ĐẠI HỌC QUỐC GIA HÀ NỘI

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

Nguyễn Song Hà

HỆ THỐNG TƯ VẤN WEBSITE CHO MÁY TÌM KIẾM DỰA TRÊN KHAI PHÁ QUERY LOG

KHOÁ LUẬN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY

Ngành: Công nghệ Thông tin

Hà Nội - 2009

ĐẠI HỌC QUỐC GIA HÀ NỘI

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

Nguyễn Song Hà

HỆ THỐNG TƯ VẤN WEBSITE CHO MÁY TÌM KIẾM DỰA TRÊN KHAI PHÁ QUERY LOG

KHOÁ LUẬN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY

Ngành: Công nghệ Thông tin

Cán bộ hướng dẫn: **PGS.TS Hà Quang Thuỵ**

Cán bộ đồng hướng dẫn: **Th.S Nguyễn Thu Trang**

Hà Nội - 2009

Lời cảm ơn

Trước tiên, tôi xin gửi lời cảm ơn và lòng biết ơn sâu sắc nhất tới Phó Giáo sư Tiến sĩ Hà Quang Thụy và Thạc sỹ Nguyễn Thu Trang, người đã tận tình chỉ bảo và hướng dẫn tôi trong suốt quá trình thực hiện khoá luận tốt nghiệp.

Tôi chân thành cảm ơn các thầy, cô đã tạo cho tôi những điều kiện thuận lợi để học tập và nghiên cứu tại trường Đại Học Công Nghệ.

Tôi cũng xin gửi lời cảm ơn tới các anh chị và các bạn sinh viên trong nhóm "Khai phá dữ liệu" đã giúp tôi rất nhiều trong việc thu thập và xử lý dữ liệu.

Cuối cùng, tôi muốn gửi lời cảm vô hạn tới gia đình và bạn bè, những người thân yêu luôn bên cạnh và động viên tôi trong suốt quá trình thực hiện khóa luận tốt nghiệp.

Tôi xin chân thành cảm ơn !

Sinh viên

Nguyễn Song Hà

Tóm tắt nội dung

Hệ tư vấn (recommender system) đã trở thành một trong những lĩnh vực nghiên cứu quan trọng kể từ khi bài báo đầu tiên về lọc cộng tác (collaborative filtering) xuất hiện vào giữa những năm 1990. Hiện nay, sự quan tâm đối với hệ tư vấn đang rất cao vì sự cần thiết của những ứng dụng có thể giúp người dùng xử lý với tình trạng quá tải thông tin & đưa ra những nội dung hoặc lời khuyên phù hợp cho từng cá nhân. Một vài ứng dụng nổi tiếng như: hệ tư vấn sách, CDs của Amazon.com, hệ tư vấn phim của MovieLens… Nhưng so với sách, phim… thì số lượng website bùng nổ mỗi ngày còn lớn hơn rất nhiều. Khóa luận đề xuất phương pháp xây dựng một hệ thống tư vấn website dựa trên việc khai phá query logs của máy tìm kiếm. Các website được tư vấn là kết quả có được dựa trên phân tích những lựa chọn của hàng nghìn người dùng trước đó. Thực nghiệm ban đầu của hệ thống cho kết quả khá tốt.

Mục lục

Danh sách bảng

Danh sách hình vẽ

Lời mở đầu

Trong thời đại bùng nổ thông tin, khi người dùng thường bị ngập trong khối lượng thông tin khổng lồ thì hệ tư vấn ngày càng có vai trò quan trọng. Có khá nhiều hệ thống tư vấn nổi tiếng, nhưng hầu hết chỉ tập trung vào một số lĩnh vực hẹp như: sách, phim, ca nhạc…Các hệ thống đó thường dựa vào đánh giá của các chuyên gia (reviewer) với những bộ tiêu chuẩn cụ thể, hoặc dựa trên việc chấm điểm sản phẩm bởi người dùng. Nhưng các lĩnh vực trong cuộc sống rất phong phú, số lượng chủng loại sản phẩm rất lớn. Để có hệ tư vấn dựa trên chuyên gia hay những bộ tiêu chuẩn cụ thể như vậy trên mọi lĩnh vực, mọi sản phẩm là điều không thể.

Khi cần tìm thông tin về một sản phẩm nào đó, giải pháp được hầu hết người dùng sử dụng là đưa câu hỏi vào máy tìm kiếm thay vì tìm đến những website/forum chuyên ngành. Tuy nhiên, máy tìm kiếm không phải lúc nào cũng hiệu quả. Máy tìm kiếm chỉ có thể đưa ra một danh sách các lựa chọn (có thể lên đến hàng triệu) chứ không thể nói được lựa chọn nào là tốt nhất.

Ví dụ, một du khách lần đầu đến Hà Nội, muốn tìm khách sạn bằng query: *"hanoi hotel",* sẽ nhận được từ Google gần hai triệu kết quả trả về. Hầu hết mọi khách sạn trong danh sách kết quả đều xa lạ và tự quảng cáo mình là tốt nhất, làm cho du khách bối rối trong biển thông tin. Không thể có thời gian để tìm hiểu lại về từng khách sạn (dù chỉ là trong 10-20 kết quả đầu); người khách cần lời khuyên cho trường hợp này. Những nhu cầu như vậy có thể bắt gặp rất nhiều trong cuộc sống hàng ngày, ngay cả khi người ta tìm kiếm những sản phẩm đơn giản như một chiếc đầu DVD, một hãng sơn, một công ty taxi …, mà vì không có thông tin nên với họ mọi thương hiệu đều như nhau. Cần có một phương pháp có thể đưa ra gợi ý, tư vấn cho người dùng đủ tốt để áp dụng cho những chủ đề rất đa dạng của cuộc sống.

Một giải pháp rất tốt và hiệu quả là gợi ý dựa trên chính kinh nghiệm của những người đã từng tìm về chủ đề này trước đó. Những thông tin được lưu lại trong log của máy tìm kiếm sẽ cho biết những người tìm về chủ đề đó thường hay truy cập vào website nào. Những website này đã qua hai lần "lọc", một của máy tìm kiếm và một của người dùng (không phải ngẫu nhiên mà nhiều người dùng lại có cùng một lựa chọn). Đôi khi những kết quả này còn tốt hơn cả kết quả máy tìm kiếm trả lại. Ví dụ: những website tin tức lớn,được nhiều người tìm & truy cập nhất của Vietnam như: VnExpress, Vietnamnet, Dân Trí… đều không xuất hiện trong top 10 khi tìm *"vietnam news"* trên cả Yahoo & Live Search (phiên bản mới của MSN).

Vì lí do đó, khóa luận đề xuất việc xây dựng một hệ thống tư vấn website cho máy tìm kiếm dựa trên khai phá query log. Bài toán khai phá query logs là bài toán phải xử lý khối lượng dữ liệu rất lớn (lên tới hàng gigabyte) nên việc chọn được một thuật toán tốt và hiệu quả về thời gian là rất khó khăn. Hệ thống này được phát triển từ đề tài nghiên cứu khoa học về hệ tư vấn website của nhóm chúng tôi [1] (thuộc phòng thí nghiệm Sislab – đại học Công Nghệ). [1] tập trung vào việc thống kê website và khai phá mẫu có thứ tự (tìm ra quy luật giữa từ khóa trong query và url được click) để đưa ra tư vấn. Khác với [1], hệ thống được đề xuất trong khóa luận tập trung vào việc xác định tập website có giá trị và xếp hạng lại chúng theo query người dùng đưa vào. Ý tưởng chính của hệ thống gồm ba bước:

Bước một: nhóm các query tương đồng vào các cụm. Mỗi cụm tương ứng với một chủ đề.

Bước hai: tìm ra tập những website (url) tốt, đại diện cho từng cụm. Tập website này gọi là tập website tư vấn.

Bước ba: khi người dùng đưa vào một query mới, query này sẽ được phân cụm. Hệ thống sẽ phân tích, và đưa ra các website trong tập website tư vấn thích hợp nhất với query đó.

Phần còn lại của khóa luận được chia thành bốn chương:

Chương 1. *Tổng quan về hệ tư vấn*: Trình bày những nội dung cơ bản về hệ tư vấn (các hệ thống nổi tiếng, mô tả bài toán tư vấn, phân loại các hệ tư vấn theo phương pháp xây dựng). Giới thiệu hệ tư vấn website được xây dựng trong khóa luận.

Chương 2. *Khai phá query log và ứng dụng*: Giới thiệu về cấu trúc query log của máy tìm kiếm, các thông tin có thể khai phá, phương pháp khai phá và các ứng dụng của việc khai phá query log.

Chương 3. *Hệ thống tư vấn website cho máy tìm kiếm dựa trên khai phá query log*: Trình bày mô hình hệ thống tư vấn website do chúng tôi đưa ra và các công trình liên quan.

Chương 4. *Thực nghiệm và đánh giá*: Xây dựng, thử nghiệm và đánh giá hệ thống với các query liên quan tới miền sản phẩm điện tử.

Phần kết luận tổng kết nội dung chính của khóa luận, các vấn đề còn tồn tại và định hướng phát triển của hệ thống.

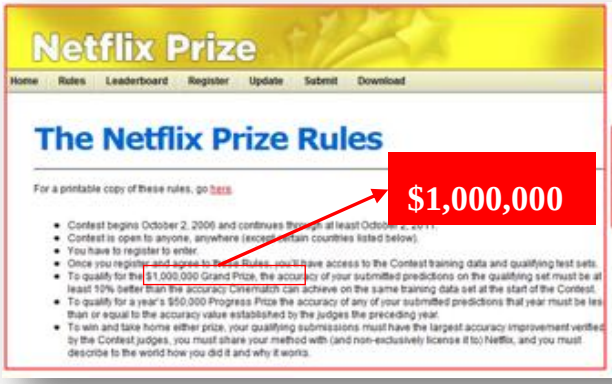
Chương 1. Tổng quan về hệ tư vấn

1.1. Giới thiệu về hệ tư vấn

Trong cuộc sống hàng ngày, trong rất nhiều trường hợp, người ta đưa ra các lựa chọn dựa trên những ý kiến hay lời khuyên của mọi người xung quanh, có thể qua lời nói, các bản đánh giá sản phẩm, khảo sát thị trường, thư giới thiệu …v..v. Nhưng trong kỉ nguyên thông tin, hàng triệu thông tin được đưa lên internet mỗi ngày, điều này dẫn tới yêu cầu phải có các phương pháp tự động thu thập thông tin và đưa ra lời khuyên để hỗ trợ cho các phương pháp truyền thống trên . Hệ tư vấn (recommender system) là một giải pháp như vậy. Hệ thống này đưa ra gợi ý dựa trên những gì người dùng đã làm trong quá khứ, hoặc dựa trên tổng hợp ý kiến của những người dùng khác. Hệ tư vấn đã trở thành một ứng dụng quan trọng và thu hút được sự quan tâm lớn của các nhà nghiên cứu cũng như các doanh nghiệp.

Một vài hệ tư vấn nổi tiếng [8] :

- Phim / TV/ âm nhạc: MovieLens, EachMovie, Morse, Firefly, Flycasting, Ringo…
- Tin tức / báo chí: Tapestry, GroupLens, Lotus Notes, Anatagonomy…
- Sách / Tài liệu: Amazon.com, Foxtrot, InfoFinder…
- Web: Phoaks, Gab, Fab, IfWeb, Let's Browse …
- Nhà hàng: Adaptive Place Advisor, Polylens, Pocket restaurent finder…
- Du lịch: Dietorecs, LifestyleFinder …



Hình 1. Giải thưởng 1 triệu USD của Netflix cho ai đưa ra được thuật toán giúp tăng độ chính xác của hệ thống tư vấn phim của họ thêm 10% [21]

Hình 2. Ba hội nghị của ACM về hệ tư vấn được tổ chức ở châu Âu và Mỹ [3]

1.2. Bài toán tư vấn

Theo Adomavicius và Tuzhilin trong [4], trong hầu hết các trường hợp, bài toán tư vấn được coi là bài toán ước lượng trước hạng (rating) của các sản phẩm (phim, cd, nhà hàng …) chưa được người dùng xem xét. Việc ước lượng này thường dựa trên những đánh giá đã có của chính người dùng đó hoặc những người dùng khác. Những sản phẩm có hạng cao nhất sẽ được dùng để tư vấn.

Một cách hình thức, bài toán tư vấn được mô tả như sau:

Gọi C là tập tất cả người dùng; S là tập tất cả các sản phẩm có thể tư vấn. Tập S có thể rất lớn, từ hàng trăm ngàn (sách, cd…) đến hàng triệu (như website). Tập C trong một số trường hợp cũng có thể lên tới hàng triệu.

Hàm *u(c,s)* đo độ phù hợp (hay hạng) của sản phẩm s với user c: $u: C \times S \to R$ với R là tập được sắp thứ tự. Với mỗi người dùng $c \in C$, cần tìm sản phẩm $s' \in S$ sao cho hàm *u(s', c)* đạt giá trị lớn nhất: $\forall c \in C, s'_c = \arg\max_{s \in S} u\,(c, s)$

Trong hệ tư vấn, độ phù hợp của một sản phẩm thường được cho bằng điểm, ví dụ người dùng A đánh giá bộ phim "Star war 3" được điểm 7/10. Tuy nhiên, nhìn chung độ phù hợp có thể là một hàm bất kì tùy thuộc vào ứng dụng cụ thể. Giá trị của hàm u có thể được xác định bởi người dùng hoặc được tính toán bởi công thức nào đó.

Mỗi người dùng trong không gian C được xác định bởi một hồ sơ (profile). Hồ sơ này có thể gồm rất nhiều loại thông tin: tuổi, giới tính, thu nhập, … hoặc có thể chỉ gồm một trường mã số người dùng (user id) duy nhất. Tương tự, mỗi sản phẩm trong không gian S cũng được xác định bởi một tập các đặc trưng. Ví dụ, trong hệ thống tư vấn phim, đặc trưng của mỗi bộ phim có thể là : tên phim, thể loại, đạo diễn, năm sản xuất, diễn viên chính …v…v.

Vấn đề chính của hệ tư vấn là hàm u không được xác định trên toàn không gian $C \times S$ mà chỉ trên một miền nhỏ của không gian đó. Điều này dẫn tới việc hàm u phải được ngoại suy trong không gian $C \times S$. Thông thường, độ phù hợp được thể hiện bằng điểm và chỉ xác định trên tập các sản phẩm đã từng được người dùng đánh giá từ trước (thường khá nhỏ). Ví dụ, bảng 1 là đánh giá của một số người dùng với các phim mà họ đã xem (thang điểm từ 0-10, kí hiệu $\emptyset$ nghĩa là bộ phim chưa được người dùng cho điểm). Từ những thông tin đó, hệ thống tư vấn phải dự đoán (ngoại suy) điểm cho các bộ phim chưa được người dùng đánh giá, từ đó đưa ra những gợi ý phù hợp nhất.

	Harry potter	Star trek	Xmen	Transformer
A	5	∅	7	9
B	9	5	5	∅
C	6	6	∅	8
D	∅	∅	8	9

Bảng 1. Đánh giá của người dùng về một số bộ phim đã xem

1.3. Phân loại hệ tư vấn

Có rất nhiều cách để dự đoán, ước lượng hạng/điểm cho các sản phẩm như sử dụng học máy, lí thuyết xấp sỉ, các thuật toán dựa trên kinh nghiệm… Theo [4], các hệ thống tư vấn thường được phân thành ba loại dựa trên cách nó dùng để ước lượng hạng của sản phẩm:

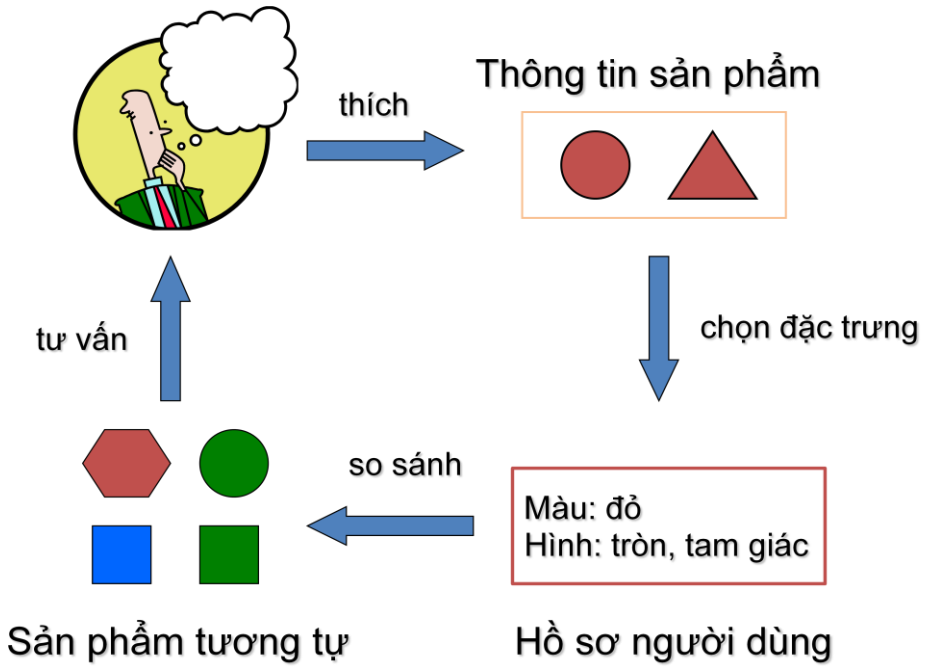
- Dựa trên nội dung (content-based): người dùng được gợi ý những sản phẩm tương tự như các sản phẩm từng được họ đánh giá cao.
- Cộng tác (collaborative): người dùng được gợi ý những sản phẩm mà những người cùng sở thích với họ đánh giá cao.
- Lai ghép (hybrid): kết hợp cả phương pháp dựa trên.

1.3.1. Phương pháp dựa trên nội dung

Theo [4], với phương pháp tư vấn dựa trên nội dung, độ phù hợp $u(c, s)$ của sản phẩm s với người dùng c được đánh giá dựa trên độ phù hợp $u(c, s_i)$, trong đó $s_i \in S$ và "tương tự" như s. Ví dụ, để gợi ý một bộ phim cho người dùng c, hệ thống tư vấn sẽ tìm các đặc điểm của những bộ phim từng được c đánh giá cao (như diễn viên, đạo diễn…); sau đó chỉ những bộ phim tương đồng với sở thích của c mới được giới thiệu.

Hướng tiếp cận dựa trên nội dung bắt nguồn từ những nghiên cứu về thu thập thông tin (IR - information retrieval) và lọc thông tin (IF - information filtering). Do đó, rất nhiều hệ thống dựa trên nội dung hiện nay tập trung vào tư vấn các đối tượng chứa dữ liệu text như văn bản, tin tức, website… Những tiến bộ so với hướng tiếp cận cũ của IR là do việc sử dụng hồ sơ về người dùng (chứa thông tin về sở thích, nhu cầu…) . Hồ sơ này được xây dựng dựa trên những thông tin được người dùng cung

cấp trực tiếp (khi trả lời khảo sát) hoặc gián tiếp (do khai phá thông tin từ các giao dịch của người dùng).



Hình 3. Tư vấn dựa trên nội dung [17]

Để cụ thể hơn, đặt *Content(s)* là tập thông tin (hay tập các đặc trưng) về sản phẩm *s*. Do hệ thống dựa trên nội dung được thiết kế chủ yếu để dành cho các sản phẩm là text, nên nội dung sản phẩm thường được biểu diễn bởi các từ khóa (keyword): $Content(s) = (w_{1s}, \ldots w_{ks})$, với $w_{1s},..w_{ks}$ là trọng số của các từ khóa từ 1 tới k (có thể được tính bằng TF-IDF). Ví dụ, hệ tư vấn website Fab biểu diễn nội dung các trang web bằng 100 từ quan trọng nhất. Tương tự, hệ thống Syskill & Webert biểu diễn văn bản bằng 128 từ có trọng số cao nhất.

Đặt *Profile(c)* là hồ sơ về người dùng *c*, bao gồm các thông tin về sở thích của *c*. Những thông tin này có được bằng cách phân tích nội dung của các sản phẩm từng được *c* đánh giá (cho điểm) trước đó. Phương pháp được sử dụng thường là các kĩ thuật phân tích từ khóa của IR, do đó, *Profile(c)* cũng có thể được định nghĩa như một vector trọng số:

$Profile(c) = (w_{1c}, \ldots, w_{kc})$ với x_{ic} biểu thị độ quan trọng của từ khóa i với người dùng *c*.

Trong hệ thống tư vấn dựa trên nội dung, độ phù hợp *u(c,s)* được xác định bởi công thức:

$u(c,s) = score(Profile(c), Content(s))$

Cả *Profile(c), Content(s)* đều có thể được biểu diễn bằng vector trọng số từ TF-IDF (tương ứng là $\overrightarrow{w_c}$, $\overrightarrow{w_s}$) nên có thể đo độ tương đồng của chúng bằng độ đo cosin:

$$u(c,s) = \cos(\overrightarrow{w_c}, \overrightarrow{w_s}) = \frac{\overrightarrow{w_c} . \overrightarrow{w_s}}{||\overrightarrow{w_c}|| \times ||\overrightarrow{w_s}||}$$

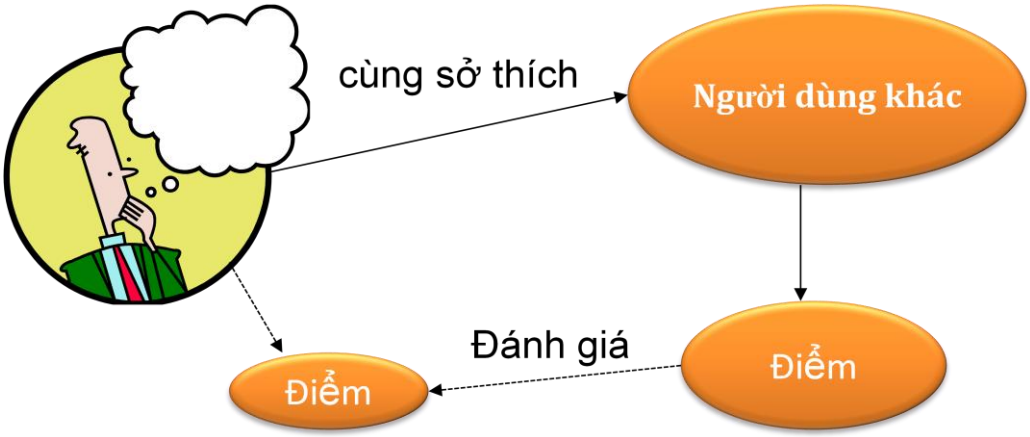
Ví dụ, nếu c đọc nhiều bài báo thuộc lĩnh vực sinh học thì các từ khóa liên quan tới sinh học (như gen, protein, tế bào, ADN…) trong *Profile(c)* sẽ có trọng số cao. Hệ quả là với các bài báo s cũng thuộc lĩnh vực này sẽ có độ phù hợp *u(c,s)* cao hơn với người dùng *c*.

Bên cạnh các phương pháp IR, hệ tư vấn dựa trên nội dung còn sử dụng nhiều phương pháp học máy khác như: phân lớp Bayes, cây quyết định, mạng nơron nhân tạo… Các phương pháp này khác với các phương pháp của IR ở chỗ nó dựa trên các mô hình học được từ dữ liệu nền. Ví dụ, dựa trên tập các trang web đã được người dùng đánh giá là có nội dung "tốt" hoặc "xấu" có thể sử dụng phân lớp Bayes để phân loại các trang web chưa được đánh giá.

1.3.2. Phương pháp cộng tác

Theo [4], không giống như phương pháp tư vấn dựa trên nội dung, hệ thống cộng tác dự đoán độ phù hợp *u(c,s)* của một sản phẩm *s* với người dùng *c* dựa trên độ phù hợp $u(c_j, s)$ giữa người dùng c_j và *s*, trong đó c_j là người có cùng sở thích với *c*. Ví dụ, để gợi ý một bộ phim cho người dùng *c*, đầu tiên hệ thống cộng tác tìm những người dùng khác có cùng sở thích phim ảnh với *c*. Sau đó, những bộ phim được họ đánh giá cao sẽ được dùng để tư vấn cho *c*.

Có rất nhiều hệ thống cộng tác đã được phát triển như: Grundy, GroupLens (tin tức), Ringo (âm nhạc), Amazon.com (sách), Phoaks (web)… Các hệ thống này có thể chia thành hai loại: dựa trên kinh nghiệm (heuristic-based hay memory-based) và dựa trên mô hình (model-based).



Hình 4. Tư vấn dựa trên cộng tác [17]

1.3.2.1. Hệ thống cộng tác dựa trên kinh nghiệm

Các thuật toán dựa trên kinh nghiệm dự đoán hạng của một sản phẩm dựa trên toàn bộ các sản phẩm đã được đánh giá trước đó bởi người dùng. Nghĩa là, hạng của sản phẩm s với người dùng c ($r_{c,s}$) được tổng hợp từ đánh giá của những người dùng khác về s (thường là N người có sở thích tương đồng nhất với c).

$$r_{c,s} = \text{aggr}\ r_{c',s} \quad \text{với } c' \in \hat{C} \text{ (tập N người dùng cùng sở thích với } c)$$

Một số ví dụ về hàm tổng hợp (aggregate):

$$(a)\ r_{c,s} = \frac{1}{N}\sum_{c' \in \hat{C}} r_{c',s}$$

$$(b)\ r_{c,s} = k \times \sum_{c' \in \hat{C}} sim(c, c') \times r_{c',s}$$

$$(c)\ r_{c,s} = \overline{r_c} + k \times \sum_{c' \in \hat{C}} sim(c, c') \times (r_{c',s} - \overline{r_{c'}})$$

Với: k = hệ số chuẩn hóa

$sim(c, c')$ = độ tương đồng (về sở thích) giữa người dùng c và c'

$\overline{r_c}$, $\overline{r_{c'}}$ = trung bình của các đánh giá được cho bởi người dùng c và c'

Có nhiều cách để tính độ tương đồng (về sở thích) giữa hai người dùng, nhưng trong hầu hết các phương pháp, độ tương đồng chỉ được tính dựa trên các sản phẩm

được cả hai người cùng đánh giá. Hai phương pháp phổ biến nhất là dựa trên độ tương quan (correlation-based) và dựa trên cosin (cosine-based).

Đặt $S_{xy} = \{s \in S | r_{x,s} \neq \emptyset \ \& \ r_{y,s} \neq \emptyset\}$ là tập các sản phẩm được đánh giá bởi cả hai người dùng *x, y*.

Công thức dựa trên độ tương quan của Pearson [27]:

$$sim\ (x,y) = \frac{\sum_{s \in S_{xy}} (r_{x,s} - \overline{r_x}) \times (r_{y,s} - \overline{r_y})}{\sqrt{\sum_{s \in S_{xy}} (r_{x,s} - \overline{r_x})^2 \times \sum_{s \in S_{xy}} (r_{y,s} - \overline{r_y})^2}}$$

Với phương pháp dựa trên cosin, hai người dùng được biểu diễn bởi 2 vector m chiều, với $m = |S_{xy}|$. Độ tương đồng giữa 2 vector được tính bởi công thức:

$$sim(x,y) = \cos(\vec{x}, \vec{y}) = \frac{\vec{x}.\vec{y}}{||\vec{x}|| \times ||\vec{y}||} = \frac{\sum_{s \in S_{xy}} r_{x,s} \times r_{y,s}}{\sqrt{\sum_{s \in S_{xy}} {r_{x,s}}^2} \times \sqrt{\sum_{s \in S_{xy}} {r_{y,s}}^2}}$$

1.3.2.2. Hệ thống cộng tác dựa trên mô hình

Khác với phương pháp dựa trên kinh nghiệm, phương pháp dựa trên mô hình (model-based) sử dụng kĩ thuật thống kê và học máy trên dữ liệu nền (các đánh giá đã biết) để xây dựng nên các mô hình. Mô hình này sau đó sẽ được dùng để dự đoán hạng của các sản phẩm chưa được đánh giá.

Breese trong [14] đề xuất hướng tiếp cận xác suất cho lọc cộng tác (collaborative filtering), trong đó công thức sau ước lượng đánh giá của người dùng c về sản phẩm s (thang điểm đánh giá từ 0 đến n):

$$r_{c,s} = E(r_{c,s}) = \sum_{i=0}^{n} i \times \Pr(r_{c,s} = i | r_{c,s'}, s' \in S_c)$$

Billsus và Pazzani trong [12] đề xuất phương pháp lọc cộng tác trên nền học máy, trong đó rất nhiều các kĩ thuật học máy (như mạng nơron nhân tạo) và các kĩ thuật trích chọn đặc trưng (như SVD – một kĩ thuật đại số nhằm làm giảm số chiều của ma trận) có thể được sử dụng.

Ngoài ra còn nhiều hướng tiếp cận khác như mô hình thống kê, mô hình bayes, mô hình hồi quy tuyến tính, mô hình entropy cực đại…

Hệ thống tư vấn cộng tác khắc phục được nhiều nhược điểm của hệ thống dựa trên nội dung. Một điểm quan trọng là nó có thể xử lý mọi loại dữ liệu và gợi ý mọi loại sản phẩm, kể cả những sản phẩm mới, khác hoàn toàn so với những gì người dùng từng xem.

1.3.3. Phương pháp lai ghép

Một vài hệ tư vấn kết hợp cả phương pháp cộng tác và dựa trên nội dung nhằm tránh những hạn chế của cả hai. Có thể phân thành bốn cách kết hợp như sau:

- Cài đặt hai phương pháp riêng rẽ rồi kết hợp dự đoán của chúng.
- Tích hợp các đặc trưng của phương pháp dựa trên nội dung vào hệ thống cộng tác
- Tích hợp các đặc trưng của phương pháp cộng tác vào hệ thống dựa trên đặc trưng
- Xây dựng mô hình hợp nhất, bao gồm các đặc trưng của cả hai phương pháp.

1.3.3.1. Kết hợp hai phương pháp riêng rẽ

Có hai kịch bản cho trường hợp này:

- Cách 1: Kết hợp kết quả của cả hai phương pháp thành một kết quả chung duy nhất, sử dụng cách kết hợp tuyến tính (linear combination) hoặc voting scheme.
- Cách 2: Tại mỗi thời điểm, chỉ chọn phương pháp cho kết quả tốt hơn (dựa trên một số độ đo chất lượng tư vấn nào đó). Ví dụ, hệ thống DailyLearner system chọn phương pháp nào đưa ra gợi ý với độ chính xác (confidence) cao hơn.

1.3.3.2. Thêm đặc trưng của mô hình dựa trên nội dung vào mô hình cộng tác

Một số hệ thống lai (như Fab) dựa chủ yếu trên các kĩ thuật cộng tác nhưng vẫn duy trì hồ sơ về người dùng (theo dạng của mô hình dựa trên nội dung). Hồ sơ này được dùng để tính độ tương đồng giữa hai người dùng, nhờ đó giải quyết được trường hợp có quá ít sản phẩm chung được đánh giá bởi cả hai người. Một lợi ích khác là các gợi ý sẽ không chỉ giới hạn trong các sản phẩm được đánh giá cao bởi những người cùng sở

thích (gián tiếp), mà còn cả với những sản phẩm có độ tương đồng cao với sở thích của chính người dùng đó (trực tiếp).

1.3.3.3. Thêm đặc trưng của mô hình cộng tác vào mô hình dựa trên nội dung

Hướng tiếp cận phổ biến nhất là dùng các kĩ thuật giảm số chiều trên tập hồ sơ của phương pháp dựa trên nội dung. Ví dụ, [29] sử dụng phân tích ngữ nghĩa ẩn (latent semantic analysis) để tạo ra cách nhìn cộng tác (collaborative view) với tập hồ sơ người dùng (mỗi hồ sơ được biểu diễn bởi một vector từ khóa).

1.3.3.4. Mô hình hợp nhất hai phương pháp

Trong những năm gần đây đã có khá nhiều nghiên cứu về mô hình hợp nhất. [10] đề xuất kết hợp đặc trưng của cả hai phương pháp vào một bộ phân lớp dựa trên luật (rule-based classifier). Popescul và cộng sự trong [25] đưa ra phương pháp xác suất hợp nhất dựa trên phân tích xác suất ngữ nghĩa ẩn (probabilistic latent semantic analysis). [6] giới thiệu mô hình hồi quy Bayes sử dụng dây Markov Monte Carlo để ước lượng tham số.

Độ chính xác của hệ thống tư vấn lai ghép có thể được cải tiến bằng cách sử dụng các kĩ thuật dựa trên tri thức (knowledge-based) như case-based reasoning. Ví dụ, hệ thống Entrée dùng những tri thức về nhà hàng, thực phẩm (như: đồ biển không phải là thức ăn chay).. để gợi ý nhà hàng thích hợp cho người dùng. Hạn chế chính của hệ thống dạng này là nó cần phải thu thập đủ tri thức, đây cũng là nút thắt cổ chai (bottle-neck) của rất nhiều hệ thống trí tuệ nhân tạo khác. Tuy nhiên, các hệ thống tư vấn dựa trên tri thức hiện đang được phát triển trên các lĩnh vực mà miền tri thức của nó có thể biểu diễn ở dạng mà máy tính đọc được (như ontology). Ví dụ, hệ thống Quickstep và Foxtrot sử dụng ontology về chủ đề của các bài báo khoa học để gợi ý những bài báo phù hợp cho người dùng.

Một vài bài báo như [9] đã thực hiện so sánh hiệu năng của hệ thống lai ghép với các hệ thống dựa trên nội dung hoặc cộng tác thuần túy và cho thấy hệ thống lai ghép có độ chính xác cao hơn.

Phương pháp	Các kĩ thuật sử dụng	
	Dựa trên kinh nghiệm	Dựa trên mô hình
Dựa trên nội dung	+TF-IDF +Phân cụm	+Phân lớp bayes +Phân cụm +Cây quyết định +Mạng nơron nhân tạo
Cộng tác	+k-Láng giềng gần nhất +Phân cụm +Lí thuyết đồ thị	+Mạng bayes +Phân cụm +Mạng nơron nhân tạo +Hồi quy tuyến tính +Mô hình xác suất
Lai ghép	+Kết hợp tuyến tính kết quả	+Tích hợp đặc trưng của một phương pháp vào mô hình của phương pháp còn lại. +Xây dựng mô hình hợp nhất hai phương pháp.

Bảng 2. Ba phương pháp tư vấn [4]

1.4. Sơ bộ về hệ tư vấn trong khóa luận

Hệ thống được xây dựng trong khóa luận là một hệ thống tư vấn website. Nhưng thay vì đứng như một ứng dụng riêng rẽ, hệ thống sẽ được tích hợp ngay vào máy tìm kiếm để trực tiếp đưa ra những tư vấn phù hợp với nội dung query của người dùng.

Phương pháp được sử dụng để đưa ra tư vấn cho một query là dựa vào các lựa chọn của những người dùng đã từng tìm về chủ đề đó. Vì thế, có thể xếp hệ thống vào nhóm các hệ tư vấn cộng tác (collaborative).

Với hầu hết các hệ tư vấn cộng tác thường thấy, từng người dùng cụ thể được xác định rõ ràng (qua hồ sơ cá nhân) và các sản phẩm thường được người dùng đánh giá

trực tiếp (ví dụ: cho điểm). Nhưng trong hệ tư vấn website cho máy tìm kiếm, cả hai việc trên đều **không thể** thực hiện được. Hầu hết tất cả các máy tìm kiếm hiện nay đều không yêu cầu người dùng phải đăng kí tài khoản vì việc buộc phải đăng nhập hệ thống là một cản trở không dễ chịu. Do đó, không thể phân biệt được người dùng với nhau mà chỉ có thể "cố gắng" phân biệt các phiên sử dụng (session) của họ bằng cách phân tích log của máy tìm kiếm (dựa vào các thông tin về IP, trình duyệt, thời gian …). Hơn nữa, do tìm kiếm đã trở thành một việc rất phổ biến và được thực hiện liên tục nhiều lần, người dùng luôn muốn nhận được kết quả thật nhanh và không muốn vướng vào các chi tiết rườm ra nên việc yêu cầu người dùng chấm điểm hay đánh giá các kết quả được trả về cũng không khả thi.

Vì những lý do trên, thay vì xác định đối tượng là người dùng, hệ thống được đề xuất trong báo cáo xác định đối tượng là các query. Hai query tương đồng có vai trò như hai người dùng cùng sở thích. Những website (url) được click tương ứng với query có vai trò như những sản phẩm được người dùng đánh giá cao (vì chỉ có một vài website được click trên tổng số kết quả trả về). Các thông tin về query tương đồng và url được click được khai thác từ query log của máy tìm kiếm.

Chương 2. Bài toán khai phá query log và ứng dụng

2.1. Cấu trúc query log

Query log bao gồm thông tin về những lượt tìm kiếm của người dùng được máy tìm kiếm lưu lại. Khác với server log thông thường, query log có thêm thông tin về nội dung query và các website được người dùng click. Mỗi máy tìm kiếm có một cách lưu log khác nhau và thường rất ít khi công bố ra ngoài (một lí do là vì vi phạm sự riêng tư của người dùng). Hình 5 & 6 là một phần query log của AOL được công bố năm 2006 [7] và cấu trúc log của Google, được công bố trên website của công ty này [18].

AnonID	Query	QueryTime	Rank	ClickURL
479	family guy movie references	2006-03-03 22:37:46	1	http://www.familyguyfiles.com
479	top grossing movies of all time	2006-03-03 22:42:42	1	http://movieweb.com
479	top grossing movies of all time	2006-03-03 22:42:42	2	http://www.imdb.com
479	car decals	2006-03-03 23:20:12	4	http://www.decaljunky.com
479	car decals	2006-03-03 23:20:12	1	http://www.modernimage.net
479	car decals	2006-03-03 23:20:12	5	http://www.webdecal.com
479	car window decals	2006-03-03 23:24:05	9	http://www.customautotrim.com
479	car window sponsor decals	2006-03-03 23:27:17	3	http://www.streetglo.net
479	bose	2006-03-03 23:30:11	1	http://www.bose.com
479	bose car decal	2006-03-03 23:31:48	1	http://stickers.signprint.co.uk
479	bose car decal	2006-03-03 23:31:48	1	http://stickers.signprint.co.uk
479	bose car decal	2006-03-03 23:31:48	7	http://www.motorcitydecals.com
479	chicago the mix	2006-03-04 22:11:31	1	http://www.wtmx.com
479	chicago the drive	2006-03-04 22:14:51	2	http://www.wdrv.com

Hình 5. Một phần query log của AOL [7]

```
q         = cars
URL       = www.google.com/search?q=cars
IP        = 72.14.253.103
Cookie    = PREF=ID=03b1d4f329293203:LD=en:NR=10…
Browser   = Firefox/2.0.0.4;Windows NT 5.1
Time      = 25 Mar 2007 10:15:32
```

Hình 6. Cấu trúc log của Google [18]

Tuy khác nhau nhưng query log thường có các trường sau:

Query:

Truy vấn mà người dùng gửi tới máy tìm kiếm. Ví dụ: *"race cars", "vietnam*

landscape", "swine flu" …Một số máy tìm kiếm giới hạn số từ trong query (Google cho phép query dài tối đa 32 từ).

Url được click và vị trí của url

Địa chỉ url người dùng click và vị trí của nó (trường ItemRank của AOL query log) trong danh sách kết quả máy tìm kiếm trả về cho query vừa được gửi.Ví dụ, với query *"champion league"*, các url được click là: *www.uefa.com* (ở vị trí 1) và *soccernet.espn.go.com* (ở vị trí 4, theo kết quả của Google).

Địa chỉ IP:

Địa chỉ IP của người dùng (ví dụ:141.243.1.172) hoặc tên DNS (ví dụ: wpbfl2-45.gate.net). Từ IP có thể biết được địa chỉ (quốc gia, vùng) của người dùng và nhà cung cấp dịch vụ internet cho họ (Internet Service Provider). Khi công bố query log ra công chúng, các máy tìm kiếm buộc phải "nặc danh hóa" (anonymizing) trường này để không làm lộ danh tính và các thông tin cá nhân của người dùng. Như ở trên, trong query log được AOL công bố, trường IP được thay thế bằng AnonID (định danh ẩn).

Phần mềm sử dụng ở máy của người dùng (user agents):

Trường này lưu thông tin về tên, phiên bản của trình duyệt cũng như tên, phiên bản của hệ điều hành được người dùng sử dụng.Ví dụ:"*Firefox/2.0.0.4;Windows NT 5.1".*

Thời gian:

Thời gian người dùng gửi query tới máy tìm kiếm. Thông thường, như trong Google hay AOL, thời gian được ghi theo định dạng *[DD/Mon/YYYY/: HH:MM:SS offset]* với:

DD/Mon/YYYY: chỉ ngày tháng năm.

HH:MM:SS : thể hiện 24h trong ngày.

Offset: chỉ độ lệch múi giờ so với giờ GMT (Greenwich Mean Time).

Ví dụ*:" 22/May/2009:16:03:00 +0700"* chỉ thời điểm 16:03:00 ngày 22 tháng 5 năm 2009, tại múi giờ GMT+7 (Bangkok-Hanoi-Jakarta). Ở một số máy tìm kiếm khác, như AltaVista, trường thời gian được lưu ở dạng **timestamp**, là số milli giây từ một mốc thời gian trong quá khứ (baseline) đến thời điểm query được gửi. Ví dụ, nếu chọn mốc thời gian là 00:00:00 ngày 1/1/1995 thì thời điểm 12:00:02 28/10/2004 có *timestamp = 20822005*

Cookie:

Được máy tìm kiếm lưu ở máy người dùng để nhận biết một số thông tin về họ. Ví dụ, trường cookie của Google lưu sở thích của người dùng về ngôn ngữ tìm kiếm và số kết quả mong muốn trong mỗi trang.

*"Cookie = PREF=ID=03b1d4f329293203:LD=**en**:NR=**10**…"*

Theo [18], để đảm bảo tính bí riêng tư, sau 18 tháng, Google sẽ xóa thông tin về cookie và IP của người dùng. Ví dụ, các thông tin đó sẽ được đưa về dạng *IP=72.14.253.XX* và *Cookie=PREF=XXXXXXXX*.

2.2. Khai phá query log

Từ những thông tin trong query log, có thể áp dụng rất nhiều các phương pháp thống kê và khai phá dữ liệu (như tìm luật liên kết, tìm mẫu có thứ tự …) để phân tích thói quen sử dụng, xu hướng, sở thích… của người dùng. Những thông tin thu được không chỉ hữu ích cho việc cải tiến chất lượng tìm kiếm mà còn giúp nghiên cứu hành vi của người dùng trên internet.

2.2.1. Một số dạng thống kê

- ***Thống kê sơ bộ***: tổng hợp những thông tin cơ bản về toàn bộ bộ query log như độ lớn, thời gian thu thập, số bản ghi, số query … Bảng 3 và 4 là ví dụ về thống kê sơ bộ với bộ query log được AOL công bố năm 2006 [7] và bộ query log lưu hành nội bộ của AltaVista [28]:

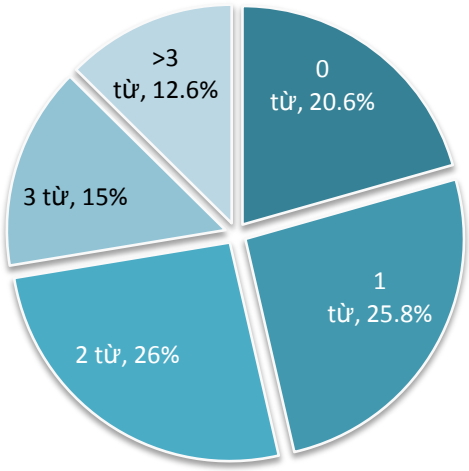
Độ lớn	**~500MB**
Thời gian thu thập	01/03/2006 – 31/05/2006
Tổng số bản ghi	36.389.567
Tổng số query	21.011.340
Số query riêng biệt (sau khi chuẩn hóa)	10.154.742
Số lần click url	19.442.629
Số query không click vào url nào	16.946.938
Số lần mở trang kết quả tiếp theo	7.887.022
Số người dùng riêng biệt	657.426

Bảng 3. Thống kê sơ bộ trên query log của AOL [7]

Độ lớn	**~280GB**
Thời gian thu thập	~6 tuần
Tổng số bản ghi	993.208.159 (~1 tỉ)
Số yêu cầu có độ dài > 0	843.445.731
Số query có độ dài > 0	575.244.993
Số query riêng biệt (độ dài > 0)	153.645.050
Số phiên làm việc	285.474.117

Bảng 4. Thống kê sơ bộ trên query log của AltaVista [28]

- ***Số từ trung bình trong query***: Theo [28], độ dài trung bình của một query trong bộ log của AltaVista là 2.35 từ. Với AOL, độ dài này là 2.34 (theo [7]). Có thể thấy query thường có độ dài rất ngắn, chủ yếu từ 2-3 từ. Sau khi phân tích query log của MSN, [15] phân loại các query dài (từ 5-12 từ) vào 5 nhóm như bảng 5:



Hình 7. Tỉ lệ từ/query trong query log của AltaVista [28]

Tổng số query: 14.921.286		
Số query dài (5 đến 12 từ): 1.423.664		
Loại	***Số lượng***	***Tỉ lệ***
Câu hỏi	106.587	7.49%
Query có chứa toán tử	78.331	5.50%
Gộp các query ngắn	918.482	64.52%
Cụm danh từ dài hoặc câu trích dẫn (quote)	320.263	22.50%

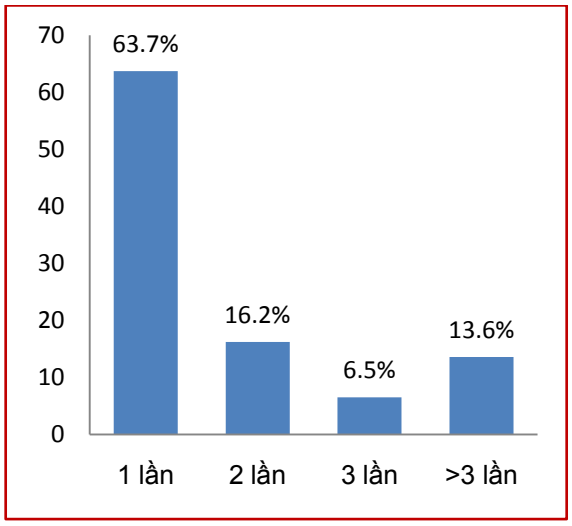
Bảng 5. Phân loại query dài trong MSN log [15]

- ***Những từ được search nhiều nhất***: thể hiện sự quan tâm và xu hướng của người dùng trong tìm kiếm thông tin trên internet. Ở các quốc gia khác nhau hay tại thời điểm khác nhau, người dùng có thể có những mối quan tâm khác nhau. Bảng 6 là những từ được tìm kiếm nhiều nhất trên Google vào năm 2006, tại Anh năm 2008 và tại Brasil năm 2008 [19]:

Google 2006	*Google tại Anh, 2008*	*Google tại Brasil, 2008*
1. bebo 2. myspace 3. world cup 4. metacafe 5. radioblog 6. wikipedia 7. video 8. rebelde 9. mininova 10. wiki	1. facebook 2. bbc 3. youtube 4. ebay 5. games 6. news 7. hotmail 8. bebo 9. yahoo 10. jobs	1. orkut 2. jogos 3. download 4. fotos 5. youtube 6. videos 7. musicas 8. musica 9. msn 10. globo

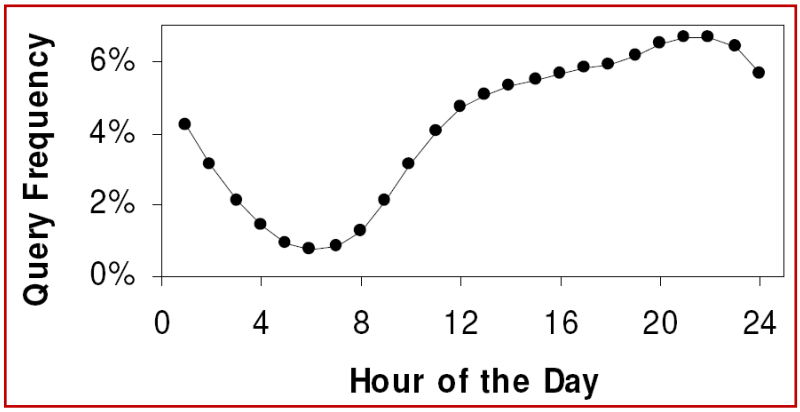
Bảng 6. Những từ được tìm nhiều nhất trên Google [19]

- ***Tỉ lệ lặp lại của query***: cho biết số lần lặp lại của một query. Hình 8 là thống kê của AltaVista [28] trong thời gian 6 tuần. Ở đây, hai query được coi là giống nhau nếu chúng chứa những từ giống nhau, không quan tâm tới thứ tự từ và các toán tử.



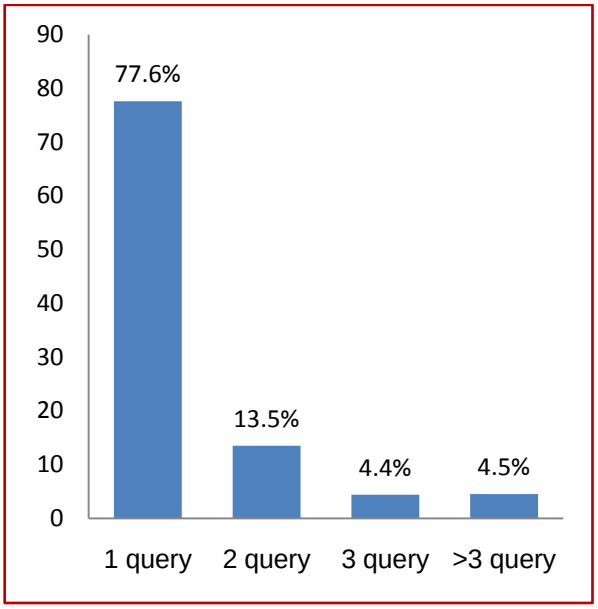
Hình 8. Tỉ lệ lặp lại query trong log của AltaVista [28]

- ***Phân bố query theo giờ trong ngày***: Hình 9 là phân bố query được gửi tới máy tìm kiếm AOL [24]. Nhận thấy tỉ lệ query cao nhất là trong khoảng thời gian từ 20h tới 24h.



Hình 9. Phân bố query trong ngày của AOL [24]

- ***Độ dài mỗi phiên***: thống kê số lượng query trong mỗi phiên tìm kiếm (session) của người dùng. Hình 10 là thống kê của AltaVista [28], số query trung bình trong một phiên là 2.02 (gần 78% các phiên chỉ có 1 query)



Hình 10. Số query trong một phiên trong query log của AltaVista [28]

- ***Nội dung query:*** phân loại nội dung query theo các chủ đề. Thông tin này giúp nắm bắt được thói quen tìm kiếm và những nội dung được nhiều người quan tâm. Bảng 7 và 8 là phân loại nội dung query của 2 máy tìm kiếm AOL và Excite. Có thể thấy các chủ đề về giải trí luôn chiếm tỉ lệ lớn.

Chủ đề	*Tỉ lệ*
Giải trí	13%
Mua sắm	13%
Sex	10%
Nghiên cứu	9%
Máy tính	9%
Sức khỏe	5%
Nhà cửa	5%
Du lịch	5%
Game	5%
Tài chính	3%
Thể thao	3%
Địa điểm ở Mỹ	3%
Lễ hội	1%
Nội dung khác	16%

Bảng 7. Phân loại chủ đề query của AOL [8]

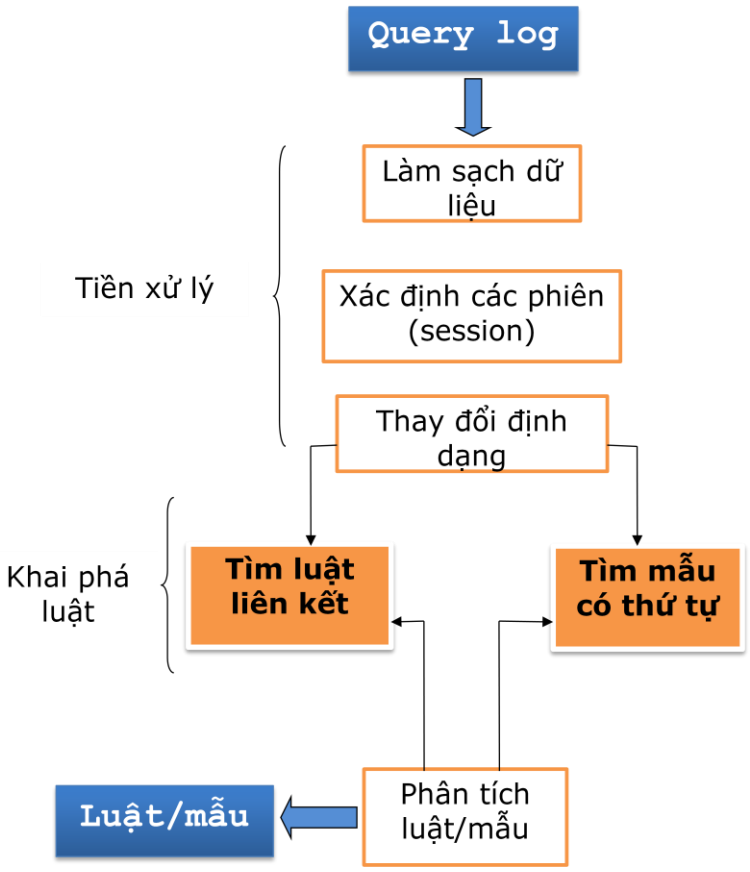
Chủ đề	*Tỉ lệ*
Giải trí	19.9%
Sex	16.8%
Thương mại, kinh tế, du lịch	13.3%
Máy tính, internet	12.5%
Sức khỏe, khoa học	9.5%
Con người, địa điểm	6.7%
Xã hội, văn hóa, tôn giáo	5.7%
Giáo dục	5.6%
Nghệ thuật	5.4%
Chính phủ	3.4%
Nội dung khác	4.1%

Bảng 8. Phân loại chủ đề query của Excite [8]

2.2.2. Khai phá luật

Phân tích các mẫu thường xuất hiện (frequent pattern mining) là một trong những phương pháp nghiên cứu trong ngành khai phá dữ liệu (data mining) với đa dạng các ứng dụng. Một trong những ứng dụng của chính là việc khám phá ra những mẫu (pattern) hay gặp trong dữ liệu log. Mục đích của việc khai phá log này nhằm lấy được những thông tin liên quan đến người dùng dựa vào những việc họ đã làm. Nó có thể phục vụ tốt cho mục đích tư vấn, quảng cáo, tạo ra những thông tin mang tính động đối với người dùng.

Hai thuật toán thường được sử dụng là *Tìm luật kết hợp* (Association Rule mining) và *Phân tích mẫu có thứ tự* (Sequential pattern mining).



Hình 11. Khai phá luật trong query log [1]

- ***Luật kết hợp*** là luật thể hiện những liên kết ẩn giữa các thuộc tính. Một luật kết hợp có dạng "Nếu có A thì có B". Phương pháp tìm luật kết hợp được áp dụng trong [1] để dự đoán quy luật tìm kiếm của người dùng. Ví dụ: nếu query trước tìm về chủ đề *"chính trị"* thì có 45% khả năng query thứ hai sẽ tìm về chủ đề *"kinh tế"*.

- ***Mẫu có thứ tự*** cũng gần giống như luật kết hợp nhưng quan tâm đến tới thứ tự xuất hiện của các thành phần trong luật. Khai phá mẫu có thứ tự được dùng trong [1] để tìm ra những url nào thường xuất hiện sau những từ khóa nhất định. Ví dụ: rất nhiều query chứa từ khóa *"game"* click vào url *computergame.com* thì luật dạng *game* → *computergame.com* có ý nghĩa cao và phản ánh được sở thích của người dùng.

2.3. Ứng dụng của khai phá query log

Những kết quả có được từ khai phá query log được ứng dụng rất nhiều không chỉ trong máy tìm kiếm (Google, Yahoo, AOL…) mà còn trong các website thương mại (Amazon, Netflix…) để phục vụ mục đích cải tiến chất lượng tìm kiếm cũng như để kinh doanh, quảng cáo… Một vài ứng dụng như:

Mở rộng query (query expansion):

Query thường rất ngắn (chỉ từ 2-3 từ) nên nó thường không cung cấp đủ thông tin cần thiết để có thể chọn ra website phù hợp với mong muốn của người dùng. Từ đó dẫn tới yêu cầu phải dự đoán và mở rộng nội dung query. Các phương pháp mở rộng query trước đó thường tập trung vào việc phân tích các văn bản. Hang Cui và cộng sự trong [16] đã đưa ra một phương pháp mới dựa trên các thông tin tương tác của người dùng được lưu lại trong query log. Ý tưởng chính của phương pháp này là tìm ra sự tương quan giữa các từ trong query và các từ trong văn bản bằng cách phân tích quan hệ giữa query và website được click .

Tìm mẫu query (query template):

Hiện nay, hầu hết các máy tìm kiếm đều là dạng dựa trên từ khóa (keyword-based), một vấn đề được đặt ra là làm sao hiểu được ý nghĩa của các từ khóa trong query? Kevin-Chang và cộng sự trong [5] đã đề xuất bài toán tìm ra các mẫu (cấu trúc) thường gặp của query, từ đó giúp hiểu được mục đích mà nó hướng tới. Ví dụ, query "jobs in boston" có mục đích tìm về công việc và "boston" là địa điểm mong muốn (#địa_điểm). Có thể bắt gặp rất nhiều những query có cấu trúc tương tự theo mẫu "jobs in #địa_điểm" chỉ khác giá trị của #địa_điểm.

Khi hiểu được mục đích mà query hướng tới, máy tìm kiếm có thể đưa người dùng đến thẳng trang web phù hợp dù có thể nó không chứa các từ khóa có trong query. Hơn nữa, khi biết cấu trúc của query (ví dụ, #địa_điểm = "boston", #thông_tin = "thời tiết"), máy tìm kiếm còn có thể thực thi ngay yêu cầu của user (sử dụng các thông tin trong query để làm tham số).

Xếp hạng lại kết quả:

Có hai vấn đề thường gặp trong máy tìm kiếm: 1) các kết quả có thứ hạng cao nhất đôi khi không chứa những thông tin phù hợp với mục đích của người dùng và 2) các trang web mới xuất hiện tuy phù hợp nhưng lại không có thứ hạng cao. Do đó,

Zhuang và Cucerzan trong [31] đã đề xuất một phương pháp xếp hạng mới (Q-Rank) để xếp hạng lại (rerank) kết quả của máy tìm kiếm dựa trên việc xác định ngữ cảnh của query nhờ query log.

Tư vấn website:

Việc xây dựng hệ tư vấn website cho máy tìm kiếm dựa trên khai phá query log đã được chúng tôi đề xuất trong nghiên cứu khoa học năm 2009 [1]. Hệ thống được xây dựng trong khóa luận là một bước phát triên của hệ thống cũ.

Chương 3. Mô hình

3.1. Các công trình liên quan

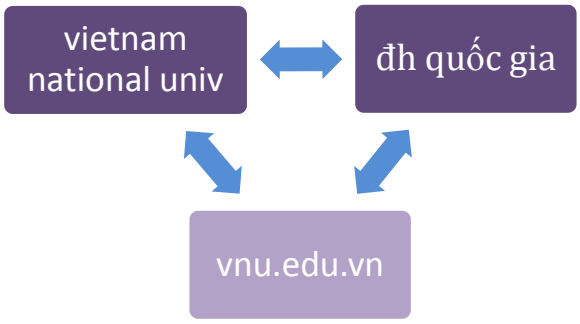
3.1.1. Phân cụm query

Việc phân cụm một tập query gặp nhiều khó khăn hơn việc phân cụm một tập văn bản thông thường (ví dụ: nội dung của trang web), do query thường ngắn, mang ít ý nghĩa nhưng lại có độ nhập nhằng cao. Ta có thể thấy, cùng một query gửi đến máy tìm kiếm nhưng lại hướng đến những mục đích hoàn toàn khác nhau.Ví dụ : query "*java*" có thể tìm về đảo Java hoặc ngôn ngữ lập trình Java. Hay các query khác nhau nhưng lại có cùng mục đích tìm kiếm.Ví dụ: *"đại học công nghệ"* và *"college of technology"* cùng hướng tới trang coltech.vnu.edu.vn.

Một vài phương pháp phân cụm cho query được sử dụng trong máy tìm kiếm (ví dụ, Encarta, AOL), dựa trên mối quan hệ giữa query và url được click:

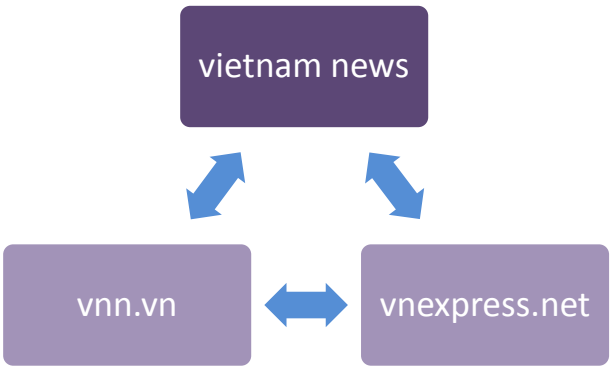
***Phương pháp 1*:** Theo Beeferman trong [11], việc phân cụm được dựa vào hai nhận xét về quan hệ giữa query và url được click:

- Nhận xét 1: *Nếu hai url khác nhau được click bởi cùng một query thì chúng có quan hệ với nhau* . Ví dụ: hình 12.



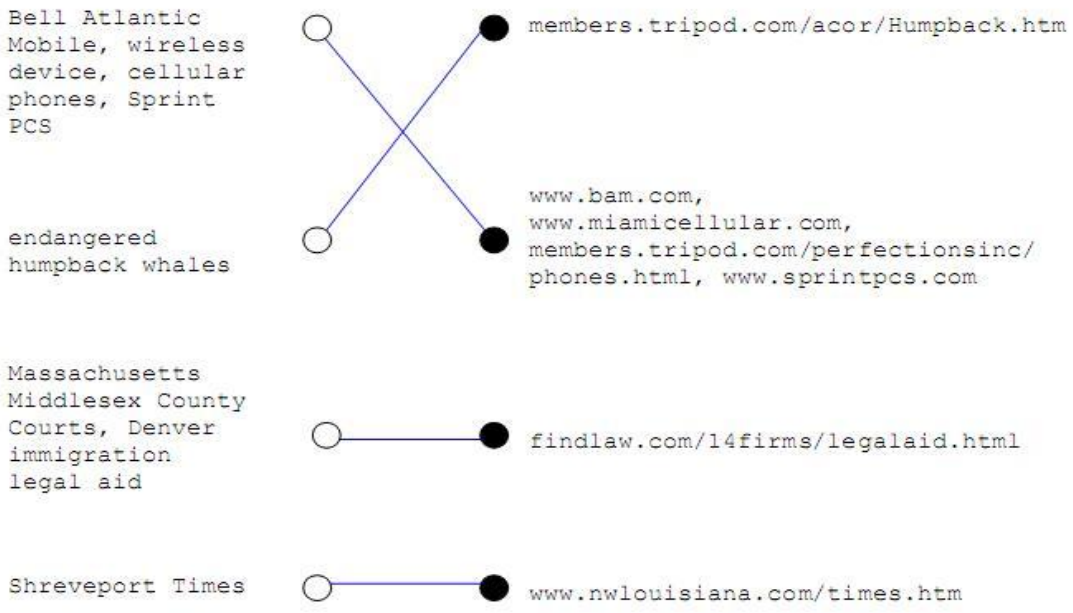
Hình 12. Quan hệ giữa 2 query cùng click 1 url

- Nhận xét 2: *Nếu hai query khác nhau cùng click vào một url thì chúng có quan hệ với nhau.* Ví dụ: hình 13.



Hình 13. Quan hệ giữa 2 url được click bởi cùng 1 query

Phương pháp này có thể phân cụm đồng thời cả query và url. Kết quả thu được có dạng : một cụm query tương ứng với một cụm url. Ví dụ: hình 14.



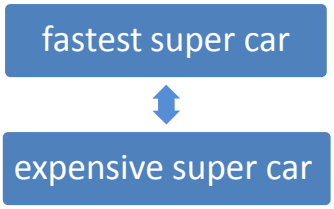
Hình 14. Đồ thị phân đôi query – url [11]

Độ tương đồng giữa các query và url được tính dựa vào độ tương đồng giữa các đỉnh trong đồ thị phân đôi. Với N(x), N(y) lần lượt là tập hợp các láng giềng (các đỉnh kề) của đỉnh x và y trong đồ thị; độ tương đồng của x và y được xác định bởi công thức:

$$\sigma(x,y) \stackrel{\text{def}}{=} \begin{cases} \dfrac{N(x) \cap N(y)}{N(x) \cup N(y)}, & if\ |N(x) \cup N(y)| > 0 \\ 0, & ngược\ lại \end{cases}$$

***Phương pháp 2*:** Được Wen, Nie và Jiang đưa ra trong [30], phương pháp này sử dụng 2 nhận xét về nội dung query và quan hệ của nó với url được click :

- Nhận xét 1 (sử dụng nội dung query)*: Nếu hai query chứa các từ giống nhau hoặc tương tự nhau, thì chúng có quan hệ với nhau.* Ví dụ: hình 15.



Hình 15. Hai query có chứa từ tương tự nhau

- Nhận xét 2 (sử dụng url được click)*: Nếu hai query khác nhau cùng click vào một url thì chúng có quan hệ với nhau.* Ví dụ: hình 12.

Độ tương tự dựa trên nội dung truy vấn (**similarity$_{\text{w-keyword}}$**) có thể sử dụng các độ đo trong các phương pháp phân cụm thông thường, như độ đo cosin:

$$similarity_{w-keyword}(p,q) = \frac{\sum_{i=1}^{k} cw_i(p) * cw_i(q)}{\sqrt{\sum_{i=1}^{m} w_i^2(p)} * \sqrt{\sum_{i=1}^{n} w_i^2(q)}}$$

Trong đó:

- **cw$_i$(p), cw$_i$(q)** là trọng số của từ khóa chung thứ i trong query *p* và *q*
- **w$_i$(p)** là trong số từ khóa thứ i trong query *q*. Trọng số từ khóa có thể sử dụng độ đo TF-IDF.

Độ tương tự dựa trên url được click (**similarity$_{\text{single-doc}}$**) được tính bởi công thức:

$$similarity_{single-doc}(p,q) = \frac{RD(p,q)}{Max(rd(p), rd(q))}$$

Trong đó:

- **RD(p,q)** là số lượng url cùng đươc click bởi cả query *p* và *q*.
- **rd(p), rd(q)** là số lượng url được click bởi mỗi query *p* và *q*.

Độ tương đồng này rất hữu ích để xác định các query khác nhau nhưng hướng tới nội dung gần nhau.

Hai phương pháp tính độ tương đồng trên tuy khác nhau, nhưng trong phân cụm query thì hai phương này lại bổ sung, hỗ trợ cho nhau. Vì vậy ta có công thức độ tương đồng tổng hợp:

similarity = a * $\textbf{similarity}_{\textbf{w-keyword}}$ + b * $\textbf{similarity}_{\textbf{single-doc}}$

(các hệ số **a, b** được xác định qua thực nghiệm).

***Phương pháp 3*:** Để giải quyết vấn đề query ngắn và ít ngữ nghĩa, query được làm giàu (bổ sung thông tin) trước khi phân cụm [1]. Có hai cách để làm giàu query:

- Sử dụng url được click: Thêm các url được click vào nội dung query nhằm làm rõ hơn mục đích mà query hướng tới.
  - Ví dụ: query *"britney spears"*, click vào *britneyspearsperfume.net* → sẽ được biểu diễn lại thành: *"britney spears britneyspearsperfume.net"*. Như vậy query này hướng tới một loại mỹ phẩm chứ không phải thông tin về một ca sĩ.
- Sử dụng phân tích chủ đề ẩn: Xác định các từ trong query thuộc vào chủ đề (topic) nào, qua đó làm rõ nội dung của query.
  - Ví dụ: query *"putin annual income"* → được bổ sung chủ đề mà các từ thuộc vào: *"putin politics annual income finance"*. Có thể thấy query này hướng tới nội dung kinh tế (finance) và chính trị (politics).

Query sau khi được bổ sung thông tin sẽ được phân cụm bởi các phương pháp thông thường như: Kmean, HAC, …

3.1.2. Phân tích chủ đề ẩn

3.1.2.1. Mô hình phân tích chủ đề

Phân tích chủ đề là một bước tiến quan trọng trong mô hình hóa văn bản. Nó dựa trên ý tưởng:

- Mỗi văn bản (document) là một phân phối xác suất theo chủ đề (topic)
- Mỗi chủ đề lại là một phân phối theo từ (word).

Biểu diễn từ và văn bản bằng phân phối xác suất có nhiều ưu điểm quan trọng so với phương pháp *Mô hình không gian đơn giản* (simple space model). Ý tưởng cơ bản của mô hình chủ đề là sử dụng một tiến trình xác suất để sinh ra văn bản mới.

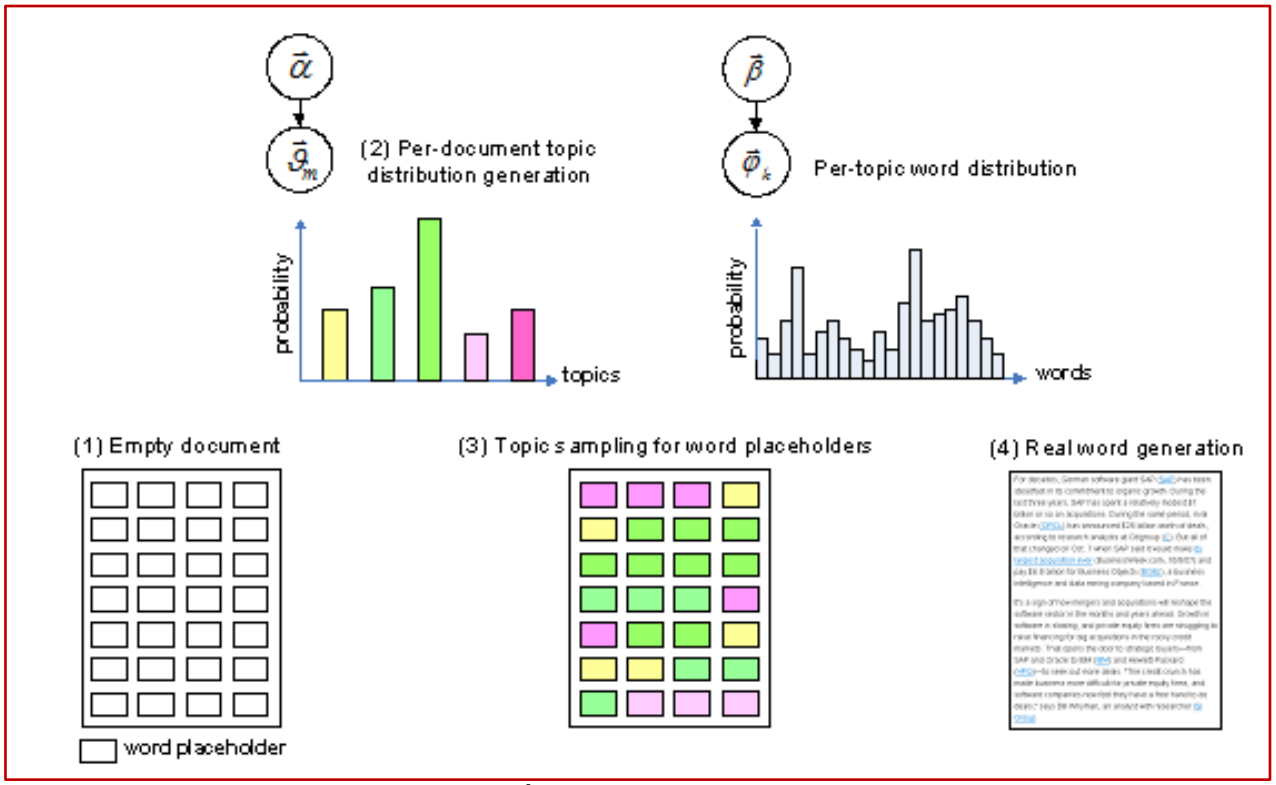
Đầu tiên, để tạo một văn bản mới, ta chọn một phân phối chủ đề cho văn bản. Nghĩa là mỗi văn bản được tổng hợp bởi nhiều chủ đề khác nhau với phân phối khác nhau.

Sau đó, để sinh ra các từ trong văn bản, ta chọn ngẫu nhiên các từ dựa trên phân phối của từ trên những chủ đề đã được chọn ở bước trước. Tiến trình sinh văn bản được minh họa trong hình sau:

Ngược lại, với một tập văn bản cho trước, ta có thể tìm ngược lại được tập chủ đề đã sinh ra các văn bản đó và tính được phân phối của các từ trong mỗi chủ đề. Các phương pháp thống kê được sử dụng để mô hình hóa tiến trình sinh văn bản và ước lượng các tham số trong mô hình. Hai ví dụ về phân tích chủ đề sử dụng mô hình ẩn là *Phân tích xác suất ngữ nghĩa ẩn* (probabilític latent semantic analysis – pLSA) và *Phân phối dirichlet ẩn* (Latent Dirichlet Allocation).

pLSA còn được biết đến như *Đánh chỉ mục xác suất ngữ nghĩa ẩn* (probabilistic latent semantic indexing – pLSI), là một kĩ thuật thống kê để phân tích các dữ liệu thường xuất hiện cạnh nhau. Nó được phát triển dựa trên LSA và được bổ sung thêm mô hình xác suất. pLSA mô hình hóa xác suất của các dữ liệu đồng xuất hiện như là một phân phối đa thức độc lập có điều kiện (conditionally independent multinomial distributions).

Theo Blei, Ng [13], dù pLSA một bước tiến trong việc mô hình hóa text theo xác suất nhưng nó chưa hoàn thiện. Lí do là pLSA chưa phải là một mô hình xác suất được xác định rõ ràng ở mức văn bản (document). Hệ quả là nó gặp vấn đề khi xác định xác suất với những văn bản nằm ngoài tập huấn luyện (trainning set). Hơn nữa, nó còn dẫn tới việc tăng tuyến tính số tham số của mô hình so với độ lớn của tập văn bản (corpus). LDA là mô hình phân tích chủ đề có thể xử lý được những vấn đề đó. Vì thế tôi đã chọn LDA để sử dụng trong khóa luận. Hình 16 giới thiệu những bước cơ bản trong tiến trình sinh của LDA.

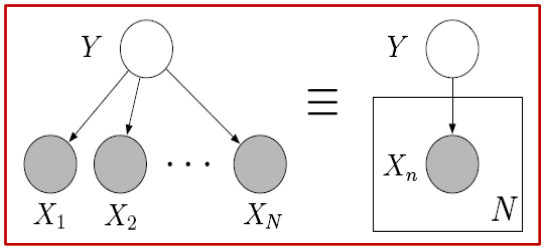


Hình 16. Tiến trình sinh văn bản LDA [2]

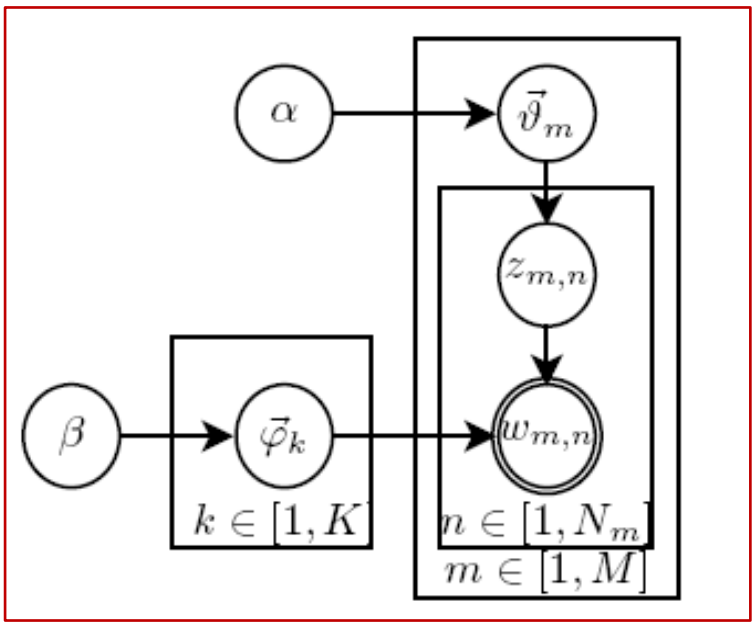
3.1.2.2. Phân phối Dirichlet ẩn (Latent Dirichlet Allocation)

LDA là mô hình sinh văn bản được giới thiệu bởi Blei, Ng và cộng sự [13] với pLSA về ý tưởng cơ bản là dựa trên việc coi văn bản là sự pha trộn của các chủ đề. Nhưng LDA là một mô hình Bayes ba mức: mức corpus, mức văn bản (document), mức từ (word).

Hình 17 & 18 mô tả tiến trình sinh văn bản bằng phương pháp LDA:



Hình 17. Kí hiệu khối lặp lại [13]



Hình 18. Mô hình LDA [13]

Các kí hiệu:

M : số văn bản trong corpus: $D = \{d_1, d_2, ..., d_M\}$

K : số chủ đề

V : số từ trong tập từ vựng

$\mathbf{N_m}$: độ dài của văn bản d_m

$\mathbf{z_{m,n}}$: chủ đề của từ w_n trong văn bản d_m

$\mathbf{w_{m,n}}$: từ thứ n trong văn bản d_m.

α và β : tham số ở mức corpus

$\vec{\vartheta}_m$: phân phối chủ đề trên mỗi văn bản d_m

$\vec{\varphi}_k$: phân phối từ trên chủ đề của k

Trong mô hình trên, mỗi khối thể hiện sự lặp lại. Khối ngoài cùng thể hiện văn bản (tập corpus gồm M văn bản). Khối trong thể hiện sự lặp lại việc chọn chủ đề ($z_{m,n}$) và từ ($w_{m,n}$) trong mỗi văn bản.Với văn bản d_m:

- Chọn $\vec{\vartheta}_m$ » Dirichlet(α)
- Với mỗi từ trong văn bản $w_{n,m}$ ($n \in [1, N_m]$):

- Chọn topic $z_{n,m}$ » Multinomial($\vec{\vartheta}_m$)
- Chọn từ $w_{n,m}$ từ xác suất $p(w_{n,m}|z_{n,m}, \vec{\varphi}_k)$, xác suất đa thức có điều kiện trên topic $z_{n,m}$.

Ngược lại, khi có văn bản cho trước, việc suy luận và ước lượng tham số cho mô hình sinh được thực hiện bằng phương pháp lấy mẫu Gibbs với công cụ JGibbsLDA.

3.2. Mô hình

3.2.1. Mô hình tổng quan

Mô hình hệ tư vấn website cho máy tìm kiếm gồm hai phần chính: xử lý online và offline.

3.2.1.1. Phần xử lý ngoại tuyến (offline)

Input: Tập query logs

Output: Các cụm query + tập url tư vấn tương ứng cho từng cụm

- Bước 1: Tiền xử lý query và url
  - Đưa query và url về một chuẩn thống nhất
  - Loại bỏ các query trùng lặp
- Bước 2: Phân cụm tập query
  - Làm giàu (bổ sung thông tin) cho query
  - Phân các query tương đồng vào cụm
- Bước 3: Xác định tập url có thể dùng để tư vấn cho từng cụm
  - Chọn ra tập url tốt để đại diện cho cụm

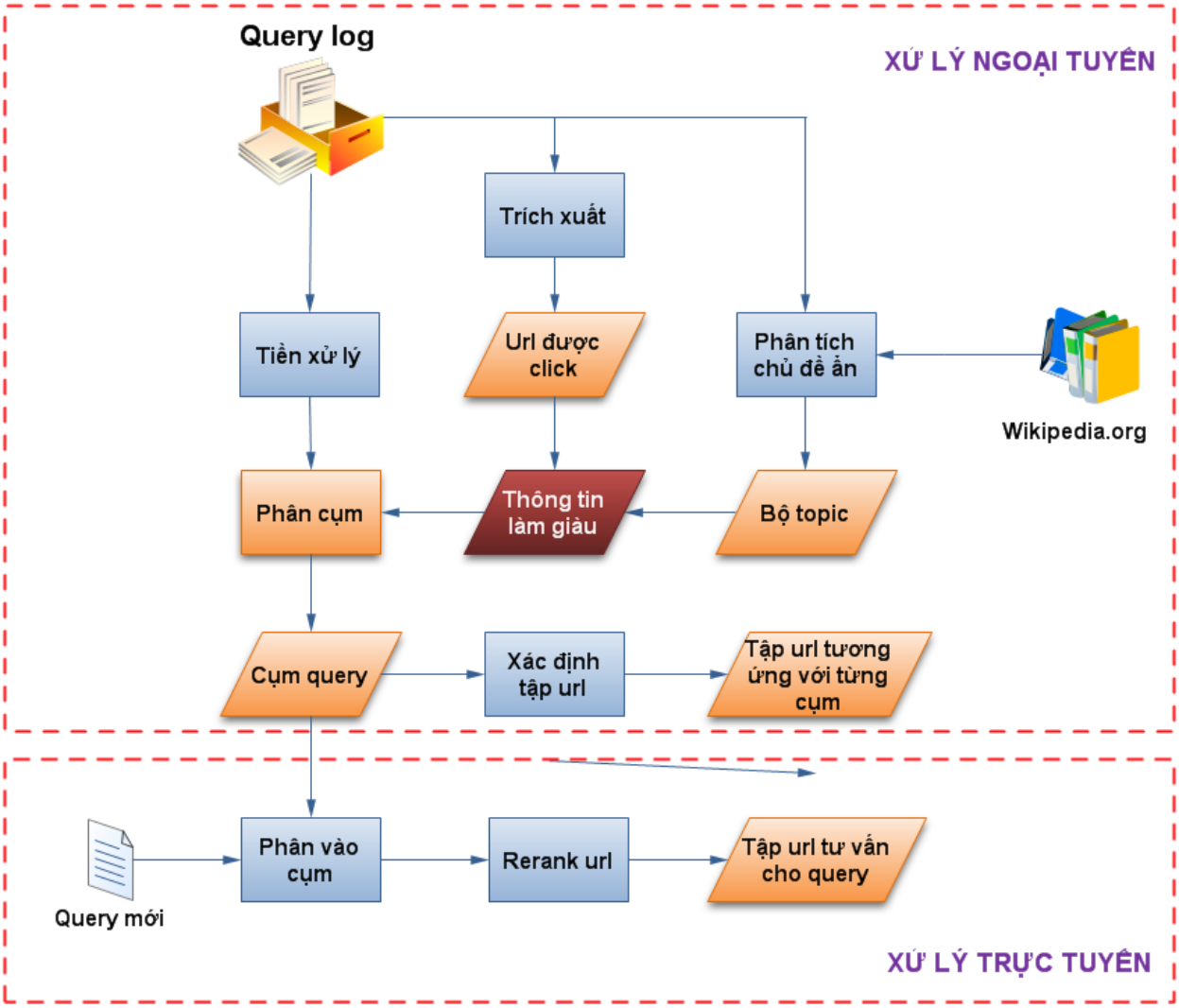
3.2.1.2. Phần xử lý trực tuyến (online)

Input: Query mới

Output: Tập url tư vấn tương ứng với query

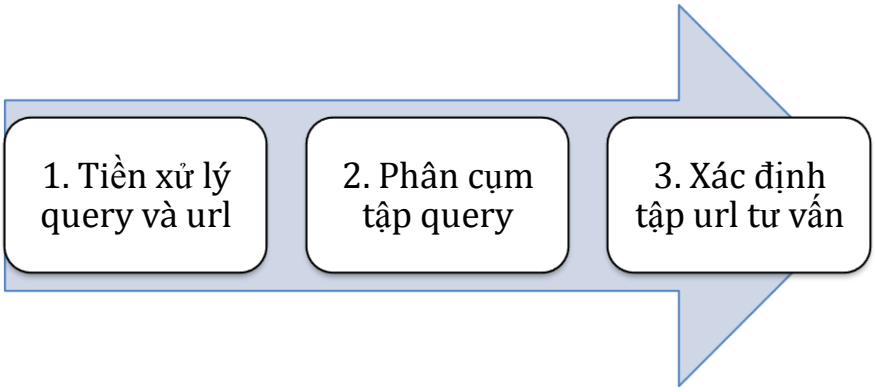
- Bước 1: Tiền xử lý query
  - Đưa query về dạng thống nhất
- Bước 2: Phân cụm query mới
  - Phân query vào một trong các cụm đã có

- Bước 3: Xếp hạng lại (rerank) tập url tư vấn trong cụm theo query mới.
  - Đưa ra N url thích hợp nhất (có hạng cao nhất) với query đó



Hình 19. Sơ đồ hệ thống tư vấn website

3.2.2. Phần xử lý ngoại tuyến



Hình 20. 3 bước xử lý ngoại tuyến

3.2.2.1. Tiền xử lý

Query và url được đưa về dạng chuẩn, thống nhất

- Query:
  - Đưa về chữ thường.
    - Ví dụ: *"New York Major"*→ *"new york major"*
  - Loại bỏ từ dừng (stop word).
    - Ví dụ: *a, an, the …*
  - Loại bỏ các kí tự đặc biệt
    - Ví dụ: **+ − ~ !** …
  - Đưa về từ gốc (stemming).
    - Ví dụ: *ladies* → *lady*, *playing* → *play*, *cooked* → *cook …*

Sau đó, các query trùng với query đã có sẽ bị loại bỏ vì chúng không có ý nghĩa trong việc phân cụm.

- Url: chỉ giữ lại domain chính, bỏ giao thức và các đường dẫn phía sau
  - Ví dụ: *http://www.vnexpress.net/thethao/...* → *vnexpress.net*

3.2.2.2. Phân cụm tập query

Query được làm giàu (bổ sung thông tin) trước khi phân cụm. Có hai cách cách làm giàu query được sử dung:

- Sử dụng url được click: Thêm các url được click vào nội dung query nhằm làm rõ hơn mục đích mà query hướng tới.

- Sử dụng bộ chủ đề ẩn: Xác định các từ trong query thuộc vào chủ đề (topic) nào, qua đó làm rõ nội dung của query.

Query sau khi được bổ sung thông tin sẽ được phân cụm bởi các phương pháp thông thường như: Kmean, HAC, … Ở đây, Kmean được chọn vì độ phức tạp chỉ là O(n*log n) (của HAC là $O(n^2)$)

- Các query được biểu diễn ở dạng vector trọng số từ TF.
- Độ tương đồng giữa hai query được tính bằng độ đo cosin. Xét 2 query: **Q** = (q_1 , q_2,…, q_n) và **P** = (p_1 , p_2,…, p_n); trong đó q_i và p_i lần lượt là trọng số của từ i trong query **Q** và **P**.Khi đó độ tương tự giữa query **P** và **Q** là:

$$similarity(P,Q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2}$$

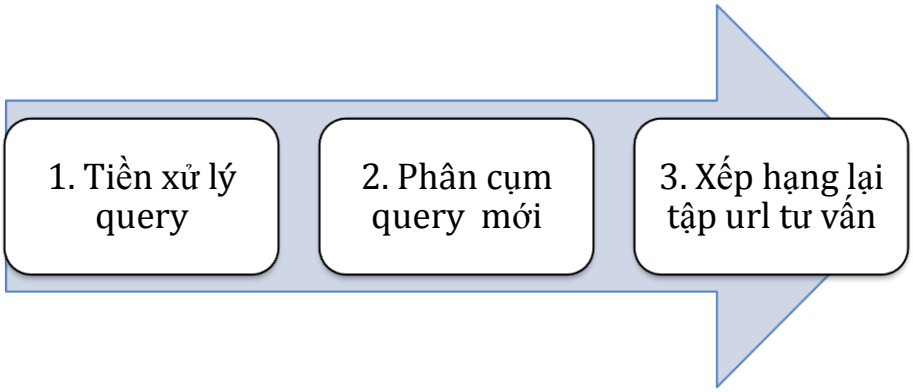
- Số cụm được xác định dựa trên khảo sát tập query đầu vào.

3.2.2.3. Xác định tập url tư vấn

Mục tiêu ở bước này là chọn ra tập url tốt để đại diện cho mỗi cụm.

Một cách đơn giản là chọn ra những url được click nhiều hơn một ngưỡng θ nào đó. Ko phải ngẫu nhiên mà người dùng lại click vào một url, họ chỉ chọn khi thấy nội dung của nó sát với những gì họ đang tìm kiếm.Vì vậy, khi một url (website) được click nhiều lần bởi nhiều người dùng, chứng tỏ nó có độ tương đồng cao với nội dung, chủ đề của cả cụm.

3.2.3. Phần xử lý trực tuyến



Hình 21. 3 bước xử lý trực tuyến

3.2.3.1. Tiền xử lý query

- Query được :
  - Đưa về chữ thường.
  - Loại bỏ từ dừng (stop word).
  - Loại bỏ kí tự đặc biệt
  - Đưa về từ gốc (stemming).

3.2.3.2. Phân cụm query mới

- Biểu diễn query dưới dạng vector trọng số từ TF
- Làm giàu thông tin cho query.
- Phân query vào một trong các cụm đã có bằng cách:
  - Tính khoảng cách từ vector biểu diễn query tới các vector tâm cụm
  - Query được phân vào cụm có khoảng cách giữa nó với tâm cụm là nhỏ nhất

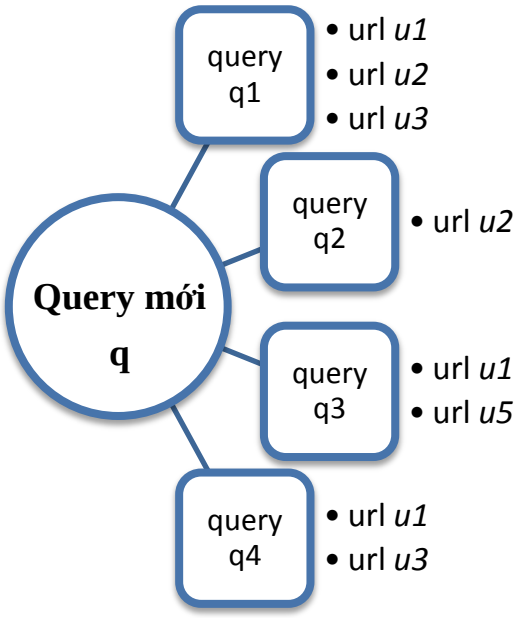
3.2.3.3. Xếp hạng lại tập url tư vấn

- Các url trong tập url tư vấn được xếp hạng lại (rerank) theo query mới. Để xếp hạng url, tôi đưa ra cách xác định giá trị hạng dựa vào:
  - Độ tương đồng giữa query mới **q** với các query $\mathbf{q_i}$ ($i = \overline{1,n}$) đã có trong cụm theo độ đo cosin, kí hiệu: **sim(q, $\mathbf{q_i}$)**
  - Giả sử url **u** trong tập url tư vấn, được chọn (click) bởi người dùng sử dụng query $\mathbf{q_1}$, $\mathbf{q_2}$ thì hạng của **u** được tính bởi công thức:

$$rank(u) = \frac{1}{sim(q, q_1)} + \frac{1}{sim(q, q_2)}$$

Các url có **rank(u)** càng cao thì càng phù hợp với query **q** và được đưa lên trước trong danh sách tư vấn. Hình 22 mô tả việc xếp hạng url dựa vào độ tương đồng giữa các query.

- Top-N url có hạng cao nhất được sử dụng để tư vấn cho người dùng



Hình 22. Sử dụng quan hệ giữa các query để tính hạng url

Chương 4. Thực nghiệm và đánh giá

4.1. Môi trường

Môi trường thực nghiệm:

Thành phần	Chỉ số
CPU	Core 2 Duo T7500 2.2Ghz
RAM	2 GB
HDD	250 GB
OS	Vista Ultimate 64 bit

Bảng 9. Môi trường thực nghiệm

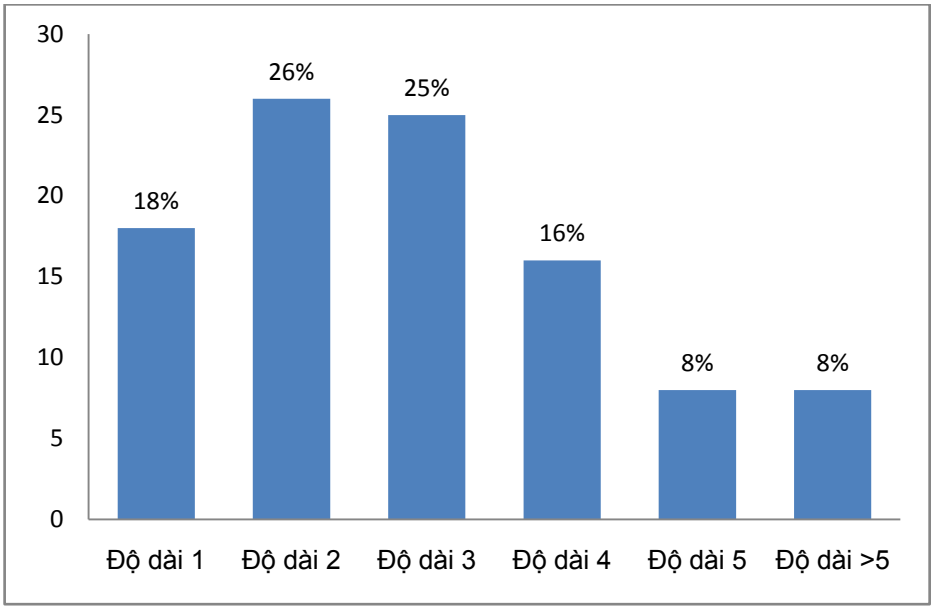
4.2. Dữ liệu và công cụ

- Dữ liệu: **1 GB** query logs được lấy từ máy tìm kiếm MSN, với **12 triệu** query & url được click. Các query đều bằng tiếng anh.

QueryID	Query	Time	URL	Position
0000003a718649f2	schwab	2006-05-11 08:07:35	http://www.schwab.com/	1
0000006d43b549c1	us geography	2006-05-04 14:23:00	http://www.enchantedlearning.com/usa/	3
0000006d43b549c1	us geography	2006-05-04 14:23:03	http://www.sheppardsoftware.com/states.html	4
0000016aa52e4fbc	wwf	2006-05-21 09:25:34	http://www.panda.org/	2
000002aa6e27443f	biggercity	2006-05-07 13:30:45	http://www.biggercity.com/chat/	1
000005aac1f6423f	shawnee studios	2006-05-09 14:21:29	http://www.shawneestudios.com/contact us.php	1
000007697d5645ab	HARRYS FLOWERS	2006-05-19 05:06:33	http://search.msn.com/local/details.aspx	5

Hình 23. Một phần query log của MSN [20]

Qua khảo sát tập query logs, nhận thấy phần lớn query có độ dài từ 2-3. Query có độ dài 1 chiếm tới 17.6 %, query có độ dài 2 chiếm 25.9% và query có độ dài 3 chiếm 25.1%. Độ dài trung bình của query là 2.79.



Hình 24. Phân bố chiều dài query trong MSN log [1]

- Công cụ:
  - Tìm chủ đề ẩn: JGibbsLDA [22]
  - Phân cụm: Lingpipe [23]
  - Các thành phần khác của hệ tư vấn (tiền xử lý, xác định tập url tư vấn, rerank url…): Bộ công cụ do tôi tự xây dựng.

4.3. Thực nghiệm

Để thử nghiệm, tôi tiến hành xây dựng hệ tư vấn cho các query liên quan tới miền **sản phẩm điện tử** vì:

- Tập **12 triệu** query rất lớn và hướng tới rất nhiều nội dung khác nhau.
- Do hạn chế về độ phức tạp nên các công cụ tìm chủ đề ẩn, phân cụm.. chỉ có thể xử lý được vài chục đến vài trăm nghìn query.
- Miền tri thức nhỏ sẽ cho kết quả phân cụm tốt hơn (do các query trong cụm gần nhau hơn).
- Miền sản phẩm điện tử được nhiều người quan tâm

Do đó cần có thêm bước lọc nội dung query.

4.3.1. Lọc nội dung query

- Tập query logs được loại bỏ các trường không liên quan (trường thời gian, query ID, vị trí của url trong danh sách kết quả), chỉ giữ lại query và các url được click.
- Xác định tập sản phẩm gồm các loại sản phẩm như: máy tính, điện thoại, ti vi, đầu cd, máy ảnh… Mỗi loại sản phẩm được xác định bởi tập các keyword tương ứng.

Sản phẩm	Từ khóa
Máy tính	computer laptop notebook netbook monitor lcd crt hdd “hard disk” “floppy disk” cdrom “dvd drive” “optical drive” cpu “dual core” “core duo” amd intel mainboard motherboard vga “graphic card” ram keyboard mouse webcam linux ubuntu fedora redhat solaris “mac os” “windows xp” antivirus router firewwall modem adsl wifi lan wan dell “hp computer” lenovo asus “sony vaio” …
Điện thoại	mobile pda “smart phone” “cell phone” nokia “samsung mobile” “lg mobile” “sony erricsion” iphone blackberry gsm cdma …
Các thiết bị khác	camera recorder nikkon kodak fujifilm “vcd player” “dvd player” tv television “plasma tv” “satellite tv” “cable tv”…

Bảng 10. Một số từ khóa liên quan tới miền sản phẩm điện tử

- Những query chứa một trong các từ khóa trên sẽ trở thành tập input cho hệ thống tư vấn. Sau khi lọc, tập kết quả thu được gồm 2639 query.

4.3.2. Xử lý offline

4.3.2.1. Tiền xử lý

- Tiền xử lý với query: đưa về chữ thường, loại bỏ từ dừng, loại bỏ các kí tự đặc biệt, đưa về từ gốc.
- Tiền xử lý với các url: chỉ giữ lại domain chính (cnn.com, bbc.com…)

4.3.2.2. Phân cụm tập query

- Làm giàu (bổ sung thông tin) cho query
  - Cách 1: Làm giàu query bằng các url được click.
  - Cách 2: Làm giàu query bằng bộ chủ đề ẩn. Hai bộ chủ đề được sử dụng:
    - Bộ 1: có sẵn, được công bố ở [22], xây dựng dựa trên các tài liệu lấy từ en.wikipedia.org.
      - Đặc điểm của các tài liệu này: dài, từ vựng phong phú, đầy đủ ngữ nghĩa.
      - Gồm **200** chủ đề, mỗi chủ đề có **200** từ
    - Bộ 2: do chúng tôi xây dựng dựa trên chính tập query có được sau bước nội dung lọc trên miền liên quan tới sản phẩm điện tử (gồm 2639 query). Công cụ sử dụng là JGibbsLDA [22].
      - Đặc điểm của query: ngắn, ít ngữ nghĩa, nhập nhằng cao.
      - Gồm **10** chủ đề, mỗi chủ đề có **100** từ.

- Phân cụm sử dụng Kmean (công cụ Lingpipe [23]):
  - Dựa trên số lượng tập query đầu vào, chúng tôi chọn số cụm là 10 do:
    - Nếu số cụm quá ít → độ gần nhau giữa các query trong một cụm sẽ giảm.
    - Nếu số cụm quá nhiều → số lượng query trong mỗi cụm giảm → tần suất lặp lại của các url thấp, dẫn đến việc không tìm được những url tốt để đại diện cho nội dung của cụm.

- Thực nghiệm phân cụm được thực hiện với 3 trường hợp:
  - Không làm giàu query
  - Làm giàu query bằng url được click
  - Làm giàu query bằng bộ chủ đề ẩn (2 bộ)

Nhận xét:

✓ Khi không làm giàu thông tin cho query; chất lượng phân cụm kém do độ gần nhau giữa các query trong cụm thấp (vì các query mang ít thông tin về mặt ngữ nghĩa) và độ tách rời giữa các cụm là không cao.

✓ Khi làm làm giàu thông tin cho query bằng cách thêm các url được click vào cuối của query thì kết quả đạt được là tốt hơn. Nó có thể nhận dạng được các query khác nhau nhưng cùng có một mục đích, hoặc query giống nhau nhưng hướng tới những mục đích khác nhau. Ví dụ: với query *"sf.net"* và *"sourcefore"* sau khi qua bước này sẽ được chuyển thành *"sf.net sourcefore.net"* và *"sourcfore sourcfore.net"*. Rõ ràng là các query này có mối quan hệ mật thiết với nhau.

Tuy nhiên phương pháp này gặp một vấn đề lớn; với một query mới mà người dùng gửi đến máy tìm kiếm thì sẽ không thể làm giàu thông tin được cho nó (vì không biết người dùng sẽ click vào url nào) và dẫn đến việc phân cụm sai.

✓ Khi dùng chủ để ẩn để làm giàu thông tin cho query; thực nghiệm được tiến hành trên cả hai bộ topic. Một bộ topic được sinh từ các văn bản lấy từ wikipedia.org; một là bộ topic được sinh trực tiếp từ các query trong query logs. Bộ topic lấy từ wikipedia do dựa trên những văn bản dài và nhiều thông tin hơn nên ngữ nghĩa của các từ trong topic khá gần nhau và tốt hơn hẳn so với bộ topic sinh từ chính query logs (do các query ngắn và mang ít thông tin).
Nhưng khi thực hiện phân cụm, bộ topic lấy từ wikipedia cho một kết quả không cân xứng: có 1 cụm có **690** query (hơn 1/4 tổng số query); 5 cụm chỉ có từ **50-90** query (không tới 1/20 tổng số query); 5 cụm còn lại trung bình **300** query/cụm. Nguyên nhân là do các từ trong tập query ít trùng lặp với các từ trong bộ topic lấy từ wikipedia nên dẫn đến trường hợp một lượng lớn query không có thuộc tính topic để bổ sung và dồn hết vào một cụm. Lúc này bộ topic thứ hai (lấy từ chính

query logs) tuy ngữ nghĩa kém hơn nhưng lại cho kết quả phân cụm tốt hơn. Với 2639 query đầu vào và 10 cụm; mỗi cụm có khoảng 200-300 query.

Bảng tổng hợp:

Query	Không làm giàu	Bổ sung url	Bổ sung chủ đề ẩn	
			Bộ chủ đề Wikipedia	Bộ chủ đề query logs
Ưu điểm		Phân biệt được: Query giống nhau nhưng khác mục đích. Query khác nhau nhưng cùng mục đích	Chất lượng (ngữ nghĩa) tốt hơn	Đều: mỗi cụm 200-300 query
Nhược điểm	Độ gần nhau trong cụm thấp. Độ tách rời giữa các cụm thấp.	Không thể bổ sung thông tin url được click cho query mới	Xuất hiện 1 cụm đột biến (700 query) và 4 cụm có ít hơn 100 query/cụm	

Bảng 11. Tổng hợp thực nghiệm phân cụm query

4.3.2.3. Xác định tập url tư vấn

Tập url tư vấn của một cụm là tập các url có số lần xuất hiện trong cụm lớn hơn ngưỡng θ. Chọn θ là số lần xuất hiện trung bình của các url trong một cụm.

Nếu **θ < 2** (số lần xuất hiện trung bình của các url trong cụm quá thấp) thì đặt lại **θ = 2**.

4.3.3. Xử lý online

Sau khi query mới được phân vào cụm; các url trong tập tư vấn của cụm được xếp hạng lại dựa trên công thức dưới đây (đã được trình bày ở mục *3.2.3.3*). Ba url có hạng cao nhất sẽ được dùng để tư vấn.

$$rank(u) = \frac{1}{sim(q, q_1)} + \frac{1}{sim(q, q_2)}$$

4.4. Đánh giá

Xây dựng bộ test gồm: 10 query, mỗi query có 5 url mà người dùng mong muốn nhận được. Các query này được đưa vào hệ thống với vai trò query mới của người dùng. Bảng 11 là kết quả được hệ thống trả lại. Độ chính xác của hệ thống được tính bằng tỉ lệ url tư vấn trùng với url mong muốn của người dùng.

Query	Url mong muốn (3-5 url/query)	Url được hệ thống tư vấn (3 url/query)	Độ chính xác của hệ thống
direct tv guide	directv.com direct-tv-guide.org online.tvguide.com tv.com tvguide.com	tv.com direct-tv-guide.org online.tvguide.com	**100%**
cell phone directory	cellpages.com phonedirectorysearch.com cellphoneshop.net reversephonedirectory.com phoneaddressdirectory.com	phonedirectorysearch.com cellphoneshop.net newyorkcellphone.com	**66%**
live tv guide	tvguide.com imdb.com tvguidemagazine.com wwitv.com tvguidemagazine.com	tv.com tvguide.com imdb.com	**66%**
internet explorer	microsoft.com msdn.microsoft.com download.cnet.com en.wikipedia.org	microsoft.com msdn.microsoft.com download.cnet.com	**100%**
lcd tv reviews	tv.com lcdtvbuyingguide.com reviews.cnet.com digitaladvisor.com lcd-tv-reviews.com	tv.com lcdtvbuyingguide.com en.wikipedia.org	**66%**
multimedia	reviews.cnet.com	microsoft.com	**33%**

keyboard	en.wikipedia.org www.computerworld.com	en.wikipedia.org techwarelabs.com	
mobile home	en.wikipedia.org mobilehomedoctor.com mobilehome.com	mobilehomeworks.com mobilehome.com 4-sale-mobile-home-park.com	**33%**
digital camera	digitalcamera-hq.com dpreview.com reviews.cnet.com kodak.com	dpreview.com kodak.com digicamera.com	**66%**
mp3 player	creative.com apple.com portableplayerz.com	creative.com apple.com anythingbutipod.com	**66%**
external hard drive	seagate.com amazon.com pcmag.com tomshardware.com	amazon.com seagate.com wdc.com	**66%**

Bảng 12. Bảng kết quả thực nghiệm

Nhận xét:

Độ chính xác trung bình của hệ thống là: **66%**. Đây là một kết quả khá tốt dù tập dữ liệu sau khi lọc khá nhỏ, chỉ vài nghìn query.

Kết luận và định hướng

- Với các kết quả đã đạt được, khóa luận đã đóng góp:
  - Xây dựng mô hình hệ tư vấn website mới dựa trên khai phá kinh nghiệm của người dùng. Các kĩ thuật được dùng:
    - Phân cụm query logs, sử dụng phân tích chủ đề ẩn để làm giàu thông tin cho query.
    - Kĩ thuật xếp hạng website (url) tư vấn theo query đầu vào.
  - Thử nghiệm ban đầu trên miền liên quan tới sản phẩm điện tử cho kết quả khá tốt.

- Những vấn đề còn tồn tại:
  - Khối lượng dữ liệu lớn dẫn tới việc những thuật toán tốt nhưng độ phức tạp cao không thể chạy được. Ví dụ HAC tốt hơn Kmean nhưng do độ phức tạp cao hơn ($O(n^2)$ so với O(n*logn)) nên phương pháp được sử dụng là Kmean.

- Định hướng phát triển:
  - Tìm cách sử dụng các phương pháp phân cụm khác có hiệu quả cao hơn.
  - Mở rộng ra các miền thông tin khác ngoài sản phẩm điện tử.
  - Tích hợp hệ thống vào máy tìm kiếm.

Tài liệu tham khảo

Tiếng việt

[1] Nguyễn Song Hà, Chu Anh Minh, Vũ Tiến Thành. Hệ tư vấn website cho máy tìm kiếm dựa trên khai phá query log, *Công trình sinh viên nghiên cứu khoa học*, Đại học Công Nghệ, ĐHQGHN, 2009

[2] Lê Diệu Thu. Online context advertising, *Luận văn tốt nghiệp đai học*, Đại học Công nghệ, ĐHQGHN, 2008.

Tiếng Anh

[3] ACM recommender system conference, http://recsys.acm.org

[4] G.Adomavicius, A.Tuzhilin. Towards the Next Generation of Recommender Systems:A Survey of the State-of-the-Art and Possible Extensions, *IEEE Transactions on Knowledge and Data Engineering, 2005*

[5] Agarwal G.Kabra Z.Zhang K.C.Chang. Mining Structured Query Templates from Search Logs. *University of Illinois at Urbana Champaign research, 2008.*

[6] Ansari, A., S. Essegaier, and R. Kohli. *Internet recommendations systems. Journal of Marketing Research, pages 363-375, 2000.*

[7] America Online (AOL) search engine log, 2006, http://www.aol.com

[8] R.Baeza, F.Silvestri. Web Query Log Mining, ACM SIGIR Conference tutorial, 2009

[9] Balabanovic, M. and Y. Shoham. Fab: Content-based, collaborative recommendation. *Communications of the ACM, 40(3):66-72, 1997*

[10] Basu, C., H. Hirsh, and W. Cohen. Recommendation as classification: Using social and content-based information in recommendation. *In Recommender Systems. Papers from 1998 Workshop. Technical Report WS-98-08. AAAI Press, 1998*

[11] D.Beeferman, A.Berger. Agglomerative clustering of a search engine query log. *In Proceedings of ACM SIGKDD International Conference . 2000*

[12] Billsus, D. and M. Pazzani. Learning collaborative information filters. *In International Conference on Machine Learning, Morgan Kaufmann Publishers, 1998.*

[13] D. Blei, A. Ng, and M. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993-1022, January 2003

[14] Breese, J. S., D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. *In Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, Madison, WI, 1998.*

[15] W.B. Croft. Query Evolution, University of Massachusetts Amherst lecture

[16] H Cui, JR Wen, JY Nie, WY Ma. Query expansion by mining user logs, - *IEEE transactions on knowledge and data engineering, 2003*

[17] HB.Deng. Introduction to Recommendation System, *China University of Hongkong seminar, 2006*

[18] Google Inc Search privacy, http://google.com/privacy.html

[19] Google Zeitgeist, http://www.google.com/intl/en/press/zeitgeist/index.html

[20] Microsoft Social Network (MSN) query log, http://www.msn.com

[21] Netflix online movie rental, http://www.netflix.com

[22] CT Nguyen, XH Phan, JGibbslda, A Java and Gibbs Sampling based Implementation of Latent Dirichlet Allocation, http://gibbslda.sourceforge.net/, 2007

[23] Lingpipe: suite of Java libraries for the linguistic analysis of human language,

http://alias-i.com/lingpipe/

[24] G Pass, A Chowdhury, C Torgeson. A picture of Search, *Proceedings of the 1st international conference on Scalable Information System, 2006*

[25] Popescul, A., L. H. Ungar, D. M. Pennock, and S. Lawrence. Probabilistic Models for Unified Collaborative and Content-Based Recommendation in Sparse-Data Environments. *In Proc. of the 17th Conf. on Uncertainty in Artificial Intelligence, Seattle, WA, 2001*

[26] K.N.Rao. Application Domain and Functional Classification of Recommender Systems—A Survey, *Journal of Library & Information Technology, Vol. 28, No. 3, pp. 17-35, 2008*

[27] Resnick, N. Iakovou, M. Sushak, P. Bergstrom, and J. Riedl. GroupLens: An open architecture for collaborative filtering of netnews. *In Proceedings of the 1994 Computer Supported Cooperative Work Conference, 1994*

[28] C Silverstein, M Henzinger, H Marais, M Moricz. Analysis of a Very Large AltaVista Query Log, *Compaq Systems Research Center, 1998*

[29] Soboroff, I. and C. Nicholas. Combining content and collaboration in text filtering. *In 43 IJCAI'99 Workshop: Machine Learning for Information Filtering, 1999*

[30] J.R.Wen, JY.Nie, H.Jiang. Query Clustering Using User Logs. *ACM Transactions on Information Systems, Vol. 20, No. 1, January 2002*

[31] Z Zhuang, S Cucerzan. Re-Ranking Search Results Using Query Logs