1. **Giới thiệu**

*Query Log Mining* là một phân nhánh của một ngành khoa học tổng quát hơn là *Web Analytics.* Thật ra có thể coi nó là một trường hợp của khai phá sử dụng web. Theo Web Analytics Association, *“Web Analytics là một cách đo lường, tập hợp, phân tích và báo cáo dữ liệu Internet cho mục đích hiểu và tối ưu hóa sử dụng Web”.*

Cụ thể hơn, khai phá query log có liên quan đến tất cả các kỹ thuật dùng để trích xuất các thông tin thú vị từ query log của bộ máy tìm kiếm với mục đích cải thiện cả hiệu quả lẫn hiệu năng của dịch vụ tìm kiếm trên mạng.

Khai phá query log không chỉ liên quan đến dịch vụ tìm kiếm (nơi xuất phát của hầu hết các câu truy vấn) mà còn với những dịch vụ tổng quát hơn như quảng cáo dựa vào tìm kiếm, hay web marketing.

* 1. **Bộ máy tìm kiếm web (Web search engines)**

Các hệ thống tương đương với các bộ máy tìm kiếm ngày nay bắt đầu xuất hiện từ năm 1994. Có thể kể đến như: *WWWW (World Wide Web Worm), AliWeb, AltaVista, Excite, Lycos, Yahoo!, Google, ASK, MSN*. Ngày nay tìm kiếm được coi là một trong những ứng dụng hữu ích nhất trên mạng internet.

Trong bài báo tổng kết lại những thách thức trong việc thiết kế các bộ máy tìm kiếm hiện đại, *Baeza-Yates et al* đã nói:

*“Thách thức chính là thiết kế ra một hệ thống phân tán ở quy mô lớn có khả năng* ***thỏa mãn những yêu cầu của người dùng*** *và đồng thời đảm bảo truy vấn phải* ***sử dụng tài nguyên một cách hiệu quả nhất****, nghĩa là phải giảm chi phí cho mỗi câu truy vấn”.*

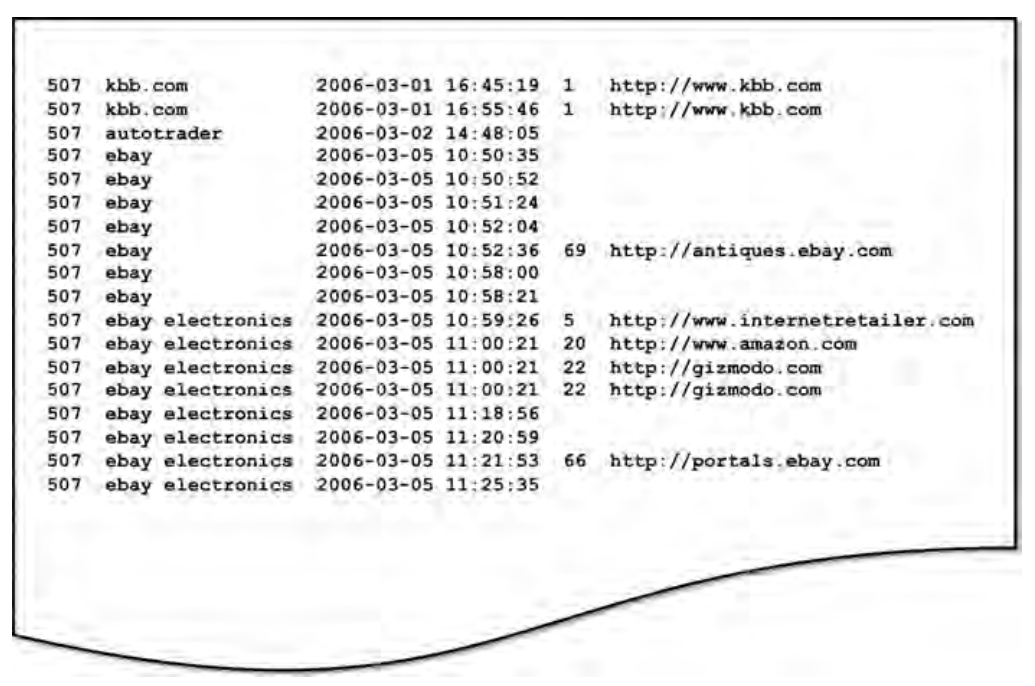
Do vậy, hai vấn đề được chỉ ra của dịch vụ tìm kiếm là: (i) chất lượng của kết quả trả về và (ii) tốc độ nhận được kết quả trả về.

Bộ máy tìm kiếm web là một dạng của một hệ thống Information Retrieval (IR). Hệ thống IR ra đời từ đầu những năm 1960 để đáp ứng hai nhu cầu chính. Thứ nhất là cho phép người dùng tìm kiếm trong các thư viện số. Thứ hai là tìm kiếm dữ liệu trong thư viện số cá nhân.

Hệ thống IR là một phần mềm với nhiệm vụ chính là trả lại một danh sách các tài liệu là kết quả của một câu truy vấn. Điều này khiến nó có vẻ tương tự với một hệ cơ sở dữ liệu (DB system). Tuy nhiên sự khác nhau cơ bản nhất giữa hệ DB và IR là hệ DB trả lại kết quả chính xác với câu truy vấn của người dùng, trong khi đó hệ IR sử dụng câu truy vấn là ngôn ngữ tự nhiên khiến cho việc trả lại một kết quả hoàn hảo là điều bất khả thi, mỗi câu truy vấn có thể có ý nghĩa khác nhau với mỗi người dùng khác nhau. Lấy ví dụ đơn giản, câu truy vấn có từ *“meta”,* vậy *meta* nghĩa là gì? Ký tự *meta* hay phím *meta* trên bàn phím máy tính, hoặc trong tiếng Tây Ban Nha thì *meta* nghĩa là bàn thắng (*goal*).

Mở rộng ra, bộ máy tìm kiếm web là một hệ thống IR ở quy mô rất lớn (*very large scale*). Điểm khác biệt với các hệ thống IR quy mô nhỏ là nó phụ thuộc vào một lượng lớn thông tin được lưu trữ trong *query log*.

Một trong những cách để nâng cao trải nghiệm của việc tìm kiếm thông tin đó là khai phá kiến thức từ những câu truy vấn cũ của người dùng. Một query log điển hình sẽ chứa thông tin về người dùng, các câu truy vấn, kết quả được người dùng chọn, ... Những kiến thức này có thể được trích chọn để nâng cao chất lượng (hiệu quả và hiệu năng) của hệ thống. Hình 1.1 cho ta thấy một đoạn query log của AOL. Mỗi câu truy vấn có năm trường thông tin: user id, câu truy vấn, thời điểm truy vấn, thứ hạng của kết quả được người dùng chọn và địa chỉ URL của kết quả đó.



*H1.1 Một đoạn query log của AOL*

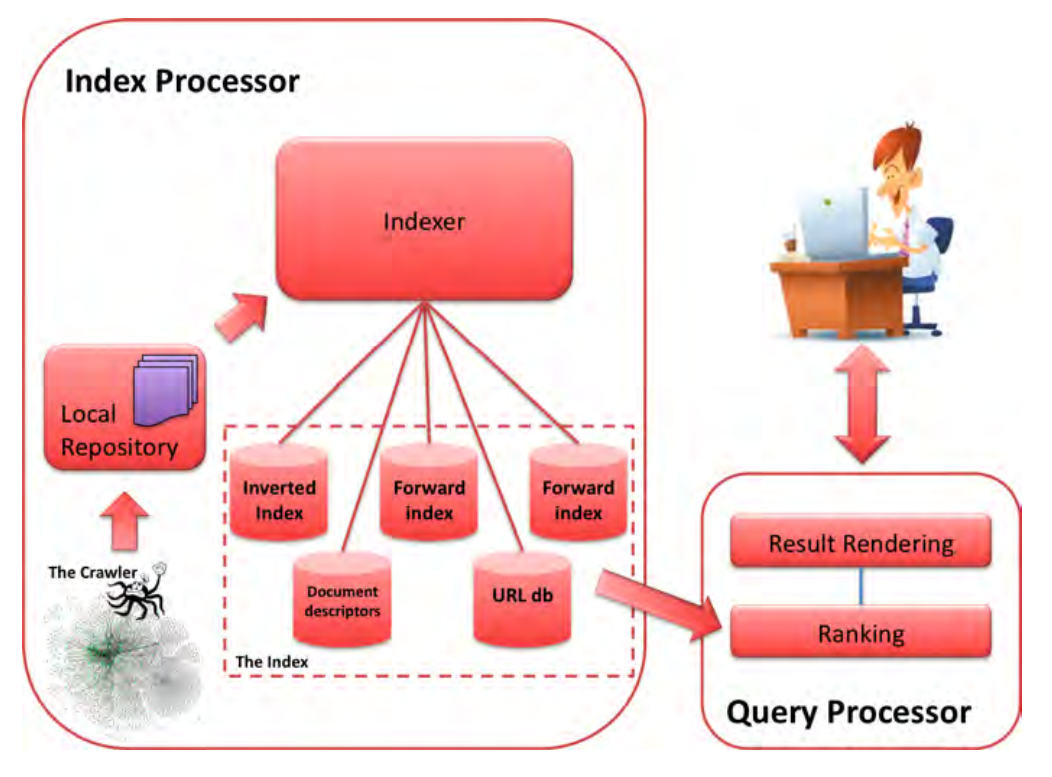
Trong bài báo này, chúng ta sẽ nghiên cứu một vài kỹ thuật làm việc với query log và cách dùng chúng để cải thiện hoạt động của bộ máy tìm kiếm. Chúng ta sẽ tổng kết lại các kết quả cơ bản liên quan đến query log: phân tích, kỹ thuật trích xuất thông tin, các kết quả thú vị nhất, các ứng dụng hữu ích nhất, các vấn đề mở và khả năng mở rộng.

* 1. **Phác họa kiến trúc của Web Search Engine**

Bộ máy tìm kiếm là một trong những phần mềm phức tạp nhất mà một công ty có thể phát triển. Với hàng chục module phụ thuộc lẫn nhau, nó được coi là một trong những thử thách khó khăn nhất trong thế giới công nghệ ngày nay.

Khi người dùng nhập vào một câu truy vấn, trình duyệt sẽ tạo ra một URL (ví dụ <http://www.google.com/search?q=foundations+trend+IR>). Sau đó nó sẽ gửi yêu cầu đến một máy chủ DNS để ánh xạ từ tên miền của bộ máy tìm kiếm (ví dụ *www.google.com*) thành địa chỉ IP của một trung tâm dữ liệu tìm kiếm. Quá trình ánh xạ này phụ thuộc vào những yếu tố sau: độ sẵn sàng, khoảng cách về địa lý và khả năng chịu tải của máy chủ DNS. Trình duyệt sau đó sẽ gửi một HTTP request đến IP này, do vậy câu truy vấn sẽ hoàn toàn được xử lý nội bộ trong trung tâm dữ liệu của bộ máy tìm kiếm. Sau khi đã xử lý xong, kết quả sẽ được trả lại cho người dùng dưới dạng một trang HTML.

Hình 1.2 chỉ ra các cách các thành phần chính của bộ máy tìm kiếm kết nối với nhau.

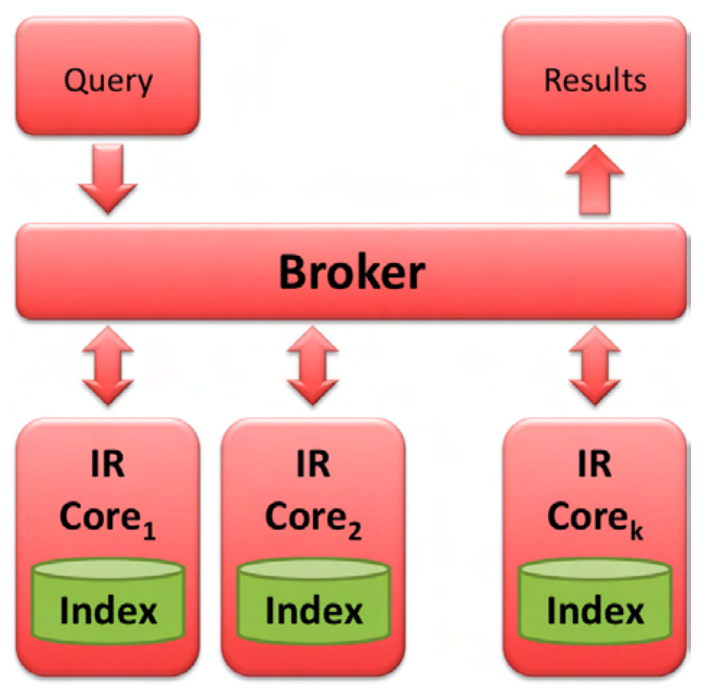


*H1.2 Cấu trúc điển hình của một web search engine*

Bộ máy tìm kiếm lấy dữ liệu từ các nguồn khác nhau: web (hầu hết), kho hình ảnh, kho video (ví dụ: Flickr hay Youtube). Cụ thể trong trường hợp nguồn dữ liệu là web, một crawler sẽ duyệt qua lần lượt các trang web theo các liên kết có trong trang web đó để tìm kiếm các nội dung mới, phát hiện và cập nhật các nội dung cũ. Crawler lưu dữ liệu các trang web đã được duyệt này vào một kho chứa nội dung (được gọi là *document cache*) và cấu trúc (đồ thị thể hiện cách các trang web được kết nối với nhau, dùng để tính thứ hạng các trang web). Trong các hệ thống hiện đại ngày nay, crawler chạy liên tục cập nhật nội dung document cache.

Nội dung được đánh chỉ mục (*index*) để tối ưu cho thao tác tìm kiếm. Chỉ mục (được xây dựng bởi bộ tạo chỉ mục *Indexer*) được chia thành nhiều loại khác nhau. Cấu trúc của một chỉ mục sẽ được nói đến ở phần 5 khi chúng ta xem xét một vài kỹ thuật để tối ưu hóa cách phân bố chỉ mục.

Trong thế giới của các bộ máy tìm kiếm thực, chỉ mục được phân bố vào các *query server* và được điều phối bởi một *broker*. Broker, sau khi nhận được một truy vấn từ người dùng, sẽ phân phối truy vấn đó đến các query server. Các server này sẽ lấy ra các document có liên quan, tính toán điểm, xếp hạng các kết quả và trả lại chúng cho broker, từ đó broker xây dựng nên trang kết quả và gửi trả lại cho người dùng. Hình 1.3 chỉ ra quá trình tương tác giữa query server và broker.



*H1.3 Cấu trúc điển hình của một hệ thống tìm kiếm phân tán*

Broker là nơi nhận các câu truy vấn và lưu trữ chúng trong query log. Do vậy đây là nơi xây dựng module khai phá query log.

* + 1. **Chỉ mục**

Chỉ mục ngược (*Inverted File index*) trên một tập hợp các trang web bao gồm nhiều thành phần có liên kết với nhau. Thành phần chính là *lexicon*: một danh sách các từ khóa (*index term*) xuất hiện trong tập hợp đó, tương ứng với mỗi từ khóa trong lexicon là một danh sách ngược (*inverted list)*. Mỗi inverted list lại chứa một tập các *posting*. Mỗi posting chứa thông tin về sự xuất hiện của từ khóa trong một tài liệu. Đề đơn giản hóa vấn đề, chúng ta coi mỗi posting chỉ chứa định danh của tài liệu (*DocID*) nơi từ khóa xuất hiện, mặc dù thực tế posting còn chứa các thông tin dùng để xếp hạng tài liệu như tần suất xuất hiện của các từ khóa trong tài liệu, …

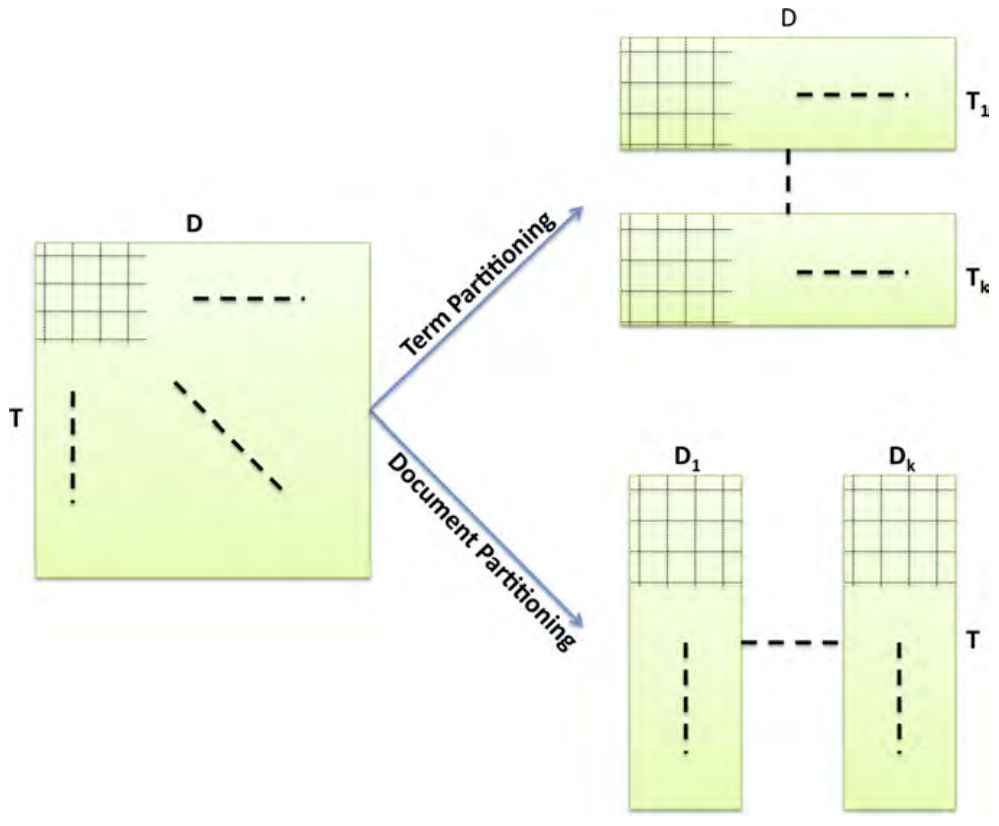
Inverted File index là một cấu trúc dữ liệu điển hình dùng để đánh chỉ mục cho web. Có ba lý do chính cho điều này. Thứ nhất, inverted file index cho phép thao tác một cách hiệu quả với các câu truy vấn trên một tập hợp lớn các dữ liệu web. Thực tế nó làm việc rất tốt với truy vấn web thông thường, nơi chỉ dùng vài từ khóa cho mỗi lượt tìm kiếm. Thứ hai, inverted file index có thể dễ dàng nén lại để giảm không gian lưu trữ. Thứ ba, inverted file index có thể dễ dàng được xây dựng dựa trên một giải thuật sắp xếp với độ phức tạp tính toán ngang với thuật toán sắp xếp đó.

Truy vấn câu trả lời sử dụng inverted file là một nhiệm vụ khá đơn giản. Chúng ta có thể minh họa điều này bởi phép toán AND. Cho một câu truy vấn gồm hai từ khóa (), quá trình giải câu truy vấn này bắt đầu bằng việc tìm kiếm và trong lexicon để cho ra các inverted list tương ứng và . Tập kết quả được xây dựng bằng cách giao giữa hai danh sách này, chính là những tài liệu có chứa cả hai từ khóa. Trong lúc thực hiện quá trình này, một hàm tính điểm sẽ được sử dụng để tính thứ hạng cho tài liệu. Top kết quả được chọn (thường là 10 kết quả có hạng cao nhất) sẽ lần lượt trả về cho người dùng. Quá trình truy vấn có thể được hoàn thành theo hai cách: *Document-At-A-Time* (DAAT) danh sách các tài liệu cho mỗi từ khóa được quét đồng thời hoặc *Term-At-A-Time* (TAAT) mỗi từ khóa được xem xét quét riêng biệt.

Một tính năng quan trọng của inverted file index đó là nó có thể dễ dàng được phân đoạn. Với một hệ thống tìm kiếm phân tán điển hình: chỉ mục có thể được phân phối trong các nút khác nhau để nâng cao thông lượng của hệ thống (số lượng truy vấn có thể trả lời được trong mỗi giây). Do vậy hai chiến thuật phân đoạn có thể được áp dụng.

Cách tiếp cận thứ nhất yêu cầu phân đoạn ngang toàn bộ inverted index trong lexicon, do đó mỗi index server sẽ chứa các inverted list liên kết với một tập con các từ khóa. Phương pháp này được gọi là *term partitioning* hay *global inverted files*. Cách tiếp cận thứ hai, được gọi là *document partitioning* hay *local inverted files*, yêu cầu mỗi index server chịu trách nhiệm với một tập con các tài liệu (phân đoạn dọc inverted index). Hình 1.4 minh họa hai cách phân đoạn trên.

Một trong những lợi ích của việc phân đoạn dọc đó là dễ dàng cho việc cập nhật nội dung tài liệu. Thực tế, một tài liệu mới có thể dễ dàng chèn vào một partition mà không làm ảnh hưởng đến partition khác.



*H1.4 Hai cách phân đoạn inverted index. Các hàng của ma trận là các từ khóa, các cột đại diện cho danh sách các posting*

Trong tài liệu này chúng ta sẽ không nhìn thấy cấu trúc thực của bất kỳ một bộ máy tìm kiếm nào. Những thông tin như vậy được coi là cực kỳ tuyệt mật và các công ty tìm kiếm sẽ không bao giờ tiết lộ chúng.

* 1. **Sự thật thú vị về các câu truy vấn**

Vì lý do về mặt thương mại, việc truy cập đến query log không phải là chuyện dễ dàng. Một số query log đã từng được công bố ra công chúng, đó là query log của AltaVista, TodoBR và AOL. Hình 1.5 cho ta thấy 250 câu truy vấn phổ biến nhất trong query log của AOL.



*H1.5 Đám mây 250 từ khóa truy vấn xuất hiện nhiều nhất trong query log của AOL*

Trong thế giới ngày này, con người rất dễ bị stress. Stress có thể dẫn đến những câu truy vấn khá ngớ ngẩn, như người dùng #427326 vào 2006-04-21 21:16:51 với từ khóa *“where is my computer”*. Truy vấn này trả về xấp xỉ 200,000 kết quả!!!

Các bộ máy tìm kiếm thường công bố danh sách những câu truy vấn thú vị nhất. Với Google danh sách này được gọi là Zeitgeist:

*“Một bản ghi các câu truy vấn thú vị được người dùng hỏi nhiều lần, trong một tên miền tìm kiếm quốc gia hoặc tại Google.com, có thể tiết lộ một chút về điều kiện sống của con người tại đây.”*

Zeitgeist không tiết lộ những câu truy vấn xuất hiện nhiều nhất mà chỉ những câu truy vấn có sự tăng lên bất thường trong cộng đồng. Ví dụ, vào cuối năm 2007, Zeitgeist của Ý xếp hạng *“federico calzolari”* là câu truy vấn “lạm phát” nhất. Nhiều người tờ báo ở Ý bắt đầu tìm hiểu xem ai là người có liên hệ với câu truy vấn này. Kết quả là một nghiên cứu sinh ở Pisa, người đã định kỳ truy vấn Google tên của mình. Kết quả là một sự tăng trưởng bất thường từ khóa này và cuối cùng nó đã xuất hiện trong Zeitgeist. Nhiều người cho rằng Federico Calzolari đã hack thuật toán xếp hạng của Google.

Điều thú vị ở trên có thể ngầm định rằng người ta có thể xác định danh tính của người dùng thông qua phiên truy vấn của họ. Tuy nhiên ở các phần sau ta có thể thấy đây là một thách thức không hề nhỏ. Mục đích của phần này là giúp chúng ta nhận ra: (i) sự đa dạng thông tin của các câu truy vấn và (ii) theo lý thuyết, các thông tin này có thể xuất phát từ một người dùng duy nhất.

* 1. **Các vấn đề quyền riêng tư trong việc khai phá query log**

Vụ bê bối nổi tiếng nhất liên quan đến quyền riêng tư của query log xảy ra vào năm 2006 với AOL. AOL đã thống kê 20 triệu câu truy vấn của 650,000 người dùng và công bố cơ sở dữ liệu này cho mục đích nghiên cứu. Mặc dù trường username trong các bản ghi này là số, tuy nhiên nếu người dùng nhập vào một vài thông tin có tính cá nhân thì dựa vào các thông tin này ta có thế định danh được người dùng đó và xem các truy vấn khác mà họ đã thực hiện. AOL đã chịu rất nhiều chỉ trích cho việc này và họ đã sa thải hai nhân viên có liên quan.

Nhiều bộ máy tìm kiếm giải quyết vấn đề này bằng cách đơn giản là không công bố log. Cách giải quyết này có thực sự tốt? Với đa số người dùng, câu trả lời là có. Một ý kiến khác, Judit Bar-Ilan đã nói:

*“[…] các kết quả thú vị có thể được lấy ra từ query log mà không gây nguy hiểm cho quyền riêng tư của người dùng.”*

Trong khi Bar-Ilan nói rằng có thể tinh chỉnh query log để ngăn chặn thông tin riêng tư bị khai thác, Jones et al lại chỉ ra rằng ngay cả các query log được coi là bảo mật nhất cũng có khả năng chứa các thông tin nhạy cảm có độ rủi ro cao.

Gần đây *ask.com* cho phép người dùng từ chối lưu trữ lại dữ liệu tìm kiếm. Google, Yahoo và Microsoft cũng liên tục hỏi quyền người dùng về việc lưu trữ lại dữ liệu truy vấn. Vậy đâu là cách hành xử đúng? Điều này phụ thuộc vào chính sách của bộ máy tìm kiếm, chúng ta sẽ không đi sâu vào chi tiết này.

Bài báo này sẽ trình bày những kết quả và tiến bộ mới nhất trong việc sử dụng query log như là nguồn thông tin. Thật không may, như đã trình bày ở trên, nguồn thông tin chính này hầu hết được giữ lại bởi các công ty làm bộ máy tìm kiếm vì nhiều lý do (vấn đề quyền riêng tư, …) và do vậy nó rất hạn chế được công bố ra ngoài, kể cả cho các nhà nghiên cứu. Vì vậy trong bài báo này các đánh giá thí nghiệm chủ yếu dựa vào kết quả của các bài báo khác.

Cuộc du hành vào thế giới kỳ diệu của các câu truy vấn bắt đầu …