

## سوال (۱)

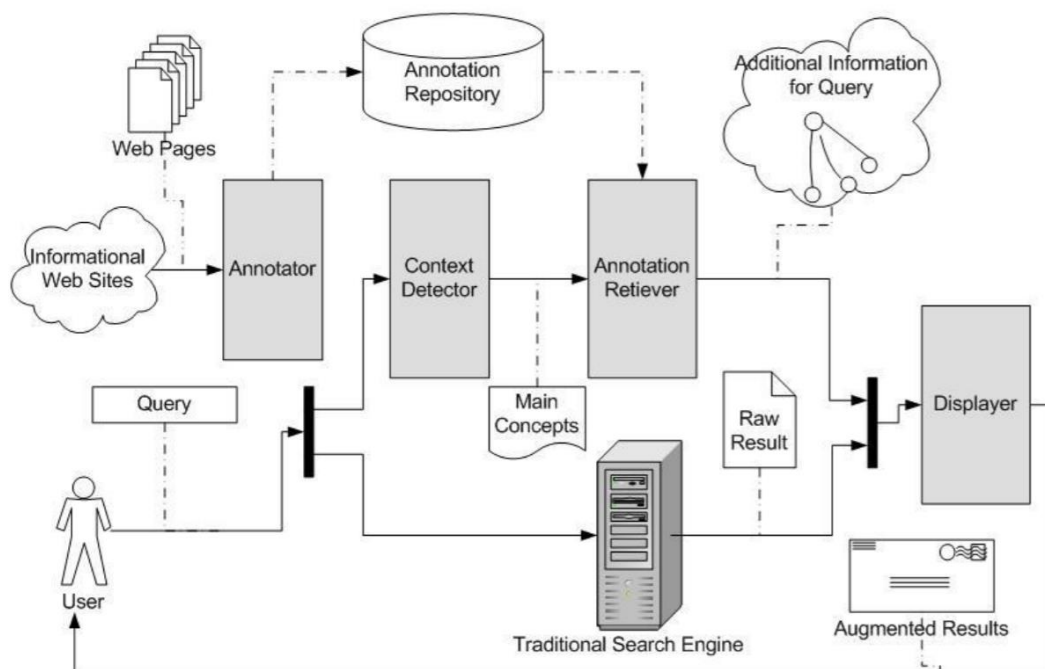
کاربر در فیلد ورودی سرچ گوگل ابتدا یک عبارت را وارد می‌کند، این عبارت که به آن query گفته می‌شود در ابتدا باید به یک سری token تبدیل شود. بزرگی و کوچکی token ها دارای اهمیت است چرا که وقتی کسی کلمه‌ای مانند Paris را ابتدا به این صورت می‌نویسد به این معناست که منظور آن شهر پاریس است. این توکن‌ها خود به سمت یک موتور سرچ traditional روانه می‌شود تا سرچ انجام شود. سپس باید از یک فیلتر نیز بگذرند که کلماتی که حرف اضافه هستند حذف شود. البته گوگل یک سری dork از قبل تعریف کرده است که آن‌ها را جداگانه پروسس می‌کند و کلماتی مانند and و or می‌تواند در آن‌ها معنی را کامل‌تر کند. برای مثال:

”filetype:xls “house prices” and “London

همچنین معنای کلمات نیز نسبت به یکدیگر نیز می‌تواند تعبیر شود برای مثال فعلی مانند change دارای چندین معنی از جمله exchange, replace و adjust می‌باشد و تنها با توجه به موقعیت آن در متن و کلمات کنار آن معنای آن سنجیده می‌شود. به همین دلیل نیاز است که کوئری به یک تشخیص دهنده متن نیز داده شود تا موضوع آن کوئری از نظر معنایی نیز مشخص شود.

پس در موتور جستجو علاوه بر بررسی لغوی به صورت معنایی نیز بررسی می‌شود تا مفهوم آن بهتر منتقل شود. تعداد زیادی patent توسط گوگل برای حل کردن این موضوع به ثبت رسیده است که از میان آن‌ها می‌توان به این [مورد](#) اشاره کرد. در شکل زیر روند کلی چنین موتورهای جستجویی مشخص شده است.

منبع شکل: اسلایدهای درس وب معنایی دکتر امینی



منبع مطالب سایت developers گوگل و مطالب کلاس وب معنایی دکتر امین

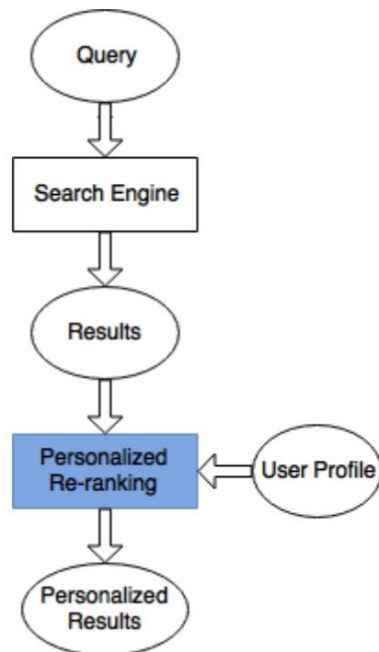
سوال ۲ و ۳)

باید گفت که گوگل با استفاده از crawler خودش دائماً در حال پایش و خزیدن پیج‌های متعدد و در حال annotate و index کردن آن‌هاست. حال که کاربر کوئری خود را وارد کرده است و این کوئری به یک سری index تبدیل شده است گوگل می‌تواند از دیتابیس خود پیج‌هایی که در آن‌ها آن index های مرتبط تکرار شده است را به کاربر نشان دهد. به این منظور الگوریتم ranking خود را بر روی آن‌ها اعمال کرده تا ترتیب نمایش به کاربر را درست کند. برای مرتب کردن آن‌ها از الگوریتم page-rank و همچنین tf-idf و میج کردن index ها استفاده می‌کند.

سوال ۴) گوگل برای هر کاربر یک نمایه تشکیل داده است که در آن نمایه میزان علاقه هر کاربر به هر بحث را مشخص کرده است. طوریکه من اگر کلمه java را بنویسم به من با پیشینه برنامه‌نویسی و سرچ‌های برنامه‌نویسی زبان java را بر میگرداند و برای یک شخص دیگر ک اهل سفر است جزیره java را نمایش می‌دهد.

گوگل این کار را با استفاده از عملیات re-ranking انجام می‌دهد به این صورت که پیج‌ها به صورت عادی rank می‌شود و نمایه کاربر که w در این الگوریتم است با استفاده از فرمول tf-idf با مجموعه رنک‌های اصلی شباهت گرفته می‌شود و بر اساس آن شباهت و رنک قبلی مجدداً sort انجام می‌شود. الگوریتم و دیاگرام تغییر در عملیات page-rank در زیر نمایش داده شده است.

منبع: اسلایدهای درس وب معنایی دکتر امینی



<b>Input:</b> User Profile $W$ and a set of search results with original ranks, $R$
<b>Output:</b> a set of search results with revised ranks $R = \{d_1, \dots, d_n\}$ , search results for query $q$ ; $d_i = (t_1, t_2, \dots, t_m)$ where $m$ is the number of distinct terms in the Web page $d_i$ and $t_k$ denotes each term
<b>For each</b> $d_i \in R$ <b>do</b> <b>For each</b> $t_k \in d_i$ <b>do</b> Calculate $tf(t_k, d_i)$ ; $d_{i_{tk}} = \frac{tf(t_k, d_i)}{\sum_{j=1}^m tf(t_j, d_i)}$ <b>End</b> Compute similarity between the user profile $P$ (vector of query topics with weight )and the feature vector of the $i_{th}$ Web page in search results, $d_i$ by dot product. $sim(W, d_i) = \frac{W \cdot d_i}{ W  \cdot  d_i }$ // for example for a user , $W$ is $\{ \{ java, 0.2333 \}, \{ mining, 0.453 \} \}$ <b>End</b> Arrange similarities according to Active User Profile in decreasing order.
<b>Output</b> -Set of search results $R$ with new Ranks