معماری یکپارچه برای موتورهای جستجوی با دامنه خاص

احمد عبداله زاده بارفروش
دانشیار

حمیدرضا مطهری نژاد
کارشناسی ارشد

دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیرکبیر

چکیده

میزان اطلاعات قابل دسترس بر روی اینترنت به صورت سرسام آوری در حال افزایش است. سرعت این افزایش چنان است که موتورهای جستجوی همه منظوره قادر به پوشش حتی نیمی از این اطلاعات نیستند. از آنجا که فقط بخش کوچکی از این اطلاعات به یک دامنه و یا موضوع مشخص ربط دارند، موتورهای جستجوی با دامنه خاص پیشنهاد شدند که یک راه حل نسبتاً مناسب برای دستیابی به میزان پوشش و دقت بالاتر به شمار می‌آیند ولی این دستاوردها کماکان در مکانیسم‌های پرس و جو از مشکلات عدیده‌ای رنج می‌برند.

در این مقاله یک معماری یکپارچه برای موتورهای جستجوی با دامنه خاص ارائه شده است که هدف آن بهبود مکانیسم‌های پرس و جوی کاربران از طریق توسعه خودکار پرس و جوی کاربر و یادگیری از جستجوهای قبلی است. به منظور توسعه خودکار پرس و جوی کاربر با اطلاعات مرتبط با حوزه مورد پرس و جو، مفهومی جدید به نام "سلسله مراتب مفهومی با دامنه خاص" معرفی و یک الگوریتم برای یادگیری آن برای دامنه‌های خاص طراحی، پیاده سازی و ارزیابی گردیده است. نتایج بکارگیری این سلسله مراتب در معماری موتورهای جستجوی با دامنه خاص بهبود قابل توجهی را در عملکرد، نسبت به معماری اصلی نشان می‌دهد. از آنجاییکه بسیاری از پرس و جوهای کاربران در یک دامنه خاص مشابه و یا تکراری هستند؛ در این معماری به منظور یادگیری از جستجوهای قبلی و استفاده از آنها در جستجوهای جدید از "استدلال بر پایه موارد" استفاده شده است. نتایج نشان می‌دهد که معماری پیشنهاد شده در جلوگیری از جستجوهای تکراری و ارائه نتیجه بهتر برای پرس و جوهای مشابه مؤثر می‌باشد.

کلمات کلیدی

موتورهای جستجو، توسعه پرس و جو، سلسله مراتب مفهومی، استدلال بر پایه موارد.

AKU Search Engine: An Integrated Architecture for Domain Specific Search Engines

A. Abdollahzadeh Barfourosh
Associate Professor

H. R. Motahari Nezhad
Graduate Student

Amirkabir University of Technology

Abstract

The amount of accessible information on Internet is increasing, rapidly, such that general-purpose search engines are unable to cover and index half of this information. A small fraction of this information represents a topic or domain. Domain specific search engines are suitable approach to reach a high precision and recall. However, these approaches suffer from querying mechanism.

In this paper, the current problems of search engines in querying mechanism is surveyed and an integrated architecture for domain specific search engines, called AK Usearch Engine, has been introduced, which its goal is to improve querying mechanism of users through automatic expanding of users' queries and learning from past searches using Case Based Reasoning (CBR). To expand

the users' query automatically and with related information, a new concept called "domain specific concept hierarchy" is introduced and an algorithm for learning this hierarchy from specific domains is designed, implemented and evaluated. The result of using this concept hierarchy in the proposed architecture shows a significant improvement in comparison to the original architecture. The implementation results show that this architecture is effective in preventing from repetitive searches and presenting results with better quality to the user.

Keywords

مقدمه

تعداد صفحات وب، تعداد سرویس دهنده‌های وب و تعداد حوزه‌های وب (مثلاً www.cisco.com) با سرعت بسیار زیادی در حال افزایش است. اکنون بیش از چندین میلیارد صفحه بر روی حدود ۹ میلیون پایگاه وب در شبکه گسترده جهانی وجود دارند [۱]. در حالیکه بزرگترین موتورهای جستجوی از لحاظ تعداد صفحات شاخص‌بندی شده (موتور جستجوی Google) کمتر از نیمی از صفحات آن را شاخص‌بندی می‌کند [۲و۳]. این موتورهای جستجو تلاشهای وسیعی برای به هنگام نگهداشتن خود انجام داده‌اند اما رشد وب فراتر از افزایش قدرت کاوشگرهای موتورهای جستجو بوده است [۴]. به دلیل رشد بسیار سریع و عدم تجانس اطلاعات قابل دسترس بر روی وب، یک موتور جستجو قادر به شاخص‌بندی تمامی اینترنت نیست و نتایج موتورهای جستجو ممکن است دارای میزان پوشش و دقت مناسبی نباشند. بعضی مشاهدات نشان می‌دهد که در یک موتور جستجوی همه منظوره که پرس و جوها در آنها می‌توانند در مورد هر موضوعی بیان شوند، بخش قابل توجهی از پرس و جوها در مورد تعداد محدودی از موضوعات مانند محصولات، سرگرمیها، حوادث و رخدادهای جاری و ... هستند [۵]. از طرف دیگر تعداد بسیار کمی از تمامی صفحات در اختیار در وب به یک موضوع خاص مربوط هستند. با این مشاهدات می‌توان گفت که یک موتور جستجوی با موضوع مشخص (و یا کاوشگر متمرکز [۶ و۷ و۸ و ۲۹و ۳۰و۳۱و ۳۲و ۳۳و ۳۴]) ابزار بهتری برای شاخص‌بندی و جستجوی وب است.

از نظر نحوه ارائه پرس و جوها، سه دسته عمده از موتورهای جستجو وجود دارند [۹]. اولین دسته بر پایه کلمات کلیدی عمل می‌کنند مانند اغلب موتورهای جستجوی فعلی (Altavista و MSN.Com) که نتایج را بر اساس تطابق دقیق کلمات کلیدی برمی‌گردانند. این موتورهای جستجو یک پرس و جوی متشکل از مجموعه‌ای از کلمات کلیدی را از کاربر دریافت می‌کنند و آن را بر روی یک و یا چند شاخص جستجو می‌کنند. این دستاورد ساده است اما این نوع از موتورهای جستجو نوعاً تعداد زیادی از اسناد را در پاسخ به یک پرس و جوی ساده برمی‌گردانند که اغلب کار یافتن اطلاعات مورد نیاز را برای کاربر مشکل می‌کنند [۹و۱۰].

تطابق دقیق کلمات کلیدی در موتورهای جستجو می تواند منجر به ایجاد مشکل شود. مخصوصا هنگامی که کلمات ورودی دارای بیش از یک معنی و کاربرد باشند. در اینصورت صفحاتی که یک موتور جستجو بر می گرداند ممکن است که در هر کدام کلمات کلیدی مورد نظر با هر کدام از معانی و یا کاربردهای متفاوت ظاهر شده باشند بدون اینکه به خواسته کاربر توجه کنند. این امر بدلیل خاصیت تطابق دقیق کلمات کلیدی و در نظر نگرفتن معنای کلمات است. در یک حوزه مشخص، اغلب کلمات دارای یک و یا تعداد معدودی معنی واضح و مشخص هستند که می‌توانند بوسیله موتورهای جستجوی با دامنه خاص (موضوع خاص) مدیریت شوند. بعنوان مثال کلمه 'Socks' نام قراردادی در شبکه‌های کامپیوتری و نیز نام یک نوع پوشاک (جوراب) در حوزه پوشاک است. یک موتور جستجوی همه منظوره صفحاتی را که برمی‌گرداند ممکن است حاوی این کلمه با هر کدام از این کاربردها باشند اما در موتورهای جستجوی با دامنه خاص مثلاً موتور جستجوی علوم کامپیوتر، موتور جستجو نتایجی را با یکی از این معانی برخواهد گرداند.

دومین دسته از موتورهای جستجو بر اساس پرس و جوهایی با استفاده از زبان طبیعی می‌باشند [۹]. این موتورها با کمک کاربر از یک پایگاه داده که حاوی پرس و جوهای بسیار پرسیده شده (FAQ) و جوابهای متناظر آنها است پرس و جوی کاربر را مشخص کرده و به آن پاسخ می‌گویند. کاربر پرس و جو را همانند اینکه از یک انسان سؤال می‌کند درون جعبه پرسش وارد

می‌کند و بر خلاف روش کلمات کلیدی نیازی به استفاده از ترکیبات منطقی (عبارات و عملگرهای منطقی) و یا ساختارهای پیچیده نیست. یک مثال از این دسته موتورهای جستجو Askjeeves.com می‌باشد. این موتور جستجو پرس و جوی کاربر را پالایش کرده و یک سری کلمات کلیدی خاص را انتخاب کرده و به موتورهای جستجوی مختلف می‌فرستد و مانند یک موتور جستجوی جانبی عمل می‌کند. این موتور جستجو فقط بر روی پرس و جوهای ساده زبان طبیعی کار می‌کند. یک فایده این دستاورد رسیدن به دقت بیشتر در مقابل موتورهای جستجوی معمولی است [۹] اما کار با ساختارهای پیچیده‌تر زبان و تطابق دو سند و یا پرس و جو و یک سند با استفاده از ساختارهای زبانهای طبیعی هنوز به عنوان زمینه‌های تحقیقاتی مطرح هستند.

کاوشگرهای متمرکز [۶و۸] (نوعی از موتورهای جستجوی با دامنه خاص) بعنوان سومین دسته، بجای اینکه پرس و جوی کاربر و یا موضوع کاوش را به فرم مجموعه ای از کلمات کلیدی در نظر بگیرند و به این منظور که بر محدودیت استراتژی تطابق دقیق کلمات کلیدی غلبه کنند، موضوع جستجو را به صورت مجموعه‌ای از اسناد در نظر می‌گیرند که باید توسط کاربر برای سیستم فراهم آید. این دستاورد تاثیر زیادی در بهبود دقت و کیفیت نتایج موتورهای جستجو می‌شود. کاوشگرهای متمرکز [۶و۸] پیشنهاد کردند که موضوع مورد تمرکز را با مجموعه‌ای از اسناد نشان دهند که به وسیله کاربر باید فراهم آید. این سیستم از مستندات استفاده می‌کند تا نسبت به میزان ربط یک سند وب به موضوع مورد تمرکز در خلال کاوش قضاوت کند. یک توسعه به این دستاورد که توسط Bharat و همکارانش [۱۱] انجام شده الگوریتمی را برای استخراج اسناد مربوط در فضای ابر پیوندی ارایه می‌دهد. این محققان معتقدند که موضوع پرس و جو از خود پرس و جو بسیار وسیع‌تر است و بنابراین آنها ۱۰۰۰ کلمه اول هر سند را در مجموعه آغازین (نمایش دهنده موضوع) را به عنوان موضوع انتخاب می‌کنند و از آن برای محاسبه میزان تشابه سند کاوش‌شده به موضوع مورد تمرکز استفاده می‌کنند. اما این دستاوردها یک بار اضافی بر کاربر تحمیل می‌کنند زیرا کاربر باید مستندات مرتبط به موضوع را برای سیستم فراهم آورد. علاوه بر آن برای کاربر کار مشکلی است که صفحاتی بامیزان ربط بالا به موضوع مورد نظر خود را بدست آورد. همچنین نمی‌توان تخمین مناسبی از میزان ربط مجموعه‌ای از کلمات انتخابی هر سند (هزار کلمه اول) به موضوع مورد تمرکز به دست آورد. به عنوان بخشی از راه حل پیشنهادی، این نیاز کاربر برای فراهم آوردن صفحات مرتبط حذف شده است در حالیکه پرس و جوی کاربر با افزودن مجموعه‌ای از عبارت‌های وابسته مفهومی به آن توسعه داده می‌شود. یک الگوریتم برای توسعه خودکار پرس و جو با استفاده از یک سلسله مراتب مفهومی با دامنه خاص برای موتورهای جستجوی با دامنه خاص پیشنهاد و نتایج پیاده‌سازی و ارزیابی ارائه شده‌اند.

بر خلاف دستاوردهای فوق، دستاورد ارائه شده دارای دو مزیت زیر می‌باشد:

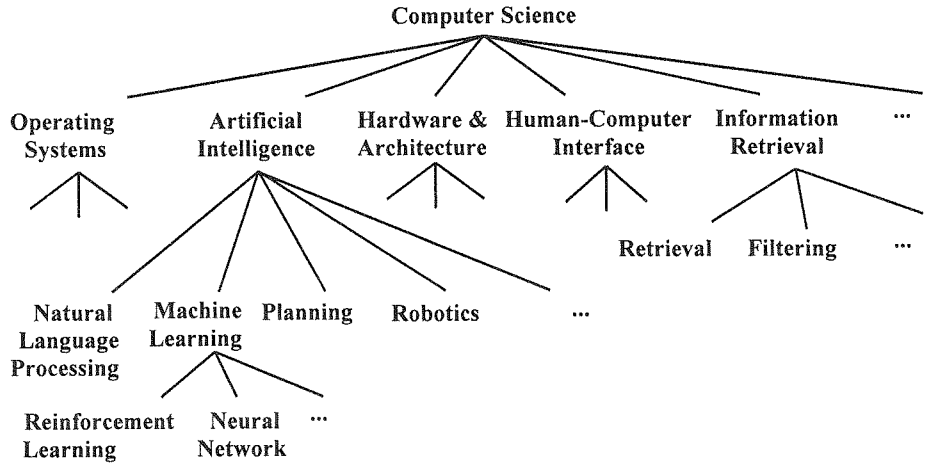
- اولاً، فرآیند توسعه پرس و جو یک فرآیند خودکار است بطوری که کاربر نیازی به فراهم آوردن اسناد مرتبط به موضوع به عنوان ورودی سیستم ندارد.
- ثانیاً، این دستاورد پرس و جو ها را با استفاده از مجموعه‌ای از لغات و عبارات مرتبط به موضوع مورد کاوش از لحاظ مفهومی توسعه می‌دهد و بنا براین دامنه معنایی و مفهومی پرس و جو را محدود می‌کند. این لغات بوسیله یک سلسله مراتب مفهومی با دامنه خاص فراهم می‌آیند و از طریق یک فرآیند آماری بر روی متن یاد گرفته می‌شوند.

در این پروژه از دستاورد فوق الذکر در یک موتور جستجو و در دو معماری متفاوت استفاده شده است. این سیستم یک موتور جستجوی مقالات علوم کامپیوتر می‌باشد که از یک سلسله مراتب مفهومی با دامنه خاص به همراه یک کاوشگر متمرکز استفاده می‌کند. در معماری اول (AKU-CS) به آزمون تاثیر توسعه خودکار پرس و جوی کاربر با استفاده از سلسله مراتب مفهومی دامنه خاص پرداخته‌ایم. معماری دوم (AKUSearch)، یک معماری پیشنهادی یکپارچه برای موتورهای جستجوی با دامنه خاص می‌باشد که علاوه بر سلسله مراتب مفهومی با دامنه خاص از "استدلال بر پایه موارد" برای یادگیری از نتایج پرس و جوهای گذشته برای بهبود کارآیی استفاده می‌کند.

بخش ۲ مقاله، سلسله مراتب مفهومی با دامنه خاص را معرفی کرده و یک الگوریتم جدید برای ساختن خودکار آن ارائه می‌کند. این بخش در ادامه، بستر آزمون و نتایج پیاده‌سازی الگوریتم و استفاده از این سلسله مراتب مفهومی را در یک معماری پیشنهادی (موتور جستجوی جانبی AKU-CS) [۱۲و۱۳] نشان می‌دهد. سپس در بخش ۳ یک معماری یکپارچه برای موتورهای جستجوی با دامنه خاص پیشنهاد شده است [۱۴] که از سلسله مراتب مفهومی با دامنه خاص و استدلال بر پایه موارد برای بهبود کیفیت نتایج موتورهای جستجو استفاده می‌کند.

۱ـ سلسله مراتب مفهومی با دامنه خاص

در ایـن بخـش سلسله مراتب مفهومی با دامنه خاص (Domain Specific Concept Hierarchy-DSCH) را بعنوان یک نوع خـاص از سلسله مراتب مفهومی معرفی می‌کنیم. DSCH یک سلسله مراتب است که شامل لغات فنی یک دامنه مشخص است. یـک سلسـله مراتب مفهومی شامل مجموعه‌ای از گره‌هاست که در یک ترتیب جزیی سازماندهی شده‌اند [۱۵]. یک گره، نهایی است اگر هیچ گره فرزندی نداشته باشد و یا در غیر اینصورت میانی است. هر گره از این سلسله مراتب دارای سبدی از مجموعه لغـات و عبارات است. این لغات، "لغات بسیار مرتبط مفهومی" به موضوع آن گره هستند. این لغات از فرهنگ داده‌های آن دامنه خاص قابل استخراج هستند. بعنوان مثالی از سلسله مراتب مفهومی با دامنه خاص، بخشی از سلسله مراتب موضوعی موتورهای جستجوی Cora [۱۶] در شکل ۱ نشان داده شده است.



شکل (۱) بخشی از سلسله مراتب موضوعی موتورهای جستجوی Cora.

۱ـ۱ـ AC-DSCH : الگوریتم پیشنهادی برای ساختن خودکار DSCH

مهندسـان دانش، خبرگان و دانشمندان یک زمینه خاص می‌توانند سلسله مراتب مفهومی مرتبط به آن موضوع را بسازنـد [۱۵]. تمرکـز اصـلی سـاختن DSCH، بدست آوردن مجموعه‌ای از لغات مرتبط به هر گره است. چندین سلسله مراتب عمومی مانـند Yahoo! و The Open Project Directory و Mining Co. وجـود دارنـد. در هـر گـره میانـی از ایـن سلسـله مراتـب‌هـا پیوندهایـی به گره های سطح بعدی وجود دارد. در گره‌های نهایی، تعدادی ابرپیوند وجود دارد که منجر به اسنادی می‌شوند که بـه موضوع مرتبط هستند. این سلسله مراتب‌ها (دروازه های وب) بوسیله عوامل انسانی نگهداری می‌شوند. تحقیقات بسیاری در سالهای اخیر صورت گرفته است تا ساختن و نگهداری دروازه های وب را خودکار سازد [۱۷و ۱۸]. هدف سلسله مراتب مفهومی و الگوریتم ارائه شده، بدست آوردن لغات و عبارات بطور مفهومی وابسته به موضوع هر گره است. شبه کد هر مرحله از الگوریتم در جعبه‌های بعد از هر مرحله ارائه شده‌اند. الگوریتم پیشنهای به صورت زیر می‌باشد:

الگوریتم AC-DSCH

ساختن خودکار یک سلسسله مراتب مفهومی با دامنه خاص با استفاده از اسناد درون گره‌های یک سلسله مراتب موضوعی

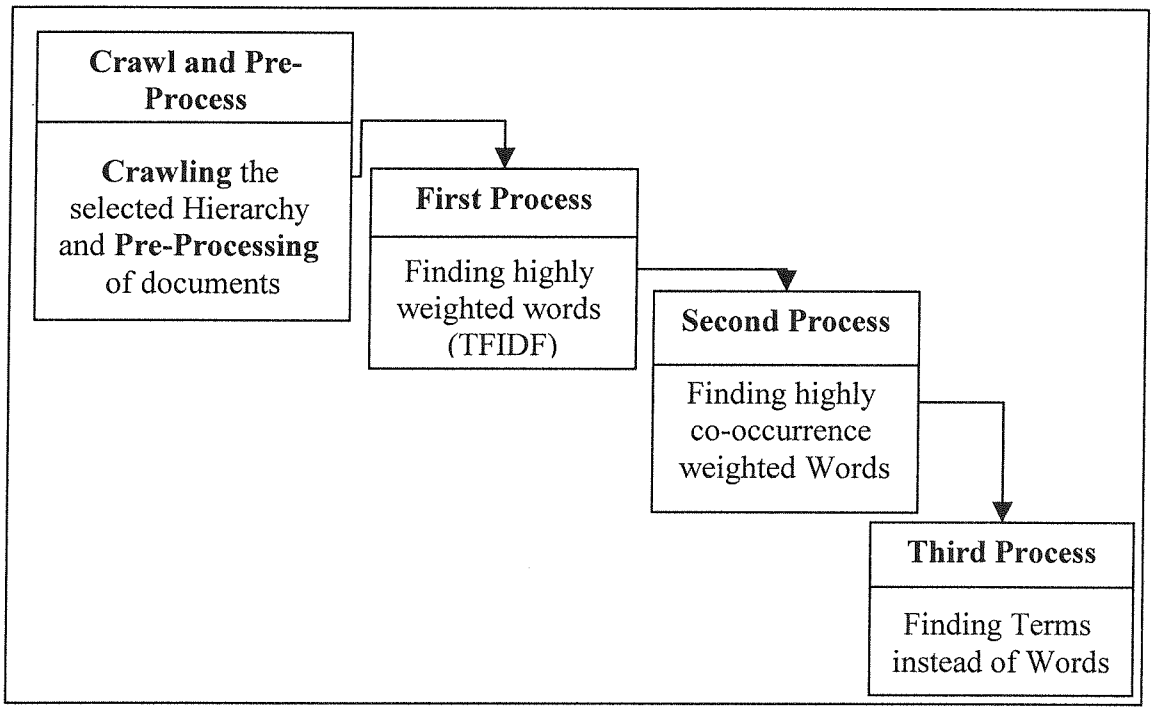
ورودیـ (۱) یـک سلسـله مراتـب از پـیش تعـریف شده H با سبد (خالی) لغات و عبارات مرتبط (۲) تعداد کلمات انتخاب شـونده از بازیابـی اسـناد مرتبط با استفاده از N, TFIDF (۳) تعداد کلمات زوجهای کلمات با رخداد همزمان در اسناد مرتبط به موضوع ، M (۴) تعداد عبارات (بیش از یک کلمه) با احتمال رخداد بالا، P .

خروجـی ـ سلسـله مراتـب بـا دامـنه خـاص 'H، در حالی که سبد لغات و عبارات مرتبط مفهومی حاوی کلمات و لغات و عبارات مرتبط می‌باشد.

AC- DSCH Algorithm (H, N, M, P) : H'

AC-DSCH: *Automatic Construction of Domain Specific Concept Hierarchy*
H: *A predefined concept hierarchy in which the baskets of nodes are empty*
N: *Number of words to be selected from TFIDF process*
M: *Number of word pairs to be selected from pairs with highest co-occurrence weight*
P: *Number of phrases to be selected with highest occurrence weight (probability)*

الگوریتم دارای چهار مرحله پردازش می‌شود که در شکل ۲ نشان داده شده و در ادامه تشریح شده‌اند.



شکل (۲) گردش کاری الگوریتم ساختن خودکار DSCH.

مرحله ۱ـ کاوش مستندات و پیش پردازش

در این مرحله، مستندات درون هر گره از سلسله مراتب موضوعی که بعنوان بستر یادگیری انتخاب شده است کاوش شده و یک پیش پردازش اولیه بر روی آنها صورت می‌گیرد. در هرگره میانی عنوان و متن ابر پیوندهای سطح بعدی به سبد آن گره افزوده می‌شوند. این لغات نواحی و موضوعات مطرح هر گره را نشان می‌دهند و دامنه مفهومی موضوع گره را می‌پوشانند. برای گره‌های نهایی، تمامی مستندات درون آنها کاوش و کپی می‌شوند. به صورت معمول، این اسناد در قالب فایلهای HTML هستند. در مرحله پیش پردازش، برچسب‌های HTML هر صفحه و نیز کلمات ثابت درون همه صفحات را که مرتبط به قالب صفحه و نه به موضوع آن هستند، حذف می‌شوند. سپس ۳ مرحله پردازش بر روی متن استخراج شده از این صفحات انجام می‌شود تا لغات و عبارات مورد نظر، استخراج شوند. این مراحل در ادامه تشریح شده‌اند.

Crawling and Pre-Processing:
Crawl the selected hierarchy
- If current node is a non-leaf node then
  - Add the name of next level nodes to the basket of the node
- If it is a leaf node, crawl all of the documents in it (.HTML) and store them

Pre-process the documents
- Remove HTML Tags
- Remove constant texts of all the pages such as fixed header, footer,….

مرحله ۲ـ اولین پردازش (یافتن کلمات با بیشترین وزن (TFIDF)

در این مرحله هدف یافتن کلماتی است که با احتمال بالایی به موضوع هرگره مربوط می‌باشند. برای یافتن وزنهای با احتمال رخداد کلمات در اسناد درون هر گره از روش وزن دهیTFIDF [۱۹] (تعداد تکرار کلمه در معکوس تعداد تکرار اسناد) استفاده می‌شود. با محاسبه این وزن برای کلمات درون اسناد هر گره، N کلمه با بالاترین وزن را بعنوان کلمات مرتبط مفهومی به موضوع گره انتخاب کرده و در سبد هر گره قرار می‌گیرد. ساختمان داده‌های سبد هر گره دارای هر کلمه و وزن TFIDF متناظر آن کلمه می‌باشد.

First Process (Words with highest TFIDF words)
Calculate TFIDF weigh of each of the words in documents of each node

- In a document d, the frequency of each word stem s is f_s, the number of documents having stem s is n_s, and the highest term frequency is called f_{dmax}. In one such TFIDF scheme [35] a word weight w_{ds} is calculated as:

$$w_{ds} = \frac{(0.5 + 0.5\frac{f_{ds}}{f_{d\max}})(\log\frac{N_D}{n_s})}{\sqrt{\sum_{j\in d}((0.5 + 0.5\frac{\int dj}{f_{d\max}})^2(\log\frac{N_D}{n_s})^2)}}$$

Where N_D is the total number of documents.

- Select N words with highest TFIDF weights for each node and add them to the basket of the node

مرحله ۳ـ دومین پردازش (یافتن کلمات با بالاترین وزن رخداد همزمان)

وزن رخداد همزمان، احتمال رخداد همزمان زوجهای کلمات را در اسناد محاسبه می‌کند. کلمات با احتمال رخداد بالا با کلمات کلیدی موضوع گره، نشان دهنده ارتباط مفهومی این کلمات با همدیگر و با موضوع هرگره هستند [۸]. به عنوان مثال، وزن رخداد همزمان کلمات "automobile" و "car" بر احتمال رخداد همزمان دو کلمه در هر سندی در مجموعه اسناد دلالت دارد، اما این کلمات ضرورتاً یک عبارت را تشکیل نمی‌دهند و آنها می‌توانند هر جایی در متن ظاهر شوند. M زوج کلمه با وزنهای رخداد همزمان بالا با کلمات موضوع گره به سبد هر گره افزوده می‌شود.

مرحله ٤ـ سومین پردازش (عبارت بجای کلمات منفرد)

در این بخش منظور از "عبارت"، مجموعه متصل به بیش از یک کلمه است. به عنوان مثال "data mining" بجای کلمات منفرد "data" و "mining" به عنوان عبارت در نظر گرفته می‌شود. در این مرحله، احتمال رخداد عبارات در اسناد محاسبه شده و هدف، یافتن عبارات بسیار مرتبط مفهومی به موضوع هر گره در سلسله مراتب مفهومی است. در اغلب موارد، نیاز به در نظر گرفتن عبارات دو کلمه ای و سه کلمه ای است و این تعداد کلمه معمولاً اغلب عبارات مد نظر را می پوشاند؛ اگر چه بعضی از عبارات با درجه بالاتر (چهار کلمه‌ای و یا بیشتر) ممکن است در بعضی از موضوعات رخ دهند ولی مشاهدات تجربی این پروژه نشان داده است که تعداد این عبارات بسیار کم است و در این پروژه فقط عبارات تا سه کلمه‌ای در نظر گرفته شده است. بر این اساس تعدادP عبارت دو کلمه‌ای و عبارت سه کلمه‌ای که با احتمال بسیار بالا در هر اسناد هر گره رخ می‌دهند با وزن متناظرشان به سبد هر گره افزوده می‌شوند.

۱ـ۲ـ پیاده سازی و ارزیابی نتایج الگوریتم DSCH

به منظور آزمون و ارزیابی الگوریتم ارائه شده، سلسله مراتب موضوعی موتور جستجوی Cora [۷و ۱۶و ۲۰] که یک موتور

جستجوی مقالات علوم کامپیوتر می‌باشد، به عنوان بستر تست در این آزمایشات انتخاب شده است. هر سند در هر گره این سلسله مراتب، شامل اطلاعاتی راجع به یک مقاله مرتبط به موضوع گره می‌باشد. الگوریتم AC-DSCH (ساختن خودکار سلسله مراتب مفهومی با دامنه خاص) بر روی این بستر تست اجرا شده است. تشریح فرآیند اجرا و تحلیل نتایج حاصله در زیربخشهای بعدی ارائه شده‌اند.

Second Process (Word Pairs with highest Co-occurrence Weight)

- For each node, find co-occurrence weight of word pairs (It is available as an option in Rainbow from Bow package) for documents in that node
- Select M word pairs with highest co-occurrence weight and add them to the basket of the node

Third Process (Finding phrases instead of single words)

- For each node, find all of (2-word and 3-word) phrases and calculate their probability of occurrence in documents of that node where
  P = probability of, p_i= phrase i, d_j = j^{th} document in the node,

$$P(p_i) = \frac{\sum_{j=1}^{n(d)} \frac{f(p_i / d_j)}{f(d_j)}}{n(d)}$$

  $f(p_i/d_j)$ is frequency of occurrence of p_i in d_j, f(dj) is frequency of all phrases in d_j, and n(d) number of all documents
- Select P (2-word and 3-word) phrases and add them to the basket of the node

۱ـ۲ـ۱ـ کاوش و پیش پردازش اسناد

در پیاده‌سازی الگوریتم، ۱۰۰ سند از هر گره موتور جستجوی Cora کاوش شده است. هر سند شامل عنوان، خلاصه و مراجع یک مقاله و نیز مشخصات مقالاتی است که به این مقاله به عنوان مرجع اشاره کرده‌اند. در هر سند بخش‌ها و کلمات ثابتی مانند برچسب‌های HTML و سرآیند و پاورقی و جمله‌های توصیفی ثابتی وجود دارند. در فاز پیش ـ پردازش، تمامی این بخش‌ها و جمله‌های ثابت و نام مولفان مقالات را حذف می‌کنیم. پس از این پردازش، هر گره دارای مجموعه‌ای از اسناد متنی می‌باشد که هر سند فقط دارای عنوان، چکیده، اطلاعات مقالات مراجع و مقالاتی (به جز نام مولفان) است که به این مقاله اشاره کرده‌اند.

۱ـ۲ـ۲ـ اولین پردازش (یافتن کلمات با بالاترین وزن TFIDF)

برای محاسبه وزن کلمات هر سند با استفاده از TFIDF از برنامه "Arrow" استفاده می‌شود. "Arrow" برنامه‌ای از بسته نرم‌افزاری پردازش آماری متن "Bow" است که بوسیله Andrew McCallum و همکارانش در دانشگاه CMU توسعه یافته است. این برنامه همچنین میزان تشابه هر سند جدید را با سندهای قبلی درون پایگاه داده خود با استفاده از TFIDF محاسبه می‌کند. جدول ۱ تعداد ۵۰ کلمه با بالاترین وزن را برای موضوع "data mining" نشان می‌دهد. این کلمات به میزان بسیار زیادی به موضوع مورد بحث ربط دارند. بعضی از کلماتی نیز که جنبه علمی دامنه مورد بحث را نشان می‌دهند مانند "conference" و "proceeding" در بین این کلمات دیده می‌شوند. همچنین نام بعضی از کنفرانسهای مشهور در موضوع مورد تمرکز مانند "KDD" و "VLDB" در بین ۱۰۰ کلمه با بالاترین وزن قرار دارند. اما بعضی از این کلمات مانند "algorithm" و "machine" و "learning"به تنهایی کلماتی عمومی هستند و در مقالات مربوط به موضوع‌های دیگر هم می‌توانند رخ دهند. این مشاهده ما را به این نتیجه رهنمایی می‌کند زوج کلماتی را بیابیم که در اسناد مربوط به موضوع (در هر گره) دارای وزن رخداد همزمان بالایی باشند و نیز عباراتی را که موضوع را توصیف می کنند؛ علاوه بر کلمات در نظر بگیریم.

۱ـ۲ـ۳ـ سومین پردازش (یافتن کلمات با بالاترین وزن رخداد همزمان)

جدول ۲ تعداد ۲۵ زوج کلمه با بالاترین وزن رخداد همزمان را برای موضوع "data mining" نشان می‌دهد. اغلب کلمات

درون جدول با کلمات "data" و "mining" دارای رخداد بالایی هستند. این کلمات، کلمات موضوع گره هستند. علاوه بر آن بعضی از این کلمات مانند "aggregation" و "association" به طور منحصر بفرد به موضوع گره مرتبط هستند. برای اینکه میزان اطلاعاتی که کلمات با وزن رخداد همزمان بالا با کلمات نمایش دهنده موضوع گره فراهم می آورند بهتر نشان داده شود، جدول ۳ تعداد ۲۰ زوج کلمه را که دارای کمترین احتمال رخداد همزمان در اسناد مرتبط به موضوع کاوش هستند را نشان می‌دهد. رخداد همزمان این زوج کلمات در اسناد مرتبط به موضوع "data mining" انتظار نمی رود و اطلاعات زیادی راجع به موضوع را نمایش نمی‌دهند. بنابراین مشاهدات می توان گفت که اگر برای محاسبه میزان تشابه دو سند و یا یک پرس و جوی توسعه یافته کاربر و یک سند، رخداد همزمان کلمات در نظر گرفته شود، می‌توان بر اساس میزان تشابه در لیست زوج کلمات با احتمال رخداد بالا برآوردی از میزان تشابه دو متن مورد نظر بدست آورد. اما فرآیند بدست آوردن کلمات با احتمال رخداد همزمان بالا برای تعداد زیادی سند در یک گره هم از لحاظ پیچیدگی فضایی و هم از لحاظ پیچیدگی زمانی پر هزینه می‌باشد.

در جدول ۲ مشاهده می‌شود بعضی از کلماتی که دارای احتمال رخداد بالایی با کلمات اصلی موضوع هر گره هستند به تنهایی کلمه‌ای هستند که ممکن است در چندین موضوع به تنهایی دیده شوند مانند کلمه "network" که هم در مبحث "computer networks" و هم در "neural network" دارای احتمال رخداد همزمان بالایی خواهد بود و کاربرد اصلی این کلمه را ترکیب آن با کلمات دیگر روشن می‌سازد. این مشاهده ما را به سمت در نظر گرفتن عبارات ـ بیش از یک ـ به جای کلمات هدایت می‌کند. زیر بخش بعدی آزمایشات انجام شده در این رابطه را تشریح می‌کند.

جدول (۱) تعداد N= ۲۵ کلمه با بالاترین وزن TFIDF برای دسته "data Mining".

Word	Weight	Word	Weight	Word	Weight
Data	0.0483205	proceedings	0.0108792	Classification	0.0062527
mining	0.0460734	algorithm	0.0102841	Trees	0.0062364
Rules	0.0335740	conference	0.0077680	Sigmod	0.0062189
databases	0.0300237	machine	0.0076521	Acm	0.0060270
knowledge	0.0290816	research	0.0075528	Relational	0.0053060
association	0.0260618	ieee	0.0074293	Intelligence	0.0049911
discovery	0.0247404	decision	0.0074105	Induction	0.0047303
Large	0.0184948	information	0.0072554		
learning	0.0141196	spatial	0.0066830		

جدول (۲) تعداد N=۲۵ زوج کلمه با بالاترین وزن رخداد همزمان (Co_o) برای دسته "data mining"
(لیست بر اساس حروف الفبای انگلیسی و نه بر اساس وزن مرتب شده‌اند)

Word1	Word2	Co_o Weight	Word1	Word2	Co_o Weight
Aggregation	Data	0.0629474	geographic	Data	0.0693333
Aggregation	mining	0.0589019	geographic	mining	0.0609187
Association	mining	0.0571912	integration	Data	0.0642788
Baskets	mining	0.0591086	query	Data	0.0587437
Custering	Data	0.0590085	relational	Data	0.0589771
Clstering	mining	0.0586942	spatial	Data	0.0683995
Correlations	mining	0.0609724	spatial	mining	0.0621446
Cubes	Data	0.0736354	technology	Data	0.0620216
Cubes	mining	0.0668675	Tools	Data	0.0602633
Dbminer	Data	0.0901771	Users	mining	0.0583686
Dbminer	mining	0.0783960	warehouse	Data	0.0950729
Dimensional	Data	0.0743458	warehouses	mining	0.0794421
Dimensional	mining	0.0716238			

۱ـ۲ـ٤ ـ سومین پردازش (عبارات به جای کلمات)

جدول ۴ تعداد ۲۵ عبارت دو کلمه ای را که با بالاترین احتمال در اسناد مربوط به موضوع "data mining" رخ داده‌اند را نشان می‌دهد. اغلب این عبارات در دامنه موضوعی "data mining" شناخته شده هستند. از این عبارات می‌توان از

"association rule" و "data mining" و "knowledge discovery" نام برد که از عبارات اصلی و تخصصی این موضوع هستند. همچنین قابل مشاهده است که چند عبارت مانند "base mining" دارای معانی کاملی نیستند و به نظر می‌رسد که بخشی از یک عبارت بزرگتر باشند. جدول۵، تعداد ۱۰ عبارت ۳ کلمه‌ای را که دارای بالاترین احتمال رخداد در اسناد گره با موضوع "data mining" را نشان می‌دهد. عبارات این جدول نیز مانند جدول ۴ به خوبی موضوع گره را توصیف می‌کنند. بر طبق جداول تعداد معدودی از عبارات مانند "level association mining" بخشی از یک عبارت بزرگتر هستند. به دلیل اینکه تعداد کم این عبارات و نیز به دلایل محاسباتی در این پیاده‌سازی به عبارات ۳ کلمه‌ای بسنده شده است.

جدول (۳) تعداد N=۲۵ زوج کلمه با کمترین وزن رخداد همزمان (Co_o) برای دسته "data mining"

Word 1	Word 2	Co_o Weight	Word 1	Word 2	Co_o Weight
classification	controlling	0.0000210	induction	generalizations	0.0000254
classification	Syntactic	0.0000210	trees	controlling	0.0000261
induction	controlling	0.0000224	trees	syntactic	0.0000261
induction	Syntactic	0.0000224	induction	occurrences	0.0000265
induction	Levelwise	0.0000242	decision	levelwise	0.0000266
decision	controlling	0.0000247	method	levelwise	0.0000266
decision	Syntactic	0.0000247	attribute	controlling	0.0000268
Method	controlling	0.0000247	attribute	syntactic	0.0000268
Method	Syntactic	0.0000247	clustering	controlling	0.0000268
Acm	Inconclusive	0.0000249	clustering	syntactic	0.0000268

جدول (۴) N=۲۵ عبارت (دو کلمه‌ای) با بالاترین وزن (احتمال) رخداد در موضوع "data mining"

Term	Weight	Term	Weight	Term	Weight
association rule	0.0023453062	spatial data	0.0003791043	mining application	0.0002056159
large database	0.0018312665	Interesting rule	0.0003469768	Level association	0.0002056159
data mining	0.0014907152	database system	0.0003469768	attribute oriented	0.0002056159
Relational database	0.0009895264	machine learning	0.0003212748	base mining	0.0001991904
decision tree	0.0006618261	multiple level	0.0002827218	learning algorithm	0.0001927649
knowledge discovery	0.0006232731	large set	0.0002698708	inductive learn	0.0001927649
relational data	0.0005461672	spatial database	0.0002570199	discovered association	0.0001927649
Mining association	0.0005140397	data cube	0.0002248924		
base system	0.0004369338	data set	0.0002120414		

جدول (۵) تعداد N= ۲۵ عبارت (سه کلمه‌ای) با بالاترین وزن (احتمال) رخداد در موضوع "data mining".

Term	Weight	Term	Weight
mining association rule	0.0000012981	attribute oriented induction	0.0000003407
Multiple level association	0.0000005192	data mining system	0.0000003083
level association rule	0.0000005192	generalized association rule	0.0000002758
discovered association rule	0.0000004868	object oriented database	0.0000002434
spatial data mining	0.0000003732	data mining techniques	0.0000001947

۱ـ۳ـ بکارگیری DSCH در موتورهای جستجوی با دامنه خاص

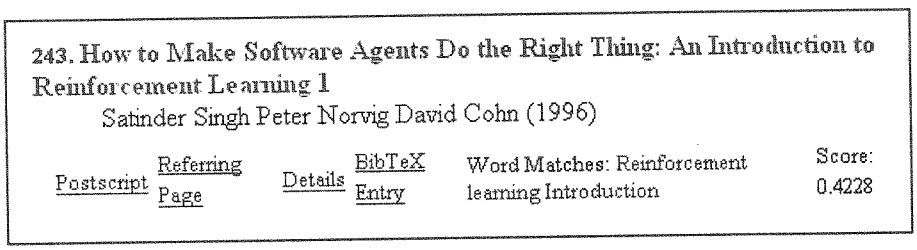
تاثیر فرستادن یک پرس و جوی غیر توسعه یافته به صورت مجموعه‌ای از کلمات کلیدی و یک پرس و جوی توسعه یافته بوسیله DSCH در معماری موتورهای جستجوی با دامنه خاص که AKU-CS [۱۲و ۱۳] نامیده شده، مطالعه و با هم مقایسه گردیده‌اند. نتایج برگردانده شده از موتور جستجو در حالت استفاده از پرس و جوی توسعه یافته تفاوت قابل توجهی را در

کیفیت نتایج ارائه شده با رتبه‌های بالا نشان می‌دهد. برای انجام آزمایشات، از کاوشگر متمرکز Cora [۷و ۱۶و ۲۰] بعنوان کاوشگر سیستم استفاده شده است. از اهداف این معماری ارائه راه حلی برای بهبود نتایج برگردانده شده توسط موتور جستجوی (مقالات علوم کامپیوتر) Cora به کاربران می‌باشد. روش کاوش این موتور جستجو سه مرتبه نسبت به روش کاوش "اول ـسطح" که توسط بسیاری از موتورهای جستجوی فعلی استفاده می‌شوند، کارآتر می‌باشد [۷]. اما علیرغم این حقیقت، در حالیکه مستندات مربوط به پرس و جوی کاربر در پایگاه داده موتور جستجو وجود دارند، در پاسخ برگردانده شده به کاربر به طور مناسب رتبه‌بندی نمی‌شوند. در زیر بخش‌های بعدی به بیان مشکل این موتور جستجو و نحوه بهبود آن توسط دستاورد توسعه پرس و جوی کاربر پرداخته شده است.

۱ـ۳ـ۱ـ پرس و جوی غیرتوسعه یافته

در موتور جستجوی Cora، تمامی مقالات از طریق یک واسط جستجوی استاندارد (جعبه متن برای دریافت پرس و جو) در دسترس هستند. این موتور جستجو از روشهای مختلف جستجو و عملگرهای معمول مانند + و ـ و جستجوی عبارت با "" پشتیبانی می‌کند. بعلاوه می‌توان جستجو را محدود به اطلاعات خاص استخراج شده از مقالات مانند مولفان مقاله و یا عنوان آنها کرد. همچنین مجموعه انتخابی از مقالات از طریق یک سلسله مراتب مانند !Yahoo برای موضوعات متفاوت علوم کامپیوتر در اختیار هستند. معیار سنجش میزان تطابق پرس و جوی کاربر با مقالات، مجموع وزنی لگاریتم تعداد تکرار لغات پرس و جو در مستندات می‌باشد. این وزن، معکوس تعداد تکرار لغت در تمامی دامنه لغات شناخته شده می‌باشد. هنگامی که یک عبارت درون پرس و جو وجود دارد، با این عبارت مانند یک لغت برخورد می‌شود ولی توسعه پرس و جو انجام نمی‌گیرد.

آزمایشات این پروژه مشکل رتبه‌بندی نتایج را در این موتور جستجو نشان می‌دهد. در یکی از این آزمایشات کلمات کلیدی "Reinforcement Learning Introduction" به این موتور جستجو ارسال شد. در لیست صفحه اول نتایج برگردانده شده، سه پاسخ اول به یک کاربرد خاص در حوزه پرس و جو مربوط بودند. بقیه نتایج صفحه اول (از ۴ تا ۱۰) به مسائل عمومی مرتبط به موضوع و نه به طور خاص به خود آن ربط دارند. اما این معنی عدم وجود سند بهتر مرتبط به موضوع نیست. عنوان مقاله ای که در رتبه ۲۴۳ ظاهر شده است در شکل ۳ دیده می‌شود. تمامی کلمات پرس و جوی کاربر در یک ترتیب معنی دار و مرتبط در عنوان مقاله وجود دارد و به نظر می‌رسد که کاربر به این مقاله بیش از مقالات گزارش شده در صفحات اول علاقمند باشد ولی این مقاله در صفحه ۲۵ نتایج لیست شده است. اگر پرس و جوی ارسال شده را به ""Rinforcement Leanrning" Introduction" تغییر دهیم رتبه مقاله فوق به ۱۴۲ کاهش می‌یابد. مشاهدات مشابهی در دیگر پرس و جوهایی که به این موتور جستجو ارائه گردید حاصل شد؛ به این معنی که مقاله بسیار مرتبط به جستجوی کاربر در پایگاه داده‌های مقالات وجود دارند ولی معمولا در چندین صفحه دورتر نسبت به صفحه اول گزارش می‌شوند.


243. How to Make Software Agents Do the Right Thing: An Introduction to Reinforcement Learning 1
Satinder Singh Peter Norvig David Cohn (1996)
Postscript Referring Page Details BibTeX Entry Word Matches: Reinforcement learning Introduction Score: 0.4228

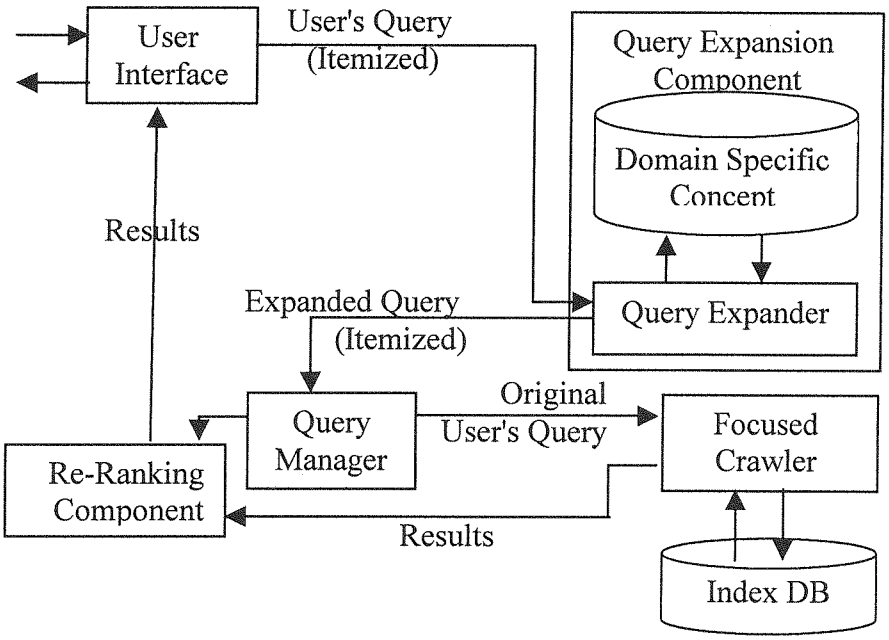
شکل (۳) نتایج رتبه ۲۴۳ برای پرس و جوی "Reinforcement Learning Introduction" در موتور جستجوی Cora.

۱ـ۳ـ۲ـ معماری پیشنهادی برای AKU-CS

معماری پیشنهادی برای سیستم جستجوگر AKU-CS [۱۲و ۱۳] در شکل ۴ نشان داده شده است. در این سیستم پرس و جوی کاربر به همان روش موتورهای جستجوی دیگر (مجموعه‌ای از کلمات کلیدی) و انتخاب یک دسته از سلسله مراتب مفهومی که کاربر آن را بسیار مرتبط به موضوع پرس و جو تشخیص می‌دهد، دریافت می‌شود. پس از این پرس و جو توسط مولفه "توسعه پرس و جو" با استفاده از مجموعه لغات و عباراتی که به صورت مفهومی مرتبط با جستجوی کاربر هستند، توسعه

داده می‌شود. این مجموعه لغات و عبارات با استفاده از سلسله مراتب مفهومی DSCH و از سبد گره متناظر با دسته انتخاب شده توسط کاربر بدست می‌آیند. سپس پرس و جوی توسعه یافته به میان افزار این موتور جستوی جانبی فرستاده می‌شود. در این میان افزار "مدیر پرس و جو" پرس و جوی اصلی کاربر را به موتور جستجوی Cora می‌فرستد. نتایج برگردانده شده به این پرس و جو به مولفه "رتبه‌بندی دوباره" فرستاده شده و در این مولفه بر اساس میزان شباهت پرس و جوی کاربر با پرس و جوی توسعه یافته نتایج دوباره رتبه‌بندی شده و به کاربر ارسال می‌شود. برای هر مقاله یک امتیاز جدید با استفاده از امتیازی که از موتور جستجوی Cora برگردانده شده و میزان تشابه TFIDF چکیده مقالات و پرس و جوی توسعه یافته و نیز وجود لغات و عبارات پرس و جوی کاربر در عنوان مقاله (با وزن بالا) محاسبه می‌شود.

در آزمایشات این پروژه، ۳۰۰ نتیجه اول موتور جستجوی Cora انتخاب و عنوان، چکیده و (در صورت وجود کلمات کلیدی) در فرآیند رتبه‌بندی دوباره بکار رفته‌اند. پس از مرتب سازی دوباره، ده نتیجه اول (با امتیاز بالا) انتخاب شده و به کاربر گزارش می‌شود. این انتخاب به این دلیل صورت گرفته است که اکثر کاربران دنبال نتیجه مطلوب خود در صفحه اول نتایج هستند [۱۳]. در مورد پرس و جوی "Reinforcement Learning Introduction" که قبلاً اشاره شد که بدون توسعه پرس و جو دارای رتبه ۲۴۳ می‌باشد، با استفاده از پرس و جوی توسعه یافته این نتیجه در بین ۵ نتیجه اول گزارش شده برای پرس و جوهای متفاوت مربوط به مفاهیم یادگیری تقویتی قرار می‌گیرد. همچنین نتایج یک پرس و جو در مورد "Association rules" در دسته "data mining" به این موتور جستجو، ده نتیجه اول گزارش شده بسیار مرتبط‌تر به موضوع در مقایسه با نتایج برگردانده شده از Cora برای همین پرس و جو می‌باشد.

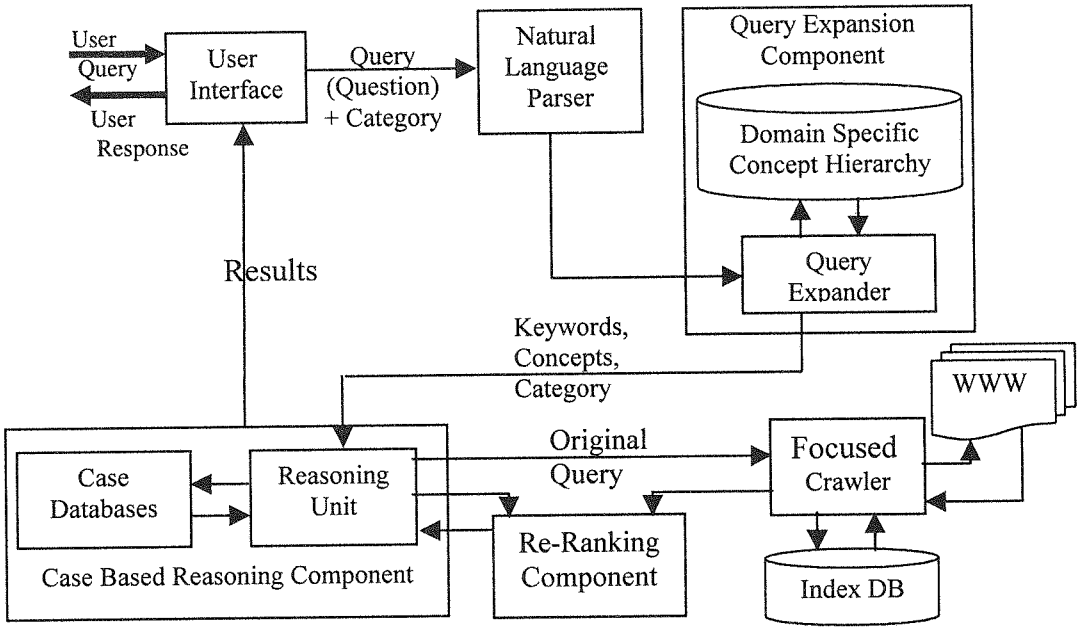


شکل (٤) معماری بکار گرفته شده برای ارزیابی کارآیی سلسله مراتب مفهومی با دامنه خاص در توسعه پرس و جو.

۲ـ یک معماری یکپارچه برای موتورهای جستجوی با دامنه خاص

در معماری پیشنهادی علاوه بر دریافت پرس و جوی کاربر به صورت مجموعه‌ای از کلمات کلیدی، اطلاعات اضافی دیگری از کاربر درخواست می‌گردد. این اطلاعات بصورت انتخاب از بین گزینه‌های پیشنهادی موجود بوده و بار اضافی بر دوش کاربر ندارد تا اطلاعاتی راجع به موضوع مدنظر فراهم آورد. همچنین این موتور قابلیت پردازش جملات بسیار ساده زبان طبیعی را فراهم می‌آورد. بسیاری از تحقیقات نشان داده است که زبان طبیعی برای بیان هدف و مقصود کاربر قوی‌تر است [۱۰] زیرا اصولاً کاربران در فرموله کردن نیازهایشان بصورت عبارات منطقی مشکل دارند حتی اگر ذهنیت روشنی از اطلاعات مورد نیاز خود داشته باشند [۲۱]. این معماری از استدلال بر پایه موارد استفاده کرده تا از تجربیات پرس و جوهای کاربران برای پاسخ ـ

گویی به پرس و جوهای مشابه آینده بهره گیرد. تحقیقات نشان داده است که استفاده از این دستاورد منجر به بهبود کارآیی در موتورهای جستجوی با دامنه خاص می‌شود [۲۲]. شکل ۵ معماری پیشنهادی یکپارچه برای موتورهای جستجوی با دامنه خاص را نشان می‌دهد [۱۴].



شکل (۵) معماری یکپارچه پیشنهادی برای موتورهای جستجوی با دامنه خاص.

این معماری که AKUSearch نام گرفته است از یک کاوشگر متمرکز و دستاورد استدلال بر پایه موارد و مولفه توسعه پرس و جوی کاربر (DSCH) استفاده می‌کند. مولفه توسعه پرس و جوی کاربر با استفاده از سلسله مراتب مفهومی با دامنه خاص در بخش قبلی به صورت مشروح بررسی گردید. در ادامه تاکید بر روی روش پیشنهادی برای بکارگیری استدلال بر پایه موارد به همراه سلسله مراتب مفهومی با دامنه خاص در جهت یادگیری از پرس و جوهای کاربران می‌باشد.

پرس و جوی کاربر از مولفه واسط کاربر به تحلیلگر زبان طبیعی فرستاده می‌شود. این تحلیلگر، یک تحلیل بسیار ساده بر روی پرس و جو انجام داده و در صورت نیاز مجموعه کلماتی به پرس و جوی کاربر می‌افزاید. در پیاده‌سازی این مولفه، یک سری کلمات خاص در نظر گرفته شد که در صورت مشاهده لغات مشخصی مانند "what" به پرس و جو اضافه شود. به عنوان مثال، فرض کنید پرس و جوی ورودی "what is reinforcement learning?" باشد، پس از بررسی این جمله در تحلیلگر، این جمله بوسیله کلمات "introduction"، "tutorial" و "survey" پشتیبانی خواهد شد و پرس و جوی نهایی به صورت "Introduction tutorial survey 'what reinforcement learning'" در خواهد آمد.

۲ـ۱ ـ استدلال بر پایه موارد برای موتورهای جستجوی با دامنه خاص

استدلال بر پایه موارد (CBR) دستاوردی برای توسعه سیستم‌های دانشی است که قادر به بازیابی و استفاده مجدد از راه حل‌هایی هستند که برای موقعیت‌های مشابه در گذشته بکار گرفته شده‌اند [۲۳]. بنابر تعریفی دیگر استدلال بر پایه موارد یک نوع از سیستم‌های استدلال است که بر پایه استفاده از تجربیات گذشته که «مورد» نامیده می‌شوند، استوار است [۲۴]. موارد، تشریح موقعیت‌هایی هستند که که در آنها عامل یادگیر با هدف مشخص با دنیای اطراف خود محاوره می‌کند. موارد با سه-تایی‌های (p, s, o) نمایش داده می‌شوند که p صورت مساله است، s راه حل بکار گرفته شده و o نتیجه (حالت نهایی فضای اطراف عامل بعد از اجرای راه حل) است. فلسفه بنیادی استدلال بر پایه موارد این است که راه حل‌های موارد موفقیت آمیز باید به عنوان پایه‌ای برای حل مسائلی در آینده استفاده شوند که به میزان مشخصی شبیه مسائل قبلی هستند [۲۴]. موارد با نتایج غیر موفقیت آمیز و یا موارد منفی دانش اضافی برای سیستم فراهم می‌کنند از این طریق که عامل را از انجام اعمالی باز

می‌دارند که منجر به حالات و یا نتایج ناموفق می‌شوند.

سناریوی حل مساله با استفاده از CBR به صورت زیر می‌باشد: مساله جدیدی که به سیستم ارائه می‌شود به عنوان بخش مساله یک مورد جدید مطرح می‌شود. از میان موارد قبلی، آنهایی را که دارای مسائلی مشابه مساله جدید هستند بازیابی شده و از بین آنها مشابه‌ترین و نزدیک‌ترین مسائل به مساله جدید انتخاب و به عنوان پایه راه حل جدید به سیستم پیشنهاد می‌شود. این راه حلها به مساله جدید تطبیق داده می‌شود. مساله جدید و راه حل و نتیجه حاصل از آن در پایگاه داده موارد ذخیره می‌شود. مرحله آخر، خاصیت یادگیری تدریجی CBR را نشان می‌دهد که قادر است با توجه به تغییرات محیط به آرامی تغییر کند و خودش را با تغییرات محیط وفق دهد [۲۳]. الگوریتم‌ها و روشهای متفاوتی برای هر کدام از فازهای بازیابی، استفاده مجدد، تطبیق و نگهداری موارد پیشنهاد و به کار گرفته شده‌اند [۲۳]. در یک سیستم CBR چهار عنصر پایه قابل شناسایی هستند:

- واژگانی که برای تشریح موارد استفاده می‌شوند.
- معیارهای سنجش تشابه که برای مقایسه موارد استفاده می‌شود.
- پایگاه داده موارد (حافظه موارد) که در آن تشریح موارد ذخیره شده‌اند.
- دانش تطابق که برای تطبیق راه حل‌های قبلی بر مساله جدید استفاده می‌شود.

استدلال بر پایه موارد برای بازه وسیعی از کاربردها بکار گرفته شده است. به عنوان مثال می‌توان از استفاده از CBR در سیستم‌های بازیابی اسناد در بازیابی اطلاعات، انتخاب محصول برای مشتری در محیط‌های تجارت الکترونیک، نرم‌افزارهای کاربردی خدمات و پشتیبانی مشتریان و محصولات، امور تشخیص بیماری در پزشکی و نقص در تجهیزات فنی ماشین‌ها، سیستم‌های پیکربندی و امور طراحی نام برد.

در مورد سیستم‌های کامپیوتری و به طور خاص موتورهای جستجو یک سری حقایق وجود دارد:

- یادگیری باعث افزایش کارآیی سیستم‌ها می‌شود [۲۲].
- موتورهای جستجو می‌توانند از جستجوهای قبلی‌شان یاد بگیرند.
- یادگیری جستجوهای قبلی می‌تواند منجر به احتراز از جستجوی دوباره در مورد جستجوهایی شود که قبلاً هم به موتورهای جستجو ارائه شده‌اند [۲۲].

یک مطالعه بر روی پرس و جوهای ارائه شده به موتور جستجوی Alta Vista در یک دوره ۴۳ روزه [۲۵] نشان می‌دهد که تقریباً دو سوم تمام پرس و جوها در طول دوره ۶ هفته‌ای فقط یکبار ارائه شده‌اند. این حقیقت نشان می‌دهد که نیازهای اطلاعاتی بر روی وب متنوع هستند و یا به موتورهای جستجو به طریق‌های گوناگون عرضه می‌شوند. مطالعه‌ای مشابه بر روی موتور جستجوی Excite بر نتایج مشابهی دلالت دارد [۲۶]. دو نکته در مورد این مطالعات باید مد نظر قرار گیرند:

- این مطالعات بر روی موتورهای جستجوی همه منظوره انجام گرفته‌اند. در این موتورهای جستجو کاربران می‌توانند در هر زمینه‌ای پرس و جو کنند.
- در این مطالعات تکرار یک پرس و جو یعنی اینکه دقیقاً همان کلمات کلیدی در دو پرس و جو موجود باشند (تطابق دقیق کلمات کلیدی) و این مطالعات پرس و جوهای مشابه و یا مرتبط به هم را مطالعه نکرده‌اند و نتایج هم فقط با توجه به تطابق دقیق گزارش شده‌اند.

تاکنون هیچ تحلیل گسترده‌ای بر روی پرس و جوهای موتورهای جستجوی با دامنه خاص منتشر نشده است. بر طبق مشاهدات فوق، حدس می‌زنیم که در یک موتور جستجوی با دامنه خاص تعداد تکرار پرس و جوهای مشابه و یا مرتبط قابل ملاحظه است زیرا پرس و جوها در این موتورهای جستجو بر روی یک دامنه (موضوع) خاص بیان می‌شوند. با در نظر گرفتن نتایج تحلیل تاریخچه پرس و جوها در موتورهای جستجوی با دامنه خاص، می‌توان انتظار داشت که حداقل بیش از یک سوم از پرس و جوها در یک دامنه خاص با هم مشابه و یا مرتبط هستند. با این ملاحظات، CBR را به عنوان روش یادگیری بسیار مفید برای موتور جستجوی دامنه خاص پیشنهاد می‌کنیم زیرا در یک دامنه خاص در موارد (پرس و جوهای) مشابه می‌توانیم استفاده مجدد داشته باشیم و می‌توانیم از جستجوهای گذشته برای جستجوهای جدیدی استفاده کنیم که حداقل به میزان مشخصی مشابهت داشته باشند.

۲ـ۲ـ استفاده از استدلال بر پایه موارد در کنار یک کاوشگر متمرکز

کاوش متمرکز بر روی کیفیت اطلاعات و تشخیص صفحات مرتبط از بین حجم عظیم اطلاعات قابل دسترس بر روی وب تاکید دارد. یک کاوشگر متمرکز [۶و۷ و ۸] بر روی اینترنت جستجو کرده، صفحات مرتبط به یک موضوع خاص را که بخش کوچکی از وب را تشکیل می‌دهند را بدست آورده، شاخص‌بندی کرده و نگهداری می‌کند. بجای اینکه تمامی صفحات قابل دسترس بر روی وب جمع‌آوری شده و شاخص‌بندی شود تا اینکه قادر به پاسخگویی به تمامی پرس و جوهای ممکن کاربران باشیم، یک کاوشگر متمرکز حوزه مورد کاوش خود را تحلیل می‌کند تا ابرپیوندهایی را بیاید که بیشترین ربط به موضوع کاوش را داشته باشند و از نواحی غیر مرتبط وب احتراز کند. کاوشگر متمرکز ایده‌آل، حداکثر تعداد صفحات مرتبط به موضوع را می‌یابد و در همان حال حداقل تعداد صفحات غیر مرتبط به موضوع را می‌پیماید. در معماری پیشنهادی برای پاسخگویی به یک پرس و جو، مولفه CBR به همراه یک کاوشگر متمرکز اینکار را انجام می‌دهد.

مولفه توسعه پرس و جو، پرس و جوی توسعه یافته را به مولفه CBR می‌فرستد. فرآیند جستجو برای پاسخگویی به این پرس و جو را در دو سناریو می‌توان بررسی کرد. در سناریوی اول، حالت آغازین سیستم را در نظر بگیریم که حافظه موارد خالی است و یا دارای تعداد بسیار کمی مورد در آن ذخیره شده‌اند. سناریوی دوم حالتی را تشریح می‌کند که تعداد قابل ملاحظه‌ای از موارد درون حافظه موارد وجود دارند و قادر به پاسخگویی و یافتن موارد مشابه برای تعداد زیادی از پرس و جوها می‌باشد. در حالت آغازین سیستم که پایگاه داده موارد خالی است، پرس و جو مستقیماً به کاوشگر متمرکز فرستاده می‌شود. کاوشگر متمرکز جستجو کرده و به پرس و جوی درخواستی پاسخ می‌گوید و پس از رتبه‌بندی مجدد پاسخ به کاربر و مولفه CBR فرستاده می‌شود و توسط این مولفه یادگرفته می‌شوند. در حالت دوم، مولفه CBR به طور مستقیم با پرس و جو کار می‌کند. اگر هیچ مورد مشابهی در پایگاه داده موارد با مساله جدید یافت نشود مانند حالت اول، پرس و جو مستقیماً به کاوشگر متمرکز فرستاده می‌شود و گرنه موارد بسیار مشابه به عنوان ورودی فاز تطبیق استفاده می‌شوند.

۲ـ۲ـ۱ـ پایگاه داده موارد

موارد در پایگاه داده موارد به صورت ساختمان داده‌های زیر ذخیره می‌شوند:

- *پرس و جوی کاربر* به همان صورتی که از ورودی دریافت شده است.
- گرهی از سلسله مراتب مفهومی که پرس و جو به آن متعلق است. این دسته توسط کاربر در ورودی انتخاب می‌شود.
- *URLهای نتایج برگشتی* (اگر وجود دارند). موارد منفی، مواردی هستند که کاوشگر متمرکز هیچ پیوندی برای آنها برنگردانده است.
- *امتیاز هر صفحه* که توسط کاوشگر متمرکز برگردانده شده است. همانطور که اشاره شد کاوشگر به هر صفحه با توجه به میزان مشابهت به پرس و جو یک امتیاز (بین صفر و یک) نسبت می‌دهد.

۳ـ۱ـ۲ ـ معیارهای سنجش میزان تشابه موارد

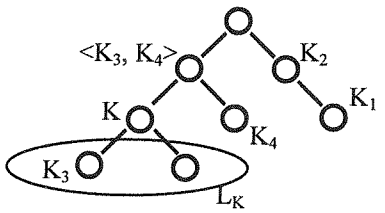
معیار سنجش تشابه ترکیبی از چند معیار دیگر است که بر اساس اطلاعاتی شکل گرفته است که در پایگاه داده موارد ذخیره شده است؛ یعنی سلسله مراتب مفهومی با دامنه خاص، مجموعه کلمات کلیدی پرس و جو و امتیاز هر صفحه برای محاسبه میزان تشابه بکار گرفته می‌شود. معیار تشابه کلی ترکیب وزنی تشابه پرس و جوی کاربران و میزان تشابه دسته‌ها (گره‌ها) در سلسله مراتب مفهومی برای مورد جدید و موارد قبلی است که بصورت زیر بیان می‌شود:

$$Sim(New\_Case, Old\_Case) = w_1 \times Query\_Similarity + w_2 \times Class\_Similarity$$

w_1 و w_2 وزنهای مکاشفه‌ای هستند که $w_1 + w_2 = 1$. از معیار ساده "فاصله ویرایشی" برای محاسبه تشابه پرس و جوی ورودی کاربر و موارد قبلی استفاده شده است. معیار "فاصله ویرایشی" ، فاصله را تعداد تفاوت میان رشته‌ها و علائم می‌داند. فاصله Levenshtein [۲۷] یک معیار محاسبه فاصله ویرایشی اولیه است که فاصله بین دو متن را بر اساس تعداد افزودنها، حذفها و یا جایگزینی‌های حروف می‌داند در حالی که سعی می‌شود یک رشته به دیگری تبدیل شود. همچنین رخداد عبارات

(با بیش از یک کلمه) هم در نظر گرفته شده است تا فاصله ویرایشی بین دو متن کاهش یابد. میزان این تشابه با عددی بین صفر و یک بیان می‌شود.

هر موردی در پایگاه داده، بجز بعضی از موارد منفی، به یک دسته در DSCH تعلق دارد. روشهایی برای محاسبه میزان تشابه گره‌ها در یک سلسله مراتب وجود دارد که در [۲۸] به آنها اشاره شده است. یک سلسله مراتب نوعی در در شکل ۵ نشان داده شده است. فرض کنید K یک گره داخلی سلسله مراتب باشد، سپس L_k بر تمامی برگهایی در زیر شاخه K اشاره دارد که از آن منشعب می‌شوند. علاوه بر آن $K_1 < K_2$ بر این دلالت دارد که K_1 گره بعدی K_2 در سلسله مراتب می‌باشد بدین معنی که K_2 در مسیر K_1 تا ریشه قرار دارد. در [۲۸] نزدیک‌ترین گره (جد) مشترک به صورت $<K_3,K_4>$ تعریف می‌شود بدین معنی که $K_3 \geq <K_3,K_4>$ و $<K_3,K_4> \geq K_4$ می‌باشد و هیچ گره K' وجود ندارد که $<K_3,K_4> > K' > K_3$ بطوریکه $K' \geq K_3$ و $K' \geq K_4$ برقرار باشد.



شکل (۵) یک سلسله مراتب نوعی.

میزان تشابه دو گره در سلسله مراتب مثلاً K_1 و K_2 را بصورت زیر تعریف می‌کنیم:

$$Sim(K_1, K_2) = Struc\_Sim(K_1, K_2) * Bag\_of\_Word\_Sim(K_1, K_2)$$

که در آن

$$Struc\_Sim = \frac{1}{2^L}\ ,\ L = Level\_diff(<K_1, K_2>, K_1) + Level\_diff(<K_1, K_2>, K_2)$$

به عنوان مثال، برای محاسبه Struc_Sim برای K_3 و K_4 در شکل ۵ داریم $L = 2+1$ و بنابراین $Struc\_Sim = 1/2^3 = 0.125$ و اگر دو گره یکسان باشند، بر اساس فرمول میزان تشابه آنها برابر ۱ محاسبه می‌شود. Bag_of_Word_Sim دو گره به صورت زیر محاسبه می‌شود:

$$Bag\_of\_Word\_Sim(K_1, K_2) = \begin{cases} 1 & K_1=K_2 \\ Term\_Sim(K_1,K_2) & Otherwise \end{cases}$$

$$Term\_Sim(K_1,K_2) = w_1 \times \frac{num(CW(K_1,K_2))}{num(W(K_1,K_2))} \times \sum_{i=1}^{n} \left| w_{K1}(CW_i(K_1,K_2)) - w_{K2}(CW_i(K_1,K_2)) \right| +$$
$$w_2 \times \frac{num(CT(K_1,K_2))}{num(T(K_1,K_2))} \times \sum_{i=1}^{m} \left| w_{K1}(CT_i(K_1,K_2)) - w_{K2}(CT_i(K_1,K_2)) \right|$$

CW= Common Words و Num = Number of و $W_{kl}(CW_i(K_1,K_2))$ برابر وزن کلمه مشترک nام گره‌های K_1 و K_2 در گره K_1. این وزنها در سبد هر گره برای هر کلمه ذخیره می‌شود. $W(K_1, K_2)$ تمامی کلمات در دسته‌های K_1 و K_2 هستند. کلمات مشترک فقط یکبار در این لیست ظاهر می‌شوند. CT = Common Terms (عبارات با بیش از یک کلمه) و $W_{kl}(TW_i(K_1,K_2))$ برابر وزن عبارت مشترک nام گره‌های K_1 و K_2 در گره K_1. این وزنها در سبد هر گره برای هر عبارت ذخیره می‌شود. $T(K_1, K_2)$ تمامی عبارات در دسته‌های K_1 و K_2 هستند. عبارات مشترک فقط یکبار در این لیست ظاهر می‌شوند. از آنجایی‌که اهمیت وجود عبارات مشترک در دو گره بیشتر از کلمات مشترک می‌باشد، وزن‌های مکاشفه‌ای w_1 و w_2 برای تمایز بین این دو استفاده شده‌اند. همواره رابطه $w_1 + w_2 = 1$ برقرار می‌باشد. میزان تشابه دسته‌ها در سلسله مراتب مفهومی می‌تواند

فقط یکبار و در فاز پیکربندی سیستم محاسبه شده و در فاز کاری سیستم به سادگی مورد استفاده قرار گیرند.

۲ـ۲ـ۳ ـ تطبیق موارد

بعد از محاسبه تشابه موارد ذخیره شده در پایگاه موارد با مورد جدید و یافتن یک حد مشخصی از تشابه، تطبیق دادن راه حل‌های قبلی بر مساله جدید مورد نیاز است. این تطبیق می‌تواند به فرم رتبه‌بندی دوباره نتایج برای مساله جدید و یا ادغام نتایج دو یا چند مورد بسیار مشابه به مورد جدید و برگرداندن تعدادی از آنها با میزان امتیاز ربط بالا به کاربر باشد. همچنین فاز تطبیق می‌تواند با استفاده از محاوره با کاربر و دریافت نظر او در انتخاب موردهای هدف انجام شود.

در حالتی که موارد یافته شده دارای تشابه بسیار بالایی به مساله جدید دارند و میزان تشابه موارد دیگر در پایگاه داده با مورد جدید بسیار کم است، مرتب‌سازی دوباره نتایج مورد با تشابه بالا و ارائه آن به کاربر پیشنهاد می‌شود. رتبه‌بندی جدید بر اساس ترکیبی از معیارها صورت می‌گیرد. وجود کلمه و یا عبارت مشابه در پرس و جوی جدید و در URL و یا عنوان صفحات در مورد قبلی، امتیاز صفحه در مورد قبلی از این پارامترهاست. هنگامی که میزان تشابه چند مورد در پایگاه داده به مساله جدید بالا و تقریباً به هم نزدیک می‌باشد، سپس URLهای موارد با میزان تشابه بالا با هم ادغام شده و مرتب‌سازی دوباره آنها مانند حالت قبلی انجام می‌شود. در حالتی که میزان تشابه بالاست و این مقادیر به هم نزدیک هستند بهتر است از کاربر کمک گرفته شده و نتیجه انتخابی او در پایگاه داده ذخیره شود. اگر با جوابهایی که به کاربر ارائه گردید، جواب مناسب را دریافت نکرده باشد، پرس و جوی توسعه یافته کاربر به کاوشگر متمرکز فرستاده شده تا مانند حالت معمول نتایج دریافت گردند.

مراجع

[1] Web Characterization (Size and Growth), http://wcp.oclc.org/stats.html, 2002.

[2] Search Engine Show Down Company, http://www.searchengineshowdown.com/stats/size.shtml, December 2002.

[3] The Search Engine Report ,http://searchenginewatch.com/sereport/index.html , December 2001.

[4] Lawrence S. and Giles C.L., "Accessibility of Information on the Web," Nature 400:107-109, July 8, 1999.

[5] Cohen W., McCallum A., Quass D. "Learning to undesrstand the web", *Bulletin of the IEEE Computer Society Technical Commitee on Data Engineering*, 2000.

[6] Chakrabarti S., Van Der Berg M., and Dom B. "Focused crawling: a new approach to topic-specific web resource discovery", *Proceedings of the 8th International World-Wide Web Conference (WWW8)*, 1999.

[7] Rennie J., McCallum A., "Using Reinforcement Learning to Spider the Web Efficiently", *Proceedings of ICML99*, 1999.

[8] Diligenti M., Coetzee F., Lawrence S., Lee Giles C., Gori M. "Focused Crawling using Context Graphs", *26th International Conference on Very Large Databases*, VLDB 2000, Cairo, Egypt, 2000, pages 527–534.

[9] Yang Q., Wang H.F., Wen J.R., Zhang G., Lu Y., Lee K.F. and Zhang H.J. "Toward a Next Generation Search Engine", *Proceedings of the Sixth Pacific Rim Artifact Artificial Intelligence Conference*, Melborne, Australia, August, 2000.

[10] Yang Y.J., Chien L.F., Lee L.S. "Speaker intention modeling for large vocabulary mandarin spoken dialogue", *Proceedings of Fourth International Conference on Spoken Language*, 1996.

[11] Bharat K. and Henzinger M. "Improved algorithms for topic distillation in a hyperlinked environment", *SIGIR Conference on Research and Development in Information Retrieval*, vol. 21, ACM, 1998.

[12] Barfourosh A. A., Motahari Nezhad H.R. "A new approach to information retrieval based on case cases reasoning and concept hierarchy in Cora", *Accepted in Third International conference in Data Mining Methods and Databases for Engineering, Finance and other Fields (Data Mining 2002)*, Bologna, Italy, 25-27 Sept. 2002.

[13] Motahari Nezhad H.R., Barfourosh A. A., "A new approach to expand user's query in Domain Specific Search Engines", *Accepted in Eight Annual Computer Society of Iran Conference (CSICC'2003)*, Mashhad - Iran, February 25-27, 2003.

[14] Barfourosh A. A., Motahari Nezhad H.R., "A Case Based Reasoning Framework for Domain Specific Search Engine", *Proceedings of The 2002 International Arab Conference on Information Technology (ACIT2002)*, Vol. 1., Qatar, December 16-19, 2002, pages 20-29.

[15] Han J. and Fu Y. "Dynamic Generation and Refinement of Concept Hierarchies for Knowledge Discovery in Databases", *AAAI'94 Workshop on Knowledge Discovery in Databases (KDD'94)*, Seattle, 1994, pages 157-168.

[16] McCallum A., Nigam K., Rennie J., Seymore K. "Automating the Contruction of Internet Portals with Machine Learning", *Information Retrieval Journal*, volume 3, Kluwer, 2000, pages 127-163.

[17] Chakrabarti S. "Data Mining for Hypertexts: A tutorial Survey", *ACM Exploration, ACM SIGKDD*, Volume 1, Issue 2, 2000, Pages 1-11.
[18] Rennie J., "Improving Multi-class Text Classification with Naive Bayes", *Master's Thesis,* Massachusetts Institute of Technology, AI Technical Report AITR-2001-004, 2001.
[19] Salton G., and Yang C., "On the specification of term values in automatic indexing", *Journal of Documentation 29,* 1973, pages 351-372.
[20] McCallum A., Nigam K., Rennie J., and Seymore K. "Building domain-specific search engines with machine learning techniques", *In AAAI-99 Spring Symposium on Intelligent Agents in Cyberspace*, 1999.
[21] Pollock A. and Hockley A. "What's wrong with Internet Searching", DLib Magazine.
[22] Olivia C., Change C., Enguix C.F., Ghose A.K. "Case-Based BDI Agents: An Effective Approach for Intelligent Search on the web", *Proceeding 1999 AAAI, Spring Symposium on Intelligent Agents in Cyberspace Stanford University*, USA, March 1999.
[23] Bartsch-Spörl B., Lenz M. and Hübner. "A. Case-Based Reasoning – Survey and Future Directions", *Knowledge-Based Systems, Lecture Notes in Artificial Intelligence*, Vol. 1570, Springer-Verlag, Berlin, 1999, pages 67-89.
[24] Kolonder, J.L. "Case-Based Reasoning". *Morgan Kauffman Publisher Inc.*, 1993.
[25] Silverstein C., Henzinger M., Marais H., and Moricz M., "Analysis of a Very Large AltaVista Query Log", *SRC Technical Note*, 1998 – 014.
[26] Wolfram D. "A Query-Level Examination of End User Searching Behaviour on the Excite Search Engine", *Proceedings of the 28th Annual Conference CAIS : Canadian Association for Information Science*, 2000.
[27] Levenshtein V. I., "Binary codes capable of correcting spurious insertions and deletions of ones", Russian Problemy Peredachi Informatsii 1, 1965, pages 12-25.
[28] Bergmann, R. "On the Use of Taxonomies for Representing Case Features and Local Similarity Measures", *Proceedings of the 6th German Workshop on Case-Based Reasoning (GWCBR'98)*, IMIB Series Vol. 7, Universität Rostock, 1998, pages 23-32.
[29] Mukherjea M. E. "WTMS: A System for Collecting and Analyzing Topic-Specific Web Information", *Proceedings of 10th World Wide Web (WWW10) Conference*, Hong Kong, 2001.
[30] Ester M., Groß M., Kriegel H., "Focused Web Crawling: A Generic Framework for Specifying the User Interest and for Adaptive Crawling Strategies", *Proceedings of VLDBD2001*, 2001.
[31] Aggarwal C., Al-Garawi F. and Yu S. "Intelligent Crawling on the World Wide Web with Arbitrary Predicates", *Proceedings of 10th World Wide Web Conference*, Hong Kong, 2001.
[32] Najork M., Wiener J. L., "Breadth-first search crawling yields high-quality pages", *Proceedings of 10th World Wide Web (WWW10) Conference*, Hong Kong, 2001.
[33] Menczer F., Pant G., Srinivasan P., Ruiz M. E., "Evaluating Topic Driven Web Crawlers", *Proceedings of SIGIR'01,* 2001.
[34] Dean J., Henzinger M. "Finding related pages in the World Wide Web", *In Proceeding of 8th Conference in World Wide Web (WWW8)*, 1999.
[35] Salton G., Buckley C., "Term-weighting approaches in automatic text retrieval", Information processing and management, 24,5(1998), 513-523.