10 năm 2016

Yêu cầu hoàn thành trước ngày 30 tháng 12 năm 2016

Đã nhận nhiệm vụ: Đ.T.T.N	Đã nhận nhiệm vụ : Đ.T.T.N
Sinh viên	*Cán bộ hướng dẫn Đ.T.T.N*

Hải Phòng, ngày.....tháng.....năm 2016

HIỆU TRƯỞNG

GS.TS.NGƯT Trần Hữu Nghị

PHẦN NHẬN XÉT TÓM TẮT CỦA CÁN BỘ HƯỚNG DẪN

1. Tinh thần thái độ của sinh viên trong quá trình làm đồ án

.....

.....

.....

.....

.....

2. Đánh giá chất lượng của đề tài (so với nội dung yêu cầu đã đề ra trong nhiệm vụ đồ án)

.....

.....

.....

.....

.....

3. Cho điểm của cán bộ hướng dẫn (điểm ghi bằng số và chữ):

.....

Ngày..... tháng..... năm 2016

CÁN BỘ HƯỚNG DẪN CHÍNH

(Ký, ghi rõ họ tên)

PHẦN NHẬN XÉT ĐÁNH GIÁ CỦA CÁN BỘ CHẤM PHẢN BIỆN ĐỀ TÀI TỐT NGHIỆP

1. Đánh giá chất lượng đề tài tốt nghiệp (về các mặt như cơ sở lý luận, thuyết minh chương trình, giá trị thực tế, ...)

.....

.....

.....

.....

.....

.....

.....

.....

.....

2**.** Cho điểm của cán bộ phản biện (*điểm ghi bằng số ,chữ*):

.....

.....

.....

Ngày..... tháng..... năm 2016

CÁN BỘ CHẤM PHẢN BIỆN

(Ký, ghi rõ họ tên)

MỤC LỤC

MỤC LỤC 9
DANH MỤC HÌNH....12
DANH MỤC BẢNG....13
DANH MỤC TỪ VIẾT TẮT....14
LỜI NÓI ĐẦU....15
Chương 1: Giới thiệu đối sánh văn bản....17
1.1 Giới thiệu....17
1.2 Phân tách tài liệu thành các từ khóa (Filter)....17
1.2.1 Các nghiên cứu về cấu trúc của các nhà nghiên cứu Việt Nam 17
1.2.2 Tách tài liệu thành các từ khóa.....22
1.2.3 Giải pháp tách từ Tiếng Anh....23
1.2.4 Giải pháp cho Tiếng Việt....23
1.3 Các hệ thống gợi ý (recommender systems - RS)....25
1.3.1 Các khái niệm về Recommender System....25
1.3.2 Xử lý tài liệu tiếng Việt....26
1.3.3 Xử lý tài liệu theo ngữ nghĩa....27
Chương 2: Phương pháp phân tích ngữ nghĩa tiềm ẩn....30
2.1 Tiền xử lý....30
Phân nhóm văn bản....30
Phương pháp phân nhóm phân cấp....30
Phương pháp phân nhóm không phân cấp....30
2.2 Tách từ....30
2.2.1 Tiếng trong tiếng Việt....31
2.2.2 Từ trong tiếng Việt....31
2.2.3 Từ dừng và từ gốc....31
***2.3* Các phương pháp tách từ phổ biến**....32
2.3.1 Phương pháp Maximum Matching....32
2.3.2 TF-IDF Term Frequency – Inverse Document Frequency..33
2.3.3 Phương pháp Transformation – based Learning (TBL)....34
2.3.4 Mô hình tách từ bằng WFST và mạng Neural....34
2.3.5 Phương pháp tách từ tiếng Việt dựa trên thống kê từ Internet và thuật giải di truyền....35
2.4 Phương pháp phân tích ngữ nghĩa tiềm ẩn....36

2.4.1 Giới thiệu 36
2.4.2 Khái niệm 37
2.4.3 Cách thức hoạt động 38
2.5 Đối sánh văn bản 46
2.5.1 Độ tương đồng 46
2.5.2 Độ tương đồng văn bản trong Tiếng Việt 49
2.6 Tính độ tương đồng cho toàn bộ văn bản 52
Chương 3: Bài toán áp dụng 53
3.1 Giới thiệu ngôn ngữ R 53
3.1.1 Giới thiệu R 53
3.1.2 Các lện trong gói phân tích ngữ nghĩa tiềm ẩn trong R 54
3.2 Cài đặt và chạy chương trình 57
3.2.1 Cài đặt 57
3.2.2 Chạy chương trình ..... **Error! Bookmark not defined.**
KẾT LUẬN 65
TÀI LIỆU THAM KHẢO 66

LỜI CẢM ƠN

Em xin chân thành cảm ơn thầy giáo Ths. Nguyễn Trịnh Đông đã tận tình chỉ bảo, định hướng, góp ý cho em trong suốt thời gian qua. Để em có thể hoàn thành đồ án tốt nghiệp. Cũng như em xin chân thành cảm ơn các thầy, cô trong Khoa công nghệ thông tin trường ĐHDL Hải Phòng giúp đỡ em. Em cũng xin gửi lời cảm ơn tới gia đình, bạn bè, những người luôn động viên, quan tâm và giúp đỡ em trong suốt thời gian em làm đồ án.

Trong đồ án này không thể tránh được sẽ có nhiều thiếu sót. Em rất mong nhận được những lời nhận xét, góp ý từ các thầy, cô và các bạn.

Hải phòng, ngày 24 tháng 12 năm 2016

Sinh viên

Nguyễn Minh Thành

DANH MỤC HÌNH

Hình 1: Sơ đồ cấu trúc từ của Nguyễn Tài Cẩn 18

Hình 2: Hình minh họa tập tách văn bản 23

Hình 3: Giải thuật tách từ từ câu 24

Hình 4: Cấu trúc giải thuật LSA 29

Hình 5: Sử dụng các khái niệm làm đại diện cho tài liệu 37

Hình 6: Sơ đồ SVD của ma trận thuật ngữ tài liệu 42

Hình 7: Sơ đồ SVD được giảm lược của ma trận thuật ngữ - tài liệu 44

Hình 8: Cửa sổ làm việc của Rstudio 58

Hình 9: Cài đặt thư viện lsa 60

Hình 10: Các thư viện lsa 61

Hình 11: File lsa_plot.R 62

Hình 12: Lệnh return 62

Hình 13: Các thuật ngữ-tài liệu 63

Hình 14: Ma trận thuật ngữ tài liệu 63

Hình 15: Ma trận giảm chiều 64

Hình 16: Ma trận tài liệu-tài liệu 64

Hình 17: Biểu đồ tương quan thuật ngữ-tài liệu 65

DANH MỤC BẢNG

Bảng 1: Bảng một số ví dụ về cấu trúc lưu trữ từ điển.....25

Bảng 2: Số lần xuất hiện của thuật ngữ trong mỗi tài liệu 41

DANH MỤC TỪ VIẾT TẮT

LSA	Latent Semantic Analysis	Phần tích ngữ nghĩa tiềm ẩn
SVD	Singular Value Decompotision	Tách giá trị số ít hoặc tách giá trị riêng
TF-IDF	Term Frequency – Inverse Document Frequency	Giải pháp đánh trọng số kết hợp tính chất quan trọng của một từ trong tài liệu chứa nó (TF-tần suất xuất hiện của từ trong tài liệu) với tính phân biệt của từ trong tập tài liệu nguồn (IDF-nghịch đảo tần suất tài liệu).

LỜI NÓI ĐẦU

Trong thời đại công nghệ số hiện nay, các nguồn tài liệu là vô cùng phong phú. Việc tìm kiếm một tài liệu trở nên đơn giản hơn bao giờ hết, rất nhiều tài liệu, thông tin tri thức mới mẻ đang phát triển từng giờ, giúp chúng ta thu nhận tri thức mọi lúc và ở bất cứ đâu. Lợi ích là không thể bàn cãi. Nhưng như hai mặt của một vấn đề, ở quy mô rộng lớn bao la như vậy, các thư viện điện tử ngày càng nhiều, một tài liệu có thể được phát hành trên internet nhiều lần từ nhiều nguồn, theo nhiều định dạng khác nhau, trong nhiều thư viện điện tử khác nhau, trên những trang web khác nhau. Tìm kiếm là dễ dàng nhưng trích trọn ra được thông tin chính xác và hữu ích lại là vấn đề không hề dễ dàng.

Làm thế nào để có thể nhận biết được đâu sẽ là tài liệu đúng, đâu sẽ là tài liệu đi sao chép, góp nhặt từ các tài liệu khác mà tính chính xác không hề được kiểm chứng. Chủ đề này đã được nghiên cứu từ gần 15 năm qua. Hiện tại, đã có một số giải pháp khá hữu hiệu cho vấn đề này và một vài công cụ phần mềm cho phép phát hiện, tìm kiếm một tài liệu hoặc một tập hợp các tài liệu nguồn phù hợp với yêu cầu. Tập hợp các tài liệu nguồn có thể là đóng- tức là các tài liệu tập hợp trước trong một thư viện điện tử hoặc là mở, chẳng hạn như các tập tài liệu văn bản trên internet.

Đã có một số nghiên cứu đề xuất các phương pháp khác nhau để xác định xem một đoạn văn bản của một tài liệu có nằm trong có nằm trong một tài liệu khác hay không. Các phương pháp này chủ yếu dựa trên tìm kiếm và so khớp chuỗi. Tuy nhiên, các phương pháp so khớp chuỗi chỉ hiệu quả nếu từ hoặc tập từ, đoạn văn là "nguyên văn". Do vậy, một yêu cầu được đặt ra là làm thế nào để phát hiện việc được các tài liệu có liên quan khi các tài liệu đó có sửa đổi như thay thế một số từ bằng từ đồng nghĩa hoặc đổi thứ tự từ, câu trong văn bản. Từ đó, một ý tưởng được đưa ra, liệu rằng tìm kiếm so sánh văn bản dựa trên nội dung, ý nghĩa sẽ cho hiệu quả cao hơn so với các phương pháp tìm kiếm và so khớp chuỗi.

Xuất phát từ những lý do trên, em chọn đề tài: **"Áp dụng phương pháp phân tích ngữ nghĩa tiềm ẩn trong đối sánh văn bản".**

Mục tiêu của đề tài là đối sánh văn bản áp dụng phương pháp phân tích ngữ nghĩa tiềm ẩn.

Đề tài được trình bày như sau:

Giới thiệu: Phát biểu bài toán

Chương 1: *Trình bày các khái niệm và kiến thức cơ bản trong lĩnh vực đối sánh văn bản.*

Chương 2: *Chương này tập trung trình bày các phương pháp phân tích xử lý và đối sánh văn bản.*

Chương 3: *Chương này trình bày phần thực nghiệm chương trình dựa trên phân tích ngữ nghĩa tiềm ẩn đã trình bày tại chương 2*

Kết luận

Tài liệu tham khảo

CHƯƠNG 1: GIỚI THIỆU ĐỐI SÁNH VĂN BẢN

1.1 Giới thiệu

Trong các loại dữ liệu thì dữ liệu văn bản là dạng phổ biến nhất. Ngày nay, với sự phát triển mạnh mẽ của Internet, dữ liệu văn bản đã trở nên phong phú về nội dung và tăng nhanh về số lượng. Chỉ bằng một vài thao tác đơn giản, tại bất kì đâu, tại bất kì thời điểm nào, ta cũng có thể nhận về một khối lượng khổng lồ các trang web và các tài liệu điện tử liên quan đến nội dung tìm kiếm. Chính sự dễ dàng này cũng mang đến cho chúng ta rất nhiều khó khăn trong việc chắt lọc ra các thông tin được coi là mới, là riêng, là hữu ích giữa các tài liệu ấy. Và việc đầu tiên ta phải làm đó là biến đổi các dạng văn bản ngôn ngữ tự nhiên thành dạng dữ liệu có cấu trúc, hay nói cách khác là xử lý dữ liệu đầu vào.

1.2 Phân tách tài liệu thành các từ khóa (Filter)

Các tài nguyên là các tài liệu được thể hiện dưới dạng văn bản như một cuốn sách, tạp chí, hay một bài báo, bài diễn văn điện tử nào đó. Với những tài liệu tiếng Anh, một từ thường có một âm tiết, ta có thể dễ dàng xác định một từ dựa vào dấu cách (space) hoặc dấu câu. Việc phân tách văn bản tiếng Anh thành các từ khóa không khó khăn. Với những văn bản tiếng Việt, mỗi từ có thể có một, hai hoặc nhiều hơn số lượng âm tiết. Việc phân tách thành từ khóa đối với văn bản tiếng Việt phải dựa trên từ điển và các thuật toán đọc từ khóa sao cho đúng nghĩa nhất của câu.

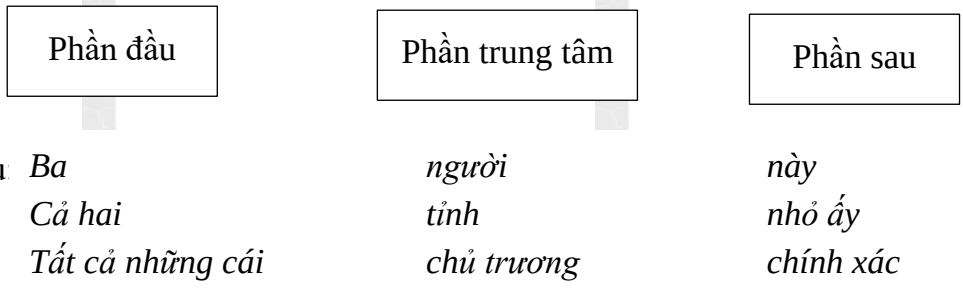
Thí dụ: “Học sinh học sinh học” thì hệ thống sẽ tách thành Học sinh/học/sinh học. Sau đó, loại bỏ các từ dừng (Stopword – Những từ mang ý nghĩa cảm thán, đại từ…như anh, bạn, do đó…), những từ không mang nhiều ý nghĩa về nội dung.

1.2.1 Các nghiên cứu về cấu trúc của các nhà nghiên cứu Việt Nam

Các quan điểm trong nghiên cứu về ngữ pháp tiếng Việt, chúng ta có thể thấy rằng chưa có một định nghĩa chuẩn thống nhất về cách gọi của từ loại cũng như cấu trúc các ngữ của tiếng Việt. Trong đồ án này, người viết luận văn sẽ chủ trương bám sát theo quan điểm được nhiều tác giả đã thống nhất, quan điểm này được đánh giá là khá phù

hợp với ngữ pháp tiếng Việt hiện tại. Đồng thời, trong quá trình xây dựng đồ án, tác giả cũng tiến hành so sánh và bổ sung thêm những phần lý thuyết thuộc hai quan điểm của Nguyễn Tài Cẩn và Diệp Quan Ban.

Nguyễn Tài Cẩn (1975) [Đặng Thị Hưởng] cho rằng cụm danh từ (danh ngữ) gồm có ba phần: phần đầu, phần trung tâm và phần cuối như sơ đồ sau:



Hình 1: Sơ đồ cấu trúc từ của Nguyễn Tài Cẩn

Trong thực tế danh ngữ còn có thể xuất hiện cả dưới dạng những dạng chỉ có hai phần: phần đầu+phần trung tâm, phần trung tâm + phần sau hoặc phần đầu + phần sau.

Phần trung tâm của danh ngữ không phải chỉ có một từ trung tâm mà bao gồm cả bộ phận trung tâm ghép gồm hai trung tâm T1 và T2, với hai vị trí T1, T2 bộ phận trung tâm có thể xuất hiện dưới 3 biến dạng :

o Có đầy đủ : T1T2, ví dụ : con chim (này)

o Dạng thiếu T1 : -T2, ví dụ : - chim (này)

o Dạng thiếu T2: T1-, ví dụ : con – (này)

- Phần đầu của danh ngữ có tất cả 3 loại thành tố phụ (3 loại định tố) :

o Định tố “cái”, ví dụ : cái cậu học sinh ấy

o Định tố chỉ số lượng, ví dụ : mấy cái cậu học sinh ấy

o Định tố chỉ ý nghĩa toàn bộ, ví dụ : tất cả mấy cái cậu học sinh ấy

- Phần cuối của danh ngữ, có thể có 2 loại định tố có tổ chức hoàn toàn khác nhau:

o Loại định tố chỉ gồm một từ, ví dụ : một quyển sách quý

o Loại định tố do một mệnh đề đảm nhiệm, ví dụ : cuốn sách tôi vừa mua hôm qua

Trong tiếng Việt có thể dùng những từ loại sau đây để làm định tố cuối :

+ Danh từ, ví dụ : vườn cau

+ Tính từ, ví dụ : ghế dài, một cái ghế rất tốt

+ Động từ, ví dụ : bàn học

+ Từ chỉ trỏ, ví dụ : sáng nay, người ấy

+ Từ chỉ vị trí, ví dụ : nhà trong, cổng trước

+ Từ chỉ con số : giường một, ngày 27

Hồ Lê (1992) cho rằng vị trí của số lượng từ, đại từ chỉ định và sự kiện từ xoay xung quanh danh từ theo mô hình sau :

Số lượng từ	D1	D2	Sự kiện từ (trừ đại từ chỉ định)	Đại từ chỉ định

Trong đó:

D1: gồm những danh từ như : con, cái, …; ông, bà…; loại, thứ, hạng, …; phía, bên, nơi, chốn, buổi, hôm, ngày, giờ, khi, lúc...

D2: gồm những danh từ còn lại.

Ví dụ:

- Con mèo đen lớn rồi
- Cô y tá
- Phía ngoài sân

Nguyễn Kim Thản (1997) cho rằng việc nghiên cứu cụm danh từ chính là việc nghiên cứu từ tổ danh từ, loại từ chiếm ưu thế tuyệt đối trong nhóm danh từ. Danh từ có thể ghép với danh từ, thời vị từ, số từ, động từ, tính từ, đại từ và một số từ phụ khác như : khi, lúc, hồi, dạo, thuở, khoảng, độ, bữa, buổi, đằng, phía, phương, nơi, bên, ngả, lối, hạng, cái, loại, cỡ, khổ, bậc, ngạnh...

-Từ tố danh từ + danh từ (N):

o Từ tố N1 N2, ví dụ: cân gạo, bó rau, hòm sách, chùm cau, tóc mây, tiền nghìn, sông Hồng, huyện Gia Lộc, nước Lào…

o Từ tố N1 z N2, ví dụ: quê của mẹ, nhà bằng gạch, kế hoạch về kinh tế, sách cho thiếu nhi, nhãn ở Hưng yên…

oTừ tố N1 (z) N2, ví dụ: quê mẹ, nhà gạch, kế hoạch kinh tế, sách thiếu nhi, nhãn Hưng yên…. (z : là giới từ).

-Từ tố danh từ + thời vị từ (E):

oTừ tổ N E, ví dụ: Ngoài nhà ngoài có cái giường mình nằm

oTừ tổ N (z) E, ví dụ: ý định (của) trên như thế nào ?

-Từ tố danh từ + số từ (F): từ chỉ số lượng bao giờ cũng đặt trước danh từ (FN), ví dụ : hai cái bàn. Từ chỉ thứ tự bao giờ cũng đặt sau danh từ (NF), ví dụ : bàn số hai, quyển thứ năm...

o Từ tố danh từ + động từ, ví dụ : cá sống, nước sôi, gió lùa, kế hoạch làm việc….

o Từ tố danh từ + tính từ, ví dụ: quả táo vàng, cái áo trắng…

o Từ tố danh từ + đại từ chỉ định, ví dụ: con mèo ấy, cái xe này, …

Diệp Quang Ban (1999) đưa ra cấu tạo chung của cụm danh từ có ba phần: phần trung tâm, phần phụ trước, phần phụ sau. Phần trung tâm thường là một danh từ hoặc một ngữ danh từ. Trong phần phụ trước người ta đã xác định được ba vị trí khác nhau sắp xếp theo một trật tự nhất định. Ở phần phụ sau thường nhận được hai vị trí có trật tự ổn định. Phần phụ trước cụm danh từ chuyên dùng chỉ mặt số lượng của sự vật nêu ở trung tâm, phần phụ sau chủ yếu dùng chỉ mặt chất lượng của sự vật nêu ở trung tâm.

Phần phụ trước			Phần phụ trung tâm	Phần phụ sau	
-3	-2	-1	0	1	2

Ví dụ:

tất cả	những	con	mèo	đen	ấy
-3	-2	-1	0	1	2

Vị trí 0 là vị trí của danh từ chính

Vị trí -1 là vị trí của từ chỉ xuất cái

Vị trí -2 là vị trí của từ chỉ số lượng, ví dụ: một, hai,…; vài, ba, dăm, dăm ba...; mỗi, từng, mọi…; những, các, một…; mấy

Vị trí -3 là vị trí của từ chỉ tổng lượng, ví dụ: hết thảy, tất cả, cả…

Vị trí 1 là vị trí của từ nêu đặc trưng miêu tả có thể gặp nhiều loại từ khác nhau như: danh từ, động từ, tính từ, số từ, đại từ và thời vị từ.

Ví dụ:

phòng tạp chí, phòng đọc, phòng hẹp, phòng chúng tôi...

Vị trí 2 là vị trí của từ chỉ định, ví dụ: cái máy này, quả táo kia...

- Tóm tắt đặc trưng của cấu trúc ngữ pháp tiếng Việt, Anh

Các đặc điểm trong câu của tiếng Việt:

- Câu được cấu tạo đa thành phần, có câu đơn, câu ghép, câu tối giản.
- Các câu được phân tách bằng các dấu chấm câu.
- Câu hoàn chỉnh có hoặc không sử dụng các trạng từ, từ cảm thán (các stopword, sẽ được loại bỏ khi phân tách để lọc thông tin).
- Câu được hình thành từ các từ, hoặc các câu đơn.
- Mỗi câu mang một ý nghĩa thông tin hoàn chỉnh.

Như đã trình bày ở trên, ngày nay, các tài liệu viết tiếng Việt đang chuyển sang khai thác trên font chữ chung, tuân theo chuẩn unicode. Điều này có lợi rất nhiều cho việc đọc chính xác các tài liệu tiếng việt của các chương trình đọc.

Với các tài liệu tiếng Anh, các từ được phân cách nhau bởi dấu cách. Việc xử lý phân tách từ từ các văn bản tiếng Anh tương đối dễ dàng.

Trong tiếng Việt không thể phân tách được thành những từ riêng bởi dấu cách. Vì từ có thể gồm một, hai hoặc nhiều hơn số lượng âm tiết (số lượng từ ghép). Vì thế, việc tách từ để chính xác đòi hỏi giải thuật tách từ tốt.

1.2.2 Tách tài liệu thành các từ khóa.

Mô hình này áp dụng cho những lần gợi ý đầu tiên, cho khi người dùng nhập từ khóa.

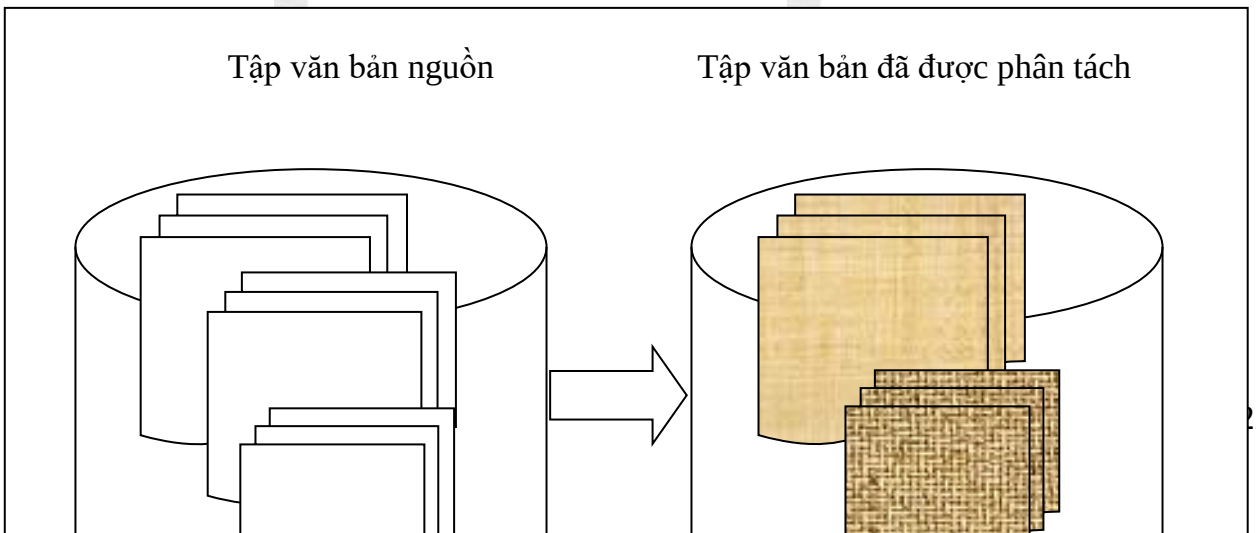
Giải thuật tách từ khóa:

Input: tập tài nguyên là sách, tạp chí, trang thông tin (html) ở các định dạng tài liệu.

Output: Tập các từ khóa với rank tương ứng.

Quá trình tìm kiếm sẽ ưu tiên với những khóa được lọc ra ở tập khóa có mức độ ưu tiên được đánh giá bằng rank tương ứng của chúng.

Giải thuật này được thiết kế để chạy offline trong phiên bản đầu tiên. Quá trình cập nhật tài liệu được người quản trị nhập. Chi phí thời gian cho modul này là khá lớn, một cuốn sách điện tử tầm 200 trang tương ứng là 3 phút cho hệ thống đọc và tách thành từ khóa.



Hình 2: Hình minh họa tập tách văn bản

1.2.3 Giải pháp tách từ Tiếng Anh

Với các tài liệu tiếng Anh, từ của tiếng Anh được phân cách với nhau bằng dấu cách. Điều đó có nghĩa là chúng ta có thể tách từ tiếng Anh bằng dấu phân tách là dấu cách. Việc xử lý các stopword của tiếng anh thì chúng tôi cũng có một file gồm hơn 300 từ stopword của tiếng Anh để phục vụ cho việc loại bỏ stopword khỏi văn bản tách để tách ra những từ quan trọng, tránh những từ mang ý nghĩa chung, hay chỉ là cảm thán.

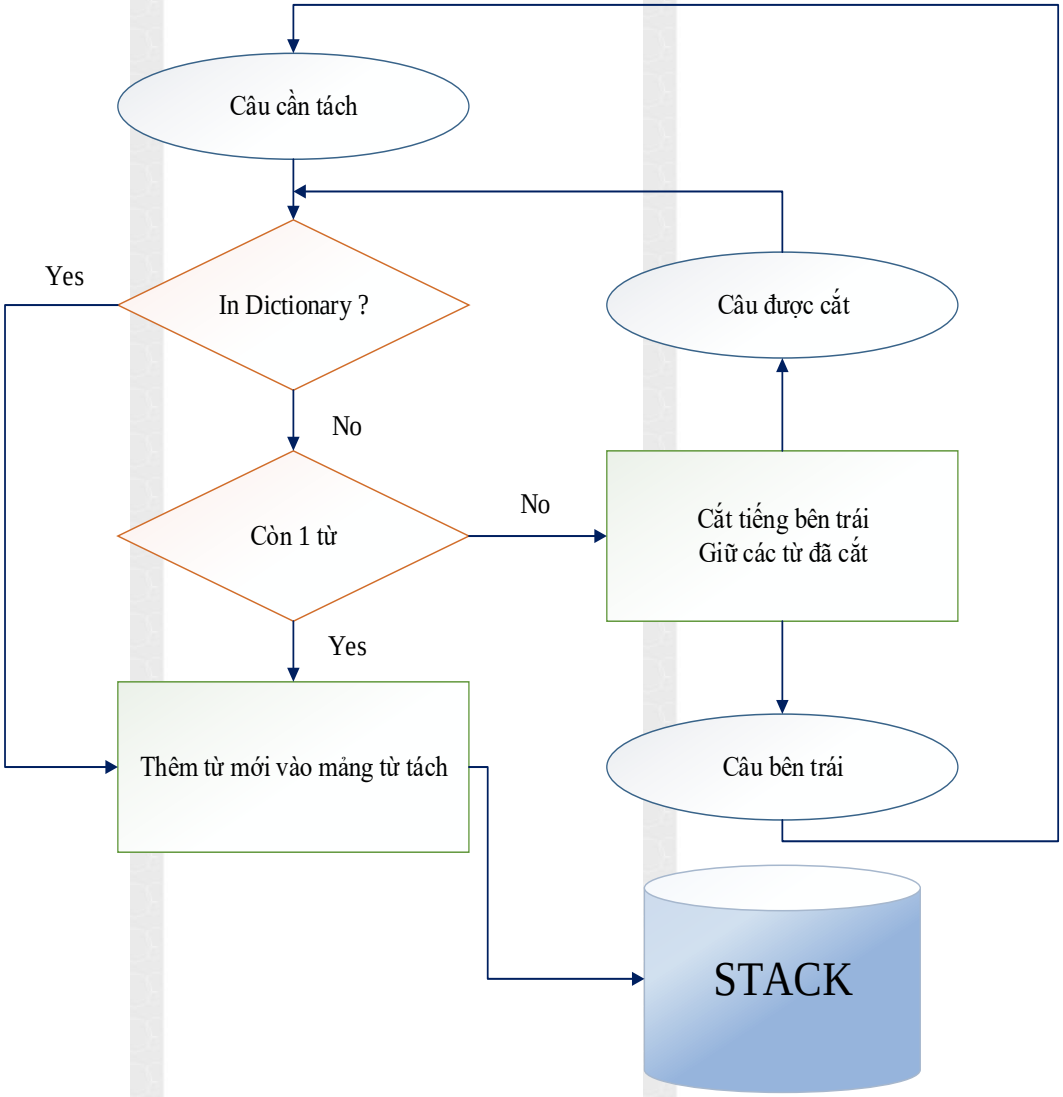
1.2.4 Giải pháp cho Tiếng Việt

1.2.4.1 Các giải pháp đã có

Hiện có rất nhiều chương trình hỗ trợ việc phân tích cú pháp, tách từ, gán nhãn từ tiếng Việt cùng với các giải thuật thuật toán khác nhau. Những đồ án được tham khảo là chương trình Code_correct (chính xác loại văn bản dựa vào tập huấn luyện), chương trình VNSegment của tác giả Phương Thái (chương trình khá hoàn chỉnh trong việc tách từ và phân loại từ tiếng Việt. Chỉ có điều viết trên nền Java và không được cung cấp sourcode và file thư viện), chương trình VNTokenizer (đề cập tới một số nghiên cứu hữu ích cho việc tác từ).

1.2.4.2 Giải pháp sử dụng và nhận xét

Giải pháp được sử dụng được mô tả bằng giải thuật phía dưới. Dựa trên một bộ từ điển tiếng Việt tương đối đầy đủ gồm hơn 99.000 từ và cụm từ.



Hình 3: Giải thuật tách từ từ câu

Tài liệu được tách thành các câu chuẩn (hoàn chỉnh). Giải thuật tách các từ khóa từ các câu đó. Các từ trong câu sẽ được kiểm tra xem có tồn tại trong từ điển không bằng các so sánh nó với các từ trong từ điển. Nếu nó "giống" từ trong từ điển, thì tách từ đó và kiểm tra tiếp. Giải thuật trên giảm thiểu tối đa tình huống nhập nhằng. Tuy nhiên trong trường hợp câu này thì giải thuật tách sai: Học sinh học sinh học →Học|sinh học|sinh học. Nhưng như tình huống này (thuộc địa bàn) thì lại giải quyết tốt (thuộc|địa bàn).

Chi phí về thời gian cho giải thuật trên là rất lớn ở bước kiểm tra từ tách ra có trong từ điển hay không. Từ điển với hơn 99.000 từ được load vào bộ nhớ trong dưới với kiểu dữ liệu được tổ chức theo 2 kiểu như sau:

- Một là mảng các string. Mỗi phần tử là từ hay cụm từ của từ điển. Mảng này được sắp xếp theo thứ tự tăng dần của mã ASCII (Trong C#, kiểu dữ liệu mảng).
- Tổ chức thành một arraylist mà mỗi phần tử của nó là mảng các string. Mảng các string thuộc 1 phần tử nó giống nhau về âm tiết đầu tiên của từ. Ví dụ

Âm tiết đầu	Từ, cụm từ						
ả	ả đào	ả đầu	ả hoàn				
ái	Ái	Ái ân	Ái chà	Ái quốc	Ái nam ái nữ	Ái thần	Ái tình
an	An cư lạc nghiệp	An dưỡng	An giấc				
Anh	Anh hàng	Anh hùng	Anh hùng chủ nghĩa	Anh kiệt	Anh linh	Anh vũ	
Ăn	Ăn	Ăn cắp	Ăn trộm				

Bảng 1: Bảng một số ví dụ về cấu trúc lưu trữ từ điển

Khi đọc một từ, ta tách âm tiết đầu tiên của nó và kiểm tra trước với các phần tử đầu tiên của mảng, sau đó, kiểm tra tiếp với các phần tử trong mảng của nó. Khi đó, nó sẽ tăng tốc độ tìm kiếm lên đáng kể.

1.3 Các hệ thống gợi ý (recommender systems - RS)

1.3.1 Các khái niệm về Recommender System

Recommender Systems (RS) là một hế thống lọc thông tin đặc biệt, hệ thống cho phép lọc thông tin dựa trên sự quan tâm của người dùng và nội dung của văn bản. có hai

kỹ thuật chính được sử dụng để xây dựng một hệ thống RS hiện nay.Một là kỹ thuật hướng nội dung – **Content based approach**, kỹ thuật này cho phép hệ thống đưa ra những gợi ý phù hợp nhất với những tiêu chuẩn đã được xác định. Hệ thống phải nắm được tất cả những đặc điểm chính được thể hiện trong đối tượng được quan tâm (Theo từ khoá của người dùng) và sắp xếp chúng theo những tiêu chuẩn tương ứng.

Kỹ thuật hướng lọc cộng tác **– Collborative filtering** CF lại làm việc dựa trên cơ chế tìm kiếm những sự đồng nhất của một cá nhân với cộng đồng mà họ tham gia để xác định gợi ý hơn là dựa trên việc xác thực nội dung của sự quan tâm. Tức là CF dựa trên sự công tác giữa một nhóm cá thể có chung một quan điểm hay một sự lựa chọn nào đó để đưa ra nhưng gợi ý cho người tìm kiếm.

Trong môt số hệ thống cũng thường áp dụng cả hai kỹ thuật này cách này gọi là hệ thống lai (Hybrid).

Về căn bản mô hình bài toán được xây dựng như sau:

Gọi C là số thành viên của hệ thống, c_i là từng người dùng cụ thể.

Gọi S là toàn bộ không gian đối tượng có thể đưa ra và s_i là một đối tượng cụ thể.

Gọi u là giá trị phù hợp của đối tượng s với người dùng c.

Vậy bài toán là sự ánh xạ u: CxS → R. Trong đó R chính là tập hợp các đối tượng được đưa ra giới thiệu. Tập R sẽ được sắp xếp theo thứ tự giảm dần của u. Công việc chính của giải thuật đơn giản chỉ là đi tìm giá trị hàm u=f(c, s) sao cho u (max) là giá trị được ưa thích nhất.

Dễ thấy độ phức tạp của bài toán là rất cao bởi không gian S là rất lớn. Ví dụ như cách ứng dụng về gợi ý sách, số lượng sách có thể lên tới hàng triệu quyển. Hoặc hệ thống gợi ý về CDs…Đồng thời không gian C cũng rất lớn nếu như mạng phát triển mang tính toàn cầu như Ebay, Google, Facebook có thể lên đến hàng tỉ thành viên.

Nhìn chúng các hệ thống truyền thống đều có mặt hạn chế nhưng chúng ta biết cách kết hợp các kỹ thuật phù hợp với từng hệ thống riêng biết. Trong nhiều hệ thống thực tế chúng ta đã thấy được khả năng mở rồng đầy tiềm năng đặc biệt trong hệ thống tìm kiếm.

1.3.2 Xử lý tài liệu tiếng Việt

Tiếng nói và chữ viết là hai yếu tố cơ bản nhất của bất kỳ ngôn ngữ nào. Trong sự phát triển của công nghệ thông tin (CNTT) ở Việt Nam, một số việc liên quan đến "tiếng Việt" đã được làm và ít nhiều có kết quả ban đầu:

(a) Trước hết là các bộ gõ chữ Việt và thành công của việc đưa được bộ mã chữ Việt vào bảng mã Unicode, cũng như việc chọn Unicode cho bộ mã chuẩn tiếng Việt (nhân đây cũng xin nói thêm, do chưa ý thức về chuẩn, rất nhiều cán bộ CNTT, nhiều cơ quan nhà nước vẫn chưa chịu đổi thói quen cũ để dùng bộ mã chuẩn Unicode, một việc rất quan trọng của xử lý tiếng Việt). Bảo tồn chữ Nôm trên máy tính cũng là một việc đầy nỗ lực và nhiều ý nghĩa được nhiều người theo đuổi lâu nay, cần được nhà nước tiếp tục ủng hộ lâu dài (http://nomfoundation.org).

(b) Tiếp theo có thể kể đến các chương trình nhận dạng chữ Việt in (OCR: optical character recognition), như hệ VnDOCR của Viện Công nghệ Thông tin, Viện Khoa học và Công nghệ Việt Nam. Các chương trình nhận dạng chữ in nhằm chuyển các tài liệu in trên giấy thành các tài liệu điện tử (dưới dạng các tệp văn bản trên máy tính).

(c) Các phần mềm hỗ trợ việc sử dụng tiếng nước ngoài, tiêu biểu là các từ điển song ngữ trên máy tính, thí dụ như các từ điển điện tử của Lạc Việt đã được dùng rộng rãi trên máy tính để tra cứu từ Anh-Việt, Việt-Anh. Điều ta cần phân biệt là các từ điển điện tử này dành cho con người sử dụng, khác với từ điển điện tử dành cho máy tính sử dụng trong xử lý ngôn ngữ tự nhiên (sẽ được đề cập ở phần sau).

(d) Các nỗ lực trong việc làm các phần mềm dịch Anh-Việt, Việt-Anh, chẳng hạn như các hệ dịch EVTRAN và VETRAN.

(e) Một loại việc nữa là Việt hóa các phần mềm mà gần đây tiêu biểu là kết quả Việt hóa Windows và Microsoft Office của Microsoft. Việc này có thể xem như việc "dịch" các thông báo tiếng Anh cố định trong các phần mềm thành các thông báo tiếng Việt.

1.3.3 Xử lý tài liệu theo ngữ nghĩa

1.3.3.1 Đặt vấn đề

Trong xử lý ngôn ngữ tự nhiên, bài toán gán nhãn ngữ nghĩa hay còn gọi là "khử sự nhập nhằng ngữ nghĩa của từ" là bài toán khó khăn nhất và cũng là bài toán trọng tâm mà đến nay trên thế giới vẫn chưa giải quyết ổn thỏa. Hiện nay, có rất nhiều mô hình với nhiều hướng tiếp cận khác nhau, chủ yếu là:

- *Dựa trên trí tuệ nhân tạo (AI-based):*

Đây là cách tiếp cận sớm nhất (1960) với những lý thuyết rất hay về mạng ngữ nghĩa, khung ngữ nghĩa và các ý niệm nguyên thủy và các quan hện như IS-A, PART-OF… Tuy nhiên, do hầu hết các tri thức về ngữ nghĩa trong cách tiếp cận này đều được xây dựng bằng tay, vì vậy các mô hình đều dừng lại ở mức độ biểu diễn trên một vài câu. Vấn đề khó khăn của cách tiếp cận này là thiếu tri thức.

- *Dựa trên Cơ sở tri thức (Knowledge-Based):*

Vào đầu thập niên 80, người ta đã chuyển sang hướng khai thác tri thức tự động từ các từ điển điện tử (MRD: Machine – Readable Dictionaries) như các từ điển đồng nghĩa… để có thể phần nào khắc phục hạn chế của hướng tiếp cận dựa trên trí tuệ nhân tạo (thiếu tri thức). Kết quả của hướng tiếp cận này là sự ra đợi của mạng WordNet – Một cơ sở tri thức khổng lồ về ngữ nghĩa theo hướng liệt kê nét nghĩa. Tuy nhiên, các cơ sở tri thức nói trên cũng chỉ là những nguồn thông tin để hệ thống chọn nghĩa tham khảo, còn chọn thông tin nào trong số những thông tin có liên quan đó thì ta phải tự xác định trong từng trường hợp cụ thể.

- *Dựa trên ngữ liệu (Corpus – Based)*

Hướng tiếp cận này sẽ rút ra các quy luật xử lý ngữ nghĩa (bằng thống kê, bằng máy học…) từ những kho dữ liệu lớn đã có sẵn và áp dụng các luật này cho trường hợp mới. Thực ra, cách tiếp cận này đã được nêu ra rất sớm (1940), nhưng do nguồn dữ liệu hạn chế, thiết bị xử lý chưa hiện đại nên không có điều kiện phát triển. Mãi tới thập niên 1990, khi mà công nghệ phát triển mạnh, đã có thể vượt qua được khó khăn của mình, cách tiếp cận này được hồi sinh và phát triển mạnh tới ngày nay.

Hiện nay, cách tiếp cận dựa trên ngữ liệu kết hợp với tri thức có sẵn là hướng tiếp cận đang được nhiều nhà ngôn ngữ học – máy tính quan tâm.

1.3.3.2 Phân tích ngữ nghĩa tiềm ẩn (Latent Semantic Analystic-LSA)

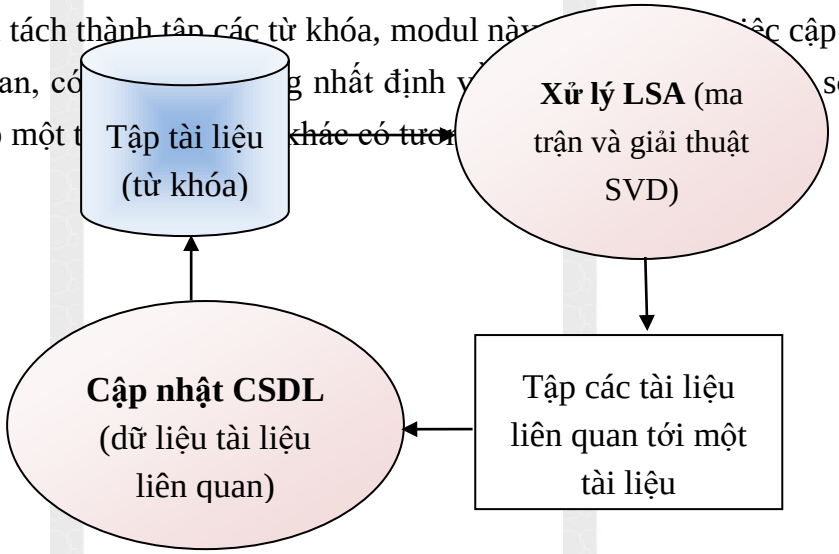
Kỹ thuật LSA là những lý thuyết và phương thức cho việc trích rút và thể hiện ngữ cảnh sử dụng ngữ nghĩa của từ dựa trên việc tính toán thông kê. Kỹ thuật này cho phép ứng dụng trên một kho dữ liệu văn bản lớn. Ý tưởng cơ bản của kỹ thuật là tổng hợp tất cả các văn cảnh của từ, trong đó, một từ được đưa ra đã và không chỉ định biểu lộ những tập ràng buộc lẫn nhau. Những tập ràng buộc này cho phép xác định sự tương đồng về nghĩa của những từ và tập hợp mỗi từ khác.

Tập các từ khóa của các tài liệu của bước phân tích trên được dùng làm đầu vào cho các hàng của ma trận. Theo đó, bộ từ khóa của một tài liệu được dùng làm cột, các tài liệu làm hàng, các ô của ma trận được khởi tạo là tần suất xuất hiện của từ khóa-thuật ngữ đó trong tài liệu. LSA dùng kỹ thuật phân tích giá trị riêng (SVD-Singular Value Decomposition) để giảm bớt kích thước ma trận thuật ngữ-tài liệu, không gian N-chiều sẽ được giảm bớt xuống một không gian K chiều, K<<N, không gian mới này được gọi là không gian khái niệm.

Sử dụng kết quả bước này, ta thu được tập các tài liệu có sự tương đồng về ngữ nghĩa nhất định với tài liệu xét. Là nguồn quan trọng trong việc đưa ra gợi ý những tài liệu tương tự với tài liệu người dùng đang đọc.

Mô hình tìm tư vấn dựa trên phân tích ngữ nghĩa tiềm ẩn

Sau khi tách thành tập các từ khóa, modul này [illegible] việc cập nhật tiếp các tài liệu liên quan, có [illegible] nhất định [illegible] sẽ được xác định cùng với nó một t[illegible] khác có tươ[illegible]



Hình 4: Cấu trúc giải thuật LSA

Phân tích ngữ nghĩa tiềm ẩn (LSA)

Có nhiều phương pháp khác nhau để đánh giá sự tương đồng về nội dung như phương pháp Định chỉ số ngữ nghĩa tiềm ẩn (LSI – Latent Semantic Index), phương pháp Phân tích ngữ nghĩa tiềm ẩn (LSA – Latent Sematic Analys).

Chỉ mục ngữ nghĩa tiềm ẩn (LSI) thêm một bước quan trọng cho việc xử lý chỉ mục tài liệu.Thêm vào việc ghi những từ khóa mà một tài liệu chứa. Phương pháp này khảo sát toàn bộ tập dữ liệu, để thấy những tài liệu khác chứa một số từ tương được với các từ đó. LSI được phát triển đầu tiên ở Bellcore trong cuối những năm 80. LSI xem các tài liệu có nhiều từ thông dụng là có nghĩa, và xem những tài liệu ít từ thông dụng là ít có nghĩa. Mặc dù thuật giải LSI không hiểu tí gì về nghĩa của các từ, nó nhận ra các khuân mẫu.

Khi tìm kiếm một CSDL chỉ mục LSI, công cụ tìm kiếm này xem xét những giá trị tương tự mà nó tính toán cho mỗi từ của nội dung, và trả về các tài liệu mà nó nghĩ là thích hợp nhất với câu truy vấn. Bởi vì hai tài liệu có thể rất gần nghĩa với nhau thậm chí nếu chúng không cùng chung một từ khóa đặc biệt, LSI không yêu cầu một sự phân tích lấy tương xứng để trả về các kết quả hữu dụng. Ở những vị trí mà một tìm kiếm theo từ khóa đơn giản sẽ không thực hiện được nếu không có phân tích lấy tương xứng, thì LSI sẽ thường trả về những tài liệu liên quan mà không chứa tất cả những từ khóa đó.

Phân tích ngữ nghĩa là một khâu rất quan trọng trong hệ thống gợi ý. Bước tách từ vựng đã tách tài liệu thành các từ khóa và nó đặc trưng cho tài liệu đó. Hệ thống sẽ tìm

kiếm trong kết quả trả về cho người dùng lần đầu tiên bằng việc so khớp các từ khóa được nhập với các từ khóa trong phần từ khóa của các tài liệu. Khâu xử lý về nội dung sẽ xác định các tài liệu nào giống tài liệu nào. Giống ở đây chỉ mức độ tương đồng về mặt nội dung giữa các tài liệu đem gợi ý. Có thể hai tài liệu không có bộ từ khóa giống nhau, nhưng nó có thể sẽ giống **về nội dung.**

CHƯƠNG 2: PHƯƠNG PHÁP PHÂN TÍCH NGỮ NGHĨA TIỀM ẨN

2.1 Tiền xử lý

Phân nhóm văn bản

Với bài toán đối sánh thực chất cũng chỉ là tìm kiếm những thông tin tiềm ẩn trong cơ sở dữ liệu rồi sau đó so sánh. Với những cơ sở dữ liệu lớn thì vấn đề đặt ra là các hệ thống tìm kiếm cần hiệu quả, một trong những kỹ thuật chủ yếu là phân nhóm văn bản nhằm giải quyết vấn đề trên.

Trong bài toán phân nhóm, một nhóm là một tập hợp các phần tử giống nhau hơn so với các phần tử thuộc nhóm khác. Mục tiêu là tìm ra một tập hợp các nhóm sao cho độ tương đồng giữa các phần bên trong mỗi nhóm cao và độ tương đồng giữa các phần tử khác nhau phải thấp.

Phương pháp phân nhóm phân cấp

Quá trình này thường có chi phí lớn. Có nhiều thuật toán được phát triển nhằm xây dựng cây phân cấp văn bản một cách hiệu quả. Các thuật toán này thường có chung phương pháp lặp quá trình phân tích hai cặp nhóm đã được xây dựng từ trước và hợp nhất cặp có độ tương đồng lớn nhất thành một nhóm văn bản.

Phương pháp phân nhóm không phân cấp

Các thuật toán phân nhóm dạng này hoạt động theo cách ngược lại với thuật toán phân nhóm phân cấp. Các thuật toán này luôn tăng số phần tử của từng nhóm và các nhóm mới có thể là kết quả của quá trình tách hay hợp các nhóm cũ. Các phương pháp phân nhóm không phân cấp có thể yêu cầu các văn bản không được trùng nhau ở các nhóm khác nhau hoặc có thể trùng nhau.

2.2 Tách từ

Tiếng Việt là ngôn ngữ đơn lập [**Nguyen Thi Minh Huyen, Vu Xuan Luong, Le Hong Phuong**][**Đặng Thị Hưởng**]. Đặc điểm này bao quát tiếng Việt cả về mặt ngữ

âm, ngữ nghĩa, ngữ pháp. Khác với các ngôn ngữ châu Âu, mỗi từ là một nhóm các ký tự có nghĩa được cách nhau bởi một khoảng trắng. Tiếng Việt và các ngôn ngữ đơn lập khác, thì khoảng trắng không phải là căn cứ để nhận diện từ.

2.2.1 Tiếng trong tiếng Việt

Trong tiếng Việt trước hết cần chú ý đến đơn vị xưa nay vẫn quen gọi là tiếng, về mặt ngữ nghĩa, ngữ âm, ngữ pháp, đều có giá trị quan trọng.

Sử dụng tiếng để tạo từ có hai trường hợp:

- Trường hợp một tiếng: đây là trường hợp một tiếng được dùng làm một từ, gọi là từ đơn. Tuy nhiên, không phải tiếng nào cũng tạo thành một từ.
- Trường hợp hai tiếng trở lên: đây là trường hợp hai hay nhiều tiếng kết hợp với nhau, cả khối kết hợp với nhau gắn bó tương đối chặt chẽ, mới có tư cách ngữ pháp là một từ, đây là trường hợp từ ghép hay từ phức.

2.2.2 Từ trong tiếng Việt

Có rất nhiều quan niệm về từ trong tiếng Việt, từ nhiều quan niệm về từ tiếng Việt khác nhau đó chúng ta có thể thấy đặc trưng cơ bản của "từ" là sự hoàn chỉnh về mặt nội dung, từ là đơn vị nhỏ nhất để đặt câu.

Người ta dùng "từ" kết hợp thành câu chứ không phải dùng "tiếng", do đó quá trình tách câu thành các "từ" cho kết quả tốt hơn là tách câu bằng “tiếng”.

2.2.3 Từ dừng và từ gốc

- Từ dừng: Đa số ngôn ngữ tự nhiên có những từ chức năng, những liên từ, giới từ xuất hiện với số lượng lớn trong các tài liệu và điển hình là ít được sử dụng khi ta xác định các tài liệu để so sánh. Các từ như vậy (ví dụ: a, an, the…) được gọi là từ dừng (stopword).

Các kỹ thuật thông thường không chỉ số hóa từ dừng nhưng có ý tưởng thay thế chúng bằng một đối tương thay thế để ghi nhớ sự xuất hiện chứa các từ dừng. Việc giảm bớt không gian chỉ số và cải thiện thực hiện là những lý do quan trọng để loại trừ các từ dừng. Tuy nhiên, việc này làm cho một số câu văn có thể bị loại bỏ mặc dù nó vẫn có nghĩa như “To be or not to be”. Và một điều nữa là từ nhiều nghĩa, một từ có rất nhiều nghĩa phụ thuộc vào văn cảnh hoặc cách nói. Ví dụ như “can”, đôi khi nó là một từ

dừng nhưng có những lúc nó lại là trung tâm của một câu văn, vì vậy từ "can" sẽ không nằm trong danh sách các từ dừng.

- Từ gốc: Trong tiếng Anh là stemming là phương thức hỗ trợ sự phù hợp cho một văn cảnh. Trong một số ngôn ngữ - điển hình là tiếng Anh - các phần của văn nói hay các cuộc đối thoại, thời và số lượng được chuyển từ những biến tố của từ. Ví dụ như từ "comparable" là từ từ "compare". Stemming có thể làm tăng số lượng văn bản trả lời, nhưng có thể bao gồm cả những tài liệu không thích hợp. Tính trọng số và phân loại từ

2.3 Các phương pháp tách từ phổ biến

2.3.1 Phương pháp Maximum Matching

Forward / Backward Phương pháp so khớp tối đa (MM-Maximum Matching) hay còn gọi là LRMM - Left Right Maximum Matching. Ở phương pháp này, chúng ta sẽ duyệt một câu từ trái sang phải và chọn từ có nhiều âm tiết nhất có mặt trong từ điển và cứ thực hiện lặp lại như vậy cho đến hết câu. Dạng đơn giản: phương pháp dùng để giải quyết nhập nhằng từ đơn. Giả sử chúng ta có một chuỗi ký tự C1, C2, …, Cn. Chúng ta sẽ áp dụng phương pháp từ đầu chuỗi. Đầu tiên kiểm tra xem C1 có phải là từ hay không, sau đó kiểm tra xem C1C2 có phải là từ hay không, tiếp tục thực hiện như thế cho đến khi tìm được từ dài nhất. Dạng phức tạp: quy tắc của dạng này là phân đoạn từ, thông thường người ta chọn phân đoạn ba từ có chiều dài tối đa. Thuật toán bắt đầu từ dạng đơn giản, cụ thể là nếu phát hiện ra những cách tách từ gây nhập nhằng, như ở ví dụ trên, giả sử C1 là từ và C1C2 cũng là một từ, khi đó chúng ta kiểm tra ký tự kế tiếp trong chuỗi C1, C2, ..., Cn để tìm tất cả các đoạn ba từ có bắt đầu với C1 hoặc C1C2. Ví dụ: Giả sử chúng ta có được các đoạn sau:

– C1 C2 C3 C4

– C1C2 C3C4 C5

– C1C2 C3C4 C5C6

Khi đó chuỗi dài nhất sẽ là chuỗi thứ ba. Do đó từ đầu tiên của chuỗi thứ ba (C1C2) sẽ được chọn. Thực hiện các bước cho đến khi được chuỗi từ hoàn chỉnh.

Nhận xét:

Phương pháp này thực hiện tách từ đơn giản, nhanh và chỉ cần dựa vào từ điển để thực hiện. Tuy nhiên, khuyết điểm của phương pháp này cũng chính là từ điển, nghĩa là độ chính xác khi thực hiện tách từ phụ thuộc hoàn toàn vào tính đủ tính chính xác của từ điển.

2.3.2 Phương pháp Term Frequency – Inverse Document Frequency

Term Frequency – Inverse Document Frequency (TF-IDF) là giải pháp đánh trọng số kết hợp tính chất quan trọng của một từ trong tài liệu chứa nó (TF-tần suất xuất hiện của từ trong tài liệu) với tính phân biệt của từ trong tập tài liệu nguồn (IDF-nghịch đảo tần suất tài liệu). Đây là một kỹ thuật cơ bản và thường được sử dụng kết hợp với các thuật toán khác để xử lý văn bản. Mục đích của kỹ thuật này là tính trọng số của một từ, qua đó đánh giá mức độ quan trọng của từ đó trong văn bản. Trong đó:

- **TF** được tính theo công thức:

$$\mathbf{tf}(t,d) = \frac{\mathbf{f(t,d)}}{\mathbf{max\{f(w,d) : w \in d\}}}$$

Với f(t,d): số lần xuất hiện của từ t trong văn bản d

Max{f(w,d) : w∈d}: số lần xuất hiện nhiều nhất của một từ bất kỳ trong văn bản.

- **IDF** được tính theo công thức:

$$\mathbf{idf}(t,D) = \log \frac{\mathbf{D}}{\mathbf{1 + \{d \in D : t \in d\}}}$$

Với D: tổng số văn bản trong tập D

{d∈D : t∈d}: số văn bản chứa từ nhất định, với điều kiện t xuất hiện trong văn bản d.

- Giá trị **TF-IDF**:

$$\mathbf{tfidf}(t,d,D) = tf(t,d) * idf(t,d)$$

Ví dụ minh họa phương pháp tính trọng số:

Có một văn bản gồm 100 từ, trong đó từ "máy tính" xuất hiện 10 lần thì độ phổ biến: tf("máy tính") = 10 / 100 = 0.1.

Bây giờ giả sử có 1000 tài liệu, trong đó có 200 tài liệu chứa từ "máy tính".

Lúc này ta sẽ tính được: idf("máy tính") = log(1000 / 200) = 0.699

Như vậy ta tính được: TF-IDF = tf*idf = 0.1 * 0.699 = 0.0699

2.3.3 Phương pháp Transformation – based Learning (TBL)

Phương pháp này tiếp cận dựa trên tập ngữ liệu đã đánh dấu. Theo cách tiếp cận này để cho máy tính có thể nhận biết ranh giới giữa các từ để có thể tách từ chính xác, chúng ta sẽ cho máy học các câu mẫu trong tập ngữ liệu đã được đánh dấu ranh giới giữa các từ đúng. Chúng ta thấy phương pháp rất đơn giản, vì chỉ cần cho máy học các tập câu mẫu và sau đó máy sẽ tự rút ra qui luật của ngôn ngữ và để từ đó sẽ áp dụng chính xác khi có những câu đúng theo luật mà máy đã rút ra. Và để tách từ được hoàn toàn chính xác trong mọi trường hợp thì đòi hỏi phải có một tập ngữ liệu tiếng Việt thật đầy đủ và phải được huấn luyện lâu để có thể rút ra các luật đầy đủ.

2.3.4 Mô hình tách từ bằng WFST và mạng Neural

Mô hình mạng chuyển dịch trạng thái hữu hạn có trọng số Weighted Finite State Transducer (WFST) đã được áp dụng trong tách từ từ năm 1996. Ý tưởng cơ bản là áp dụng WFST với trọng số là xác suất xuất hiện của mỗi từ trong kho ngữ liệu. Dùng WFST để duyệt qua các câu cần xét, khi đó từ có trọng số lớn nhất là từ được chọn để tách. Phương pháp này cũng đã được sử dụng trong công trình đã được công bố của tác giả Đình Điền năm 2001, tác giả đã sử dụng WFST kèm với mạng Neural để khử nhập nhằng khi tách từ, trong công trình tác giả đã xây dựng hệ thống tách từ gồm tầng WFST để tách từ và xử lý các vấn đề liên quan đến một số đặc thù riêng của ngôn ngữ tiếng Việt như từ láy, tên riêng,... và tầng mạng Neural dùng để khử nhập nhằng về ngữ nghĩa sau khi đã tách từ (nếu có).

Chi tiết về 2 tầng này như sau.

Tầng WFST: gồm có 3 bước sau.

Bước 1: Xây dựng từ điển trọng số: theo mô hình WFST, thao tác phân đoạn từ được xem như là một sự chuyển dịch trạng thái có xác suất. Chúng ta miêu tả từ điển D

là một đồ thị biến đổi trạng thái hữu hạn có trọng số. Giả sử:

• H là tập các từ chính tả tiếng Việt (còn gọi là "tiếng")

• P là từ loại của từ.

• Mỗi cung của D có thể là:

– Từ một phần tử của H tới một phần tử của H

– Các nhãn trong D biểu diễn một chi phí được ước lượng theo công thức:

Cost = -log(f/N) Trong đó: f là tần số của từ, N là kích thước tập mẫu.

Bước 2: Xây dựng các khả năng phân đoạn từ: Để giảm sự bùng nổ tổ hợp khi sinh ra dãy các từ có thể từ một dãy các tiếng trong câu, tác giả đã đề xuất phương pháp kết hợp dùng thêm từ điển để hạn chế sinh ra các bùng nổ tổ hợp, cụ thể là nếu phát hiện thấy một cách phân đoạn từ nào đó không phù hợp (không có trong từ điển, không có phải là tứ láy, không phải là danh từ riêng…) thì tác giả loại bỏ các nhánh xuất phát từ cách phân đoạn đoạn đó.

Bước 3: Lựa chọn khả năng phân đoạn từ tối ưu: Sau khi có được danh sách các cách phân đoạn từ có thể có của câu, tác giả đã chọn trường hợp phân đoạn từ có trọng số bé nhất.

Tầng mạng Neural

Mô hình được sử dụng để khử nhập nhằng khi tách từ bằng cách kết hợp so sánh với từ điển.

Nhận xét

Mô hình này đạt được độ chính xác trên 97% theo như công bố trong công trình của tác giả, bằng việc sử dụng thêm mạng Neural kết hợp với từ điển để khử các nhập nhằng có thể có khi tách ra được nhiều từ từ một câu và khi đó tầng mạng Neural sẽ loại bỏ đi các từ không phù hợp bằng cách kết hợp với từ điển. Bên cạnh đó, cũng tương tự như phương pháp TBL điểm quan trọng của mô hình này cần tập ngữ liệu học đầy đủ.

2.3.5 Phương pháp tách từ tiếng Việt dựa trên thống kê từ Internet và thuật giải di truyền

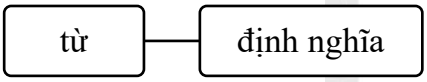
Phương pháp tách từ tiếng Việt dựa trên thống kê từ Internet và thuật giải di truyền – IGATEC (Internet and Genetics Algorithm based Text Categorization for Documents in Vietnamese) do Nguyễn Thanh Hùng đề xuất năm 2005 như một hướng tiếp cận mới trong tách từ với mục đích phân loại văn bản mà không cần dùng đến một từ điển hay

tập ngữ liệu học nào. Trong hướng tiếp cận này, tác giả kết hợp giữa thuật toán di truyền với dữ liệu thống kê được lấy từ Internet .

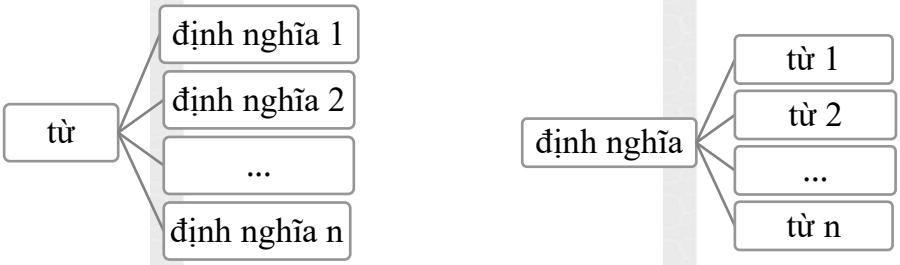
2.4 Phương pháp phân tích ngữ nghĩa tiềm ẩn

2.4.1 Giới thiệu

Nếu mỗi từ chỉ có nghĩa là một khái niệm, và mỗi khái niệm chỉ được mô tả bằng một từ, ta có thể dễ dàng tạo ra một sơ đồ thể hiện quan hệ từ - khái niệm như sau.

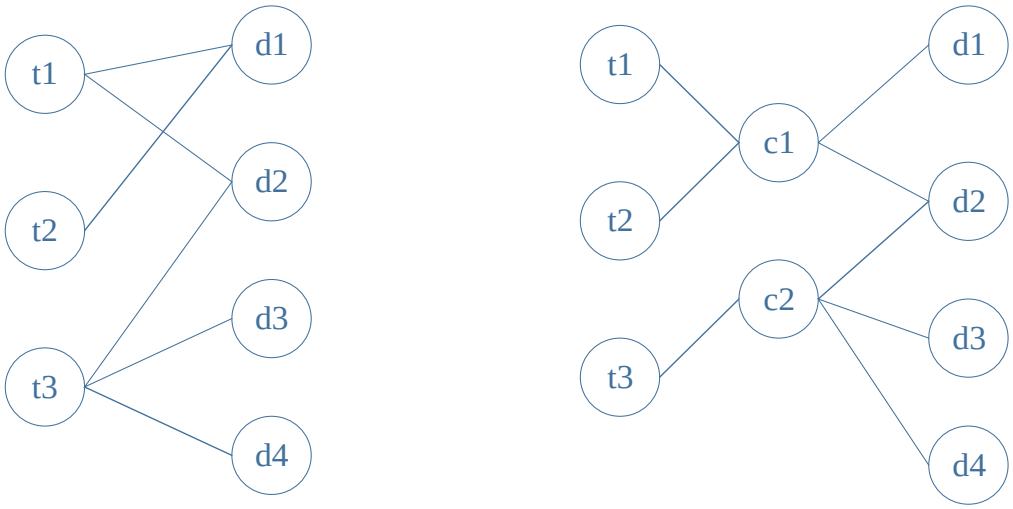


Nhưng, vấn đề này không dễ dàng như vậy bởi vì ngôn ngữ luôn luôn có những từ có nhiều nghĩa và từ đồng nghĩa.



Tất cả các sự mơ hồ nhập nhằng đó làm người đọc cũng cần 1 lúc để hiểu được các ý nghĩa mà từ hướng đến. Từ đó, một ý tưởng xuất hiện, đó là ta sẽ ánh xạ các thuật ngữ vào một không gian khái niệm và thiết lập các đối tượng tương đồng trong không gian khái niệm. Nói cách khác: ta dùng các thuật ngữ tương đồng để hình thành khái niệm làm đại diện cho tài liệu.

Sơ đồ ý tưởng sẽ là:



Hình 5: Sử dụng các khái niệm làm đại diện cho tài liệu

Xác định một tầng ở giữa thành một mối liên hệ giữa các truy vấn và tài liệu. Ta có thể thấy, các không gian khái niệm có thể có kích thước nhỏ hơn. Ví dụ, xác định truy vấn t3 với d2, d3, d4 trong tập trả lời dựa vào việc chúng có liên quan đến khái niệm c2. Đây chính là ý tưởng của LSA.

2.4.2 Khái niệm

Phân tích ngữ nghĩa tiềm ẩn (Latent Semantic Analysis-LSA) là một lý thuyết và phương pháp có thể chiết xuất, biểu diễn ngữ cảnh theo nghĩa của từ, bằng cách tính toán thống kê đối với một tập lớn văn bản (Landauer and Dumais, 1997). Ý tưởng là tập hợp tất cả các từ trong bối cảnh, từ đó đưa ra tập các ràng buộc, chủ yếu là về sự tương đồng ý nghĩa của từ và bộ từ với nhau.[**Thomas K Landauer, Peter W. Foltz, Darrell Laham, 1998**]

Trong các năm sau đó, nhiều nhà khoa học đã trình bày các báo cáo nghiên cứu, khai thác một lý thuyết mới về cảm ứng kiến thức và đại diện, cung cấp một phương thức để xác định sự tương đồng về ý nghĩa của từ và các đoạn văn bằng cách phân tích các tập lớn văn bản. Sau khi xử lý một mẫu lớn các ngôn ngữ máy có thể đọc được, LSA đại diện cho các từ được sử dụng trong đó, và tập hợp những từ bất kỳ này, ví dụ như một câu, đoạn văn, hoặc bài luận là điểm mấu chốt trong chiều " không gian ngữ nghĩa ".

Các đại diện của các đoạn văn mà LSA tạo nên có thể được hiểu như là các "tập" trừu tượng, đôi khi các tập có nội dung đơn thuần là lời nói như lập luận triết học, đôi khi từ cuộc sống thật hay tưởng tượng được mã hóa vào các mô tả bằng lời.

LSA khác với một số phương pháp thống kê khác ở **hai khía cạnh quan trọng**.

Thứ nhất, các dữ liệu đầu vào mà sau đó được LSA biểu diễn (xử lý) không đơn thuần chỉ là hội của các từ kế tiếp nhau, mà là hội của các dữ liệu thống nhất về nghĩa (ví như các từ hoặc đoạn có ý nghĩa hoàn chỉnh). Nghĩa là, các dữ liệu ban đầu LSA sử dụng không chỉ là các cặp giao của các từ đồng xuất hiện mà là mô hình chi tiết của sự trùng lặp nhiều lần của nhiều từ trên một số lượng lớn ý nghĩa cục bộ của cả văn bản, giống như là các câu hoặc là các đoạn văn, LSA xử lý tất cả như một thể thống nhất. Do đó, nó bỏ qua cách thứ tự các từ làm nên ý nghĩa của câu văn để nắm bắt sự khác biệt trong cách lựa chọn từ ngữ và trong ý nghĩa của các đoạn văn có liên quan với nhau.

Thứ hai, không như nhiều phương pháp khác, LSA có một bước tiền xử lý, trong đó sự phân phối tổng thể của một từ trong các ngữ cảnh sử dụng của nó (độc lập với các mối tương quan của nó với các từ khác) đầu tiên sẽ được đưa vào một bản kê; thực tế, bước này đã cải thiện đáng kể kết quả của LSA.

Cơ chế giải quyết vấn đề của LSA là sử dụng kỹ thuật SVD (Singular Value Decomposition) nghĩa là kỹ thuật tách giá trị số ít (hoặc là tách giá trị riêng), nhằm làm giảm kích thước của ma trận tần số. LSA xem mỗi tài liệu là một vector có độ dài là k, bởi vì sau khi thực hiện phân tích thành giá trị riêng chỉ giữ lại k hàng để biểu diễn ý nghĩa quan trọng. Kỹ thuật này cho phép loại bỏ những cụm từ và nhóm cụm từ mà phân biệt được giữa những tài liệu khác nhau.

Để củng cố cho những tuyên bố trên, LSA đã được sử dụng để ước lượng sự giống nhau về ý nghĩa của các từ trong văn bản.

Các kết quả cho thấy:

(1) sự tương đồng ý nghĩa gần phù hợp với cách hiểu của con người,

(2) tỷ lệ thu nhận kiến thức từ văn bản của LSA xấp xỉ của con người,

(3) những kết quả đó phụ thuộc rất nhiều vào số chiều đại diện vectơ.

Theo các cách khác nhau, LSA thể hiện khả năng quy nạp kiến thức chính xác và mạnh mẽ, gần xấp xỉ với khả năng của con người. Nó mô phỏng một loạt các hiện tượng nhận thức khác mà phụ thuộc vào ý nghĩa của từ và đoạn văn.

2.4.3 Cách thức hoạt động

LSA là một kỹ thuật thống kê/toán học tự động hoàn toàn dùng để trích rút và suy luận các quan hệ của việc dự kiến sử dụng ngữ cảnh của các từ trong đoạn văn nghị

luận. Nó không phải là phương pháp truyền thống xử lý ngôn ngữ tự nhiên hoặc chương trình trí tuệ nhân tạo.

Với đầu vào là văn bản thô đã được phân tích thành các định nghĩa từ-các chuỗi ký tự đặc biệt và tách thành các đoạn có ý nghĩa hoặc các mẫu câu hoặc đoạn văn.

Bước đầu sẽ là thể hiện văn bản như một ma trận, trong đó mỗi hàng là tượng trưng của một từ duy nhất và mỗi cột là tượng trưng của một đoạn văn bản hoặc ngữ cảnh. Mỗi ô sẽ là các tần số xuất hiện của từ (hàng) trong một đoạn văn(cột). Tiếp theo, các ô ban đầu sẽ được biến đổi sơ bộ trong đó mỗi tần số trong ô sẽ đc suy xét bởi một hàm thể hiện cả tầm quan trọng của từ trong đoạn văn bản cụ thể và mức độ mang thông tin của các từ loại trong các văn bản.

Tiếp theo, LSA áp dụng Phân Tích Giá Trị Số Ít (Singular Value Decomposition - SVD) với ma trận. Sau khi áp dụng SVD, một ma trận ban đầu được phân rã thành ba ma trận. Một ma trận thành phần mô tả các thực thể hàng gốc như là vectơ chuyển hóa các giá trị hệ số trực giao, một ma trận là các thực thể cột gốc, và một ma trận đường chéo chứa giá trị tỉ lệ. Như vậy mà khi nhân ba ma trận lại sẽ được ma trận ban đầu. Kỹ thuật này nhằm mục đích giảm kích thước của ma trận ban đầu, tập trung vào các liên kết mạnh nhất và loại bỏ các nhiễu.

Tóm lại, LSA thực hiện các bước cơ bản sau:

Bước 1- Tạo ma trận tần số của thuật ngữ-tài liệu (ma trận gốc).

Bước 2- Áp dụng SVD: Phân tích ma trận gốc thành 3 ma trận với số chiều nhỏ hơn.

Bước 3- Nhận dạng vectơ: Mỗi tài liệu sẽ được đặt tương ứng với 1 vectơ.

Bước 4- Tạo mục chỉ dẫn: Lưu trữ các vectơ khái niệm được chỉ số bởi một khái niệm nào đó.

Khi khai thác tài liệu với truy vấn, ta chỉ đơn giản ánh xạ truy vấn vào không gian vectơ đã thực hiện ở bước 4 và tìm tài liệu trong tập không gian đó sao cho vectơ tài liệu gần với vectơ truy vấn.

2.4.3.1 Tạo ma trận tần số thuật ngữ-tài liệu

Ví dụ sau sử dụng tài liệu gồm chín bản ghi về kỹ thuật với các chủ đề khá khác nhau, năm bản về vấn đề tương tác máy tính con người (c1-c5), bốn bản về lý thuyết đồ thị toán học (m1-m4). Như vậy ma trận ban đầu có chín cột, 12 hàng, mỗi hàng tương ứng với một thuật ngữ được sử dụng trong ít nhất hai tài liệu (các từ in nghiêng).

Tài liệu của một số bản ghi nhớ kỹ thuật:

c1: Human machine *interface* for ABC *computer* applications

(*Giao diện* máy cho các ứng dụng *máy tính* Lab ABC với con người)

c2: A *survey* of *user* opinion of *computer system response time*

(*Nghiên cứu* sự đánh giá của *người sử dụng* về thời gian *hệ thống máy tính trả lời*)

c3: The *EPS user interface* management *system*

(*Hệ thống* quản lý *giao diện người dùng EPS*)

c4: *System* and *human system* engineering testing of *EPS*

(Kiểm thử kỹ thuật xây dựng *hệ thống* và *con người EPS*)

c5: Relation of *user* perceived *response time* to error measurement

(Mối quan hệ của người *sử dụng-thời gian trả lời* thấy được độ sai lệch đo lường)

m1: The generation of random, binary, ordered *trees*

(Sinh ra các *cây* ngẫu nhiên, nhị phân, không thứ bậc)

m2: The intersection *graph* of paths in *trees*

(*Đồ thị* tác động qua lại của đường dẫn trong các *cây*)

m3: *Graph minors* IV: Widths of *trees* and well-quasi-ordering

(*Thứ bậc đồ thị* IV: Chiều rộng của *cây* và hầu như được sắp thứ tự tốt)

m4: *Graph minors*: A *survey*

(*Thứ bậc đồ thị*: Sự *nghiên cứu*)

{X}=

Thuật ngữ	Tài liệu								
	c1	c2	c3	c4	c5	m1	m2	m3	m4
con người	1			1					
giao diện	1		1						
máy tính	1	1							
người sử dụng		1	1		1				

hệ thống		1	1	1					
trả lời		1			1				
thời gian		1			1				
eps			1	1					
nghiên cứu		1							1
cây						1	1	1	
đồ thị							1	1	1
thứ bậc								1	1

Bảng 2: Số lần xuất hiện của thuật ngữ trong mỗi tài liệu

Mỗi thuật ngữ trong bảng trên, được lấy từ 9 tài liệu. Giá trị mỗi ô là số lần mà một thuật ngữ (hàng) xuất hiện trong tài liệu (cột) (ô trống là giá trị 0).

Sử dụng truy vấn: "Sự tương tác giữa con người với máy tính" (human computer interaction). Từ bảng trên ta có thể thấy kết quả sẽ là các tài liệu c1, c2, c4 vì chúng có chứa một hay nhiều thuật ngữ trong câu truy vấn. Còn c3 và c5 bị bỏ sót vì không có các thuật ngữ chung nào với truy vấn.

2.4.3.2 Áp dụng SVD

Khái niệm

Kỹ thuật SVD (tách giá trị số ít hoặc tách giá trị riêng) được sử dụng nhiều trong lý thuyết ma trận nhằm làm giảm kích thước của tần số. Thông thường, bất kỳ giảm chiều nào đều dẫn tới mất mát thông tin, do vậy phải đảm bảo rằng SVD phải có "năng lực thông tin" (information efficient) cao nhất có thể. Có nghĩa là chúng chỉ mất đi phần bảng tần số ít ý nghĩa nhất.

Cơ sở lý thuyết

Ma trận thuật ngữ-tài X liệu sẽ được tách thành 3 ma trận:

$$\{X\} = \{ T_0 \}\{ S_0 \}\{ D_0^T \}$$

Trong đó:

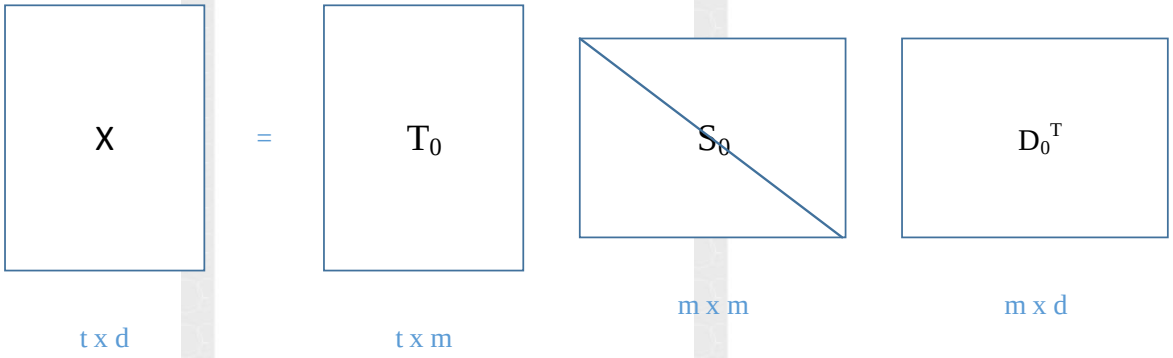
- *X là ma trận chữ nhật t*d của các thuật ngữ và tài liệu.*

- *T_0 và D_o là ma trận có cột trực giao.*

- *S_0 là ma trận chéo (m*m) của các giá trị số ít sắp xếp giảm dần, trong đó m=min(t,d) là hạng của X.*

- *T_0 là ma trận của các vectơ riêng (giá trị số ít) nhận được từ phép nhân ma trận X với ma trận chuyển vị X^T. $T_0=X*X^T$*

- *D_0* là ma trận của các vectơ riêng *(giá trị số ít) nhận được từ phép nhân ma trận chuyển vị X^T với ma trận X. $D_0=X^T*X$*



Hình 6: Sơ đồ SVD của ma trận thuật ngữ tài liệu

Ví dụ, phân tích SVD với ma trận: $X=\begin{bmatrix}4 & 0\\3 & -5\end{bmatrix}$

$$XX^T=\begin{bmatrix}16 & 12\\12 & 34\end{bmatrix} \quad \text{và} \quad X^TX=\begin{bmatrix}25 & -15\\-15 & 25\end{bmatrix}$$

- Tính D_0 dựa vào ma trận X^TX :

Trước tiên, các giá trị riêng được tính theo công thức: $Det(X-c_i)=0$. Trong đó c là giá trị riêng và I là ma trận đơn vị.

$$Det\begin{bmatrix}25-c & -15\\-15 & 25-c\end{bmatrix} <=> (25-c)(25-c)-(-15)(-15)=0$$

$$=> c^2 - 50c + 400=0$$

$$=> c_1=40 \text{ và } c_2=10$$

Dựa vào các giá trị riêng để tính các vectơ riêng theo công thức: $(X-c_i)x=0$ trong đó x là vectơ riêng cần tìm.

- Với $c_1=40$ thì

$$\begin{bmatrix}25-40 & -15\\-15 & 25-40\end{bmatrix}\begin{bmatrix}x1\\x2\end{bmatrix}=\begin{bmatrix}0\\0\end{bmatrix}$$

<=> $-15x_1 - 15x_2 = 0$ và $-15x_1 - 15x_2 = 0$

=> $-x_1 = x_2$

Suy ra $\begin{bmatrix} x1 \\ x2 \end{bmatrix} = \begin{bmatrix} x1 \\ -x1 \end{bmatrix}$

Tính được $L = \sqrt{(x1)^2 + (x2)^2} = x_1\sqrt{2}$

Và $x_1 = \begin{bmatrix} \frac{x1}{L} \\ \frac{-x1}{L} \end{bmatrix} = \begin{bmatrix} \frac{x1}{\sqrt{2}} \\ \frac{-x1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} 0.707 \\ -0.707 \end{bmatrix}$

- Với $c_2 = 10$ ta cũng có $x_1 = x_2$

- Tính ma trận T_0 theo cách tương tự với các giá trị riêng c_1 và c_2 như trên nhưng là với ma trận XX^T

- Ma trận chéo các giá trị riêng của S_0 được tính:

$S_0 = \begin{bmatrix} 40^{\frac{1}{2}} & 0 \\ 0 & 10^{\frac{1}{2}} \end{bmatrix} = \begin{bmatrix} 6.32 & 0 \\ 0 & 3.16 \end{bmatrix}$

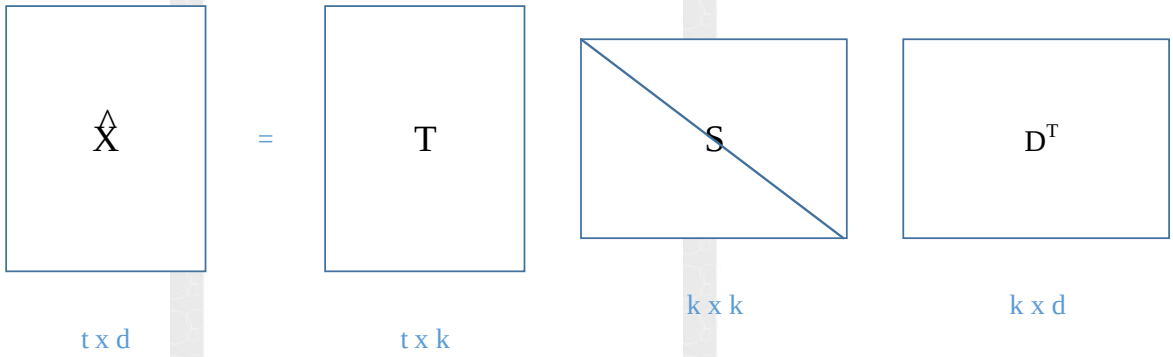
Vậy từ A sau khi áp dụng SVD ta có 3 ma trận sau:

$$\begin{bmatrix} -0.447 & -0.894 \\ -0.894 & 0.447 \end{bmatrix} \begin{bmatrix} 6.32 & 0 \\ 0 & 3.16 \end{bmatrix} \begin{bmatrix} 0.707 & 0.707 \\ -0.707 & 0.707 \end{bmatrix}$$

Sử dụng SVD có thể nhận được một ma trận xấp xỉ của X bởi chỉ các giá trị số ít lớn nhất trong ma trận S_0. Tích của các ma trận là một ma trận $\hat{X}$ xấp xỉ bằng X có hạng k, việc lựa chọn k xác định có bao nhiêu khái niệm quan trọng, với giả định rằng các khái niệm với giá trị số ít nhỏ hơn trong S_0 được coi là nhiễu và có thể bỏ qua. Các giá trị số ít trong S_0 được sắp xếp, đầu tiên k lớn nhất được giữ lại và những tập nhỏ hơn còn lại nhận giá trị 0. Khi đó, các số 0 được đưa vào S_0, ta có đơn giản hóa việc

biểu diễn bằng việc xóa các giá trị bằng 0 trong S_0 để được một ma trận đường chéo S, và sau đó xóa các cột tương ứng của T_0 và D_0 để được T và D tương ứng. Sau khi đã giảm lược:

$$X \approx \hat{X} = TSD^T$$



Hình 7: Sơ đồ SVD được giảm lược của ma trận thuật ngữ - tài liệu

Trong đó số chiều được chọn k≤ m. Giá trị k phải đủ lớn để phù hợp với mọi đặc tính của cấu trúc dữ liệu nhưng đủ nhỏ để lọc ra các chi tiết không phù hợp hay các chi tiết không quan trọng.

Quy trình thực hiện

Từ bảng giá trị ban đầu, phân tích giá trị số ít SVD được hiển thị như hình dưới; trừ lỗi làm tròn số, khi nhân 3 ma trận lại ta sẽ được ma trận gốc như minh họa ban đầu.

Trong ví dụ ban đầu, ma trận X (12*9) sẽ được phân tích thành 3 ma trận như sau:

{ T_0 } =

0.22	-0.11	0.29	-0.41	-0.11	-0.34	0.52	-0.06	-0.41
0.20	-0.07	0.14	-0.55	0.28	0.50	-0.07	-0.01	-0.11
0.24	0.04	-0.16	-0.59	-0.11	-0.25	-0.30	0.06	0.49
0.40	0.06	-0.34	0.10	0.33	0.38	0.00	0.00	0.01
0.64	-0.17	0.36	0.33	-0.16	-0.21	-0.17	0.03	0.27
0.27	0.11	-0.43	0.07	0.08	-0.17	0.28	-0.02	-0.05
0.27	0.11	-0.43	0.07	0.08	-0.17	0.28	-0.02	-0.05
0.30	-0.14	0.33	0.19	0.11	0.27	0.03	-0.02	-0.17
0.21	0.27	-0.18	-0.03	-0.54	0.08	-0.47	-0.04	-0.58
0.01	0.49	0.23	0.03	0.59	-0.39	-0.29	0.25	-0.23
0.04	0.62	0.22	0.00	-0.07	0.11	0.16	-0.68	0.23
0.03	0.45	0.14	-0.01	-0.30	0.28	0.34	0.68	0.18

{ S_0 } =

$$\begin{bmatrix} 3.34 & & & & & & & & \\ & 2.54 & & & & & & & \\ & & 2.35 & & & & & & \\ & & & 1.64 & & & & & \\ & & & & 1.50 & & & & \\ & & & & & 1.31 & & & \\ & & & & & & 0.85 & & \\ & & & & & & & 0.56 & \\ & & & & & & & & 0.36 \end{bmatrix}$$

$\{ D_0^T \} =$

$$\begin{bmatrix} 0.20 & 0.61 & 0.46 & 0.54 & 0.28 & 0.00 & 0.01 & 0.02 & 0.08 \\ -0.06 & 0.17 & -0.13 & -0.23 & 0.11 & 0.19 & 0.44 & 0.62 & 0.53 \\ 0.11 & -0.50 & 0.21 & 0.57 & -0.51 & 0.10 & 0.19 & 0.25 & 0.08 \\ -0.95 & -0.03 & 0.04 & 0.27 & 0.15 & 0.02 & 0.02 & 0.01 & -0.03 \\ 0.05 & -0.21 & 0.38 & -0.21 & 0.33 & 0.39 & 0.35 & 0.15 & -0.60 \\ -0.08 & -0.26 & 0.72 & -0.37 & 0.03 & -0.30 & -0.21 & 0.00 & 0.36 \\ 0.18 & -0.43 & -0.24 & 0.26 & 0.67 & -0.34 & -0.15 & 0.25 & 0.04 \\ -0.01 & 0.05 & 0.01 & -0.02 & -0.06 & 0.45 & -0.76 & 0.45 & -0.07 \\ -0.06 & 0.24 & 0.02 & -0.08 & -0.26 & -0.62 & 0.02 & 0.52 & -0.45 \end{bmatrix}$$

Tiếp theo, ta tìm $\hat{X}$ bằng cách chỉ giữ lại 2 giá trị số ít đầu tiên trong S_0 và các cột tương ứng với ma trận T_0 và D_0. Áp dụng công thức giảm lược:

$$X \approx \hat{X} = TSD^T$$

Ta có ma trận xấp xỉ $\hat{X}$:

$\{\hat{X}\} =$

$$\begin{bmatrix} 0.16 & 0.40 & 0.38 & 0.47 & 0.18 & -0.05 & -0.12 & -0.16 & -0.09 \\ 0.14 & 0.37 & 0.33 & 0.40 & 0.16 & -0.03 & -0.07 & -0.10 & -0.04 \\ 0.15 & 0.51 & 0.36 & 0.41 & 0.24 & 0.02 & 0.06 & 0.09 & 0.12 \\ 0.26 & 0.84 & 0.61 & 0.70 & 0.39 & 0.03 & 0.08 & 0.12 & 0.19 \\ 0.45 & 1.23 & 1.05 & 1.27 & 0.56 & -0.07 & -0.15 & -0.21 & -0.05 \\ 0.16 & 0.58 & 0.38 & 0.42 & 0.28 & 0.06 & 0.13 & 0.19 & 0.22 \\ 0.16 & 0.58 & 0.58 & 0.42 & 0.28 & 0.06 & 0.13 & 0.19 & 0.22 \\ 0.22 & 0.55 & 0.51 & 0.63 & 0.24 & -0.07 & -0.14 & -0.20 & -0.11 \\ 0.10 & 0.53 & 0.23 & 0.21 & 0.27 & 0.14 & 0.31 & 0.44 & 0.42 \\ -0.06 & 0.23 & -0.14 & -0.27 & 0.14 & 0.24 & 0.55 & 0.77 & 0.66 \\ -0.06 & 0.34 & -0.15 & -0.30 & 0.20 & 0.31 & 0.69 & 0.98 & 0.85 \\ -0.04 & 0.25 & -0.10 & -0.21 & 0.15 & 0.22 & 0.50 & 0.71 & 0.62 \end{bmatrix}$$

Ý nghĩa của việc giảm lược:

- Kích thươc của bảng tần số gốc giả sử là t*d, trong đó t là tổng số thuật ngữ và d là tổng tài liệu. Có thể ước chừng t $\approx 1\ triệu$ và d $\approx$ 10,000 (CSDL dạng nhỏ).

- Sau khi giảm chiều thì kích thước ba ma trận đơn giản giả sử còn 200: kích thước ma trận thứ nhất là t*k sẽ là 1 triệu * 200 = 200 triệu dữ liệu đầu vào.

Kích thước ma trận đơn là 200*200 = 40,000 dữ liệu đầu vào nhưng trong đó thì chỉ cần lưu trữ 200, còn lại sẽ nhận giá trị 0.

Kích thước ma trận cuối cùng là k*d sẽ là 200*10,000 = 2 triệu dữ liệu đầu vào.

Tổng tất cả giữ liệu đầu vào sẽ là vào khoảng 202 triệu khi ta áp dụng SVD.

- Nếu không áp dụng thì sẽ là 1 triệu*10,000 $\approx$ 10 tỉ. Như vậy có thể thấy, SVD làm giảm không gian sử dụng xuống 50 lần so với dữ liệu thô.

2.5 Đối sánh văn bản

2.5.1 Độ tương đồng

Các phép đo độ tương tự giữa văn bản và văn bản đã được quan tâm nghiên cứu từ rất lâu trong các ứng dụng của xử lý ngôn ngữ tự nhiên và các lĩnh vực liên quan. Một trong các ứng dụng sớm nhất của độ tương tự văn bản là tìm kiếm thông tin, ở đó các tài liệu có liên quan tới câu truy vấn được xếp hạng theo thứ tự của độ tương tự. Ngoài ra, độ tương tự văn bản còn được dùng cho phân lớp văn bản, trích chọn hay tóm tắt văn bản, phương pháp cho đánh giá dịch máy tự động hay đánh giá tính chặt chẽ của văn bản. Nội dung chương trình bày một số phép tính độ tương đồng văn bản điển hình và phương pháp để xác định độ tương đồng trên văn bản.

2.5.1.1 Khái niệm

Độ tương đồng là một đại lượng dùng để so sánh hai hay nhiều đối tượng với nhau, phản ánh cường độ của mối quan hệ giữa các đối tượng với nhau. Ví dụ: xét 2 câu “Tôi là nam” và “Tôi là nữ”, ta có thể nhận thấy hai câu trên có sự tương đồng khá cao.

Phát biểu bài toán tính độ tương đồng như sau:

Xét 2 văn bản d_i và d_j. Mục tiêu là tìm ra một giá trị $S(d_i,d_j)$, $S \in (0,1)$, thể hiện độ

tương đồng giữa 2 văn bản d_i và d_j. Giá trị càng cao thì sự giống nhau về nghĩa của hai văn bản càng nhiều.

Ví dụ trong mô hình không gian vector, ta sử dụng độ đo Cosine để tính độ tương đồng giữa hai văn bản, mỗi văn bản được biểu diễn bởi một vector. Độ tương tự ngữ nghĩa là khái niệm thể hiện tỷ lệ dựa trên sự giống nhau về nội dung ý nghĩa của tập các tài liệu hoặc các thuật ngữ trong một danh sách các thuật ngữ [**Đỗ Thị Thanh Nga**]. Độ tương đồng ngữ nghĩa phản ánh mối quan hệ ngữ nghĩa giữa các câu, các tài liệu văn bản.

2.5.1.2 Độ tương đồng văn bản dựa trên tập từ chung

- Khoảng cách Jaro

Khoảng cách Jaro định nghĩa độ đo tương tự giữa hai chuỗi. Cho hai câu s_1 và s_2, khoảng cách Jaro d giữa s_1 và s_2 được tính như sau:

$$d = \frac{1}{3}\left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m}\right)$$

Trong đó m là số từ giống nhau, t là ½ số bước chuyển.

Phép chuyển vị trí sẽ được thực hiện khi hai từ giống nhau trong hai câu s_1 và s_2 có khoảng cách không lớn hơn giá trị:

$$\left\lfloor \frac{\max(|s_1|, |s_2|)}{2} \right\rfloor - 1$$

Mỗi từ trong câu s_1 được so sánh với tất cả các từ trong s_2. Số bước chuyển được định nghĩa là số lượng từ giống nhau giữa hai câu (nhưng thứ tự trong chuỗi khác nhau) chia cho 2.

- Mô hình tương phản (Contrast model)

Mô hình tương phản do Tversky đề xuất để tính độ tương tự giữa hai câu A và B như sau:

$$Sim(A,B) = \propto * g(A \cap B) - \beta * g(A - B) - \gamma * g(B - A)$$

Trong đó g(A∩B) biểu diễn cho các từ chung giữa A và B, g(A-B) biểu diễn cho các từ riêng của A và g(B-A) biểu diễn cho các từ riêng của B. Hệ số $\propto,\beta,\gamma$ được xác định trong quá trình thử nghiệm thuật toán.

- Hệ số Jaccard

Hệ số Jaccard là một độ đo tương tự của các tập hợp dựa trên phương pháp thống kê. Theo đó, độ tương tự giữa hai câu A và B như sau:

$$Sim(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

2.5.1.3 Độ tương đồng văn bản dựa trên vector biểu diễn

- Độ tương đồng Cosine:

Trong phương pháp này, các văn bản được biểu diễn theo mô hình không gian vector, mỗi thành phần của vector chỉ đến một từ tương ứng trong danh sách mục từ đã thu được từ quá trình tiền xử lý văn bản đầu.

Không gian vector hay số chiều của vector có kích thước bằng số mục từ trong danh sách mục từ. Giá trị mỗi phần tử của vector là độ quan trọng của mục từ trong câu. Độ quan trọng của từ được tính theo một trong các phương pháp đã trình bày ở trên, phần mô hình vector biểu diễn văn bản, ví dụ:

$w_{ij} = \frac{tf_{ij}}{\sqrt{\sum_j tf_{ij}^2}}$ với tf_{ij} là tần số xuất hiện của từ i trong câu j.

Giả sử vector biểu diễn cho hai văn bản lần lượt có dạng:

$D_i = <w^i_1, \ldots, w^i_t>$ với w^i_t là trọng số của từ thứ t trong không gian vector i.

$D_j = <w^j_1, \ldots, w^j_t>$ với w^j_t là trọng số của từ thứ t trong không gian vector j.

Độ đo tương đồng được tính là Cosine của góc giữa hai vector biểu diễn cho hai văn bản D_i và D_j. Độ tương tự của chúng được tính theo công thức [Trần Ngọc Phúc]:

$$Sim(D_{ij}) = \frac{\sum_{k=1}^{t} w_k^i w_k^j}{\sqrt{\sum_{k=1}^{t}(w_k^i)^2 * \sum_{k=1}^{t}(w_k^j)^2}}$$

Nhận xét: vector biểu diễn cho các câu chưa quan tâm đến mối quan hệ ngữ nghĩa giữa các từ mục, do đó các từ đồng nghĩa sẽ không được phát hiện, kết quả tín độ tương tự chưa cao.

- Độ tương đồng dựa vào khoảng cách Euclide:

Khoảng cách Euclide cũng là một phương pháp khá phổ biến để xác định mức độ tương đồng giữa các vector đặc trưng của hai văn bản [Trần Ngọc Phúc].

Cho hai vector v_a và v_b, khoảng cách Euclide được định nghĩa như sau:

$$E\_dist(\vec{v}_a, \vec{v}_b) = \sqrt{\sum_{i=1}^{n}(w_{ai} - w_{bi})^2}$$

Mức độ tương đồng giữa hai vector được xác định bằng công thức:

$$E\_sim(\vec{v}_a, \vec{v}_b) = 1 - \frac{E\_dist(\vec{v}_a, \vec{v}_b)}{n} = 1 - \frac{1}{n}\sqrt{\sum_{i=1}^{n}(w_{ai} - w_{bi})^2}$$

- Độ tương đồng dựa vào khoảng cách Manhattan:

Khoảng cách Manhattan là phương pháp tính độ tương đồng giữa các vector đặc trưng biểu diễn cho hai văn bản [Trần Ngọc Phúc].

Cho hai vector v_a và v_b, khoảng cách Manhattan được định nghĩa như sau:

$$man\_dist(\vec{v}_a, \vec{v}_b) = \sum_{i=1}^{n}|w_{ai} - w_{bi}|$$

Mức độ tương đồng giữa hai vector được xác định bằng công thức:

$$man\_sim = 1 - \frac{man\_dist(\vec{v}_a, \vec{v}_b)}{n} = 1 - \frac{1}{n}\sum_{i=1}^{n}|w_{ai} - w_{bi}|$$

2.5.2 Độ tương đồng văn bản trong Tiếng Việt

Thông thường khi đánh giá độ tương tự văn bản, chúng ta cần phân tích văn bản thành các đơn vị nhỏ hơn và thực hiện đánh giá dựa trên các đơn vị này. Việc xử lý văn bản và tách từ vựng đã được đề cập trong chương 2, phương pháp cho bài toán so tự sánh văn bản tiếng Việt. Một số bộ công cụ tách từ tiếng Việt như vnTokenizer, JvnTextPro đã được xây dựng và cho kết quả khả quan có thể sử dụng làm bước tiền xử lý cho hệ thống so sánh văn bản. Sau khi tách từ, mỗi văn bản T_i sẽ được biểu diễn bằng một vector các từ có dạng: $T_i = \{w_1, w_2, ..., w_{ni}\}$ với n_i là số từ tách được của T_i.

Trong nhiều trường hợp, độ tương tự giữa hai đoạn văn bản có thể xác định dựa trên so khớp từ đơn giản, điểm tương tự được xác định dựa trên số đơn vị từ vựng xuất hiện ở cả hai đoạn văn bản đầu vào. Tuy nhiên, phương pháp này không thể khẳng định được độ tương tự ngữ nghĩa của văn bản do chưa quan tâm tới hiện tượng đồng nghĩa của từ, tầm quan trọng của từ như tần suất xuất hiện, vị trí xuất hiện của từ và câu trong văn bản.

Các phương pháp đánh giá độ tương tự văn bản chủ yếu dựa trên hai

yếu tố: độ tương tự ngữ nghĩa giữa các từ và độ tương tự theo trật tự của các từ trong văn bản.

- **Đánh giá độ tương tự ngữ nghĩa giữa các từ**

Dựa trên **phân tích ngữ nghĩa tiềm ẩn**. Một số phương pháp sử dụng mạng từ (WordNet), một số khác dựa trên kho ngữ liệu Web.

- **Đánh giá độ tương tự theo trật tự của từ trong văn bản**.

Đánh giá về độ tương tự ngữ nghĩa của từ được dùng để tạo ra các vector đặc trưng ngữ nghĩa của văn bản. Vector đặc trưng này sử dụng các công thức trong mục **" 2.5.1.2 Độ tương đồng văn bản dựa trên tập từ chung"** cho ta một phép đo độ tương tự giữa hai văn bản tương ứng.

2.5.2.1 Độ tương tự ngữ nghĩa từ - từ

Sử dụng phương pháp Phân tích ngữ nghĩa tiềm ẩn đã trình bày ở mục 2.4.

2.5.2.2 Độ tương tự trật tự từ trong văn bản

Độ tương tự về trật tự của từ là một yếu tố quan trọng ảnh hưởng đến độ tương tự của văn bản. Các văn bản cùng chứa một tập từ vựng giống nhau nhưng khác nhau về vị trí có thể có ý nghĩa hoàn toàn khác nhau.

Ví dụ cho hai câu:

T1 = { con_chó, cắn, con_mèo }

T2 = { con_mèo, cắn, con_chó }

Hai câu trên cùng chứa một tập các từ giống nhau và gần giống nhau về thứ tự từ, chỉ sai khác thứ tự của cặp từ "con_mèo" và "con_chó". Nếu chỉ dựa trên độ tương tự ngữ nghĩa của văn bản thì hai văn bản nếu cùng chứa một tập từ giống nhau sẽ cho kết quả là hoàn toàn giống nhau, có nghĩa là sim(T1,T2)=1. Tuy nhiên, hai câu trên có ý nghĩa không giống nhau, sự khác nhau của hai câu chính là do sự sai khác về vị trí của các từ trong câu.

Tác giả Dương Thăng Long trong đề tài nghiên cứu của mình [Dương Thăng Long] đã đưa ra phương pháp đánh giá độ tương tự của văn bản dựa trên thứ tự của từ như sau:

- Với mỗi cặp văn bản T_1 và T_2, xác định tập các từ vựng phân biệt của cả hai văn bản $T = T_1 \cup T_2$.
- Vector đặc trưng thứ tự từ của hai văn bản, kí hiệu $R_1 = (r_{11}, r_{12}, \ldots, r_{1m})$ và $R_2 = (r_{21}, r_{22}, \ldots, r_{2m})$, được tính dựa trên tập T.
  - Vector thứ tự từ biểu diễn thứ tự của mỗi từ thuộc T nằm ở vị trí nào trong văn bản tương ứng.
  - Với mỗi từ $w_i \in T$, tìm một từ đúng hoặc gần nghĩa nhất trong T_1 để xác định trọng số cho phần tử r_{1i} trong R_1 theo một trong ba trường hợp sau:
    - Nếu từ w_i có trong T_1 thì r_{1i} là số thứ tự của từ đó trong T_1.
    - Tìm từ trong T_1 gần nghĩa nhất với w_i, sử dụng phương pháp đo độ tương tự ngữ nghĩa giữa hai từ. Nếu độ đo này vượt ngưỡng θ cho trước thì r_{1i} là số thứ tự của từ đó trong T_1.
    - Nếu không tìm thấy hoặc độ tương tự giữa từ các trong T_1 và w_i không vượt ngưỡng thì đặt r_{1i} là 0. Vector đặc trưng thứ tự của từ

biểu diễn thông tin về cấu trúc của từ trong văn bản. Mức độ giống nhau về cấu trúc của hai văn bản được tính toán dựa trên vector đặc trưng thứ tự từ bằng công thức sau:

$$SimR = 1 - \frac{|R_1 - R_2|}{|R_1 + R_2|} = 1 - \frac{\sqrt{\sum_{i=1}^{m}(r_{1i} - r_{2i})^2}}{\sqrt{\sum_{i=1}^{m}(r_{1i} + r_{2i})^2}}$$

2.6 Tính độ tương đồng cho toàn bộ văn bản

Trong các phân tích ở trên, độ tương tự ngữ nghĩa thể hiện phép đo dựa trên nghĩa từ vựng, còn độ tương tự cấu trúc của từ thể hiện mối quan hệ về thứ tự giữa các từ, các từ đứng trước hoặc sau các từ khác. Cả hai độ tương tự này đều có vai trò quan trọng trong xác định độ tương tự của văn bản. Do đó, để đo sự tương tự của các văn bản cần phải kết hợp của hai loại độ đo trên, độ tương tự về ngữ nghĩa và độ tương tự về thứ tự các từ trong văn bản. Biểu thức kết hợp giữa hai độ đo có dạng

$$sim = a*simS + b*simR \quad \textbf{với a+ b = 1}$$

Việc xác định bộ trọng số của mỗi độ đo tương tự (a,b) chưa có một công thức chung nào, chỉ có thể sử dụng phương pháp quan sát và thử nghiệm qua các dữ liệu thực tế để có lựa chọn tốt nhất cho hệ thống. Phương án cân bằng các tiêu chí là một lựa chọn, trong đó trọng số của các tiêu chí kết hợp có giá trị giống nhau, tức là a = 0,5 và b = 0,5.

CHƯƠNG 3: BÀI TOÁN ÁP DỤNG

3.1 Giới thiệu ngôn ngữ R

3.1.1 Giới thiệu R

Phân tích số liệu và biểu đồ thường được tiến hành bằng các phần mềm thông dụng như SAS, SPSS, Stata, Statistica, và S-Plus. Đây là những phần mềm được các công ti phần mềm phát triển và giới thiệu trên thị trường khoảng ba thập niên qua, và đã được các trường đại học, các trung tâm nghiên cứu và công ti kĩ nghệ trên toàn thế giới sử dụng cho giảng dạy và nghiên cứu. Nhưng vì chi phí để sử dụng các phần mềm này tương đối đắt tiền (có khi lên đến hàng trăm ngàn đô-la mỗi năm), một số trường đại học ở các nước đang phát triển (và ngay cả ở một số nước đã phát triển) không có khả năng tài chính để sử dụng chúng một cách lâu dài. Do đó, các nhà nghiên cứu thống kê trên thế giới đã hợp tác với nhau để phát triển một phần mềm mới, với chủ trương mã nguồn mở, sao cho tất cả các thành viên trong ngành thống kê học và toán học trên thế giới có thể sử dụng một cách thống nhất và **hoàn toàn miễn phí**.

Năm 1996, trong một bài báo quan trọng về tính toán thống kê, hai nhà thống kê học Ross Ihaka và Robert Gentleman, lúc đó thuộc Trường đại học Auckland, New Zealand phác họa một ngôn ngữ mới cho phân tích thống kê mà họ đặt tên là R. Sáng kiến này được rất nhiều nhà thống kê học trên thế giới tán thành và tham gia vào việc phát triển R.

Cho đến nay, qua chưa đầy 10 năm phát triển, càng ngày càng có nhiều nhà thống kê học, toán học, nghiên cứu trong mọi lĩnh vực đã chuyển sang sử dụng R để phân tích dữ liệu khoa học. Trên toàn cầu, đã có một mạng lưới hơn một triệu người sử dụng R, và con số này đang tăng rất nhanh. Có thể nói trong vòng nhiều năm nữa, vai trò của các phần mềm thống kê thương mại sẽ không còn lớn như trong thời gian qua nữa.

Vậy R là gì? Về bản chất, R là ngôn ngữ máy tính đa năng, có thể sử dụng cho nhiều mục tiêu khác nhau, từ tính toán đơn giản, toán học giải trí (recreational mathematics), tính toán ma trận (matrix), đến các phân tích thống kê phức tạp và vẽ biểu đồ. Vì là một ngôn ngữ, cho nên ta có thể sử dụng R để phát triển thành các phần mềm chuyên môn cho một vấn đề tính toán cá biệt.

3.1.2 Các lện trong gói phân tích ngữ nghĩa tiềm ẩn trong R

- alnumx: là chuỗi ký tự bao gồm một biểu thức chính quy sử dụng trong các gsub triển khai trong textvector xác định tất cả các ký tự chữ và số (bao gồm cả các ký tự đặc biệt trong một số ngôn ngữ). Câu lệnh: ***data(alunumx)***

- as.textmatrix: Trả về một không gian ngữ nghĩa tiềm ẩn (tạo ra bởi createLSAspace) ở định dạng textmatrix: hàng là những thuật ngữ, cột là văn bản. Câu lệnh: ***as.textmatrix(LSAspace)***

- associate: trả lại những thuật ngữ dựa trên một ngưỡng, gần với thuật ngữ ban đầu nhất, sắp xếp theo thứ tự giảm dần về sự tương đồng. Câu lệnh: ***associate(textmatrix, term, measure = "cosine", threshhold = 0.7)*** *(tất cả các thuật ngữ >= ngưỡng 0,7 sẽ được chọn sau đó sắp xếp theo thứ tự giảm dần về độ tương đồng, nếu đặt ngưỡng bằng 0 sẽ trả lại tất cả các thuật ngữ)*

- corpora: Đây bộ dữ liệu chứa tập sao ví dụ đánh giá bài luận. Dùng tập lệnh: ***data(corpus_training), data(corpus_essays), hoặc data(corpus_scores)***

- cosine: tính cosine giữa 2 vectơ hoặc là giữa tất cả các vectơ cột của ma trận x. Dùng lệnh**: *cosine(x,y = NULL)*** *(Ma trận **x** có thể là ma trận thuật ngữ-tài liệu, cột là tài liệu, hàng là thuật ngữ. Khi thực thi trên 2 vectơ x,y thì sẽ là tính tương đồng cosine của 2 vectơ.)*

- dimcalc: phương thức dùng để chọn một giá trị hợp lý cho phân tách giá trị số ít SVD trong LSA. Dùng tập lệnh:

dimcalc_share(share=0.5) *(giá trị "share")*

dimcalc_ndocs(ndocs) *(giá trị "ndocs": số tài liệu)*

dimcalc_kaiser()

dimcalc_raw()

dimcalc_fraction(frac=(1/50)) *(giá trị "frac")*

- fold_in: các tài liệu được bổ sung được ánh xạ vào một không gian ngữ nghĩa tiềm ẩn có trước mà không ảnh hưởng đến sự phân phối của không gian. Dùng lệnh: ***fold_in(docvecs, LSAspace)*** *(**docvecs** là ma trận thuật ngữ, **LSAspace** là không gian ngữ nghĩa tiềm ẩn tạo ra bởi lệnh **createLSAspace**)*

- lsa: tính không gian tiềm ẩn ngữ nghĩa từ ma trận thuật ngữ-tài liệu. Câu lệnh: ***lsa(x,dims=dimcalc_share())*** *(trong đó **x** là ma trận thuật ngữ tài liệu, **dims** là số chiều hoặc là cấu trúc hàm.*)

- print.textmatrix: in ma trận text. Lệnh:

print(x, bag_lines, bag_cols, ...) *(**x** là ma trận text, **bag_lines** số hàng, **bag_cols** số cột, dấu **…** là đối số được truyền vào)*

- query: tạo một truy vấn trong định dạng của ma trận text đã có. Lệnh:

query (qtext, termlist, stemming=FALSE, language="german") *(**termlist** là danh sách các thuật ngữ của ngữ cảnh không gian tiềm ẩn ngữ nghĩa, **stemming**: xác định liệu tất cả các thuật ngữ có được hiển thị nghĩa gốc, **qtext** là chuỗi truy vấn, các từ được ngăn cách bởi cách trống)*

- sample.textmatrix: Tạo ra một tập hợp các văn bản của một ngữ liệu để giúp giảm bớt kích thước ngữ liệu thông qua việc lấy mẫu ngẫu nhiên. Lệnh: ***sample.textmatrix(textmatrix, samplesize, index.return=FALSE)*** *(**samplesize** số file mong muôn- Desired number of files, **index.return** nếu để true sẽ trả lại vị trí tập con trong vectơ cột gốc)*

- specialchars: danh sách các ký tự đặc biệt của thực thể html và các ký tự thay thế. Câu lệnh: ***data(specialchars)***

- stopwords : tập từ dừng. Câu lệnh :

data(stopwords_de)

stopwords_de

data(stopwords_en)

stopwords_en

data(stopwords_nl)

stopwords_nl

data(stopwords_fr)

stopwords_fr

data(stopwords_ar)

stopwords_ar

- summary.textmatrix: Trả lại một bản tóm tắt với một số thông tin thống kê về một textmatrix nhất định. Câu lệnh :

summary(object,…) *(**object** là ma trận text, dấu **…** là đối số được đưa ra)*

- textmatrix: tạo ma trận thuật ngữ-tài liệu từ file text trong thư mục nhất định. Tập lệnh :

textmatrix(mydir, stemming=FALSE, language="english",
minWordLength=2, maxWordLength=FALSE, minDocFreq=1,
maxDocFreq=FALSE, minGlobFreq=FALSE, maxGlobFreq=FALSE,
stopwords=NULL, vocabulary=NULL, phrases=NULL,
removeXML=FALSE, removeNumbers=FALSE)

textvector(file, stemming=FALSE, language="english",
minWordLength=2, maxWordLength=FALSE, minDocFreq=1,

maxDocFreq=FALSE, stopwords=NULL, vocabulary=NULL,

phrases=NULL, removeXML=FALSE, removeNumbers=FALSE)

- triples: Cho phép lưu trữ, quản lý và lấy SPO- (chủ đề, vị ngữ, đối tượng) ba liên kết với các cột tài liệu của một ma trận thuật ngữ tài liệu. Tập lệnh:

getTriple(M, subject, predicate)

setTriple(M, subject, predicate, object)

delTriple(M, subject, predicate, object)

getSubjectId(M, subject)

- weightings: tính trọng số sủa ma trận thuật ngữ-tài liệu để chọn ra …

lw_tf(m)

lw_logtf(m)

lw_bintf(m)

gw_normalisation(m)

gw_idf(m)

gw_gfidf(m)

entropy(m)

gw_entropy(m)

*(**m** là ma trận thuật ngữ tài liệu)*

3.2 Cài đặt và chạy chương trình

3.2.1 Cài đặt

Để cài đặt R, ta phải truy nhập vào mạng và vào website "Comprehensive R Archive Network" (CRAN) có đường link như sau:

http://cran.R-project.org

Tài liệu cần tải về, tùy theo phiên bản và hệ điều hành, nhưng thường có tên bắt đầu bằng R và theo sau là số phiên bản. Phiên bản mới nhất hiện nay cho Windows là phiên bản R 3.3.2 cho cả hệ điều hành 32 và 64bit.

Tài liệu này khoảng 70 MB, và địa chỉ cụ thể để tải là:

https://cran.r-project.org/bin/windows/base/

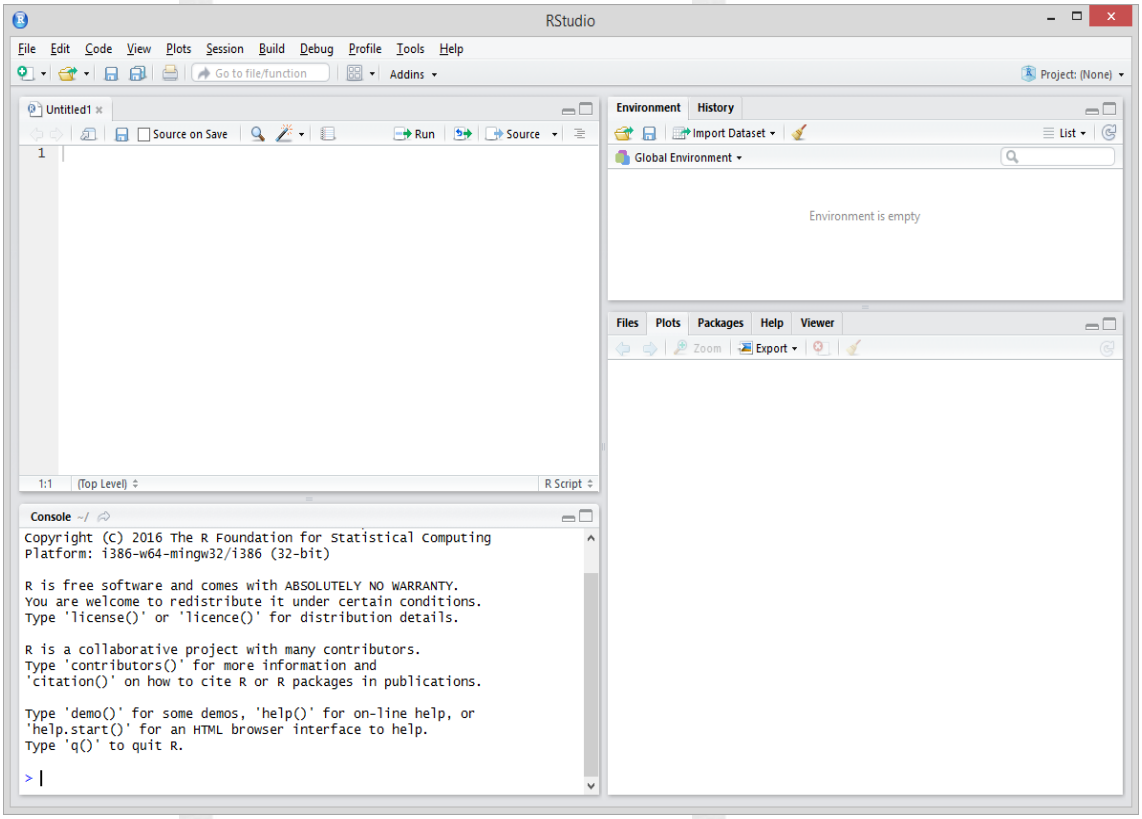
Sau khi hoàn tất quá trình tải và cài đặt, *icon R*

sẽ xuất hiện trên *desktop* của máy tính. Sau đó ta nên cài thêm Rstudio để làm việc với R dễ dàng hơn. Rstudio có phiên bản chạy trên desktop và cả phiên bản chạy trên server, cả 2 đều hoàn toàn miễn phí. Địa chỉ để tải Rstudio:

http://www.rstudio.org/download/

File tải về có tên RStudio-1.0.44 là phiên bản mới nhất hiện tại. Sau khi cài đặt thành công, icon Rstudio sẽ xuất hiện trên desktop

Cửa sổ làm việc của Rstudio:



Hình 8: Cửa sổ làm việc của Rstudio

3.2.2 Thực nghiệm

Sử dụng phần mềm Rstudio để thực hiện LSA.

Input:

- Dữ liệu gồm chín tiêu đề tài liệu về kỹ thuật với các chủ đề khá khác nhau, năm về vấn đề tương tác máy tính con người (c1-c5), bốn về lý thuyết đồ thị toán học (m1-m4).

  c1: Human machine *interface* for ABC *computer* applications

  (*Giao diện* máy cho các ứng dụng *máy tính* Lab ABC với con người)

  c2: A *survey* of *user* opinion of *computer system response time*

  (*Nghiên cứu* sự đánh giá của *người sử dụng* về thời gian *hệ thống máy tính trả lời*)

  c3: The *EPS user interface* management *system*

  (*Hệ thống* quản lý *giao diện người dùng EPS*)

  c4: *System* and *human system* engineering testing of *EPS*

  (Kiểm thử kỹ thuật xây dựng *hệ thống* và *con người EPS*)

  c5: Relation of *user* perceived *response time* to error measurement

  (Mối quan hệ của người *sử dụng-thời gian trả lời* thấy được độ sai lệch đo lường)

  m1: The generation of random, binary, ordered *trees*

  (Sinh ra các *cây* ngẫu nhiên, nhị phân, không thứ bậc)

  m2: The intersection *graph* of paths in *trees*

  (*Đồ thị* tác động qua lại của đường dẫn trong các *cây*)

  m3: *Graph minors* IV: Widths of *trees* and well-quasi-ordering

  (*Thứ bậc đồ thị* IV: Chiều rộng của *cây* và hầu như được sắp thứ tự tốt)

  m4: *Graph minors*: A *survey*

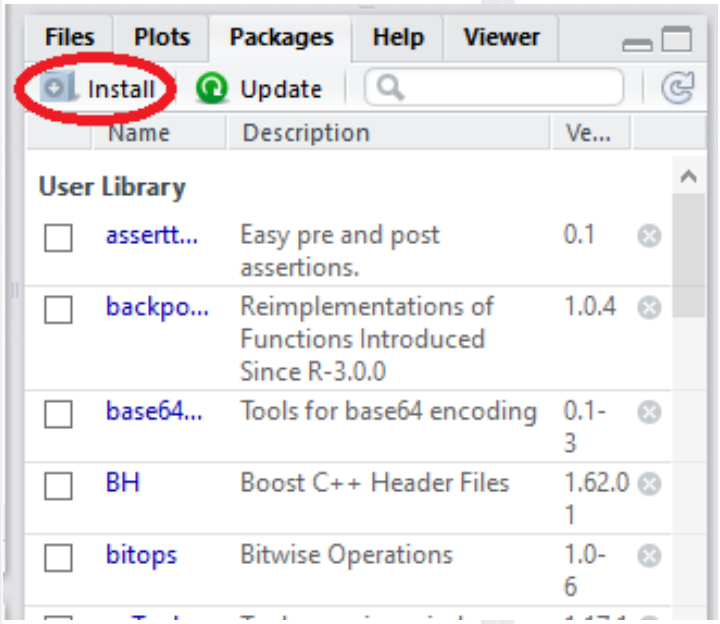
  (*Thứ bậc đồ thị*: Sự *nghiên cứu*)

Output:

- Tương quan thuật ngữ-tài liệu.

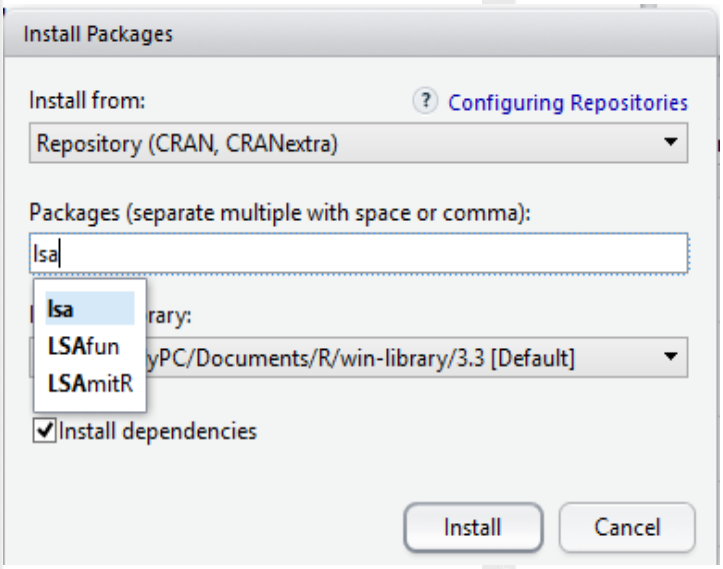
Kết quả: Ma trận, biểu đồ tương quan thuật ngữ tài liệu sau khi chạy "LSApackage".

Cài thư viện lsa trước để có thể chạy được "LSApackage". Để cài đặt, ta mở Rstudio, tại phần Packages, chọn Install:



Hình 9: Cài đặt thư viện lsa

Sau khi ấn Install, ta gõ "lsa" vào ô Packages, sau đó nhấn Install, Rstudio sẽ tự động tải và cài đặt thư viện lsa:



Hình 10: Các thư viện lsa

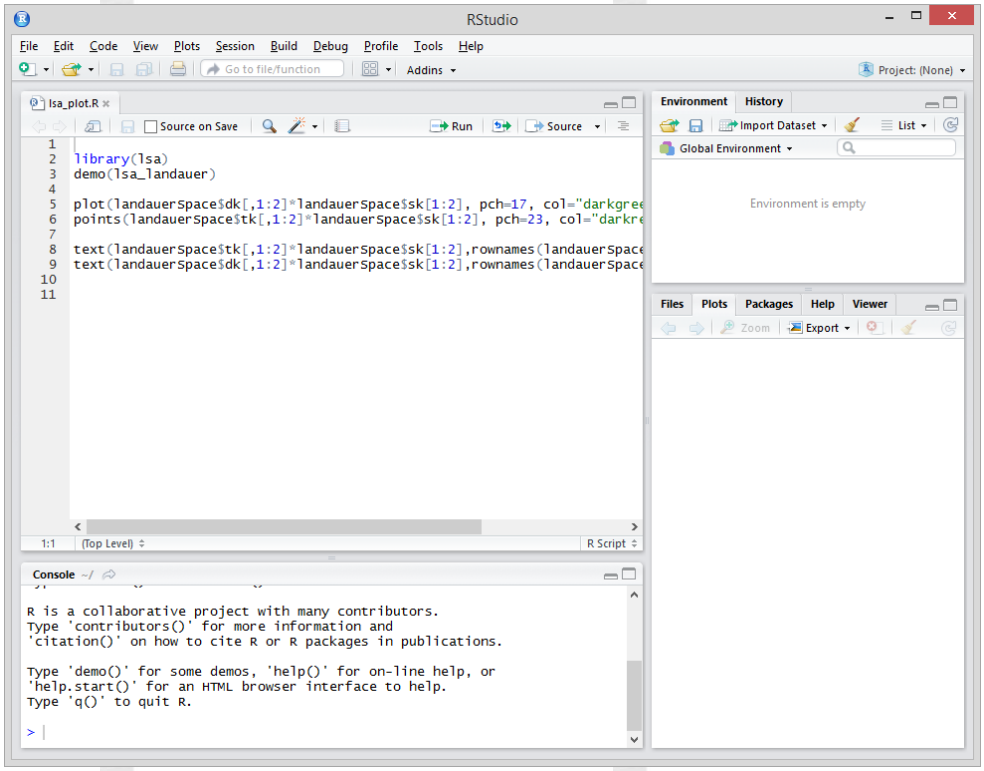
Sau khi cài đặt thư viện lsa, ta tải gói LSA "LSA package" tại địa chỉ:

https://CRAN.R-project.org/package=lsa

Khối lệnh của phần Input (file lsa_landauer.R):

```
ldir = tempfile()
dir.create(ldir)
write( c("human", "interface", "computer"), file=paste(ldir, "c1", sep="/"))
write( c("survey", "user", "computer", "system", "response", "time"), file=paste(ldir, "c2", sep="/"))
write( c("EPS", "user", "interface", "system"), file=paste(ldir, "c3", sep="/"))
write( c("system", "human", "system", "EPS"), file=paste(ldir, "c4", sep="/"))
write( c("user", "response", "time"), file=paste(ldir, "c5", sep="/"))
write( c("trees"), file=paste(ldir, "m1", sep="/"))
write( c("graph", "trees"), file=paste(ldir, "m2", sep="/"))
write( c("graph", "minors", "trees"), file=paste(ldir, "m3", sep="/"))
write( c("graph", "minors", "survey"), file=paste(ldir, "m4", sep="/"))
```

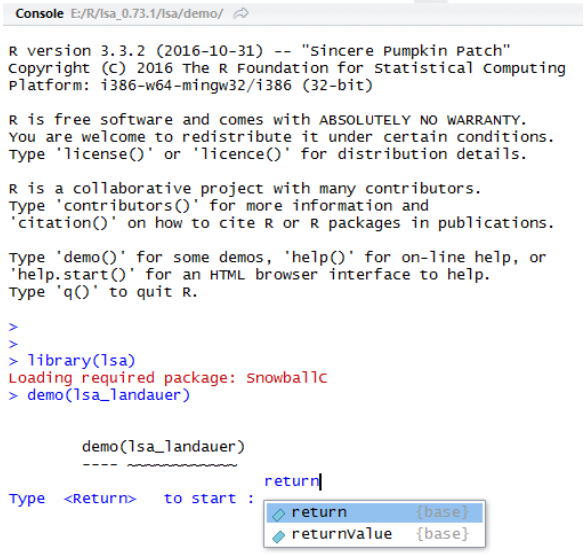
File lsa_plot trong thư mục demo bằng Rstudio:



Hình 11: File lsa_plot.R

Ta thực thi lần lượt từng dòng lệnh bằng cách để trỏ chuột tại đầu dòng, sau đó nhấn biểu tượng Run hoặc tổ hợp CTRL+Enter.

Tại đây, ta gõ lệnh “return” để bắt đầu:



Hình 12: Lệnh return

Sau khi thực thi, file lsa_landauer.R trong thư mục demo sẽ được chạy, thực hiện các bước phân tích ngữ nghĩa tiềm ẩn.

```
Console ~/
> ldir = tempfile()

> dir.create(ldir)

> write( c("human", "interface", "computer"), file=paste(ldir, "c1", sep="/"))

> write( c("survey", "user", "computer", "system", "response", "time"), file=paste(ldir, "c2",
 sep="/"))

> write( c("EPS", "user", "interface", "system"), file=paste(ldir, "c3", sep="/"))

> write( c("system", "human", "system", "EPS"), file=paste(ldir, "c4", sep="/"))

> write( c("user", "response", "time"), file=paste(ldir, "c5", sep="/"))

> write( c("trees"), file=paste(ldir, "m1", sep="/"))

> write( c("graph", "trees"), file=paste(ldir, "m2", sep="/"))

> write( c("graph", "minors", "trees"), file=paste(ldir, "m3", sep="/"))

> write( c("graph", "minors", "survey"), file=paste(ldir, "m4", sep="/"))
> # -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
> # generate doc term matrix from landauer files
>
> dtm = textmatrix(ldir, minWordLength=1)

> dtm
          docs
terms      c1 c2 c3 c4 c5 m1 m2 m3 m4
```

Hình 13: Các thuật ngữ-tài liệu

```
Console ~/
> # generate doc term matrix from landauer files
>
> dtm = textmatrix(ldir, minWordLength=1)

> dtm
          docs
terms      c1 c2 c3 c4 c5 m1 m2 m3 m4
  computer  1  1  0  0  0  0  0  0  0
  human     1  0  0  1  0  0  0  0  0
  interface 1  0  1  0  0  0  0  0  0
  response  0  1  0  0  1  0  0  0  0
  survey    0  1  0  0  0  0  0  0  1
  system    0  1  1  2  0  0  0  0  0
  time      0  1  0  0  1  0  0  0  0
  user      0  1  1  0  1  0  0  0  0
  eps       0  0  1  1  0  0  0  0  0
  trees     0  0  0  0  0  1  1  1  0
  graph     0  0  0  0  0  0  1  1  1
  minors    0  0  0  0  0  0  0  1  1

> # -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
> # make a space, reconstruct original
>
> landauerOriginalSpace = lsa(dtm, dims=dimcalc_raw())

> X = as.textmatrix(landauerOriginalSpace)

> # X should be equal to dtm (beside rounding errors)
> all( (round(X,2) == dtm) == TRUE)
[1] TRUE
```

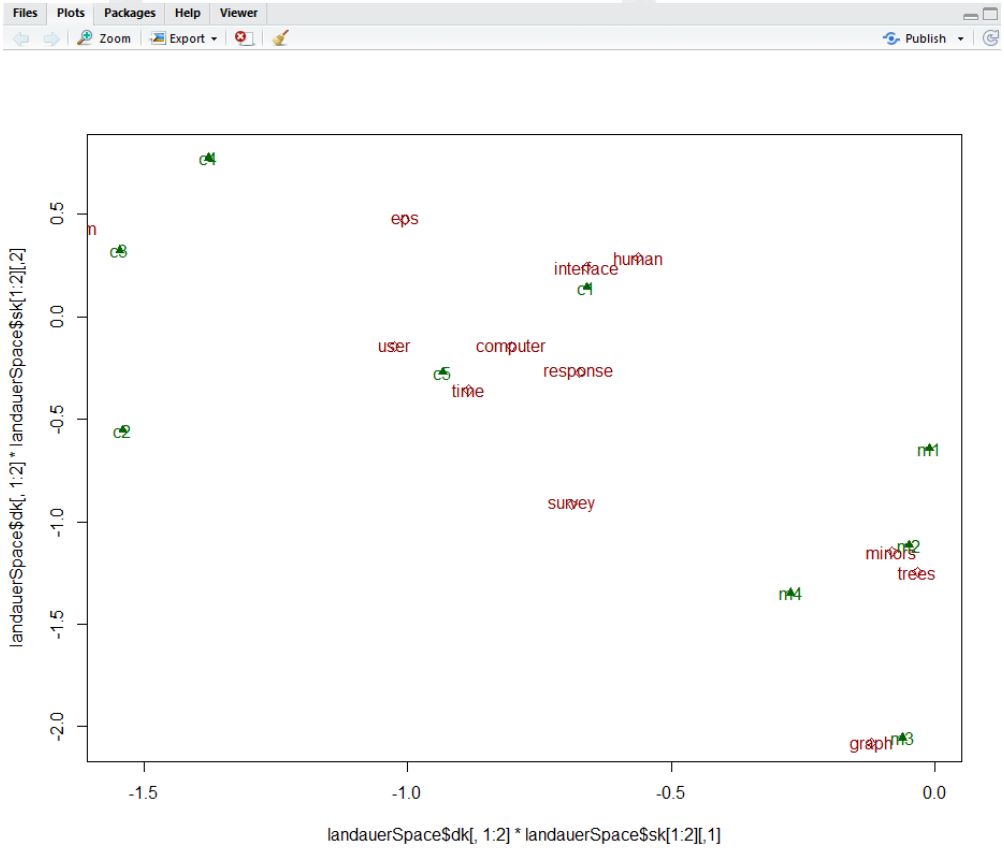
Hình 14: Ma trận thuật ngữ tài liệu

```
Console ~/
> round(Y,2)
            c1   c2    c3    c4   c5    m1    m2    m3    m4
computer  0.15 0.51  0.36  0.41 0.24  0.02  0.06  0.09  0.12
human     0.16 0.40  0.38  0.47 0.18 -0.05 -0.12 -0.16 -0.09
interface 0.14 0.37  0.33  0.40 0.16 -0.03 -0.07 -0.10 -0.04
response  0.16 0.58  0.38  0.42 0.28  0.06  0.13  0.19  0.22
survey    0.10 0.53  0.23  0.21 0.27  0.14  0.31  0.44  0.42
system    0.45 1.23  1.05  1.27 0.56 -0.07 -0.15 -0.21 -0.05
time      0.16 0.58  0.38  0.42 0.28  0.06  0.13  0.19  0.22
user      0.26 0.84  0.61  0.70 0.39  0.03  0.08  0.12  0.19
eps       0.22 0.55  0.51  0.63 0.24 -0.07 -0.14 -0.20 -0.11
trees    -0.06 0.23 -0.14 -0.27 0.14  0.24  0.55  0.77  0.66
graph    -0.06 0.34 -0.15 -0.30 0.20  0.31  0.69  0.98  0.85
minors   -0.04 0.25 -0.10 -0.21 0.15  0.22  0.50  0.71  0.62

> # -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
> # now read in again the landauer sample (but
> # with the vocabulary of the existing matrix)
>
> pdocs = textmatrix(ldir, vocabulary=rownames(dtm))

> # -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
> # now calc a pseudo SVD on the basis of dtm's SVD
>
> Y2 = fold_in(pdocs, landauerSpace)

> round(Y2,2)
            c1   c2    c3    c4   c5    m1    m2    m3    m4
computer  0.15 0.51  0.36  0.41 0.24  0.02  0.06  0.09  0.12
human     0.16 0.40  0.38  0.47 0.18 -0.05 -0.12 -0.16 -0.09
interface 0.14 0.37  0.33  0.40 0.16 -0.03 -0.07 -0.10 -0.04
response  0.16 0.58  0.38  0.42 0.28  0.06  0.13  0.19  0.22
survey    0.10 0.53  0.23  0.21 0.27  0.14  0.31  0.44  0.42
system    0.45 1.23  1.05  1.27 0.56 -0.07 -0.15 -0.21 -0.05
time      0.16 0.58  0.38  0.42 0.28  0.06  0.13  0.19  0.22
user      0.26 0.84  0.61  0.70 0.39  0.03  0.08  0.12  0.19
eps       0.22 0.55  0.51  0.63 0.24 -0.07 -0.14 -0.20 -0.11
trees    -0.06 0.23 -0.14 -0.27 0.14  0.24  0.55  0.77  0.66
graph    -0.06 0.34 -0.15 -0.30 0.20  0.31  0.69  0.98  0.85
minors   -0.04 0.25 -0.10 -0.21 0.15  0.22  0.50  0.71  0.62
```

Hình 15: Ma trận giảm chiều

```
Console ~/
[1] TRUE

> # calc pearson doc2doc correlation
>
> rawCor = cor(dtm)

> lsaCor = cor(Y)

> # you should clearly see, that the "computer" documents (starting with "c")
> # can in lsaCor be much better be differentiated from the "math" documents
> # (starting with "m"). Moreover, the computer and math documents respectively
> # have become more similar within their group.
>
> round(rawCor,2)
      c1    c2   c3   c4    c5   m1   m2   m3   m4
c1  1.00 -0.19 0.00 0.00 -0.33 -0.17 -0.26 -0.33 -0.33
c2 -0.19  1.00 0.00 0.00  0.58 -0.30 -0.45 -0.58 -0.19
c3  0.00  0.00 1.00 0.47  0.00 -0.21 -0.32 -0.41 -0.41
c4  0.00  0.00 0.47 1.00 -0.31 -0.16 -0.24 -0.31 -0.31
c5 -0.33  0.58 0.00 -0.31  1.00 -0.17 -0.26 -0.33 -0.33
m1 -0.17 -0.30 -0.21 -0.16 -0.17  1.00  0.67  0.52 -0.17
m2 -0.26 -0.45 -0.32 -0.24 -0.26  0.67  1.00  0.77  0.26
m3 -0.33 -0.58 -0.41 -0.31 -0.33  0.52  0.77  1.00  0.56
m4 -0.33 -0.19 -0.41 -0.31 -0.33 -0.17  0.26  0.56  1.00

> round(lsaCor,2)
      c1    c2   c3   c4    c5   m1   m2   m3   m4
c1  1.00  0.91 1.00 1.00  0.84 -0.86 -0.85 -0.85 -0.81
c2  0.91  1.00 0.91 0.88  0.99 -0.57 -0.56 -0.56 -0.50
c3  1.00  0.91 1.00 1.00  0.84 -0.86 -0.85 -0.85 -0.81
c4  1.00  0.88 1.00 1.00  0.81 -0.89 -0.88 -0.88 -0.84
c5  0.84  0.99 0.84 0.81  1.00 -0.44 -0.44 -0.43 -0.37
m1 -0.86 -0.57 -0.86 -0.89 -0.44  1.00  1.00  1.00  1.00
m2 -0.85 -0.56 -0.85 -0.88 -0.44  1.00  1.00  1.00  1.00
m3 -0.85 -0.56 -0.85 -0.88 -0.43  1.00  1.00  1.00  1.00
m4 -0.81 -0.50 -0.81 -0.84 -0.37  1.00  1.00  1.00  1.00

> # clean up
> unlink(ldir, recursive=TRUE)
>
```

Hình 16: Ma trận tài liệu-tài liệu



Hình 17: Biểu đồ tương quan thuật ngữ-tài liệu

KẾT LUẬN

Qua lần thực hiện đồ án này, tìm hiểu về lĩnh vực được rất động các cộng đồng khoa học trong và ngoài nước tham gia nghiên cứu và phát triển là phân tích ngữ nghĩa tiềm ẩn. Giúp em học hỏi và hoàn thiện hơn các kỹ năng từ lý thuyết đến thực hành.

Tìm hiểu về phân tích ngữ nghĩa tiềm ẩn trong đối sánh văn bản đem đến cho em các hiểu biết về các phương pháp phân tích, đối sánh văn bản, các phương pháp biến đổi, xử lý ngôn ngữ tự nhiên…

Đồ án đã phần nào giải quyết được vấn đề sau:

- Nghiên cứu sơ lược các phương pháp phân tích, xử lý ngôn ngữ tự nhiên

- Nghiên cứu về phương pháp phân tích ngữ nghĩa tiềm ẩn, độ tương tự thứ tự từ, qua đó áp dụng vào việc đánh giá độ tương tự văn bản.

Ứng dụng phân tích ngữ nghĩa tiềm ẩn trong đối sánh văn bản đã và đang được áp dụng trên rất nhiều nơi, với nhiều các mức độ khác nhau, từ trích chọn thông tin, đến đánh giá quan điểm người dùng, qua đó hướng người dùng đến các vấn đề mà họ quan tâm, hay là vấn đề về sao chép tài liệu, ngoài ra còn ứng dụng trong các vấn đề an ninh quốc phòng, kinh tế chính trị và rất nhiều vấn đề khác. Hướng phát triển của đề tài sẽ là nghiên cứu chuyên sâu và hoàn thiện hơn các công đoạn xử lý tiếng Việt như các từ đồng nghĩa, các từ ghép, cụm từ… để làm cho kết quả của hệ thống so sánh được chính xác và hoàn thiện hơn.

TÀI LIỆU THAM KHẢO

[Thomas K Landauer**,** Peter W. Foltz**,** Darrell Laham] **An Introduction to Latent Semantic Analysis.** Thomas K Landauer**,** Peter W. Foltz**,** Darrell Laham, 1998.

[Đặng Thị Hưởng] Đặng Thị Hưởng. ***Semantics***, TP.Hồ Chí Minh, 1997.

[Đỗ Thị Thanh Nga] "***Tính toán độ tương tự ngữ nghĩa văn bản dựa vào độ tương tự giữa từ với từ***", Đỗ Thị Thanh Nga, Đại học Công nghệ Đại học Quốc gia Hà Nội, 2010.

[TS. Dương Thăng Long] "***Nghiên cứu độ tương đồng văn bản trong tiếng Việt và ứng dụng hỗ trợ đánh giá việc sao chép bài điện tử***", TS. Dương Thăng Long, Viện Đại học Mở Hà Nội, 2014.

[Trần Ngọc Phúc] **“Phân loại nội dung tài liệu Web”**, Trần Ngọc Phúc, Đại học Lạc Hồng, 2012.

[Nguyen Thi Minh Huyen, Vu Xuan Luong, Le Hong Phuong] Nguyen Thi Minh Huyen, Vu Xuan Luong, Le Hong Phuong. **A case study of the probabilistic tagger QTAG for Tagging Vietnamese Texts.**