KHOA KỸ THUẬT VÀ CÔNG NGHỆ

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**

**BÁO CÁO TUẦN**

**THỰC TẬP ĐỒ ÁN CƠ SỞ NGÀNH**

**Tên đề tài:** Tìm hiểu và mô phỏng thuật toán Topic-Sensitive PageRank.

**Giảng viên hướng dẫn:** Nguyễn Nhứt Lam

**Thời gian thực hiện:** từ ngày 04/11/2024 đến ngày 29/12/2024

**Sinh viên thực hiện:** Nguyễn Thị Thanh Phụng

**Mã số sinh viên:** 110122143 - **Mã lớp:** DA22TTB.

**Nội dung :**

1. **Đã hoàn thành đề cương chi tiết.**
2. **Tìm hiểu về đồ thị của thuật toán.**
3. **Thuật toán PageRank** là thuật toán xếp hạng đồ thị dựa trên cấu trúc tổng quátt, chúng ta có cách để tính được chính xác thứ hạng của từng đỉnh, tuy nhiên khi số đỉnh càng lớn, việc tính toán chính xác là điều gây mất thời gian và không cần thiết.
4. **Thuật toán PageRank** ban đầu để cải thiện thứ hạng của kết quả truy vấn tìm kiếm, 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, chúng ta đề xuất tính toán một tập hợp các vector PageRank, có thiên vị 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.
5. Đồ thị trong **thuật toán Topic-Sensitive PageRank** (TSPR) được sử dụng để mô hình hóa các trang web và mối quan hệ giữa chúng qua các liên kết. Đồ thị này là nền tảng để tính toán điểm PageRank tùy theo từng chủ đề. Để hiểu rõ hơn, hãy cùng xem xét cấu trúc và đặc điểm của đồ thị này:

* **Cấu trúc đồ thị**
* **Đỉnh (Node)**: Mỗi trang web trong hệ thống được biểu diễn bằng một đỉnh. Mỗi đỉnh đại diện cho một trang, và sẽ có nhiều đỉnh tương ứng với tất cả các trang web trong hệ thống.
* **Cạnh (Edge)**: Các cạnh trong đồ thị biểu thị các liên kết giữa các trang. Một cạnh từ đỉnh A đến đỉnh B cho biết trang A có liên kết đến trang B.
* **Trọng số cạnh**: Đôi khi các liên kết có thể có trọng số để thể hiện mức độ quan trọng của liên kết, nhưng trong PageRank và TSPR, các cạnh thường không có trọng số rõ ràng; thay vào đó, mức độ ảnh hưởng được tính dựa trên số lượng liên kết của trang nguồn và số lượng trang mà nó liên kết đến.
* **Đồ thị phân loại theo chủ đề**
* Đối với TSPR, đồ thị được phân loại theo các chủ đề khác nhau, và mỗi chủ đề sẽ có một *vector PageRank* riêng, chỉ ra mức độ quan trọng của các trang đối với chủ đề đó. Để thực hiện điều này, mỗi chủ đề sẽ có một "điểm khởi đầu" khác nhau trong đồ thị, nhằm đảm bảo rằng thuật toán ưu tiên các trang web liên quan hơn đến chủ đề đó.
* **Cách tính toán trên đồ thị của TSPR**
* **Tính nhiều vector PageRank**: Đồ thị sẽ được phân tích nhiều lần cho từng chủ đề riêng biệt. Mỗi lần chạy thuật toán PageRank, thuật toán sẽ điều chỉnh để chỉ ưu tiên các trang web liên quan đến chủ đề được chọn.
* **Tính điểm chủ đề cho mỗi trang**: Với mỗi trang web, TSPR sẽ lưu lại điểm PageRank từ nhiều vector chủ đề khác nhau, để khi có truy vấn theo chủ đề cụ thể, hệ thống có thể nhanh chóng truy cập vào các điểm này để xếp hạng.
* **Ví dụ minh họa đồ thị trong TSPR**

Giả sử có một đồ thị nhỏ với các trang web như sau:

**Trang A** có liên kết đến **Trang B** và **Trang C**

**Trang B** có liên kết đến **Trang C** và **Trang D**

**Trang C** có liên kết đến **Trang A**

**Trang D** có liên kết đến **Trang C**

Trong TSPR, mỗi trang sẽ có một điểm PageRank khác nhau tùy theo chủ đề của truy vấn. Chẳng hạn, nếu người dùng tìm kiếm thông tin về "công nghệ", thuật toán sẽ ưu tiên các trang nào có nội dung công nghệ cao hơn trong đồ thị khi tính toán điểm PageRank.

* **Đặc điểm đồ thị của TSPR so với PageRank thông thường**
* **Nhiều vector PageRank**: Thay vì một vector điểm duy nhất, đồ thị của TSPR sẽ chứa nhiều vector PageRank, mỗi vector tương ứng với một chủ đề.
* **Phân tán điểm khởi tạo theo chủ đề**: Mỗi chủ đề có một điểm khởi tạo khác nhau, giúp kết quả phản ánh chính xác mức độ liên quan của trang đến chủ đề đó.

Như vậy, đồ thị của TSPR là một biến thể phức tạp hơn so với đồ thị trong PageRank gốc, bởi vì nó không chỉ xếp hạng dựa trên cấu trúc liên kết mà còn tính đến sự liên quan về chủ đề. Điều này giúp cải thiện khả năng xếp hạng và tìm kiếm theo ngữ cảnh của các công cụ tìm kiếm.

1. Để tính toán điểm xếp hạng của các trang trong đồ thị của **thuật toán Topic-Sensitive PageRank** (TSPR), ta thực hiện một số bước tương tự như trong PageRank gốc nhưng với sự tùy chỉnh theo từng chủ đề. Sau đây là các bước cơ bản để tính toán trên đồ thị của TSPR:

* **Xây dựng đồ thị**
* **Đỉnh (Node)**: Mỗi trang web trong tập dữ liệu sẽ được biểu diễn bằng một đỉnh.
* **Cạnh (Edge)**: Mỗi liên kết từ trang này đến trang khác sẽ được biểu diễn bằng một cạnh có hướng. Nếu trang A liên kết đến trang B, ta vẽ một cạnh từ A đến B.
* **Xác định các chủ đề (Topic)**
* **Xác định các chủ đề chính** mà hệ thống sẽ sử dụng để xếp hạng, ví dụ: thể thao, công nghệ, du lịch, sức khỏe, v.v.
* **Gán chủ đề cho các trang web**: Hệ thống có thể gán chủ đề dựa trên nội dung của các trang hoặc dựa trên danh mục mà chúng thuộc về. Những trang có nội dung phù hợp với chủ đề nào sẽ được ưu tiên trong tính toán của chủ đề đó.
* **Tạo vector khởi tạo theo từng chủ đề.**
* Để mỗi chủ đề có một trọng tâm riêng, ta cần khởi tạo các *vector chủ đề*. Mỗi vector chủ đề là một vector xác suất, với các giá trị lớn hơn cho các trang thuộc về chủ đề đó.
* Ví dụ, nếu đang tính toán điểm PageRank cho chủ đề "công nghệ", vector khởi tạo sẽ có giá trị cao hơn cho các trang về công nghệ so với các trang thuộc chủ đề khác.
* **Tính toán điểm PageRank theo từng chủ đề**
* Với mỗi chủ đề, ta chạy thuật toán PageRank dựa trên vector khởi tạo của chủ đề đó.
* **Thuật toán PageRank** được tính bằng cách lặp lại công thức:
* : Điểm PageRank của trang .
* d: Hệ số điều chỉnh, thường lấy giá trị khoảng 0.85, dùng để điều chỉnh tỉ lệ nhảy ngẫu nhiên.
* : Tập các trang liên kết .
* : Số lượng liên kết của trang .
* **Điều chỉnh theo vector chủ đề**: Khi tính toán PageRank theo từng chủ đề, thuật toán sẽ điều chỉnh giá trị PageRank ban đầu của các trang theo vector khởi tạo tương ứng.
* **Kết hợp điểm PageRank từ các vector chủ đề.**
* Khi có một truy vấn từ người dùng, hệ thống sẽ xác định các chủ đề liên quan đến truy vấn đó và kết hợp điểm PageRank từ các vector chủ đề phù hợp.
* **Tính điểm xếp hạng tổng hợp**: Giả sử, điểm PageRank của trang cho chủ đề "công nghệ" là và cho chủ đề "sức khỏe" là . Khi đó, điểm xếp hạng của trang có thể là tổ hợp tuyến tính của các điểm này, ví dụ:

…

Trong đó, là trọng số của từng chủ đề, có thể được xác định dựa trên truy vấn của người dùng.

* **Cập nhật điểm PageRank và lặp lại.**
* **Lặp lại**: Quá trình tính toán PageRank được lặp lại cho đến khi các giá trị hội tụ, nghĩa là không thay đổi nhiều giữa các lần lặp.
* **Điểm hội tụ**: Sau khi quá trình hội tụ, ta sẽ có một tập hợp điểm PageRank cho mỗi chủ đề, thể hiện mức độ quan trọng của các trang trong từng lĩnh vực.

1. **Tìm hiểu và tải về phần mềm mô phỏng thuật toán.**
2. Python là gì?

Python là một [ngôn ngữ lập trình bậc cao](https://topdev.vn/blog/ngon-ngu-lap-trinh-bac-cao/), mã nguồn mở và đa nền tảng. Python được sử dụng rộng rãi để phát triển các ứng dụng web, phát triển phần mềm, khoa học dữ liệu và máy học (ML).

1. Lợi ích của Python:

Các nhà phát triển có thể dễ dàng đọc và hiểu một chương trình Python vì ngôn ngữ này có cú pháp cơ bản giống tiếng Anh.

Python giúp cải thiện năng suất làm việc của các nhà phát triển vì so với những ngôn ngữ khác, họ có thể sử dụng ít dòng mã hơn để viết một chương trình Python.

Python có một thư viện tiêu chuẩn lớn, chứa nhiều dòng mã có thể tái sử dụng cho hầu hết mọi tác vụ. Nhờ đó, các nhà phát triển sẽ không cần phải viết mã từ đầu.

Các nhà phát triển có thể dễ dàng sử dụng Python với các ngôn ngữ lập trình phổ biến khác như Java, C và C++.

Cộng đồng Python tích cực hoạt động bao gồm hàng triệu nhà phát triển nhiệt tình hỗ trợ trên toàn thế giới. Nếu gặp phải vấn đề, bạn sẽ có thể nhận được sự hỗ trợ nhanh chóng từ cộng đồng.

Trên Internet có rất nhiều tài nguyên hữu ích nếu bạn muốn học Python. Ví dụ: bạn có thể dễ dàng tìm thấy video, chỉ dẫn, tài liệu và hướng dẫn dành cho nhà phát triển.

Python có thể được sử dụng trên nhiều hệ điều hành máy tính khác nhau, chẳng hạn như Windows, macOS, Linux và Unix.