NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....

Trà Vinh, ngày ….. tháng …… năm ……

Giáo viên hướng dẫn

(Ký tên và ghi rõ họ tên)

NHẬN XÉT CỦA THÀNH VIÊN HỘI ĐỒNG

....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....
....

Trà Vinh, ngày ..... tháng …… năm ……

Thành viên hội đồng

(Ký tên và ghi rõ họ tên)

LỜI CẢM ƠN

Trước hết, em xin được bày tỏ lòng biết ơn sâu sắc đến thầy Nguyễn Nhứt Lam, là người hướng dẫn em trong đồ án này. Thầy đã luôn tận tình hướng dẫn, giúp đỡ em từ khi bắt đầu chọn đề tài đến khi hoàn thành đồ án. Thầy đã cung cấp cho em kiến thức chuyên sâu và kinh nghiệm quý báu để giúp em hoàn thành đồ án.

Em cũng muốn gửi lời cảm ơn đến những người bạn của em, nhờ có sự giúp đỡ và hỗ trợ từ các bạn, em đã có thể vượt qua những khó khăn trong quá trình nghiên cứu và hoàn thành đồ án. Các bạn đã cùng nhau chia sẻ kiến thức và kinh nghiệm, đóng góp ý kiến để em có thể phấn đấu hoàn thành đồ án.

Một lần nữa, em xin chân thành cảm ơn thầy và bạn bè đã giúp đỡ em trong quá trình nghiên cứu và hoàn thành đồ án. Em hy vọng rằng những kiến thức và kinh nghiệm em đã học được trong quá trình này sẽ trở thành nền tảng cho sự phát triển và thành công của em trong tương lai.

MỤC LỤC

DANH MỤC HÌNH ẢNH – BẢNG BIỂU

TÓM TẮT ĐỒ ÁN CƠ SỞ NGÀNH

Đồ án cơ sở ngành được thực hiện với mục tiêu tìm hiểu về thuật toán Topic-Sensitive PageRank và đánh giá hiệu suất của nó. Trong thuật toán PageRank ban đầu để cải thiện thứ hạng của kết quả truy vấn tìm kiếm, một vector duy nhất được tính toán, sử dụng cấu trúc liên kết của các trang web để đánh giá tầm quan trọng của các trang web, độc lập với bất kì truy vấn tìm kiếm cụ thể nào. Để tạo ra kết quả tìm kiếm chính xác hơn, chúng ta cần tính toán một tập hợp các vector PageRank, bằng cách sử dụng một tập hợp các chủ đề đại diện, để nắm bắt chính xác hơn khái niệm về tầm quan trọng đối với các chủ đề cụ thể. Bằng cách sử dụng các vector PageRank có thiên vị được tính toán trước để tạo điểm số tầm quan trọng đối với một chủ đề cụ thể cho truy vấn các trang tại thời điểm truy vấn, cho thấy rằng chúng ta có thể tạo ra kết quả chính xác hơn với thứ hạng chính xác hơn so với một PageRank chung duy nhất. Các điểm PageRank theo chủ đề được tính bằng cách xác định chủ đề dựa trên các từ khóa được dùng để truy vấn dựa trên chủ đề ngữ cảnh thực hiện tìm kiếm để tính toán PageRank.

MỞ ĐẦU

1. Lý do chọn đề tài

Trong thời đại công nghệ phát triển hiện nay, Internet đã trở thành một kho không tin khổng lồ, xã hội càng phát triển, thì lưu lượng thông tin được lưu trữ ngày càng lớn hơn. Vậy làm thế nào người ta có thể thộng tin chính xác một cách nhanh chóng, phù hợp với nhu cầu của họ trong khi có một biển thông tin? Thuật toán tìm kiếm như PageRank ra đời để đáp ứng nhu cầu tìm kiếm thông tin của người dùng bằng cách xếp hạng kết quả tìm kiếm, dù đã có những thành công nhất định, tuy nhiên vẫn còn nhiều hạn chế, đặc biệt là trong kết quả tìm kiếm khi nó chỉ cho ra kết quả tổng quát thay vì xem xét ngữ cảnh hoặc chủ đề cụ thể mà người dùng quan tâm.

Thuật toán Topic-Sensitive PageRank là một trong những thuật toán dùng trong việc xếp hạng kết quả tìm kiếm, là một biến thể của thuật toán PageRank được dùng trong Google Search để xếp hạng các trang web dựa trên chủ đề và ngữ cảnh tìm kiếm.

Với sự quan tâm dành cho thuật toán xếp hạng này, em đã chọn đề tài "Tìm hiểu và mô phỏng thuật toán Topic-Sensitive PageRank". Bài viết sẽ trình bày các kiến thức cơ bản về thuật toán PageRank, biến thể của nó là Topic-Sensitive PageRank và ứng dụng của nó trong tìm kiếm thông tin, cũng như giới thiệu phương pháp tiếp cận để giải quyết bài toán xếp hạng.

2. Mục đích

Mục đích nghiên cứu nhằm phát triển và cải tiến thuật Topic-Sensitive PageRank để nâng cao độ chính xác và liên quan của kết quả tìm kiếm một cách cụ thể.

3. Đối tượng

Đối tượng nghiên cứu bao gồm thuật toán PageRank và biến thể của nó (Topic-Sensitive PageRank), các phương pháp phân loại chủ đề, các chỉ số dùng để đánh giá hiệu suất của thuật toán.

4. Phạm vi nghiên cứu

Phạm vi nghiên cứu sẽ bao gồm tìm hiểu về lí thuyết của PageRank, Topic-Sensitive PageRank và một số khái niệm liên quan, phân tích các phương pháp hiện có, thực hiện cài đặt và kiểm thử để đánh giá hiệu suất.

CHƯƠNG 1: TỔNG QUAN

1.1 Sơ lược về sự hình thành và phát triển Topic-Sensitive PageRank

Thuật toán Topic-Sensitive PageRank là một biến thể của thuật toán PageRank gốc, được tính toán dựa trên chủ đề hoặc ngữ cảnh của trang web khi tính thứ hạng của trang web đó. Ý tưởng chính là cá nhân hóa phép tính PageRank cho từng chủ đề hoặc truy vấn, để thứ hạng của các trang thiên về những trang có liên quan hơn đến chủ đề đã cho.

Trong những nghiên cứu trước đây về Topic-Sensitive PageRank và ứng dụng trong việc tìm kiếm theo chủ đề, có thể kể đến một số nghiên cứu như sau.

- Nghiên cứu của Taher H.Haveliwala(2002), một nhà nghiên cứu tại đại học Stanford, đã phát triển thuật toán PageRank theo chủ đề (Topic-Sensitive PageRank)
- Nghiên cứu của Sergey Brin và Lawrence Page (1990), người đã phát triển ra thuật toán PageRank là một trong những tiền đề để phát triển thuật toán Topic-Sensitive PageRank.

Những nghiên cứu này đã chứng minh tính khả thi và hiệu quả của thuật toán Topic-Sensitive PageRank trong việc xếp hạng trong tìm kiếm theo chủ đề. Tuy nhiên, vấn đề này vẫn còn nhiều thách thức và đòi hỏi sự nghiên cứu và phát triển liên tục.

Đồ án này sẽ tập trung vào tìm hiểu và phỏng thuật toán Topic-Sensitive PageRank. Vấn đề này đang được quan tâm bởi tính ứng dụng của nó trong tìm kiếm thông tin cũng như cá nhân hóa người dùng dựa trên các chủ đề liên quan đến nội dung tìm kiếm của người dùng và nhiều ứng dụng khác.

1.2 Một số ứng dụng của Topic-Sensitive PageRank

Tìm kiếm thông tin: Topic-Sensitive PageRank có thể được sử dụng để cải thiện kết quả tìm kiếm dựa trên ngữ cảnh và chủ đề tìm kiếm bằng cách sử dụng các vector PageRank. Ví dụ, Một người muốn tìm hiểu về ”địa điểm du lịch nổi tiếng ở Sapa”. Topic-Sensitive PageRank có thể được sử dụng để phân tích các trang web du lịch, blog cá nhân và các diễn đàn về du lịch để xác định các trang có nội dung liên quan đến Sapa. Kết quả tìm kiếm sẽ ưu tiên các vài viết về hướng dẫn du lịch và đánh giá từ

những người đã từng đến Sapa giúp người dùng có cái nhìn tổng quan và lựa chọn tốt hơn cho chuyến đi của mình.

Cá nhân hóa người dùng: dựa trên những chủ đề tìm kiếm của người dùng, thuật toán sẽ đề xuất những bài viết hay video có nội dung liên quan đến chủ đề đã tìm kiếm trước đây. Ví dụ, cá nhân A thích xem bóng đá khi A tìm kiếm video liên quan về bóng đá trên Youtube với tần suất nhiều thì Youtube sẽ dựa trên lịch sử tìm kiếm Topic-Sensitive PageRank có thể phân tích những nội dung mà người đã xem để xuất ra nhiều video liên quan đến bóng đá cho người dùng A.

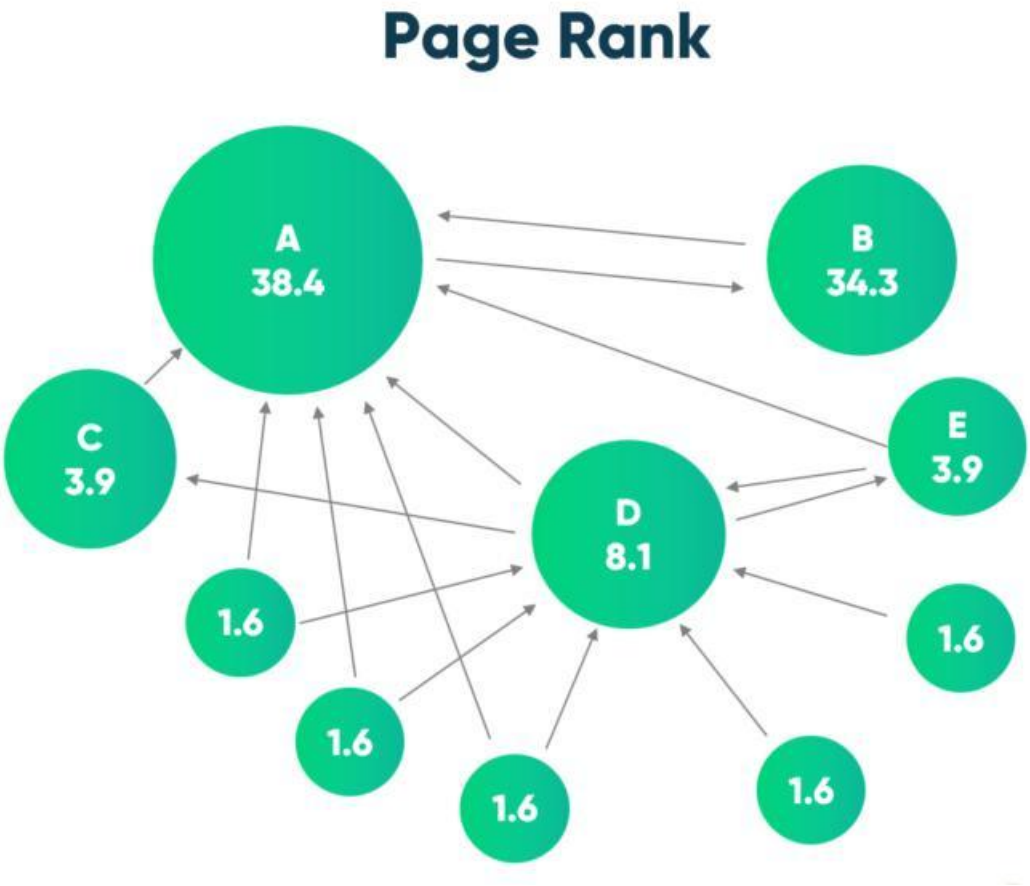
Tối ưu hóa quảng cáo: TopicSensitive PageRank có thể được sử dụng để tối ưu hóa quảng cáo bằng cách xác định các trang web có liên quan nhất cho các chiến dịch quảng cáo, từ đó tối ưu hóa việc phân phối quảng cáo đến đúng đối tượng mục tiêu. Ví dụ, một công ty bán giầy thể thao muốn quảng bá sản phẩm của họ, công ty sử dụng Topic-Sensitive PageRank để phân tích các trang web thể thao, blog về sức khỏe và diễn đàn thể thao. Topic-Sensitive PageRank giúp xác định các trang nội dung liên quan đến giày thể thao như các bài viết về “giày đá bóng tốt nhất”, “giày chạy bộ tốt nhất” hoặc “hướng dẫn chọn giày thể thao phù hợp”. Kết quả là quảng cáo của công ty sẽ xuất hiện trên các trang này, tăng khả năng người dùng nhấp vào quảng cáo vì học đang tìm kiếm thông tin liên quang

CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT

2.1 Cơ sở lý thuyết

2.1.1 Giới thiệu sơ lược về PageRank

Thuật toán PageRank là thuật toáng xếp hạng đồ thị được nghiên cứu và phát minh bởi Sergey Brin và Larry Page. Dựa trên cấu trúc tổng quát của trang web, chúng ta có thể tính được chính xác thứ hạng của từng đỉnh thuật toán này được ứng dụng trong công cụ tìm kiếm Google.



Hình 1. Minh hoạ đồ thị của thuật toán PageRank.[5]

Trong báo cáo của Lê Kim Dung Trường Đại học Công Nghệ đã nghiên cứu về thuật toán PageRank đơn giản như sau:

Gọi G là một đồ thị của trang web. Đặt G = (V, E) với V = {1,2,…n} là tập n đỉnh của dồ thị G (mỗi đỉnh là một trang web cần tính hạng trang) còn E là tập các cạnh, E = {(i, j)/ nếu có siêu liên kết từ trang i đến trang j}. Chúng ta có thể đưa ra giả thuyết rằng đồ thị của trang web là liên thông, nghĩa là từ một trang bất kì có thể có đừng liên kết tới một trang web khác trong đồ thị đó.

Cho một đồ thị trang web G như trên. Với mỗi trang web i, ký hiệu N(i) là số liên kết đi ra từ trang web thứ i và B(i) là số trang web có liên kết đến trang i.

Khi đó hạng trang r(i) của trang web được định nghĩa như sau:

$$r(i) = \sum_{j \epsilon B(i)} \frac{r(j)}{N(j)}$$

Việc ta chia N(j) cho thấy rằng những trang liên kết tới trang i sẽ phân phối hạng của chúng cho các trang web mà chúng liên kết tới.

Các phương trình này được viết lại dưới dạng ma trận r = rP trong đó:

- $r = [r_1, r_2, \dots, r_n]$ là vector PageRank, với r_i là hạng của trang web i trong đồ thị trang web
- P là ma trận chuyển $n \times n$ với giá trị các phần tử được xác định:

$$a_{ij} = \begin{cases} \frac{1}{N_i} nếu\ có\ liên\ kết\ từ\ i\ đến\ j \\ 0\ \ ngược\ lại \end{cases}$$

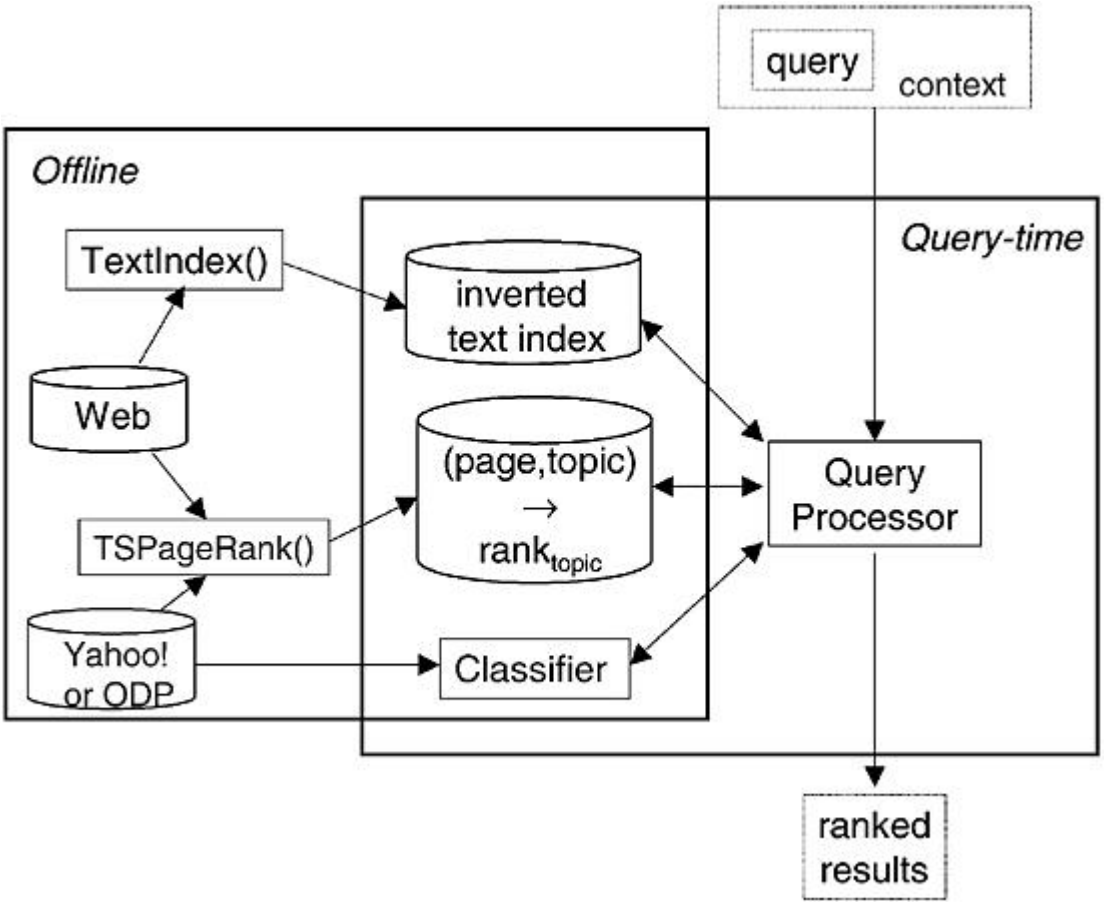
Từ đó các công thức PageRank được viết lại:

r = rP

Phương trình trên cho thấy vector PageRank r chính là vector riêng của ma trận chuyển P tương ứng với giá trị riêng $\lambda = 1$. Trong đại số tuyến tính có một số phương pháp tính vector riêng của ma trận, tuy nhiên do kích thước quá lớn của ma trận đang xét, khi thi hành người ta sử dụng phương pháp lặp để tính toán vector PageRank.[1]

2.1.2 Sơ lược về thuật toán Topic-Sensitive PageRank

Thuật toán Topic-Sensitive PageRank là một biến thể của thuật toán PageRank gốc được thiết kế để cải thiện độ chính xác của kết quả tìm kiếm bằng cách tính toán một tập hợp các vector PageRank có thiên hướng theo một chủ đề đại diện. Điều này giúp làm tăng sự chính xác trong kết qur trả về, phù hợp với ngữ cảnh cũng như nội dung mà người dùng cần.



Hình 2. Sơ đồ thuật toán Topic-Sensitive PageRank

Để cải thiện thứ hạng của kết quả truy vấn tìm kiếm thuật toán PageRank ban đầu, một vector PageRank duy nhất được tính toán, sử dụng cấu trúc liên kết, để nắm bắt tầm quan trọng tương đối của các trang Web, độc lập với bất kỳ truy vấn tìm kiếm cụ thể nào. Để tạo ra kết quả tìm kiếm chính xác hơn, bằng cách sử dụng một tập hợp các chủ đề đại diện, để nắm bắt chính xác hơn khái niệm về tầm quan trọng đối với một chủ đề cụ thể. Bằng cách sử dụng các vector PageRank có thiên vị (được tính toán trước) này để tạo điểm số tầm quan trọng cụ thể cho truy vấn cho các trang tại thời điểm truy vấn, cho thấy rằng chúng ta có thể tạo ra thứ hạng chính xác hơn so với một vector PageRank chung duy nhất. Đối với các truy vấn tìm kiếm từ khóa thông thường, chúng ta tính toán điểm PageRank theo chủ đề cho các trang đáp ứng truy vấn bằng cách sử dụng chủ đề của các từ khóa truy vấn, cho thấy rằng chúng ta có thể tạo ra thứ hạng chính xác hơn so với một vector PageRank chung duy nhất.Ta có thể mô tả các kỹ thuật để triển khai hiệu quả một hệ thống tìm kiếm quy mô lớn dựa trên lược đồ PageRank theo chủ đề. Đối với các tìm kiếm được thực hiện trong ngữ cảnh , chúng ta có thể tính toán điểm PageRank theo chủ đề bằng cách sử dụng chủ đề của ngữ cảnh mà truy vấn xuất hiện

2.2 Phương pháp.

Trong phương pháp tiếp cận Topic-Sensitive PageRank theo Taher H.Haveliwala, chúng ta tính toán trước các điểm tầm quan trọng ngoại tuyến, như với PageRank thông thường. Tuy nhiên, chúng ta cần tính toán nhiều điểm tầm quan trọng cho mỗi trang, ta tính toán một tập hợp các chỉ số đánh giá tầm quan trọng của một trang web đối với các chủ đề khác nhau. Tại thời điểm truy vấn, những điểm quan trọng được kết hợp dựa trên các chủ đề của truy vấn để tạo nên điểm PageRank tổng hợp cho các trang khớp với truy vấn. Chỉ số này có thể được sử dụng cùng với các lược đồ chấm điểm dựa trên IR khác để tạo ra thứ hạng cuối cùng cho các trang kết quả liên quan đến truy vấn.

a) Định hướng ODP

Bước đầu tiên trong phương pháp này chính là tạo một tập hợp các vector PageRank mà mỗi vector sẽ phản ánh tầm quan trọng của các trang web trong bối cảnh của một chủ đề cụ thể. Taher H.Haveliwala trong nghiên cứu của mình đã tạo ra 16 vector PageRank, mỗi vector tương ứng với một trong 16 danh mục hàng đầu của ODP. Điều này cho phép hệ thống có khả năng phân loại và đánh giá các trang web theo nhiều chủ đề khác nhau. Vậy ODP là gì?

ODP là một nguồn tài nguyên phong phú với hàng triệu URL được phân loại theo các chủ đề khác nhau.Việc sử dụng ODP giúp đảm bảo rằng các vector PageRank được tạo ra có độ chính xác và độ tinh cậy cao.

Bằng cách sử dụng vector cá nhân hóa không đồng nhất thay vì sử dụng một vector cá đồng nhất (tức là mỗi trang điều có cùng một trọng số). Điều này có nghĩa là các trang trong danh mục cụ thể sẽ nhận được trọng số cao hơn, trong khi các trang không thuộc danh mục đó sẽ nhận được trọng số thấp hơn hoặc bằng không

Để tạo ra vector cá nhân hóa cho mỗi danh mục ta có công thức sau:

$$v_{ji} = \begin{cases} \dfrac{1}{|T_j|} & i \in T_j \\ 0 & i \notin T_j \end{cases}$$

Trong đó T_j là taaoj hợp các URL trong danh mục c_j. Công thức này đảm bảo rằng các trang trong danh mục sẽ có trọng số cao hơn, từ đó ánh hưởng đến điểm số PageRank của chúng.

Qua đó giảm thiểu sự chênh lệch từ các cá nhân trong điều chỉnh điểm số. ODP được biên soạn bởi nhiều biên tập viên tình nguyện nên có ít bị ảnh hưởng bởi các yếu tố bên ngoài. Các vector bị thiên lệch này sẽ phản ánh chính xác hơn tầm quan trọng của các trang trong bối cảnh của các chủ đề cụ thể, từ đó cải thiện độ chính xác của kết quả tìm kiếm

b) Tầm quan trọng của thời gian truy vấn

Bước thứ hai trong phương pháp này được thực hiện tại thời điểm truy vấn. Cho một truy vấn q, đặt q' là ngữ cảnh của q. Nói cách khác, nếu truy vấn được đưa ra bằng cách tô sáng thuật ngữ trong u. Sử dụng mô hình ngôn ngữ unigram, với các tham số được đặt thành ước lượng hợp lý tối đa của chúng, ta cần tính xác xuất của 16 lớp ODP cấp cao nhất, có điều kiện trên q'. Đặt q' là số hạng thứ i trong truy vấn (hoặc ngữ cảnh truy vấn) q'. Sau đó, với truy vấn q, chúng ta tính cho mỗi c_j như sau:

$$P(c_j|q') = \frac{P(c_j).P(q'|c_j)}{P(q')} \propto P(c_j).\prod_i P(q'_i|c_j)$$

$P(c_j|q')$ được tính toán dễ dàng từ vector thuật ngữ $D \to j$. Đại lượng $P(c_j)$ không đơn giản như vậy. Chúng ta cần làm cho nó đồng nhất mặc dù chúng ta có thể cá nhân hóa kết quả truy vấn cho những người dùng khác nhau bằng cách thay đổi phân phối này. Nói cách khác, đối với một số người dùng k, chúng ta có thể sử dụng phân phối tiên nghiệm $P_k(c_j)$ phản ánh sở thích của người dùng k. Phương pháp này cung cấp một khuôn khổ thay thế cho cá nhân hóa dựa trên người dùng, thay vì thay đổi trực tiếp vector giảm chập $\vec{p}$ như đã được đề xuất.

Sử dụng chỉ mục văn bản, chúng tôi truy xuất các URL cho tất cả các tài liệu chứa các thuật ngữ truy vấn gốc q. Cuối cùng, chúng tôi tính toán điểm quan trọng nhạy cảm với truy vấn của mỗi URL được truy xuất này như sau. Đặt $rank_{jd}$ là hạng của tài liệu d được đưa ra bởi vectơ hạng $\overrightarrow{PR}(\alpha,\overrightarrow{v_j})$ (tức là vectơ hạng cho chủ đề c_j). Đối với tài liệu Web d, chúng tôi tính toán điểm quan trọng nhạy cảm với truy vấn s_{qd} như sau.

$$s_{qd} = \sum_j P(c_j|q').rank_{jd}$$

Phương pháp PageRank theo chủ đề có thể cho kết quả được tính toán chính xác hơn vì nó dựa trên những liên kết và nội dung trang Web. Tuy nhiên, phương pháp này cũng gặp phải những trở ngại là: việc phân chia các chủ đề có thể không đầy đủ, không bao hàm được tất cả các chủ đề; vấn đề này có thể giải quyết bằng cách tăng thêm các chủ đề nhưng việc tăng thêm các chủ đề chắc chắn sẽ làm tăng thời gian tính toán…"[1]

Kết quả được xếp hạng dựa trên điểm số tổng hợp S_{qd}.

Phép tính PageRank nhạy cảm với truy vấn ở trên có cách giải thích xác suất sau, theo mô hình "random surfer". Với w_j là hệ số được sử dụng để trọng số cho vector hạng j, với

$\sum_j w_j = 1$. Sau đó lưu ý rằng

$$\sum_j \left[w_j \overrightarrow{PR}(\alpha, \overrightarrow{v_j})\right] = \overrightarrow{PR}\left(\alpha, \sum_j \left[w_j \overrightarrow{v_j}\right]\right)$$

sự tương đồng về độ tương đương của hai bảng xếp hạng, vì nó không chỉ ra mức độ mà các thứ tự tương đối của n URL hàng đầu trong hai bảng xếp hạng tương đồng với nhau. Do đó, chúng tôi cũng sử dụng một biến thể của thước đo khoảng cách Kendall's τ. Xem [9] để thảo luận về các thước đo khoảng cách cho danh sách xếp hạng trong bối cảnh kết quả tìm kiếm trên Web. Để nhất quán với OSim, chúng ta sẽ trình bày định nghĩa của mình dưới dạng một thước đo tương đồng (thay vì khoảng cách), để các giá trị gần 1 chỉ ra sự đồng ý gần hơn. Xem xét hai danh sách URL có thứ tự một phần, τ_1 và τ_2, mỗi danh sách có độ dài n. Gọi U là hợp nhất của các URL trong τ_1 và τ_2. Nếu δ1 là U − τ_1, thì gọi τ_1' là sự mở rộng của τ_1, trong đó τ_1' chứa δ_1= xuất hiện sau tất cả các URL trong τ_1. Chúng tôi mở rộng τ_2 theo cách tương tự để thu được τ_2'. Chúng ta định nghĩa thước đo tương đồng của chúng tôi là KSim như sau:

$$KSim(\tau_1, \tau_2) = \frac{|(u,v) : \tau_1', \tau_2' \text{ agree on order of } (u,v), u \neq v|}{|U||U-1|}$$

Nói cách khác, KSim(τ_1, τ_2) là xác suất mà τ_1' và τ_2' tương đồng về thứ tự tương đối của một cặp nút khác biệt ngẫu nhiên $(u, v) \in U \times U$.

CHƯƠNG 3: ĐÁNH GIÁ KẾT QUẢ

3.1 Mô tả công việc nghiên cứu

3.1.1 Xây dựng mô hình Topic-Sensitive PageRank

Đồ án này đề xuất mô hình PageRank theo chủ đề như sau:

Ta có ma trận

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Chuyển mỗi cột A thành ma trận PageRank. Đầu tiên chún ta cần tính số luongj liên kết từ mỗi trang.

- Trang 1 có hai liên kết ra (đến trang 2 và trang 3).
- Trang 2 có hai liên kết ra (đến trang 3 và trang 4).
- Trang 3 có hai liên kết ra (đến trang 1 và trang 4).
- Trang 4 không có liên kết ra.

PageRank ma trận (A) sẽ được tính toán bằng cách chia từng phần tử của liên kết ma trận cho số lượng liên kết ra của trang tương ứng. Nếu một trang không có liên kết ra, chúng ta sẽ sử dụng một giá trị mặc định (thường là $\frac{1}{N}$ với N là số lượng trang).

PageRank ma trận

$$A = \begin{bmatrix} 0 & 0.5 & 0.5 & 0 \\ 0 & 0 & 0.5 & 0.5 \\ 0.5 & 0 & 0 & 0.5 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Tính PageRank, chúng ta sử dụng công thức

$$PR(i) = (1-d) + d. \sum_{j \epsilon L(i)} \frac{PR(j)}{C(j)}$$

Trong đó:

D là hệ số giảm (thường là 0.85).

PR(i) là PageRank của trang i.

L(i) là tập hợp các liên kết đến trang (i).

C(j) là số lượng liên kêt của trang (j).

Giả sử (d=0.85)

Lần lặp đầu tiên ta được:

Tính PR(1):

$$PR(1) = (1 - 0.85) + 0.85.\frac{PR(3)}{C(3)}$$

$$PR(1) = (1 - 0.85) + 0.85.\left(\frac{0.25}{2}\right) = 0.15 + 0.85.0.125 = 0.15 + 0.10625 = 0.25625$$

Tính PR(2):

$$PR(2) = (1 - 0.85) + 0.85.\frac{PR(1)}{C(1)}$$

$$PR(2) = (1 - 0.85) + 0.85.\left(\frac{0.25}{2}\right) = 0.15 + 0.85.0.125 = 0.15 + 0.10625 = 0.25625$$

Tính PR(3):

$$PR(3) = (1 - 0.85) + 0.85.\left(\frac{PR(1)}{C(1)} + \frac{PR(2)}{C(2)}\right)$$

$$PR(3) = (1 - 0.85) + 0.85.\left(\frac{0.25}{2} + \frac{0.25}{2}\right) = 0.15 + 0.85.0.25 = 0.15 + 0.2125 = 0.3625$$

Tính PR(4):

$$PR(4) = (1 - 0.85) + 0.85.\left(\frac{PR(2)}{C(2)} + \frac{PR(3)}{C3)}\right)$$

$$PR(4) = (1 - 0.85) + 0.85.\left(\frac{0.25}{2} + \frac{0.25}{2}\right) = 0.15 + 0.85.0.25 = 0.15 + 0.2125 = 0.3625$$

3.2 Kết quả nghiên cứu

3.2.1 Huấn luyện và đánh giá mô hình

Dưới đây là đoạn code thể hiện cho mô hình trên:

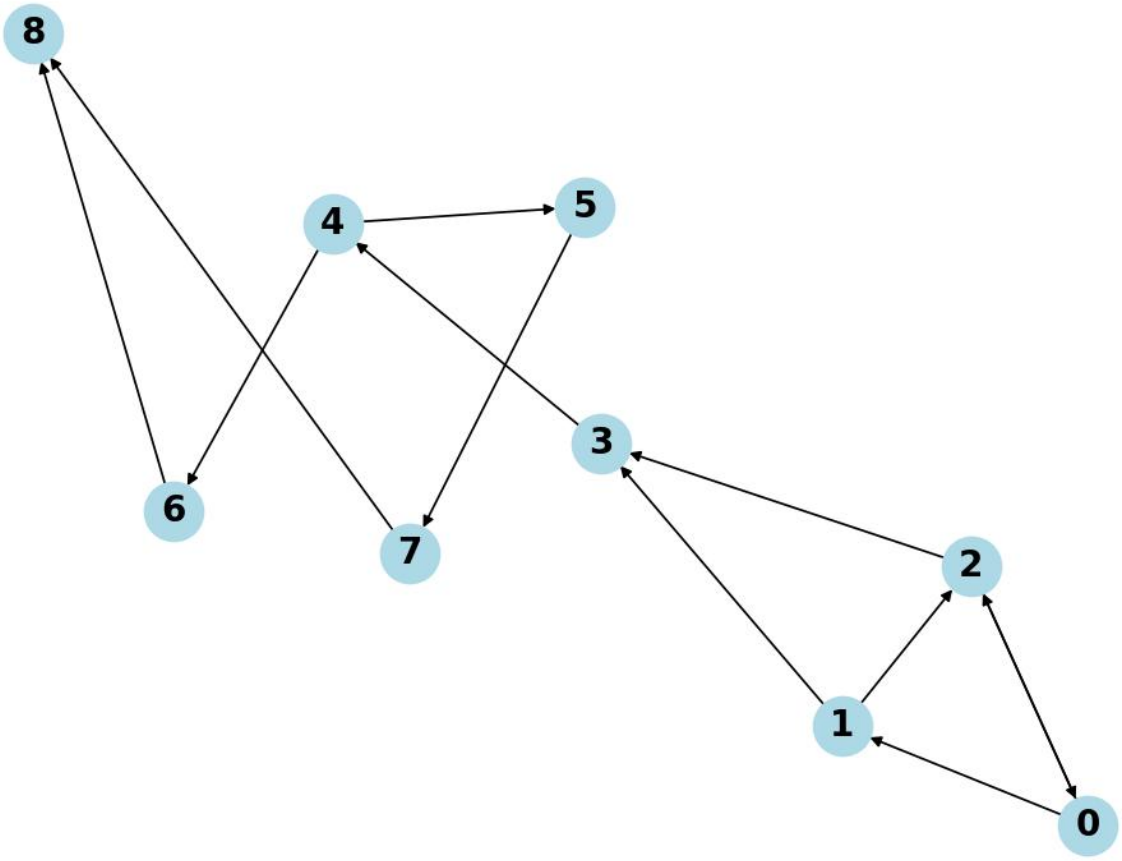
```
import numpy as np
import matplotlib.pyplot as plt
import networkx as nx

def topic_sensitive_pagerank(links, topic_vector, alpha=0.85, max_iter=100, tol=1e-6):
    n = links.shape[0]
    pagerank = np.ones(n) / n
    for _ in range(max_iter):
        new_pagerank = np.zeros(n)
        for i in range(n):
            incoming_links = np.where(links[:, i] > 0)[0]
            for j in incoming_links:
                new_pagerank[i] += pagerank[j] / np.sum(links[j])
            new_pagerank[i] = (1 - alpha) * topic_vector[i] + alpha * new_pagerank[i]
        if np.linalg.norm(new_pagerank - pagerank, 1) < tol:
            break
        pagerank = new_pagerank
```

```
    return pagerank
links = np.array([[0, 1, 1, 0, 0, 0, 0, 0, 0],
                  [0, 0, 1, 1, 0, 0, 0, 0, 0],
                  [1, 0, 0, 1, 0, 0, 0, 0, 0],
                  [0, 0, 0, 0, 1, 0, 0, 0, 0],
                  [0, 0, 0, 0, 0, 1, 1, 0, 0],
                  [0, 0, 0, 0, 0, 0, 0, 1, 0],
                  [0, 0, 0, 0, 0, 0, 0, 0, 1],
                  [0, 0, 0, 0, 0, 0, 0, 0, 1],
                  [0, 0, 0, 0, 0, 0, 0, 0, 0]])
topic_vector = np.array([0.8, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1])
pagerank_scores = topic_sensitive_pagerank(links, topic_vector)
print("Topic-Sensitive PageRank Scores:", pagerank_scores)
G = nx.from_numpy_array(links, create_using=nx.DiGraph)
plt.figure(figsize=(8, 6))
pos = nx.spring_layout(G)
nx.draw(G, pos, with_labels=True, node_size=700, node_color='lightblue', font_size=16,
font_weight='bold', arrows=True)
plt.title('Đồ thị liên kết')
plt.show()
```

3.2.2 Kết quả thực nghiệm

```
Topic-Sensitive PageRank Scores: [0.17382532 0.08887573 0.1266479  0.10659749 0.10560778 0.05988324
 0.05988324 0.06590066 0.12191607]
```

Hình 3.Các thông số đạt được



Hình 4.Đồ thị của thuật toán sau khi mô phỏng.

affirmative action	lipari
alcoholism	lyme disease
amusement parks	mutual funds
architecture	national parks
bicycling	parallel architecture
blues	recycling cans
cheese	rock climbing
citrus groves	san francisco
classical guitar	shakespeare
computer vision	stamp collecting
cruises	sushi
death valley	table tennis
field hockey	telecommuting
gardening	vintage cars
graphic design	volcano
gulf war	zen buddhism
hiv	zener
java	

Hình 5.Các truy vấn có thể được sử dụng.

	NoBias	Arts	Business	Computers	Games	Health	Home	Kids & Teens	News	Recreation	Reference	Regional	Science	Shopping	Society	Sports	World
NoBias	1																
Arts	0.09	1															
Business	0.08	0.06	1														
Computers	0.10	0.08	0.08	1													
Games	0.07	0.12	0.08	0.11	1												
Health	0.07	0.07	0.08	0.06	0.09	1											
Home	0.07	0.07	0.07	0.06	0.09	0.12	1										
Kids & Teens	0.08	0.08	0.04	0.06	0.09	0.11	0.09	1									
News	0.07	0.09	0.07	0.07	0.11	0.09	0.07	0.09	1								
Recreation	0.09	0.09	0.06	0.08	0.09	0.06	0.08	0.08	0.06	1							
Reference	0.07	0.07	0.05	0.08	0.08	0.09	0.06	0.10	0.06	0.05	1						
Regional	0.12	0.09	0.07	0.06	0.06	0.08	0.08	0.08	0.07	0.10	0.07	1					
Science	0.11	0.08	0.08	0.07	0.09	0.11	0.06	0.09	0.08	0.06	0.10	0.08	1				
Shopping	0.05	0.07	0.07	0.06	0.09	0.06	0.07	0.05	0.05	0.08	0.04	0.06	0.04	1			
Society	0.10	0.10	0.06	0.06	0.07	0.10	0.09	0.11	0.09	0.08	0.09	0.11	0.10	0.05	1		
Sports	0.07	0.09	0.07	0.07	0.13	0.09	0.10	0.08	0.10	0.10	0.07	0.09	0.07	0.09	0.07	1	
World	0.10	0.06	0.06	0.07	0.07	0.06	0.05	0.06	0.06	0.07	0.06	0.08	0.07	0.05	0.07	0.06	1

Hình 6.Biểu đồ bảng so sánh các cặp của chủ đề của Taher H. Haveliwala.

Kết quả của mô hình trên về thứ hạng của các trang như sau:

Ví dụ: Ta có 4 trang web tương đương với 4 nút trên đồ thị với chủ đề sản phẩm công nghệ, ta có:

Trang web 0:

- Trang 0 có trọng số là 0.17 vậy nên trang web 0 có nội dung liên quan nhiều đến chủ đề sản phẩm công nghệ.
- Trong đồ thị liên kết, trang 0 nhận liên kết từ trang 2 và có liên kết đến trang 1 và trang 2

Trang web 1:

- Trang web 1 có trọng số là 0.088 nên trang web có nội dung liên quan ít đến chủ đề sản phẩm công nghệ
- Trong đồ thị liên kết, trang 1 nhận liên kết từ trang 0 và có liên kết đến trang 2 và trang 3.

Trang web 2:

- Trang web 2 có trọng số là 0.1266 nên trang web có nội dung liên quan tương đối nhiều đến chủ đề.
- Trong đồ thị thì trang web nhận liên kết từ trang 0 và trang 1 và liên kết

đến trang 0 và 3.

Trang web 3 :

- Trang web có trọng số là 0.106 nên trang web có liên quan tương đối nhiều về chủ đề.
- Trong đồ thị trang web nhận liên kết từ trang 1 và trang 2 và liên kết đến trang 4.

Trang web 4:

- Trang web 4 có trọng số là 0.1056 nên trang web có liên quan tương đối nhiều về chủ đề.
- Trong đồ thị trang web nhận liên kết từ trang 3 và liên kết đến trang 5 và trang 6.

Trang web 5:

- Trang web 5 có trọng số là 0.5988 nên trang web có liên quan tương đối ít về chủ đề.
- Trong đồ thị trang web nhận liên kết từ trang 4 và liên kết đến trang 7.

Trang web 6:

- Trang web 6 có trọng số là 0.5988 nên trang web có liên quan tương đối ít về chủ đề.
- Trong đồ thị trang web nhận liên kết từ trang 4 và liên kết đến trang 8.

Trang web 7:

- Trang web 7 có trọng số là 0.0659 nên trang web có liên quan tương đối ít về chủ đề.
- Trong đồ thị trang web nhận liên kết từ trang 5 và liên kết đến trang 8.

Trang web 8:

- Trang web 8 có trọng số là 0.1219 nên trang web có liên quan tương đối nhiều về chủ đề.
- Trong đồ thị trang web nhận liên kết từ trang 6 và trang 7 và không có liên kết ra..Vì không có liên kết ra nên sẽ được phân phối lại trong đồ thị

Khi tìm kiếm về chủ đề sản phẩn công nghệ thì kết quả lần lượt là:

- Trang 0.

- Trang 2.

- Trang 8.
- Trang 3.
- Trang 4.
- Trang 1.
- Trang 7.
- Trang 5.
- Cuối cùng là trang 6.

3.3 Ưu điểm và nhược điểm của mô hình

- Ưu điểm:
    - Cải thiện độ chính xác của kết quả tìm kiếm, cho phép tính toán các điểm số tầm quang trọng của trang web dựa trên chủ đề cụ thể, giúp cung cấp kết quả tìm kiếm chính xác hơn và phù hợp hơn với nhu cầu của người dùng.
    - Nhạy cảm với ngữ cảnh, bằng cách sử dụng ngữ cảnh của truy vấn, thuật toán có thể điều chỉnh các kết quả tìm kiếm để phản ánh tố hơn yêu cầu của người dùng.
    - Giảm thiểu thiên lệch trong việc điều chỉnh điểm số.
    - Hệ thống có thể điều chỉnh các vector PageRank dựa trên các chủ đề cụ thể, cho phép nó thích ứng với các loại truy vấn khác nhau và ngữ cảnh tìm kiếm.
- Nhược điểm
    - Việc tính toán nhiều vector PageRank cho các chủ đề khác nhau có thể tốn nhều thời gian và tài nguyên, đặc biệt khi mô hình web ngày càng lớn
    - Độ chính xác của thuật toán phụ thuộc vào chất lượng của nguồn dữ liệu từ ODP. Nếu dữ liệu không chính xác hoặc không đầy đủ, kết quả tìm kiếm có thể bị ảnh hưởng.

- Việc triển khai thuật toán này có thể phức tạp hơn so với các thuật toán PageRank truyền thống, do cần phải xử lí nhiều vector và xác định ngữ cảnh cho từng truy vấn.
- Việc xác định chủ đề phù hợp cho một truy vấn có thể gặp khó khăn, đặc biệt là với các truy vấn đa nghĩa hoặc không rõ ràng. Điều này có thể dẫn đến việc sử dụng các vector PageRank không phù hợp
- Không hoàn toàn loại bỏ được spam mặc dù thuật toán có thể giảm thiểu ảnh hưởng của spam, tuy nhiên nó không hoàn toàn loại bỏ được những trang web có nội dung không liên quan nhưng vẫn có nhiều liên kết.

CHƯƠNG 4: KẾT LUẬN

Đồ án đã thực hiện giới thiệu Topic-Sensitive PageRank và mô phỏng đồ thị của thuật toán, từ đó đánh giá hiệu suất của thuật toán.

Dựa trên kết quả thực nghiệm và đánh giá, chúng tôi đã thực hiện thành công việc mô phỏng thuật toán Topic-Sensitive PageRank. Đồ thị của thuật toán đã mô phỏng lại liên kết của các trang dựa trên chủ đề tìm kiếm.

Với kết quả đạt được, chúng tôi đề xuất sử dụng mô hình Topic-Sensitive PageRank và đưa ra những cải tiến trong phương pháp tiền xử lý dữ liệu để tăng độ chính xác của thuật toán.

CHƯƠNG 5: HƯỚNG PHÁT TRIỂN

Một trong những hướng phát triển tiếp theo có là cải thiện và tăng cường chất lượng nguồn dữ liệu, kết hợp nhiều nguồn dữ liệu khác nhau ngoài ODP để tạo ra các vector PageRank nhạy cảm với chủ đề. Điều này bao gồm việc sử dụng dữ liệu từ các trang mạng xã hội, các diễn đàn hoặc các trang web chuyên ngành. Cùng với đó phát triển các phương pháp để làm sạch và chuẩn hóa dữ liệu, đảm bảo các liên kết URL và thông tin liên quan là chính xác và đáng tin cậy.

Ngoài ra, ta có thể nghiên cứu các phương pháp tối ưu hóa thuật toán nhằm giảm thiểu thời gian cũng như là tài nguyên cần thiết cho việc tính toán nhiều vector PageRank

Cuối cùng, để nâng cao khả năng các nhân hóa, cần tạo ra các hệ thống có khả năng cá nhân hóa việc tìm kiếm dựa trên sở thích và hành vi của người dùng

DANH MỤC TÀI LIỆU THAM KHẢO

1. http://www.uet.vnu.edu.vn/~thuyhq/Student_Thesis/K51_Le_Kim_Dung_Thesis.pdf
2. http://www-cs-students.stanford.edu/~taherh/papers/topic-sensitive-pagerank.pdf.
3. (2003). Topic-Sensitive PageRank: A Context-Sensitive Ranking Algorithm for Web Search. *IEEE Trans. Knowl. Data Eng.*, 15, 784-796. https://doi.org/10.1109/TKDE.2003.1208999.
4. Chen, Z., & Lang, Y. (2003). Topic-Sensitive PageRank : A Context-Sensitive RankingAlgorithm for Web Search. .
5. https://tungphat.com/page-rank-la-gi/
6. Sergey Brin, Rajeev Motwani, Larry Page, and Terry Winograd. What can you do with a web in your pocket. In Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, 1998.
7. Sergey Brin and Larry Page. The anatomy of a large-scale hypertextual web search engine. In Proceedings of the Seventh International World Wide Web Conference, 1998.
8. S. Chakrabarti, B. Dom, D. Gibson, J. Kleinberg, P. Raghavan, and S. Rajagopalan. Automatic resource compilation by analyzing hyperlink structure and associated text. In Proceedings of the Seventh International World Wide Web Conference, 1998.
9. Cynthia Dwork, Ravi Kumar, Moni Naor, and D. Sivakumar. Rank aggregation methods for the web. In Proceedings of the Tenth International World Wide Web Conference, 2001.