بسمه تعالی

دانشکده مهندسی

گروه کامپیوتر

پایان‌نامه کارشناسی ارشد

خزش متمرکز وب با هدف کشف اسناد وب معنایی

ریحانه امامدادی

استاد راهنما: دکتر محسن کاهانی

بهمن ماه ۱۳۹۲

چکیده

با توجه به گسترش روزافزون وب‌معنایی و افزایش حجم داده‌های معنایی بر روی وب، لازم است تا این داده‌ها با کمک ابزارهایی از سطح وب جمع‌آوری شوند تا بتوان از آنها در کاربردهای مختلف استفاده کرد. یکی از ابزارهای مهم در این زمینه، خزنده وب است. خزنده وب، برنامه‌ای است که اسناد را به طور خودکار و با دنبال کردن پیوندهای داخل آنها از سطح وب جمع‌آوری می‌کند.

درحوزه وب‌معنایی، هدف خزنده جمع‌آوری نوع خاصی از اسناد یعنی اسناد وب‌معنایی است. این در حالی است که عدم اتصال کافی بین اسناد وب‌معنایی و احاطه شدن آنها توسط اسناد HTML، باعث شده است تا بسیاری از خزنده های وب‌معنایی، علاوه بر اسناد وب معنایی، اسناد HTMLرا نیز مورد خزش قرار دهند. اما با توجه به حجم بالای اسناد HTML و غیرمعنایی بودن بسیاری از پیوندهای داخل آنها، واکشی این اسناد و پیمودن همه مسیرهای ایجاد شده توسط آنها، باعث اتلاف منابع و زمان و پایین آمدن سرعت دستیابی به اسناد وب معنایی می‌شود.

در این پایان نامه، یک خزنده متمرکز ارائه شده است که بر خلاف خزنده‌های متمرکز موجود که به دنبال جمع‌آوری اسناد در مورد یک موضوع خاص هستند، خزنده پیشنهادی تلاش می‌کند تا بر مبنای دو تابع ارتباط،میزان توانایی پیوندهای استخراج شده از اسناد HTML را در دستیابی به اسناد وب‌معنایی تخمین بزند و بر اساس آن، اولویت پیوندها را برای خزش تعیین کند. نتایج ارزیابی‌ها نشان می‌دهد که بکارگیری فرآیند خزش متمرکز و توابع ارتباط پیشنهادی در حوزه خزش وب معنایی، موجب دستیابی سریعتر به اسناد وب معنایی و کاهش حجم واکشی اسناد غیر معنایی می‌شود.

کلمات کلیدی:خزش وب، خزنده وب‌معنایی، اسناد وب‌معنایی، اسناد HTML، خزش متمرکز، تابع ارتباط.

فهرست مطالب

فصل ۱-مقدمه ۱

۱-۱- تعریفمساله ۱

۱-۲- روشپیشنهادی ۴

۱-۳- نوآوریهایروشپیشنهادی ۵

۱-۴- ساختارپایاننامه ۶

فصل ۲-مرورادبیات ۷

۲-۱- خزشوب ۷

۲-۱-۱- معماریخزندهوب ۸

۲-۱-۲- چالشهایاصلیدرفرآیندخزشوب ۱۴

۲-۲- خزشوبمعنایی ۱۷

۲-۲-۱- اسنادوبمعنایی ۱۸

۲-۲-۲- سیاستانتخاباسناددرخزندههایوبمعنایی ۲۰

۲-۲-۳- تجزیهاسنادوبمعنایی ۲۴

۲-۲-۴- شیوهپیونداسناددروبمعنایی ۲۵

۲-۲-۵-روشهایجمعآوریSeedدرخزندههایوبمعنایی ۲۶

۲-۲-۶- سیاستموازیسازیدرخزندههایوبمعنایی ۲۹

۲-۲-۷- خلاصهکارهایگذشته ۳۳

۲-۳- خزشمتمرکزوب ۳۶

۲-۳-۱- مفاهیممرتبطباموضوع ۳۸

۲-۳-۲- تابعارتباط ۴۰۳۹
۳-۳-۲- پویاییعمقخزش ۴۳۴۲
2-3-4-آموزشخزندهمتمرکزهمزمانبافرآیندخزش ۴۴۴۲
۵-۳-۲- استفادهازخزشمتمرکزدرحوزهوبمعنایی ۴۵۴۳
۶-۳-۲- استفادهازخزشمتمرکزبرایپیشبینینوعاسناد ۴۵۴۴
۷-۳-۲- خلاصهکارهایگذشته ۴۷۴۵

۴-۲- خلاصهفصل ۴۹۴۶

فصل ۳-روشپیشنهادی ۴۹۴۸

۱-۳- خزندهمتمرکزHTMLباهدفدستیابیبهاسنادوبمعنایی ۵۰۴۸

۱-۱-۳- تابعارتباطبرایشناساییپیوندهایRDF ۵۲۵۰
۲-۱-۳- تابعارتباطبرایپیشبینیپیوندهایHTMLوالدRDF ۵۵۵۳
۳-۱-۳- تابعارتباطنهایی ۵۹۵۷
۴-۱-۳- الگوریتمخزشمتمرکز ۶۰۵۸

۲-۳- چارچوبهاپیشنهادیبرایخزشوبمعنایی ۶۱۶۰

۳-۳- خلاصهفصل ۶۵۶۴

فصل ۴-پیادهسازیوارزیابی ۶۷۶۵

۱-۴- پیادهسازی ۶۷۶۵

۲-۴- معیارهایارزیابی ۶۷۶۵

۳-۴- ارزیابیخزندهمتمرکزپیشنهادی ۶۹۶۷

۴-۳-۱- محیطارزیابی ۷۰۶۸
۴-۳-۲- خزندههایموردمقایسه ۷۰۶۸
۴-۳-۳- روشارزیابی ۷۲۷۰
۴-۳-۴- تحلیلنتایج ۷۳۷۱

۴-۴- ارزیابیچارچوبهایخزشپیشنهادیبرایوبمعنایی ۸۱۷۸

۴-۴-۱- محیطارزیابی ۸۱۷۹
۴-۴-۲- خزندههایموردمقایسه ۸۱۷۹
۴-۴-۳- روشارزیابی ۸۱۷۹
۴-۴-۴- تحلیلنتایج ۸۱۷۹

۴-۵- خلاصهفصل ۸۱۷۹

فصل ۵-نتیجهگیریوکارهایآینده ۸۲۸۰

مراجع ۸۳۸۱

فهرست شکل‌ها

شکل (۲-۱): نمودارجریانکاریکخزندهوب ۹۷

شکل (۲-۲): یکسندHTMLودرختتگهایداخلآن (ساختارDOM) ۱۳۱۱

شکل (۲-۲): چارچوبخزشترکیبیدرموتورجستجویSWOOGLE ۲۹۲۵

شکل (۲-۳): مدلخطلولهپنجمرحلهای[HAR2006] ۳۱۳۷

شکل (۲-۴): معماریخزنده[SLUG[DOD2006 ..... ۳۲۲۸

شکل (۲-۵): یکمعماریکلیبرایخزندههایمتمرکز ۳۷۳۳

شکل (۲-۶): یکگرافزمینهباعمقدو..... ۴۰۳۶

شکل (۳-۱): شبهکدتابعارتباطبرایخزندهمتمرکزپیشنهادی..... ۶۰۵۵

شکل (۳-۲): معماریخزندهمتمرکزپیشنهادی ۶۱۵۶

شکل (۳-۳): چارچوبخزشپیشنهادیبرایوبمعناییشاملیکخزندهRDFویکخزندهمتمرکزHTML..... ۶۳۵۹

شکل (۴-۱): میانگیندرصدپشتیبانیواطمینانهریکازقوانیندرتابعارتباطخزندهمتمرکزپیشنهادی ۷۸۶۸

شکل (۴-۲): میانگیننرخدستیابیبهاسنادوبمعناییبرایسهخزندهموردمقایسه..... ۷۹۶۹

فهرست جدول‌ها

جدول ۴-۱: PLDهایانتخابیبرایارزیابیخزندهمتمرکزپیشنهادی..... ۷۰۶۱

جدول (۴-۲): آمارحاصلازخزشفضایانتخابیتوسطخزندههایموردمقایسه..... ۷۴۶۵

جدول (۴-۳): دقتخزندهمتمرکزپیشنهادیبهتفکیکنوعپیوندهاپیشبینیشده ۷۶۶۷

فصل ۱- مقدمه

در این فصل،ابتدا به تعریف مسئله پرداخته می‌شود و درآن علت نیاز به خزش وب معنایی و چالش‌های آن توضیح داده می‌شود. در بخش بعدی، راه‌حل پیشنهادی برای برطرف کردن بعضی از چالش‌های موجود به اختصار شرح داده می‌شود. در ادامه نیز نوآوری‌های روش پیشنهادی و ساختار بقیه مطالب پایان‌نامه بیان می‌شود.

۱-۱- تعریف مساله

دنیای وب اطلاعات سودمند زیادی را به صورت الکترونیکی و عمدتا در قالب اسناد HTML[۱] در اختیار کاربرانش قرار می‌دهد. اما قابل فهم نبودن اطلاعات موجود در اسناد HTMLبرای ماشین‌ها، باعث شده است که حتی موتورهای جستجوی قدرتمندی مانند گوگل قادر به پاسخگویی به پرس‌و-جوهای ساخت‌یافته کاربران نباشند. در این راستا در سال ۱۹۹۸، وب معنایی[۲]به عنوان نسل جدید وب معرفی شد[Ber01].

در وب معنایی، تلاش می‌شود تا محتوای اسناد وب به گونه‌ای نمایش داده شود تا علاوه بر انسان‌ها، برای ماشین‌ها (برای مثال موتورهای جستجو) نیز قابل فهم باشد. برای رسیدن به این هدف، از زبان‌های استاندارد وب معنایی مانند RDF/S[۳]و OWL[۴]و به طور کلی مدل داده‌ی‌RDFبرای انتشار داده‌های معنایی بر روی وب استفاده می‌شود. در مدل داده‌ی‌RDF، اشیا و روابط میان آنها با

---

[۱] Hyper Text Markup Language

[۲] Semantic Web

[۳] Resource Description Framework / RDF Schema

[۴] Web Ontology Language

Formatted: Line spacing: single

استفاده از URIنام‌گذاری شده و سپس به صورت سه‌گانه[۲] در قالب (شی، صفت، مقدار) توصیف می‌شوند[Ber01]. به سندی که به طور کامل به یکی زبان‌های وب معنایی نوشته شده است، سندوب معنایی گفته می‌شود[Ber01,Din06].

امروزه، حجم داده‌های معنایی منتشرشده بر روی وب به سرعت در حال افزایش است. برای مثال، در پروژه ابر داده‌های پیوندی[۳] که در واقع تلاشی در جهت تحقق وب معنایی است، مطابق با آخرین آمار بروز شده در سال ۲۰۱۱، در حدود ۲۹۵ دیتاست[۴] معنایی، شامل ۳۱ میلیارد سه‌گانه منتشر شده است.[۵]

در حال حاضر، موتورهای جستجوی وب معنایی یکی از کاربردی‌ترین حوزه‌ها برای استفاده از داده‌های معنایی منتشر شده بر روی وب هستند. در هر موتور جستجو، اولین مولفه موردنیاز برای شاخص گذاری[۶] و بازیابی داده‌ها، بخش جمع‌آوری داده است که برای این منظور از یک خزنده وب[۷] استفاده می‌شود[Din06,Bat12,Hog11,Che09,Sab07,Ore08].

خزنده وب، یک برنامه (عامل نرم‌افزاری) است که اسناد را به طور خودکار از وب واکشی[۸] (دانلود)کرده و محتوای آنها را در یک مخزن ذخیره می‌کند. سپس، پیوندهای داخل اسناد را استخراج

---

۱ Uniform Resource Identifier

۲ Triples

۳ Linking Open Data Cloud

۴ Dataset

۵ http://lod-cloud.net/state/

۶ Indexing

۷ Web Crawler

۸ Fetch

کرده و این مراحل را برای پیوندهای بدست آمده تکرار می‌کند[Pan04,Dhe11].

با توجه به حجم روزافزون اطلاعات و سرعت بالای تغییر اسنادبر روی وب و از سوی دیگر محدودیت‌های پهنای باند شبکه و هزینه‌های زمانی، یک خزنده وب در طول فرآیند خزش با چالش‌های مختلفی روبرو می‌شود. شناسایی و انتخاب اسناد با اهمیت تر و تعیین اولویت آنها برای خزش، تعیین اولویت و زمان برای خزش مجدد اسناد، تلاش برای کاهش سربار ناشی از خزش بر روی سایت‌ها و تعیین سیاست موازی‌سازی برای دسترسی به حجم بالاتری از اسناد در زمان کمتر، از جمله این چالش‌ها است [Pan04,Dhe11].

در صورتی که خزنده به دنبال جمع‌آوری اسناد وب معنایی از سطح وب باشد، خزنده‌وب معنایی نامیده می‌شود. در این حالت، خزنده با دنبال کردن URIهای داخل اسناد وب معنایی،فرآیند خزش را ادامه می‌دهد[Din06, Bat12, Hog11, Che09, Sab07, Ore08, Dod06, Hae06, Ise10, Las12].یک خزنده وب معناییباید در سیاست انتخاب اسناد،ضمن تلاش برای دستیابی سریعتر به اسنادوب معنایی، حجم واکشی اسناد غیرمعنایی را نیز کاهش دهد و بدین ترتیب در مصرف منابع از جمله پهنای باند شبکه صرفه‌جویی کند[Bat12,Hog11]. اما برای رسیدن به این هدف، خزنده‌های وب معنایی با مشکلاتیروبرو هستند که در ادامه به آن می‌پردازیم.

بر خلاف اسناد HTML، اتصال و پیوند کافی بین اسناد وب‌معنایی وجود ندارد. همچنین، اسناد وب‌معنایی به صورت پراکنده در سطح وب قرار دارند و بسیاری از آنها توسط اسناد HTML احاطه شده‌اند. از این رو، در بعضی از خزنده‌های وب‌معنایی، اسناد HTML نیز مورد خزش قرار می-گیرندو از اتصال بین آنها برای دستیابی به حجم بیشتری از اسناد وب‌معنایی استفاده می‌شود [Sab07,Hae06,Ore08,Bat12,Las12]. اما با توجه به حجم بالای اسناد HTML و غیرمعنایی بودنبسیاری از پیوندهای داخل آنها، واکشی این اسناد و پیمودن همه مسیرهای ایجاد شده توسط آنها، باعث اتلاف منابع و زمان وپایین آمدن سرعت دستیابی به اسناد وب معنایی می-

Commented [m1]: همه پاورقی ها خطوطشان با فاصله ۱ باشد

Commented [m2]: لزومی ندارد برای این موضوع کوچک اینقدر ارجاع بیاورید. یک یا دو تا کافی است

شود[Bat07,Hog11].

در این راستا، بعضی از خزنده‌های وب معناییضمن خزش اسناد HTML، تلاش می‌کنند تا با اعمال سیاست‌های تنبیهی، حجم واکشی اسناد غیر معنایی را کاهش دهند. برای مثال با تنظیم عمق خزش، از پیشروی خزنده در مسیرهایی که احتمال رسیدن به اسناد وب معنایی در آنها کم است، جلوگیری می‌شود[Bat12]. در این حالت، خزنده بر اساس تجربیات گذشته در مورد خزش پیوندها تصمیم‌گیری می‌کند. در حالی کهممکن است در مسیر جاری امکان دستیابی به یک سند وب معنایی وجود داشته باشد و تنها به علت غیر معنایی بودن اسناد قبلی، خزنده امکان دسترسی به آن سند را پیدا نمی‌کند.

از سوی دیگر، در روش‌های موجود برای خزش وب‌معنایی، پیوندها اولویت یکسانی برای خزش دارند و یااولویت خزش آنها بر مبنای دامنه‌ای تعیین می‌شود که متعلق به آن هستند. برای مثال، در طول فرآیند خزش، هر چه قدر از یک دامنه اسناد وب معنایی بیشتری بدست آید، خزنده پیوندهای بیشتری از آن دامنه را مورد خزش قرار می‌دهد [Bat07,Hog11]. اما در هیچ یک از روش‌های موجود، تمایزی بین پیوندهای داخل یک دامنه برای تعیین اولویت خزش آنها وجود ندارد. در حالی که ممکن است خزنده با خزش یک پیوند و پیمودن مسیرهای ایجاد شده توسط آن، به اسناد وب معنایی زیادی دست پیدا کند و یا در مقابل، با اسناد غیر معنایی زیادی روبرو شود.

موارد بیان شده نشان می‌دهد کهشیوه برخورد با اسناد و پیوندهای HTML در فرآیند خزش وب معنایی و همچنین اولویت بندی پیوندها ، تاثیر زیادی بر عملکرد خزنده از نظر میزان دستیابی به اسناد وب معنایی و حجم واکشی اسناد غیر معنایی دارد.

۲-۱- روش پیشنهادی

با توجه به چالش‌های مطرح شده و اهمیت خزش اسناد HTML در خزش وب معنایی، راهکار

پیشنهادی این است که با رویکرد مشابه به خزنده‌های متمرکز[1](موضوعی)، پیوندهای استخراج شده از اسناد HTML مورد تحلیل قرار گرفته و بر مبنای آن در مورد اولویت خزش پیوندها تصمیم‌گیری شود.

در خزش متمرکز موضوعی، خزنده به دنبال جمع‌آوری اسنادی در مورد یک یا چند موضوع خاص است. در این راستا، خزنده متمرکز با استفاده از یک تابع ارتباط، پیوندها را تحلیل کرده و میزان ارتباط آنها رابا موضوع مورد نظر تخمین می‌زند و بر این اساس، اولویت پیوندهارا برای خزش تعیین می‌کند[Cha99].

انگیزه راهکار پیشنهادی این است که یک خزنده وب معنایی می‌تواند با بکارگیری روش‌های موجود در خزنده‌های متمرکز، از آنها برای تحلیل پیوندهای استخراج شده از اسناد HTMLاستفاده کند و میزان توانایی پیوندها را در دستیابی به اسناد وب معنایی، تخمین بزند. این کار باعث می‌شود تا با شناسایی پیوندهای اشاره‌کننده به اسناد وب معنایی در مرحله استخراج پیوند و دادناولویت بالاتربه آنها برای خزش،خزنده دستیابی سریعتری به اسناد وب معنایی داشته باشد. همچنین، شناسایی اسناد HTMLای که شامل پیوندهایی به اسناد وب معنایی هستند و دادن اولویت بالاتر به آنها نسبت به سایر اسناد HTML، می‌تواند به خزنده برای رسیدن به این هدف کمک کند. در صورت شناساییاسناد غیر معنایی قبل از مرحله واکشی نیز، خزنده می‌تواند از واکشی آنها جلوگیری کرده و بدین ترتیب حجم واکشی اسناد غیر معنایی را کاهش دهد.

۳-۱- نوآوری‌های روش پیشنهادی

نوآوری‌های روش پیشنهادی را می‌توان در موارد زیر خلاصه کرد:

---

[1] Focused Crawler

۱- اولویت‌بندی پیوندها بر اساس میزان توانایی آنها در هدایت خزنده به سمت اسناد وب معنایی

۲- ارائه یک خزنده متمرکز با هدف دستیابیبه اسناد وب‌معنایی

۳- ارائه یک تابع ارتباط برای شناسایی پیوندهای اشاره‌کننده به اسناد وب معنایی

۴- ارائه یک تابع ارتباط برای شناسایی اسنادHTMLای که شامل پیوندهایی به اسناد وب معنایی هستند.

۵- ارائه چارچوب جدید برای خزش وب معناییبر مبنای خزش متمرکز وب

۴-۱- ساختار پایان‌نامه

ساختار مطالب این پایان‌نامه در ادامه توضیح داده شده است. در بخش بعدی، ابتدا مفاهیم پایه در مورد خزنده‌های وب و سپس کارهای مرتبط در حوزه‌های خزش وب معنایی و خزش متمرکز وب مورد بررسی قرار گرفته است. در فصل سوم، روش پیشنهادی و چگونگی تحلیل پیوندهای استخراج-شده از اسناد HTML، توضیح داده شده است. در فصل چهارم، جزئیات پیاده‌سازی سیستم پیشنهادی، نحوه ارزیابی و نتایج آن بیان شده است. فصل پنجم نیز به نتیجه‌گیری و کارهای آینده اختصاص دارد.

فصل ۲- مرور ادبیات

سیستم پیشنهادی این پایان نامه، یک خزنده وب است که با بکارگیری روش‌های موجود در خزنده‌های متمرکز وب، تلاش می‌کند تا سیاست انتخاب اسناد در خزنده‌های وب معنایی را بهبود ببخشد. از این رو، در این فصل پس از بیان مفاهیم پایه در زمینه "خزش وب"، کارهای مرتبط موجود در حوزه‌های "خزش وب معنایی" و "خزش متمرکز وب"، مورد بررسی قرار می‌گیرد.

۱-۲- خزش وب

هر سند وب دارای یک آدرس یا شناسه یکتا (URI) است که از طریق آن قابل دسترسی می-باشد.URI، شامل نام پروتکل انتقال برای دستیابی به سند (مانند HTTP[۱])، نام سایت میزبان سند، شماره درگاه، بخش مسیر (نام پوشه‌ها، نام فایل و پسوند فایل) و همچنین بخش پرس‌وجواست. مطابق با استاندارد سازمان جهانی وب (RFC2396[۲]) هر URIبه صورت زیر بیان می‌شود:

[پرس‌وجو]?[مسیر]/[درگاه]:[سایت میزبان]//:[پروتکل]

محتوای یک سندوب علاوه بر اطلاعات متنی و چندرسانه‌ای، می‌تواند شاملآدرس اسناد دیگر نیز باشد که به آنها پیوند گفته می‌شود.خزنده وب، برنامه‌ای است که به صورت خودکار، با دسترسی به هر سند وب و دنبال کردن پیوندهای داخل آن، به جمع‌آوری اسناد وب می‌پردازد. نتیجه این کار، بدست آوردن مجموعه‌های بزرگی از اسناد است که موتورهای جستجو، کتابخانه‌های دیجیتال، پرتال‌های عمومی و غیره،با توجه به اهدافشان از آناستفاده می‌کنند[Pan04,Dhe11].

---

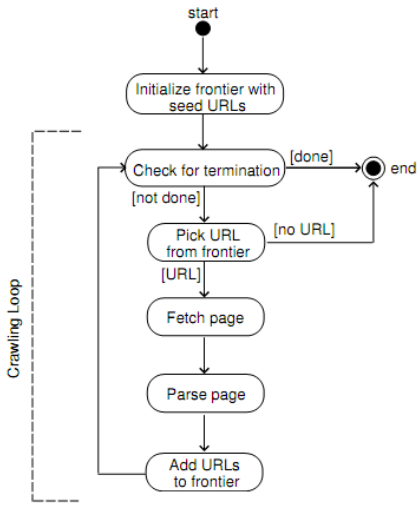
[۱] Hypertext Transfer Protocol

[۲] http://www.ietf.org/rfc/rfc2396.txt

در ادامه این بخش، جزئیات بیشتری از معماری خزنده وبو وظایف مولفه‌های اصلی آن بیان می‌شود. سپس چالش‌های اصلی در فرآیند خزش وب مورد بررسی قرار می‌گیرد.

۱-۱-۲- معماری خزنده وب

شکل ۲-۱ نمودار جریان کار یک خزنده وب را نشان می‌دهد. برای شروع فرآیند خزش، ابتدا آدرس چند سند وب به خزنده داده می‌شود. به این آدرس‌ها که خارج از چرخه خزش به خزنده داده می‌شود، seedگفته می‌شود.خزنده پس از قرار داردنseedها در یک صف، وارد چرخه خزش می‌شود. در هر چرخه، یک آدرس از صف برداشته می‌شود و سند مربوط به آنبا استفاده از پروتکل HTTPاز وب واکشی (دانلود)می‌شود. در مرحله بعد سند تجزیه[۱]می‌شود و پیوندهای داخل آن استخراج می-شود. خزنده آدرس هر پیوند بدست آمده را درصف قرار می‌دهد و این چرخه رابرای آدرس‌های جدید تکرار می‌کند.



[۱]Parse

شکل (۲-۱): نمودار جریان کار یک خزنده وب[Pan04]

مجموعه‌ی اسناد وب و پیوندهای میان آنها را می‌توان به صورت یک گراف بسیار بزرگ در نظر گرفت کهوظیفه خزنده وب، پیمایش این گراف است. از سوی دیگر، خزنده با انجام فرآیند خزش، گرافی از اسنادخزش شده را تشکیل می‌دهد که به آنگراف خزش گفته می‌شود. بنابراینگراف خزش، زیر مجموعه‌ای از گراف وب است. اسناد، گره‌های باز شده و پیوند ها، گره‌های باز نشده‌ی گراف خزش هستند. در گراف خزش، به سندی که به یک سند دیگر پیوند می‌دهد، والد آن سند گفته می‌شود. آدرس‌هایی نیز که در ابتدای فرآیند خزش به خزنده داده می‌شوند، گره‌های شروع گراف هستند؛ با شروع از این گره‌ها، گراف وب با اجرای چرخه‌های خزش پیمایش می‌شود و به صورت همزمان گراف خزش ساخته می‌شود.

اما به دلیل این که گراف وب، یک گراف دارای حلقه است، باید راه‌هایی را برای کشف حلقه‌ها در نظر گرفت تا از تکرار بی نهایت چرخه خزش جلوگیری شود؛ یکی از این راه‌ها، اجتناب از خزش دوباره اسنادقبلا واکشی شده در هر دور از فرآیند خزش است.

برای اتمام چرخه خزش نیز می‌توانشرایط مختلفی رادر نظر گرفت. برای مثال، می‌توان خزنده را به گونه ای تنظیم کرد که پس از واکشی تعداد مشخصی از اسناد و یا رسیدن به یک عمق مشخصی از گراف خزش، متوقف شود[Pan04].

۱-۱-۱-۲- صف خزش

صف خزش، شامل همه آدرس‌های واکشی نشده توسط خزنده و یا همان پیوندها است. برای خزنده‌های معمولی می‌توان از یک صفFIFO[1]برای ذخیرهپیوندها استفاده کرد. در این

[1] First In First Out

حالتخزندهگراف وب رابه صورت اول-سطح[1]پیمایش می‌کند.روش دیگر، آن است که از یک صف اولویتبرای ذخیرهپیوندها استفاده شود. در این روش، ابتدا خزندهبر اساس چند معیار، امتیاز هر سند را محاسبه می‌کند. سپس اسناد به ترتیب امتیازشان در صف قرار می‌گیرند. با این روش،خزنده گراف وب را به صورت اول-بهترین[2] پیمایش می‌کند. در بعضی از خزنده‌ها نیز از روش اول-عمق[3]برای پیمایش گراف وب استفاده می‌شود. دراین حالت، صف‌های خزش از نوع LIFO[4]هستند. به عبارت دیگر، در هر لحظه، اولویت با پیوندهایی است که اخیرا در صف قرار گرفته‌اند [Pan04].

از سوی دیگر در بعضی از خزنده‌ها، به URIهای هر PLD[5] زیر دامنه و یا سایت، یک صف جداگانه اختصاص داده می‌شود. از دلایل اصلی این کار می‌توان به کنترل سایت‌های با حجم بالا و جلوگیری از تاثیر سایت‌های اسپم[6] رعایت عدالت میان PLD ها و کاهش سربار ناشی از خزش بر روی PLD ها اشاره کرد[Bat07,Ise10,Cyg11,Hog2011].

همچنین در بیشتر خزنده‌ها، به دلیل محدودیت‌های حافظه اصلی، صف خزش بر روی حافظه جانبی ذخیره می‌شود. این کار علاوه بر افزایش مقیاس‌پذیری، یک وضعیت پایدار را برای خزش فراهم می‌کند که از آن می‌توان برای ادامه یک فرایند خزشاز قبل انجام شده و یا بازگشت از حالت شکست استفاده کرد. در اینجا، لازم است تا در ابتدای هر دور، به میزان فضای قابل استفاده از حافظه اصلی،

---

[1] Breadth-First

[2] Best-First

[3] Depth-First

[4] Last In First Out

[5] Pay Level Domain: به دامنه‌های اینترنتی خریداری شده، PLD گفته می‌شود؛ مانند amazon.com و um.ac.ir.

[6] spam

مجموعه‌ای از پیوندهای موجود در صف حافظه جانبی به صف حافظه اصلی منتقل شوند[Pan04].

۲-۱-۱-۲- واکشی سند

در مرحله واکشی، خزنده باید محتوای سند را از وب دانلود کند. برای این کار، خزنده ابتدا یک یک اتصال را با سایت میزبان سند برقرار می‌کند.بدین ترتیب که از طریقپروتکلHTTPیک درخواست به سایت میزبان[۱] آن سند ارسال شده و پاسخ آن را دریافتمی‌شود. پاسخ این درخواست، شامل اطلاعاتی مانندکد وضعیت واکشی، آدرس‌های تغییر مسیرو نوع محتوایسند است. خزنده پس از بررسی کد وضعیت، در صورت موفق بودن برقراری اتصال (کد وضعیت=۲۰۰)، می‌تواند محتوای سند را دانلود کند.همچنین، در صورتی که کد وضعیت واکشی، نشان دهنده تغییر آدرس باشد (کد وضعیت برابر با 3xx باشد)، خزنده می تواند آدرس جدید را از سرآیند[۲]پاسخ برداشته و آن را به عنوان پیوند در صف خزش قرار دهد تا در دور بعد مورد واکشی قرار بگیرد[Pan04]..

از سوی دیگر، اگر خزنده به دنبال جمع‌آوری نوع خاصی از اسناد مانند عکس باشد، می تواند فیلد نوع محتوا در سرآیند پاسخ را بررسی کند و در صورتی که مقدار آن برای مثال برابر با"image/gif"باشد، آن را دانلود کند و از واکشی سایر اسناد جلوگیری کند.به این عمل در اصطلاح صافی واکشی گفته می‌شود[Pan04,Dhe11].

۲-۱-۱-۳- تجزیهسند

وقتی که خزنده، محتوای یک سند را از وب واکشی می‌کند، باید عملیاتی بر روی سند انجام شود تا بتوان اطلاعات موجود در سند را استخراج کرده و از آنها استفاده کرد. به این فرآیند تجزیه سند گفته می‌شود. امروزه پارسرهای مختلفی برای تجزیه انواع اسناد وب پیاده‌سازی شده‌اند. عملیات

۱Host

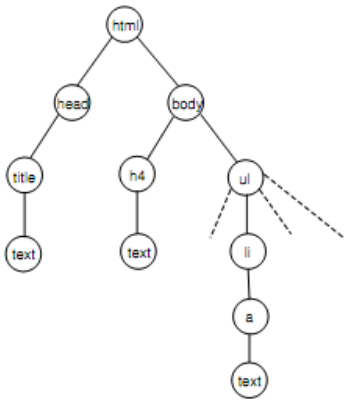
۲header

تجزیه یک سند می‌تواند شامل مراحل، حذف کلمات تکراری و بی ارزش(مانند it، the و غیره)، ریشه-یابیکلمات و غیره باشد. بعضی از پارسرها نیز قابلیت مرتب‌سازی[1]و استخراج درخت تگ (ساختار DOM[2]) رااز سند HTML دارند. در ساختار DOM، تگ‌های داخل سند HTML به صورت یک نمودار درختی نمایش داده می‌شوند. شکل ۲-۲ یک نمونه سند HTML و ساختار DOM معادل با آن را نشان می‌دهد. داده‌های بدست آمده در مرحله تجزیه سند، در مخزنی بر روی حافظه جانبی ذخیره می‌شوند تا عملیات دیگری همچون شاخص‌گذاری، استدلال، رتبه‌بندی و پرس و جو بر روی آنها انجام شود[Pan04].

[1] Tidying

[2] Document Object Model

```
<html>
<head>
<title>Projects</title>
</head>
<body>
<h4>Projects</h4>
<ul>
  <li> <a href="blink.html">LAMP</a> Linkage analysis with multiple processors.</li>
  <li> <a href="nice.html">NICE</a> The network infrastructure for combinatorial exploration.</li>
  <li> <a href="amass.html">AMASS</a> A DNA sequence assembly algorithm.</li>
  <li> <a href="dali.html">DALI</a> A distributed, adaptive, first-order logic theorem prover.</li>
</ul>
</body>
</html>
```



شکل (۲-۲): یک سند HTML و درخت تگ های داخل آن (ساختار DOM)[Pan04]

۴-۱-۱-۲- استخراج پیوند

پس از تجزیه یک سند، پیوندهای داخل آن استخراج شده و به صف خزش افزوده می شوند. برای مثال در مورد اسناد HTML،می‌توان از یک پارسربرای یافتن پیوندهای داخل تگ‌لنگر[1] و بدست آوردن مقادیر مربوط به صفات href استفاده کرد.

اما URIهای استخراج شده از سند، همیشه فرمتدرستی ندارند.برای مثال، URIهای نسبی باید بهURIهای مطلق تبدیل شوند. همچنین،خزندهباید همهURIها را به یک شکل منحصر فرد استاندارد نگاشت دهد. به مجموعه این کارها نرمال سازی URI گفته می‌شود.هدف از نرمال سازی URIها،

[1] Anchor

جلوگیری از واکشی منابع یکسان در طول فرآیند خزش می‌باشد[Pan04,Dhe11].

گاهی اوقات با توجه به هدف خزش، لازم نیست تا همه پیوندهای استخراج شده از یک سند، به صف خزش افزوده شوند. به این مرحله در اصلاح صافی پیوند[۱]گفته می‌شود. برای مثال خزنده می‌تواند از پیوندهایی که از پروتکل FTPاستفاده می‌کنند، صرف نظر کند و یا فقط پیوند های را برای خزش انتخاب کند که دارای پسوند فایل".rdf" هستند[Ise10].

۵-۱-۱-۲- فراداده‌های خزش

در طول فرآیند خزش، مجموعه‌ای از فراداده‌ها[۲]در مورد وضعیت واکشی، بازیابی و پردازش اسناد نگهداری می‌شود. معمولا از این فراداده‌ها برای ارئه گزارش و یا تحلیل پیشرفت فرآیند خزشاستفاده می‌شود. در صورت اولویت‌بندی پیوندها، اولویت هر پیوند نیز در مخزن فراداده ثبت می‌شود. همچنین می‌توان برای تصمیم‌گیری در مورد بروز رسانی و ملاقات دوباره پیوندها، از فراداده‌هایی مانند آخرین تاریخ دستیابی و هش[۳]محتوا استفاده کرد[Pan04,Han06, Ore08].

Commented [m3]: کلمه فارسی مناسب

۲-۱-۲- چالش‌های اصلی در فرآیندخزش وب

حجم روزافزون و سرعت بالای تولید و تغییر اسناد بر روی وب، ویژگی‌های مهم دنیای وب هستندکه خزش آن را با مشکل روبرو می‌کند.با توجه به حجم بسیار زیاد وب، یک خزندهتنها می‌تواند بخشی از اسنادوب را در یک زمان مشخص واکشی کند. از این رو، نیاز به اولویت بندیاسناد برای واکشی دارد. همچنین، زمانی که خزنده در حال واکشی آخرین سند یک سایت است، احتمال زیادی وجود دارد که اسناد جدید به آن سایت اضافه شوند و یا اسناد از قبل واکشی‌شده، دوباره بروز

---

[۱] Link Filter

[۲] Metadata

[۳] hash

شده و یا حتی حذف شوند. بنابرین یک خزنده باید به طور مداوم وب را خزش کند تا اسناد جدید و یا بروز شده را واکشی کند. این موارد یک خزنده با چندین سوال اساسی روبرو می‌کند:

- کدام اسناد برای خزش انتخاب شوند؟ اولویت واکشی اسناد بر اساس چه معیاری تعیین شود؟
- کدام اسناد باید بروز شوند؟ اولویت بروز رسانی اسناد بر اساس چه معیاری تعیین شود؟
- چگونه باید سربار ایجاد شده ناشی از فرآیند خزش را بر روی سایت‌ها به حداقل رساند؟
- چگونه فرآیند خزش را موازی‌سازی کرد؟

رفتاریک خزنده وب، نتیجه سیاست‌هایی است که برای پاسخ دهی به سوالات مطرح شده، در نظر می‌گیرد[Pan04,Dhe11].

۱-۲-۱-۲- سیاست انتخاب اسناد

با توجه به حجم روزافزون و تغییر پویای وب، امکان خزش تمام وب وجود ندارد؛بنابراین لازم است تا خزنده با اولویت‌بندی پیوندها در صف خزش تلاش کند تا اسناد با اهمیت‌تر را زودتر از سایر اسناد انتخاب و واکشی کند. روش‌ها و معیارهای انتخاب سند را می‌توان به دو دسته محبوبیت محور و علاقه محور تقسیم کرد:

۱) محبوبیت محور: در این معیارها، اهمیت سند وابسته به میزان محبوبیت آن است. برای تعریف محبوبیت می‌توان از تعداد پیوندهای ورودی سند و یا معیار PageRank استفاده کرد.
۲) علاقه محور: در این معیارها، هدف، جمع‌آوری اسناد مورد علاقه کاربر یا سیستم است. علاقه می‌تواند بر روی موضوع، دامنه و یا نوع سند تعریف شود. برای مثال ممکن است که خزنده تنها نیاز به واکشی اسناد در یک دامنه خاص مانند ".com" را داشته باشد؛ و یا علاقه کاربر به اسنادی که در مورد علم کامپیوتر هستند، نسبت به سایر اسناد بیشتر باشد[Pan04,Dhe11].

۲-۲-۱-۲- سیاست بروز رسانی

پس از اینکه که تعداد اسنادواکشی شده توسط خزنده به تعداد مشخصی رسید، خزنده باید شروع به ملاقات دوباره اسنادواکشیشده کند تا تغییرات جدید را کشف کرده و مخزن دادههایش را بروز نگه دارد. معمولا از فرادادههایی مانند آخرین زمان ملاقات ، فیلد تغییر محتوا (last-modified-since) در سرآیند پاسخ درخواست واکشی و یا هش محتوای سند، برای تصمیم گیری در مورد بروز رسانی سند استفاده میشود[Han06,Ore08, Dhe11].

اما از آنجایی که اسناد وب با نرخهای بسیار متفاوتی تغییر میکنند، خزنده باید به دقت در مورد اینکه کدام اسناد دوباره ملاقات شوند، تصمیمگیری کند. برای مثال اولویت ملاقات دوباره اسنادی که به ندرت تغییر میکند، باید نسبت به اسنادیکه به صورت مکرر بروز میشود، پایین تر باشد و یا اسنادی که از نظر خزنده اهمیت بیشتری دارند، میتوانند بیشتر مورد خزش قرار بگیرند. [Pan04,Dhe11,Din06]

۳-۲-۱-۲- سیاست موازی سازی

به دلیل اندازه وسیع وب، خزندهها اغلب بر روی چندین ماشین اجرا میشوند و اسناد را به صورت موازی و چند نخی[۱] از سطح وبواکشی میکنند. این کار به افزایش توان عملیاتی خزنده کمک می کند. اما واضح است که این خزندههای موازی باید به درستی با هم هماهنگ شوند تا یک سند یکسان چندین بار توسط خزندههای مختلف واکشی نشود.از سوی دیگر، این هماهنگی خود موجب سربار ارتباطی و محدودیت در تعداد خزندههای موازی میشود.بنابراین، خزندهها باید تعداد نخها و یا در حالت توزیع شده، تعداد ماشینها را بر اساس محدودیتهای موجودتعیین کنند. [Pan04, Dhe2011]

---

[۱] Multi thread

۴-۲-۱-۲- سیاست مودبانهبودن[1]

یک خزنده باید مجموعه‌ای از سیاست‌ها را در زمان واکشی سایت‌ها رعایت کند تا از سربار تحمیلی ناشی از خزش بر روی آنها جلوگیری شود. برای مثال، با استفاده از فایل robot.txt، یک سایت می‌تواند امکان واکشی بخشی یا همه اطلاعات مربوط به خودش را از خزنده سلب کند. بنابراین قبل از واکشی سایت، خزندهباید ابتدا فایلrobot.txt مربوط به سایت را بررسی کند تا در صورت عدم اجازه از سوی سایت میزبان، آن سایت را مورد خزش قرار ندهد[Pan04, Dhe2011].

همچنین، خزنده باید یک تاخیر زمانی را بین ارسال هر دو درخواست متوالی به یک سایت در نظر بگیرد. به صورت پیش فرض به ازای هر سایت، خزنده تنها می‌تواند در هرثانیه یک سند از آن را واکشی کند. سایت‌ها نیز می‌توانند این مقدار را با استفاده از فیلدهای Crawl-Delay و Request-rate در فایل robot.txt تنظیم کنند[Cyg11]. از سوی دیگر، با مبتنی بر PLD کردن صف ها و توزیع نخ-های اجرایی بین به صف ها نیز می‌توان به رعایت عدالت بین PLD ها و کاهش سربار بر روی یک PLD کمک کرد[Hog11].

Commented [m4]: کمی توضیحات زیاد بود. ده صفحه برای مقدمات زیاد است.

۲-۲- خزش وب معنایی

یک سندوب معنایی، نوع خاصی ازاسناد وب است که در آن از مدل داده RDF برای انتشار داده‌ها استفاده می‌شود.به داده‌های موجود در اسناد وب معنایی، داده‌های معنایی گفته می‌شود [Din06].

برای جمع‌آوری داده‌های معنایی از سطح وب، روش‌های مختلفی وجود دارد. معمولا سایت‌های

[1] Politness Policy

با حجم داده بالا مانند dbpedia[1] و geonames[2] تمام داده‌های معنایی خود را به صورت فایل‌هایرونوشت[3] منتشر می‌کنند. برخی از سایت‌ها نیز با ارائه Sparql Endpoint، به کابران یا عامل‌های نرم‌افزاری این امکان را می‌دهند تابا ارسال پرس و جو، به داده‌های معنایی آنها دسترسی پیدا کنند[Ore08,Cyg11].

اما با توجه به عدم ارائه این دو امکان توسط بسیاری از سایت‌ها، لازم است تا با استفاده از یک خزنده وب، اسناد وب معنایی متعلق به این سایت‌ها را کشف و داده‌های معنایی موجود در آنها را بازیابی کرد. در این راستا، به خزنده‌ای که به دنبال جمع آوری اسناد وب معناییاز سطح وب باشد، خزنده وب معنایی گفته می‌شود.البته در بیشتر موارد، پردازش فایل رونوشت و همچنین ارسال پرس‌و جو به Sparql Endpointها نیز به خزنده‌وب معنایی واگذار می‌شود[Las12,Cyg11].

تاکنون، خزنده‌های مختلفی برای خزش وب معناییپیاده‌سازی شده‌اند که بیشتر آنها متعلق به موتورهای جستجوی معنایی [Din06,Bat12,Hog11,Che09,Sab07,Ore08]هستند. چند خزنده متن‌باز نیز در این حوزه وجود دارد [Dod06,Ise10,Las12].

در ادامه این بخش، بهجزئیات بیشتری در مورد اسنادوب معنایی،روش واکشی و تجزیه این اسناد، چگونگی پیوند میان آنها، سیاست برخورد با اسناد غیر معنایی، روش‌های جمع‌آوری seed و مدل‌های موازی‌سازی در خزنده‌هایوب معناییمی‌پردازیم.

۱-۲-۲- اسناد وب معنایی

در وب سنتی، داده‌ها در قالب مستنداتی که برای انسان قابل فهم هستند ارائه می‌شوند، اما

۱

۲

dump

هدف اصلی در وب معنایی داشتن وبی از داده‌ها است که این داده‌ها برای ماشین نیز قابل فهم باشد.در این راستا، برای توصیف اشیا (منابع) و روابط میان آنها از مدل داده RDF استفاده می‌شود.عناصر اصلی در مدل دادهRDF، سه گانه‌هاییبه صورت (شی، صفت، مقدار) هستند که با استفاده از URIها نام‌گذاری می‌شوند[Ber01].

از مدل داده RDF، فرمت (دستور زبان) های مختلفی ارائه شده استکه رایجترین آنها عبارتند از:RDF/XML،Turtle،Notation3 (N3)،N-Triples و N-Quads. به سند وبی که به طور کامل با یکی از دستور زبان های وب معنایی نوشته شده باشد، سند وب معنایی گفته می-شود[Din06,Cyg11]. در شکل ۲-۱ یک سند وب معنایی و در شکل ۲-۲ گراف RDF معادل آن نمایش داده شده است کهدرآن دانشکده مهندسی دانشگاه فردوسی مشهد با استفاده از صفات owl:sameAs، rdfs:label و foaf:homepage توصیف شده است.

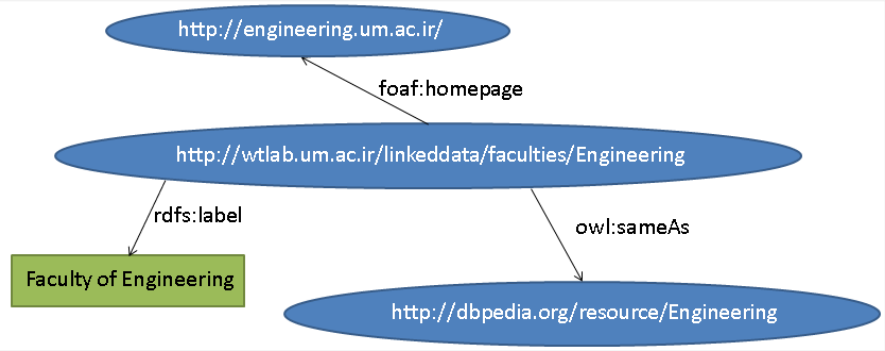
از سوی دیگر، اسنادی هم وجود دارند که داده‌های معنایی، در محتوای متنی آنها جاسازی شده است. برای مثال، در یک سند HTML، می‌توان توصیف بیشتری از متن یک پاراگراف برای ماشین ارائه داد. برای این منظور از دستور زبان‌هایی مانندRDFa[1]استفاده می‌شود. به چنین اسنادی، اسناد وب معنایی جاسازی شده گفته می‌شود [Din06,Cyg11].

[1] Resource Description Framework in Attributes

```
<?xml version="1.0"?>
<rdf:RDF
        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
        xmlns:foaf="http://xmlns.com/foaf/0.1/"
        xmlns:owl="http://www.w3.org/2002/07/owl#"
        xmlns:dbpprop="http://dbpedia.org/property#"

        <rdf:Description rdf:about="http://wtlab.um.ac.ir/linkeddata/faculties/Engineering">
           <rdfs:label xml:lang="en">Faculty of Engineering</rdfs:label>
           <owl:sameAs rdf:resource="http://dbpedia.org/resource/Engineering"/>
           <foaf:homepage>http://engineering.um.ac.ir/</foaf:homepage>
        </rdf:Description>
</rdf:RDF>
```

شکل (۱-۲): یک نمونه سند وب معنایی که با دستور زبان RDF/XML نوشته شده است.



شکل (۲-۲): یک نمونه گراف RDF

۲-۲-۲- سیاست انتخاب اسناد در خزنده های وب معنایی

۱-۲-۲-۲- واکشی اسناد وب معنایی

یک خزنده وب معنایی باید تلاش کند تا در طول فرآیند خزش، تنها اسناد وب معنایی را مورد خزش قرار دهد و از واکشی اسناد غیر معنایی صرف‌نظر کند. بنابراین تشخیص صحیح نوع محتوای اسناد، تاثیری زیادی در میزان دستیابی خزنده به اسناد وب معنایی و کاهش حجم واکشی اسناد غیر معنایی دارد.

خزنده‌های وب معناییپس از ارسال درخواست به سایت میزبان برای واکشی یک سند، از فیلد

نوع محتوا در سرآیند پاسخ، برای تشخیص معنایی بودن سنداستفاده می کنند (صافی واکشی). برای مثال، اسناد با نوع محتوای "application/rdf+xml" و یا "text/n3" یک سند وب معنایی هستند[]. اما مقدار فیلد نوع محتوا در سرآیند پاسخممکن است مشخص نشده باشد و یا مقدار آن با نوع محتوای واقعی سند مطابقت نداشته باشد [Umb08]. برای مثال، در بعضی از موارد با وجود اینکه سند، یک سند وب معنایی است اما مقدار فیلد نوع محتوای آن برابر با "text/html" و یا "application/xml" است و بنابراین خزنده به اشتباه سند را از نوع HTML و یا XML[1]در نظر می-گیرد.

با این وجود، بیشتر خزنده‌های وب معنایی، به فیلد نوع محتوا در سرآیند پاسخ در خواست واکشی اعتماد کرده و مسئولیت اشتباه بودن آنها را به منتشر کنندگان اسناد وب معنایی واگذار می-کنند. اما برای حل این مشکل، خزنده MultiCrawlerدر مرحله واکشی، از پسوند فایل موجود در آدرس اسناد نیز مانند ".rdf" و یا ".owl" برای تشخیص معنایی بودن یک سند استفاده می-کند[Har06]. اما پسوند فایل در آدرس بسیاری از اسناد مشخص نیست. همچنین، اسنادی با انواع محتوای مختلف وجود دارند که از پسوند فایل مشترک استفاده می‌کنند. برای مثال، از پسوند فایل ".xml" علاوه بر فایل‌های XML، برای فایل‌های RDF و RSS[2]نیز استفاده می‌شود؛ این در حالی استکه خزنده فقط باید اسناد RDF را به عنوان اسناد وب معنایی در نظر بگیرد. در مورد پسوند فایل-های فشرده مانند ".zip"، ".bz2" و غیره نیز این مشکل وجود دارد.بنابراین در صورت متکی بودن به فیلد نوع محتوا و پسوند فایل برای تشخیص وب معنایی بودن یک سند، خزنده ممکن است امکان دستیابی به بعضی از اسنادوب معنایی را پیدا نکند و یا بعضی از اسناد غیر معنایی را به عنوان اسناد

[1] Extensible Markup Language

[2] Rich Site Summary

وب معنایی در نظر بگیرد.

۲-۲-۲-۲- سیاست برخورد با اسناد غیر معنایی

به طور کلی می‌توان گفت که خزنده‌های وب معنایی با کمک صافی واکشی و صافی پیوند از واکشی اسناد غیر معنایی مانند عکس، ویدئو، PDFو غیره صرف‌نظر کرده و تنها اسناد وب معنایی را واکشی می‌کنند. اما شیوه برخورد این خزنده‌ها با اسناد HTML متفاوت است.اسناد وب معنایی به صورت پراکنده در سطح وب قرار دارند و برخلاف اسناد HTML، اتصال و پیوند کافی بین آنها وجود ندارد. بنابراین در صورت صرف نظر کردن از اسناد HTML، خزنده نمی‌تواند به حجم بالایی از اسناد-وب معنایی دست پیدا کند[Sab07,Hae06,Ore08,Bat12,Las12].

بنابراین، چگونگی برخورد یک خزنده وب معنایی با اسناد غیرمعنایی به ویژه اسناد HTML، تاثیر زیادی بر عملکرد خزنده و میزان دستیابی آن به اسناد وب معنایی دارد. در این راستا، خزنده‌های وب معنایی موجود را می‌توان به سه دسته تقسیم کرد.

خزنده‌های دسته اول، در طول فرآیند خزش همانند سایر اسناد غیرمعنایی ، اسناد HTML را نیز خزش نمی‌کنند. برای این منظور، در مرحله استخراج پیوند، پیوندهایی با پسوند فایل"html."در صف خزش قرار نمی‌گیرند. اما به دلیل مشخص نبودن پسوند فایل در آدرس بسیاری از اسناد HTML، این اسناد به مرحله واکشی می‌رسند. در این مرحله خزنده ابتدا فیلد نوع محتوا در سرآیند پاسخ درخواست واکشی را بررسی می‌کند و در صورتی که مقدار آن برای مثال برابر با "text/html" باشد، از واکشی محتوای آن سند صرف‌نظر می‌کند. در این حالت، خزنده می‌تواند به شیوه‌ی دیگری نیز عمل کند و اجازه واکشی به اسناد با نوع محتوای "text/html" را بدهد اما پس از استخراج پیوندهای داخل آنها، تنها پیوندهایی را انتخاب کند که پسوند فایل آنها از نوع معنایی است (برای مثال ".rdf"، ".owl" و غیره) [Ore08,Che09,Dod06,Ise10].

اما با توجه به عدم اتصال کافی بین اسناد وب معنایی و همچنین قرار گرفتن آنها در بین اسناد HTML، خزنده‌های وب معنایی دسته دوم در صورت برخورد با پیوندهای HTML، آنها را نیز مورد خزش قرار می‌دهند. به بیان بهتر، خزنده علاوه بر اسناد وب معنایی، محتوای اسناد HTML را نیز واکشی می‌کند. سپس پیوندهای داخل آنها را استخراج کرده و در صف خزش قرار می‌دهد. تنها تفاوت این است که بر خلاف اسناد وب معنایی، محتوای اسناد HTML در مخزن ذخیره نمی-شود[Din06,Sab07,Las12,Cyg11].

اما با توجه به حجم بالای اسناد HTML و غیرمعنایی بودن بسیاری از پیوندهای داخل این اسناد، واکشی همه آنها و پیمودن مسیرهای ایجاد شده توسط آنها، تنها باعث اتلاف منابع و زمان می‌شود. در این‌راستا، خزنده‌های دسته سوم تلاش می‌کنند تا با اعمال سیاست‌های تنبیهی حجم واکشی اسناد غیر معنایی را کاهش دهند.

برای مثال، خزنده موتور جستجوی وب معنایی SWSE ، نه تنها اسناد HTML را مورد خزش قرار نمی دهد؛ بلکه اگر تعداد اسناد وب معنایی بدست‌آمده از یک دامنه (برای مثال یک سایت) نسبت به تعداد کل اسناد واکشی‌شده از آن، از یک مقدار مشخصی کمتر باشد، خزنده آن دامنه را جریمه کرده و تا یک مدت زمان معین آن را مورد خزش قرار نمی دهد [Hog11]. اما خزنده وب معنایی BioCrawlerبرای کاهش حجم واکشی اسناد غیر معنایی به گونه‌ای دیگر عمل می کند. این خزنده اسناد HTML را مورد خزش قرار می‌دهد اما در صورتی که در مسیر جاری تعداد اسناد غیر معنایی بدست آمده بسیار بیشتر از تعداد اسناد وب معناییبدست آمده باشد، خزنده در آن مسیر متوقف شده و از یک نقطه تصادفی دیگر شروع به خزش می‌کند [Bat12]. مشکل اصلی هر دو روش این است که در مسیر یا دامنه جاری ممکن است یک سند وب معنایی وجود داشته باشد و تنها به علت غیر معنایی بودن اسناد قبلی، خزنده امکان دسترسی به آن سند را پیدا نمی‌کند.

۳-۲-۲-۲- اولویت واکشی اسناد

Commented [m5]: اینها به صفحه بعد بروند تا با متنشان باشند

نکته دیگر در مورد سیاست انتخاب اسناد، روشتعیین اولویت اسناد برای واکشی است. در خزنده‌های وب معنایی موجود، روش اولویت بندی، بر روی URI و یا PLD تعریف می‌شود.

در صف‌های مبتنی بر PLD، معمولا انتخاب PLDها به صورت نوبت چرخشی (ترتیبی) است. در واقع خزنده، به ترتیب از هر صف PLD، یک پیوند(URI) را بر می‌دارد. این کار موجب رعایت عدالت میان PLDها می‌شود[Cyg11]. بعضی از خزنده‌ها نیز به هنگام برداشتن URI از صف هرPLD، معیارهایی را در نظر می‌گیرند و در واقع PLDها را اولویت‌بندی می‌کنند. از جمله این معیارها می‌توان به تعداد URIهای موجود در صف [Ise10] و یا میزان غنی بودن PLD از نظر معنایی [Bat07, Hog11] اشاره کرد. برای مثال هر چه از یک PLD تعداد اسناد وب معنایی بیشتری بدست آمده باشد، خزنده URI های بیشتری را از صف مربوط به آن PLD، انتخاب می‌کند.

در مورد شیوه اولویت‌دهی به URIها، خزنده‌ی موتورجستجویSindice، به URIهایمعرفی شده توسط کاربران، اولویت بالاتری برای خزش داده می‌دهد[Cyg11]. از سوی دیگر خزنده‌ی موتور جستجوی Swoogle، زمانی که می‌خواهد یک سند وب معنایی از قبل واکشی شده را مورد بروزرسانی قرار دهد، آن را با اولویت بالاتری در صف خزش قرار می‌دهد [Din06].

اما در روش‌های موجود، تمایزی بین پیوندهای داخل یک دامنه برای تعیین اولویت خزش آنها وجود ندارد. در حالی که ممکن است خزنده با خزش یک پیوند و پیمودن مسیرهای ایجاد شده توسط آن، به اسناد وب معنایی زیادی دست پیدا کند و یا در مقابل، با اسناد غیر معنایی زیادی روبرو شود.

۳-۲-۲- تجزیه اسناد وب معنایی

پس از واکشی یک سند وب معنایی، لازم است است تا سند تجزیه شده و داده‌های معنایی یا همان سه‌گانه های داخل آن استخراج شود. در حال حاضر، پارسرهای مختلفی برای تجزیه اسناد وب

معنایی وجود دارند که از جمله آنها می‌توان به پارسر Jena[۱] Any23[۲]و Nxparser[۳]اشاره کرد. هر یک از این پارسرها از بعضی از فرمت‌های اسناد وب معنایی پشتیبانی می‌کنند.

در بیشتر خزنده های وب معنایی، سه گانه های استخراج شده از هر سند با استفاده از پارسربه فرمت N-Triples(سه گانه) و یاN-Quads(چهارگانه) تبدیل می‌شوند تا از آنها در کاربردهای مختلف استفاده شود. فرمت N-Quads مشابه به فرمت N-Triples است با این تفاوت که هر سه‌گانه‌به همراه URI سندی که از آن استخراج شده است، در قالب یک چهارگانه ذخیره می‌شود.

۴-۲-۲- شیوه پیوند اسناد در وب معنایی

همانطور که در بخش ۲-۲-۱ بیان شد، در اسناد وب معنایی برای نام‌گذاری هر اشیا و روابط بین آنها از URI استفاده می‌شود. بنابراین یک خزنده وب معنایی، با دنبال کردن URIهای موجود در یک سند،می تواند فرآیند خزش را ادامه می‌دهد [Din06,Bat12,Hog11,Che09,Ise10,Las12].

اما همه URIهای داخل یک سند وب معنایی اشاره به یک سند وب نمی‌کنند. به عبارت دیگر، از یک URIمی‌توان تنها برای نام‌گذاری یک شی نیز استفاده کردو به ازای آن، سندی بر روی وب تعریف نکرد.بر این اساس، در بعضی از خزنده‌های وب معناییتنها URIهایی مورد خزش قرار می‌گیرند که با احتمال بیشتری می‌توانند آدرس یک سند وب باشند. از جمله این URIها می‌توان به مقادیر صفاتowl:sameas،rdfs:seealso، rdfs:isDefinedBy و owl:import اشاره کرد[Hae06,Din06,Dod06,Sab07].اما در این روش، با توجه به عدم اتصال کافی بین اسناد وب معنایی خزنده نمی‌تواند به حجم قابل قبولی از اسناد وب معنایی دست پیدا کند.

۱http://www.apache.org/dist/jena/binaries/

۲http://any23.org/

۳http://sw.deri.org/2006/08/nxparser/

۵-۲-۲- روش‌های جمع‌آوری Seed در خزنده‌های وب معنایی

بعضی از خزنده‌هایوب معناییبرای حل مشکل عدم اتصال کافی بین اسناد وب معناییو پراکنده بودن آنها در سطح وب، سعی می‌کنند تا در کنار خزش اسناد HTML، seedهای مناسبی را نیز برای خود جمع‌آوری کنند. در ادامه به بررسی روش‌های جمع‌آوری seed در خزنده‌های وب معنایی می-پردازیم.

۱-۵-۲-۲- دریافت Seed از کاربران

یکی از رایج‌ترین روش‌ها در موتورهای جستجوی معنایی، طراحی یک فرم ورودURI است که از طریق آن، کاربران،آدرس اسنادی را که شامل داده‌های معنایی هستند، به خزنده معرفی می‌کنند. آدرس وارد شده توسط کاربر، علاوه بر آدرس یک سند، می‌تواند آدرسیکفهرست وب و یا آدرس مربوط به یک نقشه سایت[Cyg11] نیز باشد. آدرس نقشه سایت، ارزش زیادی دارد؛ به این دلیل که خزنده با کمک آن می‌تواند به صورت قابل اعتماد و کامل به URIهای یک سایت دسترسی داشته باشد.علاوه براین، خزنده موتور جستجوی Sindice[Ore08, Cyg11] امکانی به نام Ping API را در اختیار کاربرانش قرار داده است. با استفاده از این API، کاربران می‌توانند اعلان‌های خودکاری را برای این موتور جستجو ارسال کنند؛ تا بدین ترتیب اسناد مربوط به سایت آنها توسط خزنده Sindice مورد خزشقرار بگیرد.

دریافت seed از کاربران، هر چند که به فرآیند جمع‌آوری seed کمک می‌کند اما هیچ تضمینی وجود ندارد که همه کاربران، با دادن آدرس سایت خود، از یک موتور جستجو بخواهند تا سایت آنها را مورد خزش قرار دهد. به همین دلیل معمولا از این روش در کنار روش‌های دیگر استفاده می‌شود.

Submission Form

sitemap

۲-۵-۲-۲- استفاده از خروجی بعضی از موتورهای جستجو

بعضی از خزنده‌های وب معنایی، از خروجی بعضی از موتورهای جستجو برای جمع‌آوری seedاستفاده می‌کنند. برای مثال با ارسال چند پرس‌وجو[۱] به موتور جستجوی گوگل[۲] اسناد با نوع rdf مورد بازیابی قرار می‌گیرند. سپس نتایج حاصل از اینپرس‌وجوها که فهرستی ازآدرس اسناداست، به عنوان seed به خزنده وب معناییداده می‌شود [Din06,Sab07]. به این نوع روش جمع‌آوری seed، در اصطلاح فراجستجو[۳] به خزنده‌هایی که با ارسال پرس‌وجو به موتورهای جستجو، اسناد و داده‌های موردنظر را بدست می‌آورند، فراخزنده[۴] گفته می‌شود.

اما در این روشچون بر مبنای یک سری پرس وجو عمل می‌شود، نمی‌توان همه اسناد معنایی را به خوبی تحت پوشش قرار داد. از سوی دیگر تضمینی هم وجود ندارد که موتورجستجوی مورد نظر، همه اسناد معنایی را در مخزن خود نگهداری کرده باشد. مسئله مهم‌تر نیز این است که در صورت استفاده از موتورهای جستجوی وب سنتی مانند گوگل،اولویت خزش و بروز رسانی اسناد در این موتورها،بر اساس محبوبیت اسناد است؛ این در حالی است که در خزش وب معنایی، علاوه بر محبوبیت، هر چه یک سند از نظر معنایی غنی تر باشد، باید بیشتر مورد خزش قرار بگیرد.

بنابراین هر چند که این روش تا حدی به خودکارسازی فرآیند جمع‌آوری seed کمک می‌کند اما با توجه به محدودیت‌های بیان شده، باید در کنار آن از روش‌های دیگر نیز استفاده کرد.

۲-۵-۲-۳- استفاده از گراف خزش

---

[۱] query

[۲] google

[۳] MetaSearch

[۴] MetaCrawler

بعضی از خزنده‌ها، با به کارگیری روش‌های مبتنی بر گراف، به جمع‌آوری seed می‌پردازند. در این شیوه، با اعمال الگوریتم‌هایی بر روی گراف خزش، سایت‌هایی به عنوان seed، انتخاب می‌شوند. برای مثال،در خزنده موتور جستجوی معنایی Swoogle[Din06]، بخشی با نام InductiveLearner وجود دارد. در این بخش بر اساس گراف خزش بدست‌آمده و با در نظر گرفتن معیارهایی همچون میزان دقت[۱] فراخوانی[۲] سایت‌هایی که به تعداد زیادی از اسناد معنایی پیوند می‌دهند و یا با شروع از آنها می‌توان به تعداد زیادی از اسناد معنایی رسید، به عنوان seed برای دور بعدی خزش انتخاب می‌شوند.

استفاده از این روش در فرآیند بروز رسانی اسناد خزش شده و مخزن داده‌ها، می‌تواند بسیار موثر باشد؛ چرا که در این حالت، با انتخاب seedهای مناسب، می‌توان با سرعت بیشتری به تعداد زیادی از سایت‌هایی معنایی از قبل بازیابی شده دست پیدا کرد و آنها را مورد خزش مجدد و بروز رسانی قرار داد. اما در صورتی که از این روش برای انتخاب seed در دور اول خزش استفاده شود، باید از مخازن خزش موجود استفاده کرد که در این صورت، میزان جامعیت این مخازن و بروز بودن آنها، محدودیت‌هایی را ایجاد خواهد کرد.

۴-۵-۲-۲- خزش محدود بعضی از سایت ها

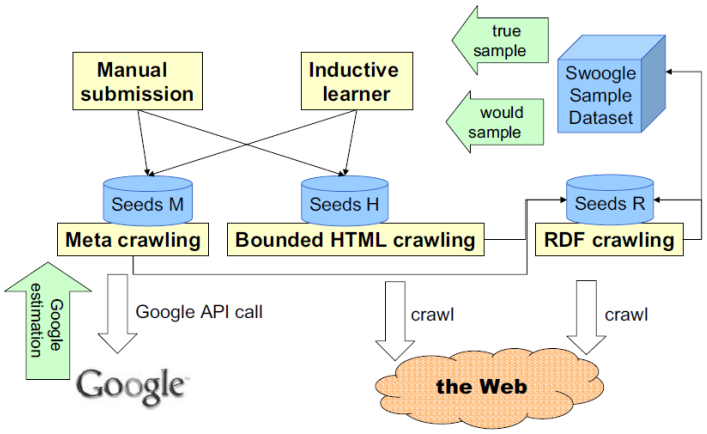
موتور جستجوی معنایی Swoogle[Din06] در چارچوب خزش خود علاوه بر یک خزنده RDF، از یک خزنده HTML نیز استفاده می‌کند. منظور از خزنده RDF، خزنده جمع‌آوری کننده اسناد وب معنایی است. در این روش، سایت‌هایی که علاوه بر اسناد HTML شامل تعداد مناسبی اسناد وب معناییبنیز هستند، شناسایی شده وآدرس آنها به عنوان Seed، به خزندهHTML داده

۱precision

۲recall

می‌شود تا با خزش همه اسناد آن سایت، به اسناد وب معناییآن دست پیدا کند.پس از تجزیه اسناد وب معنایی بدست آمده و استخراج پیوندهای داخل آنها، این پیوندها به عنوان Seed در اختیار خزنده RDF قرار داده می‌شود.اما در خزنده HTML، با اعمال محدودیت‌هایی بر روی آدرس پیوند، دامنه پیوند، عمق خزش، تعداد اسناد بازیابی شده و درصد اسناد وب معنایی بدست‌آمده، سعی می‌شود تا به نحوی فضای خزشمحدود شود. اما ممکن است در فضای محدود شده تعداد اسناد وب معنایی کمی وجود داشته باشد و در مقابل در خارج از این فضا اسناد وب معنایی نیز وجود داشته باشد.

در شکل ۲-۳، چارچوب خزش مورد استفاده در موتور جستجوی Swoogle نشان داده شده است.در این چارچوب به منظور تقویت فرآیند جمع‌آوری Seed برای خزنده RDF، از هر چهار روش معرفی‌شده، استفاده می‌شود.



شکل (۳-۲): چارچوب خزش ترکیبی در موتور جستجوی Swoogle

۲-۲-۶- سیاست موازی سازی در خزنده های وب معنایی

۲-۲-۶-۱- چارچوب توزیع شده

یکی از رایج‌ترین شیوه‌های موازی سازی فرآیند خزش، استفاده از نخ های اجرایی

است[Pan04]. اما به ازای تعداد مشخصی نخ، سرانجام خزنده با محدودیت‌های CPUوI/Oروبرو می‌شود.از این رو، خزنده‌های موتور جستجوی معنایی Sewer[Las12]و SWSE[Hog11] از چارچوب توزیع شده نگاشت-کاهش برای افزایش مقیاس پذیری و توان عملیاتی خود استفاده می کنند. برای مثال، فرآیند خزش توزیع شدهدر خزنده‌ی موتور جستجوی SWSE، که بر اساس معماری Master/Slaveپیاده سازی شده است، به صورت زیر است:

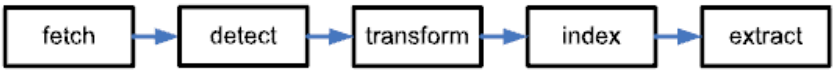
۱. ماشینMaster،آدرس‌های شروع (Seedها) را با استفاده از یک تابع توزیع مبتنی بر هش، بین ماشین‌های Slave، تقسیم می‌کند.
۲. هر ماشین Slave، آدرس‌های جدید را به صف اضافه می‌کند. سپس یک دور فرآیند خزش را اجرا می‌کند و محتوای اسناد واکشی شدهرا در مخزن ذخیره می‌کند. همچنین پیوندهای جدید را در داخل یک صف قرار می‌دهد.
۳. هر ماشین Slave با استفاده از تابع توزیع، URIهای جدید داخل صف را بین همتاهای خودتقسیم می‌کند.

مراحل ۲ و ۳ به صورت بازگشتی و به تعداد دورهای مورد نظر تکرار می‌شود.

۲-۶-۲-۲- مدل خط لوله

در طول فرآیند خزش اسناد وب معنایی، به دلیل متنوع بودن دستورزبان‌ها و نیاز به استفاده از پارسرهای مختلف، پردازش اسناد نسبت به زمان واکشی آن‌هامدت زمان بیشتری به طول می‌انجامد. به همین دلیل برای افزایش مقیاس‌پذیری و بالا رفتن توان عملیاتی، می‌توان بخش پردازش را از بخش واکشی جدا کرد و مراحل مختلف فرآیند خزش را به صورت موازی انجام داد. به این مدل پیاده‌سازی، مدل خط لوله گفته می‌شود که در خزنده‌های وب معنایی[Har06]MultiCrawlerو Sindice[Ore08]از آن استفاده می‌شود.

مدل خط‌لوله، شامل پنج مولفه است که در شکل ۲-۳ نشان داده شده است. هر مولفه از خط-لوله دارای یک صف از URIها است وهمچنین مسئول اجرای یک بخش از فرآیند خزشمی‌باشد. به طور کلی، در طول فرآیند خزش، هر مولفه به صورت جداگانه مراحل زیر را تکرار می‌کند:



شکل (۲-۳): مدل خط لوله پنج مرحله ای [Har06]

- یک URIرا از صف خود برداشته و یک نخ اجرایی به آن اختصاص می‌دهد.
- نخ، کارهای لازم را بر روی URI انجام می‌دهد.
- URI در صف مولفه بعدی قرار داده می‌شود.

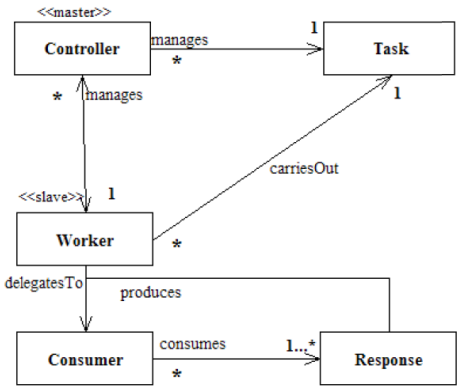
داده‌های معنایی بدست آمده از فرآیند خزشدر یک مخزن ذخیره می‌شوند که این مخزن بر روی یک ماشین جداگانه قرار دارد.به منظور بالا بردن توان عملیاتی می‌توان مولفه‌های خط لوله را به صورت توزیع شده بر روی چند ماشین اجرا کرد و یا چند خط لوله را به صورت موازی با هم پیاده‌سازی کرد.

۳-۶-۲-۲- مدل ترکیبی کنترل‌کننده – کارگر و تولیدکننده-مصرف‌کننده

روش دیگر برایجدا کردن مرحله واکشی اسناد از مرحله پردازش آنها، استفاده از دو الگوی کنترل‌کننده-کارگر و تولید کننده-مصرف کننده است که در خزنده وب معناییSlug[Dod06]پیشنهاد داده شده است. معماری کلی این خزنده (شامل مولفه‌های اصلی و روابط میان آنها) به صورت نمودار UML در شکل ۲-۴ نشان داده شده است.

بخش کنترل‌کننده – کارگر، مسئول واکشی اسناد از وب است. در واقع مولفه کنترل کننده، ابتدا چند کارگر (نخ اجرایی) را ایجاد نموده و سپس به هر کدام از آنها،وظیفه واکشی چند سند از وب را واگذار می‌کند. این وظایف در صف کنترل کننده قرار دارد. بنابراین، الگوی کنترل کننده –کارگر

مشابه به الگوی طراحی Master-Slaveاست با این تفاوت که در اینجا کنترل کننده، خودش نتایج بدست آمده از هر وظیفه را جمع نمی‌کند و کارگرها هم مسئول انجام این کار نیستند.



شکل (۲-۴): معماریخزنده[Dod2006]SLUG

برای پردازش نتایج بدست آمده از واکشی اسناد، از الگوی تولیدکننده-مصرف‌کننده استفاده می‌شود. هر کارگر ابتدا، نتایج حاصل از واکشی یک سند (پاسخ‌ها) را تولید می‌کند و سپس یک یا چند مصرف‌کننده (نخ اجرایی) را تعیین می‌کند تا نتایج بدست آمده را مدیریت و پردازش کنند. مصرف‌کننده، پاسخ‌ها یا همان اسناد وب معنایی بدست آمده را تجزیه‌کرده و با استخراج پیوندهای آنها وظایف جدیدی را تولید می‌کند. پیوندهای استخراج شده پس از عبور از صافی‌های موجود در کنترل کننده، به صف وظایف اضافه می‌شوند.

۴-۶-۲-۲- محیط چند عامله

خزنده وب معنایی BioCrawler[Bat12]، یک چارچوب خزش جدید را بر اساس دیدگاه محیط‌های چندعامله ارائه داده است. در واقع BioCrawler، مجموعه‌ای از خزنده‌های (عامل های) هوشمند است که به صورت توزیع شده و مستقل بر روی وب، فرآیند خزشرا انجام می‌دهند. مدل دانش این خزنده‌ها به صورت مجموعه‌ای از قوانین است. این قوانین که مسیرهای دستیابی به

سایت‌های معنایی را نشان می‌دهد، در طول فرآیند خزش، توسط خزنده‌ها ایجاد و بهینه می‌شود. در واقع خزنده‌ها با استفاده از این قوانین، مسیر های دستیابی به اسناد وب معنایی را یاد می گیرند تا آنها را نسبت به سایر اسناد بیشتر مورد بروز رسانی قرار دهند.

در این چارچوب، بخشی برای مدیریت قوانین وجود دارد. هر خزنده، قوانین بدست آمده را به بخش مدیریت می‌دهد. در هنگام انتخاب PLD، خزنده از بخش مدیریت درخواست می‌کند تا بهترین قانون را بر اساس بردار دید خزنده، به آن معرفی کند. بردار دید هر خزنده، شامل مجموعه دامنه‌هایی است که خزنده می‌تواند اسناد داخل آنها را واکشی کند. بنابراین، خزنده با کمک مسیرهلی از قبل شناسایی شده (قوانین) بهترین دامنه را انتخاب کرده و اسناد مربوط به آن را خزش می کند. برای پیاده‌سازی BioCrawler از زیر ساختJADE[1] استفاده شده است.

۷-۲-۲- خلاصه کارهای گذشته

یک خزند وب‌معنایی در طول فرآیند خزش با چالش‌های خاص خود روبرو است که بعضی از آنها عبارتند از:

۱- عدم امکان دستیابی به بسیاری از اسناد وب‌معنایی در صورت صرف‌نظر کردن از اسناد HTML در طول فرآیند خزش

۲- حجم بالای اسناد HTML و غیرمعنایی بودن بسیاری از پیوندهای داخل این اسناد

۳- امکان تشخیص نادرست اسناد وب‌معنایی در صورت متکی بودن به فیلد نوع محتوا و پسوند فایل

۴- اختلاف زمانی زیاد بین مرحله واکشی و پردازش سند به دلیل تنوع دستور زبان در اسناد وب معنایی و همچنین حجم بالای سه‌گانه‌ها (داده‌های معنایی) در بعضی از اسناد

[1] Java Agent Development Framework

۵- عدم دستیابی کافی به اسناد وب‌معنایی در صورت متکی بودن به مقدار بعضی از صفات خاص (مانند rdfs:seealso) در مرحله استخراج پیوند

۶- نیاز به اولویت بندی پیوندها بر مبنای میزان توانایی آنها در هدایت خزنده به سمت اسناد وب‌معنایی به منظور دستیابی سریعتر به اسناد وب معنایی

خزنده‌های وب معنایی، برای حل این چالش‌ها از سیاست‌های مختلفی استفاده می‌کنند که در بخش‌های قبلی به آن پرداخته شد. جدول۲-۱، خلاصه‌ای از ویژگی‌هایی خزنده‌هایبررسی شدهدر حوزه وب معنایی را نشان می دهد.

جدول(۱-۲):خلاصه کارهای گذشته درحوزه خزش وب معنایی

مرجع	نام خزنده	محیط اجرایی	انتخاب Seed	اسناد HTML	تشخیص معنایی بودن سند	انتخاب پیوند از اسناد وب معنایی	نوع صف	مدل موازی سازی	پارسر
[Hae06]	MultiCrawler		کاربر	خزش	پسوند فایل فیلد نوع محتوا	مقادیر صفات خاص	مبتنی بر URI	خط لوله توزیع شده	
[Hog11]	SWSE	موتور جستجو	کاربر	عدم خزش سیاست تنبیهی	فیلد نوع محتوا	همه URIها	مبتنی بر PLD	خط لوله توزیع شده	Jena
[Ore08, Cyg11]	Sindice	موتور جستجو	کاربر	خزش	فیلد نوع محتوا	همه URIها	مبتنی بر PLD	خط لوله توزیع شده	Any23
[Din06]	Swoogle	موتور جستجو	هر چهار روش	خزش	فیلد نوع محتوا	مقادیر صفات خاص	مبتنی بر URI		Jena
[Bat09,Bat12]	BioCralwer	موتور جستجو	کاربر- فراجستجو	خزش سیاست تنبیهی		همه URIها	مبتنی بر PLD	محیط چند عامله	Jena
[Ise10]	LDSpider	منبع‌باز	کاربر	عدم خزش	فیلد نوع محتوا	همه URIها	مبتنی بر PLD	چند نخی	Any23 NX
[Dod06]	Slug	منبع‌باز	کاربر	عدم خزش		مقادیر صفات خاص	مبتنی بر URI	تولید-مصرف، کنترل-کارگر	Jena
[Las12]	Sewer	منبع‌باز	کاربر	خزش		همه URIها	مبتنی بر URI	توزیع شده	
[Che09]	Falcons	موتور جستجو	کاربر- فراجستجو	عدم خزش	فیلد نوع محتوا	همه URIها	مبتنی بر URI	چند نخی	Jena
[Sab07]	Watson	موتور جستجو	کاربر- فراجستجو	خزش		مقادیر صفات خاص	مبتنی بر URI		Jena

۳-۲- خزش متمرکز وب

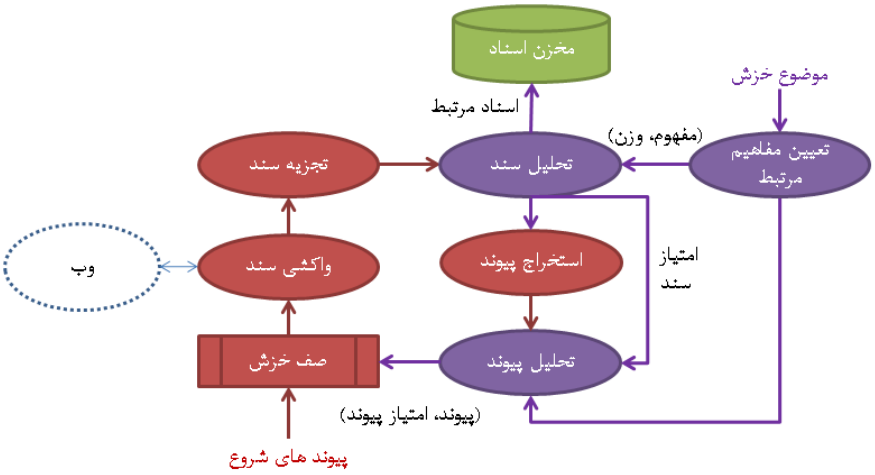
رشد روزافزون وب و تغییر سریع محتوای آن باعث شده است تا خزنده‌های وب، قادر به تحت پوشش قرار دادن تمام مقیاس وب نباشند. از سوی دیگر بسیاری از کاربران وب، به دنبال اسناد با کیفیت و مرتبط با موضوع موردنظر خود هستند؛ این در حالی است که پرس وجو هایمبتنی بر کلمات کلیدی در موتورهای جستجو، برای پاسخ‌دهی به این نیاز کاربران، کارایی لازم را ندارد و کاربر خودش مجبور است تا از میان نتایج بدست‌آمده، مرتبط‌ترین آنها را انتخاب کند. همچنین، در بیشتر خزنده‌ها، اولویت بروز رسانی اسناد، بر مبنای محبوبیت آنها است و به میزان ارتباط اسناد با موضوع موردنظر توجهی نمی‌شود.

با توجه به مشکلات و محدودیت‌های بیان شده، لازم است تا جستجوی کاربران و خزش اسناد وب، توسط ابزارهای هوشمندی پشتیبانی شود. یکی از این ابزارها، خزش متمرکز وب می‌باشد.

یک خزنده متمرکز، به دنبال جمع‌آوری اسناد در مورد یک یا چند موضوع از پیش تعریف شدهاست.در این راستا، خزنده متمرکز تلاش می‌کند تا با اولویت‌بندی پیوندهابر مبنای میزان ارتباط آنها با موضوع موردنظر و عدم واکشی اسناد نامرتبط، ضمن صرفه‌جویی در مصرف منابع و زمان، دستیابی سریع‌تری به اسناد مرتبط با موضوع داشته باشد[Bat09,Cha99].

شکل ۲-۵ یک معماری کلی را برای خزنده‌های متمرکز نشان می‌دهد. ابتدا موضوعاتموردنظر بایدتوسط مجموعه‌ای از مفاهیم برای خزنده تعریف شود که به آن مجموعه مفاهیم مرتبط گفته می-شود. سپس خزنده در طول فرآیند خزش، به ازای هر سند واکشی شده،مفاهیم موجود در محتوای سند را با مفاهیم مرتبط مقایسه می‌کند و بر اساس یک تابع ارتباط، میزان مرتبط بودن سند را با موضوعتخمین زده و به آن امتیاز می‌دهد (بخش تحلیل سند).اگر امتیاز سند از یک مقدار مشخص بیشتر باشد، به عنوان یک سند مرتبط در نظر گرفته شده و در مخزن ذخیره می‌شود، و در غیر این

صورت از آن سند صرف نظر می‌شود[Bat09,Cha99].



شکل (۵-۲): یک معماری کلی برای خزنده‌های متمرکز

پس از محاسبه میزان ارتباط سند، پیوندهای استخراج شده از آن می‌توانند با امتیاز سند والدشان در صف خزشاولویت‌بندی شوند[Zhe08,Cha99,Jal11]. در بعضی از خزنده‌ها نیز با انجام تحلیل بیشتر بر روی پیوندها، بر اساس کلمات موجود در متن لنگر[۱] متن پیرامون و یاآدرس پیوند، میزان ارتباط پیوند محاسبهشده و بر مبنای آن اولویت‌بندی می‌شود (بخش تحلیل پیوند) [Yuv06, Ha10].

بنابراین به طور کلی می توان گفت که پیاده‌سازی هر خزنده متمرکز شامل دو مرحله اصلی است: ۱) تعیین مجموعه مفاهیم مرتبط با موضوع ۲) تعیین تابع ارتباط[Bat09]. در ادامه، ابتدا به جزئیات بیشتری از روش‌های پیاده‌سازی این دو مرحله در خزنده‌های متمرکز و سپس کاربردهای آنها در حوزه‌های دیگر می‌پردازیم.

[۱] Anchor Text

۱-۳-۲- مفاهیم مرتبط با موضوع

از آنجایی که یک خزنده متمرکز، به دنبال جمع آوری اسناد در مورد یک موضوع خاص است، ابتدا باید موضوع موردنظر برای خزنده تعیین شود. در خزنده‌های متمرکز، از روش های مختلفی برای انجام این کار استفاده می‌شود.

ساده ترین و رایج ترین روش، دریافت کلمات کلیدی از کاربر است [Li05,Agg01]. اما یک کاربر، معمولا دامنه محدودی از کلمات را می‌تواند برای توصیف یک موضوع به کار ببرد. برای حل این مشکل، بعضی از خزنده های متمرکز با به کارگیری روش‌هایی به بسط موضوع می‌پردازند.

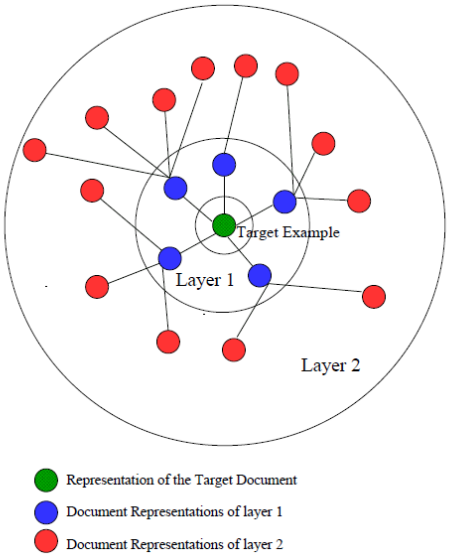
برای مثال، پس از دریافت کلمات کلیدی در مورد موضوع مورد نظر، این کلمات به صورت پرس‌وجو به موتورهای جستجوی معروف همچون گوگل، یاهو و یا MSN داده می‌شود و k نتیجه برتر بازیابی می‌شودو سپس محتوای اسناد موجود در لیست نتایج واکشی شده و کلمات داخل آنها استخراج می‌شود. پس از محاسبه مقدار TF-IDF به ازای هر کلمه، n کلمه با امتیاز TF-IDF بالاتر، به عنوان مفاهیم مرتبط با موضوع انتخاب می‌شوند [Hat10,Zhe08].

این کار با این فرض انجام می‌شود که اسناد بدست آمده از خروجی این موتورهای جستجو، مرتبط‌ترین اسناد برای موضوع موردنظر هستند [Hat10].اما معیار رتبه بندی در نتایج این موتورها برمبنای میزان محبوبیت اسناد است. .همچنین، بر اساس تشابه لغوی (متنی) اسناد با کلمات کلیدی پرس‌وجو، در مورد آنها تصمیم‌گیری می شود و تشابه معنایی در نظر گرفته نمی‌شود. بنابراین هیچ تضمینی وجود ندارد که نتایج حاصل از این موتورهای جستجو، مرتبط‌ترین اسناد برای موضوع مورد-نظر باشند.

برای در نظر گرفتن تشابه معنایی در بسط موضوع خزش، می‌توان از آنتولوژی ها استفاده کرد. در این حالت لازم است تا خزنده متمرکز مخزنی از آنتولوژی‌های مختلف داشته باشد و پس از دریافت

کلمات کلیدی در مورد موضوع، مرتبط‌ترین آنتولوژی را انتخاب کند[Yuv06]. اما میزان جامع بودن مخزن آنتولوژی و چگونگی تعیین مرتبط‌ترین آنتولوژی، چالش‌های جدیدی است که در این روش برای خزنده ایجاد می‌شود. برای این منظور در بعضی از خزنده‌ها، کاربر باید به جای کلمات کلیدی، آنتولوژی موضوع را نیز به صورت مستقیم به خزنده بدهد [Mae08, Kum12]. اما یافتن آنتولوژی مناسب برای یک موضوع کار آسانی نیست. همچنین، ممکن است که برای موضوع مورد نظر، آنتولوژیوجود نشده باشد و وظیفه ایجاد آن به کاربر واگذار می‌شود. در این روش، به جای آنتالوژی، از فهرست‌های موضوعی و دانش‌نامه نیز استفاده می‌شود[Jal11,Zhe08, Rav09, Yuv06, Hat10].

در روش دیگر، مجموعه‌ای از اسناد مرتبط با موضوع تحت عنوان اسناد هدف به خزنده داده می‌شود. در این حالت، خزندهبرای بسط موضوع و بدست آوردن مفاهیم مرتبط بیشتر، والدین این اسناد را تا سطح n ام از وب بازیابی می‌کند.اسناد هدف به همراه اجدادشان یک گراف زمینه (Context Graph)را تشکیل می‌دهند.نمونه‌ای از این گراف در شکل ۲-۶ نشان داده شده است. سپس مفاهیم موجود در اسناد هدف و والدین آنها استخراج شده و K مفهوم با TF-IDF بالاتر به عنوان مفاهیم مرتبط در نظر گرفته می شوند[Dil00].



شکل (۶-۲): یک گرافزمینهبا عمق دو

به ازای همه این روش‌ها، در نهایت مجموعه‌ای از مفاهیم مرتبط با موضوع بدست می‌آید. این مجموعه به صورت C={c1,c2,c3,…,cn} نمایش داده می‌شود که در آن هر ci نماینده یک مفهوم مرتبط است[Bat09]. اما می‌توان به هر مفهوم مرتبط، یک وزن نیز انتساب داد که بیان کننده میزان اهمیت آن باشد. در این حالت مجموعه مفاهیم مرتبط به صورت C={(c1,w1),(c2,w2),…,(cn,wb)} بیان می‌شود که در آن wi برابر با وزن مفهوم ci است. برای وزن‌دهی به مفاهیم مرتبط از روش‌های مختلفی استفاده می‌شود که معروفترین آنها TF-IDF است.

۲-۳-۲- تابع ارتباط

مهمترین بخش در خزنده‌های متمرکز، بخش محاسبه ارتباط سند واکشی شده با موضوع موردنظر است. در واقع خزنده سعی می‌کند تا میزان تشابه متنی یا معنایی سند را با موضوعمورد نظر تخمین بزند.برای این منظور از یک تابع ارتباط استفاده می‌شود. ورودی تابع ارتباط، اطلاعاتی در مورد

سندواکشی شده و خروجی آن میزان ارتباط سند با موضوع است.

برای مثال، می‌توان از تشابه کسینوسی [Hat10]، و یا از روش‌های یادگیری ماشین مانندروش‌های دسته بندی [Dil00, Rav09]، شبکه عصبی [Zhe208]، محاسبات فازی [Jal11]و غیره، برای پیاده‌سازی تابع ارتباط استفاده کرد. همچنین در صورت استفاده از آنتولوژی، روش‌های مختلفی برای تخمین میزان ارتباط معنایی سند با موضوع وجود دارد؛ بعضی از این روش‌ها عبارتند از: امتیاز دادن به کلمات موجود در سند یا پیوند بر اساس نوع رابطه [Yuv06] یا فاصله [Ehr03] آنها با مفاهیم مرتبط با موضوع بر مبنای ساختار آنتولوژی، نگاشت بین آنتولوژی موضوع و آنتولوژی سند[Mae08, Kum12] و غیره.

برای مثال مقاله [Zhe08]، یک خزنده متمرکز مبتنی بر آنتالوژی ارائه داده است که در آن، یک شبکه عصبی بر مبنای مجموعه‌ای از اسناد مرتبط و غیر مرتبط با موضوع، آموزش داده می شود و سپس از آن برای محاسبه ارتباط اسناد استفاده می شود. در این خزندهبه ازای هر سند موجود در مجموعه آموزش، کلماتداخل محتوای سند، استخراج می شوند. سپس از میان آنها، کلمات مشترک با مجموعه مفاهیم مرتبط به همراهتعداد تکرار آنها به یک شبکه عصبی داده می‌شود. بدین ترتیب شبکه بر اساس مجموعه آموزش، آموزش داده می شود.سپس خزنده در طول فرآیند خزش، به ازای هر سند، مفاهیم مرتبط موجود در محتوای سند را استخراج کرده و آنها را به همراه فرکانس‌شان به شبکه عصبی می دهد تا میزان ارتباط سند را پیش بینی کند.

پس از محاسبه میزان ارتباط یک سند، پیوندهای استخراج شده از آن می‌توانند با امتیاز سند والدشان در صف خزشاولویت‌بندی شوند. این کار با این فرض انجام می‌شود که اسناد مرتبط به هم پیوند می‌دهند که البته همیشه درست نیست[Ehr03, Mae08]. چرا که اسناد نامرتبط با یک موضوع می‌توانند شامل پیوندهایی به اسناد مرتبط باشند و از سوی دیگر یک سند مرتبط ممکن است شامل هیچ پیوند مرتبطی نباشد.

به همین دلیل در بعضی از خزنده‌های متمرکز با فرضاین که پیوند اشاره‌کننده به یک سند می‌تواند توصیف کننده ارتباط آن سند با موضوع باشد، پیوندهای استخراج شده از اسنادنیز مورد تحلیلقرار می‌گیرند. بدین ترتیب کهتابع ارتباطبر روی متن پیرامون، متن لنگر وآدرس پیوند اعمال می‌شود و میزان ارتباط پیوند با موضوع محاسبه می‌شود[Yuv06, Hat10]. همچنین بعضی از خزنده‌های متمرکز با در نظر گرفتن هر دو فرض فوق، امتیاز پیوند را با امتیاز سند والد (یا اجداد) پیوند ترکیب می‌کنند. با این کار خزنده تا حدی امکان پیشروی در مسیر های نامرتبط را نیز پیدا می‌کند.

برای محاسبه ارتباط پیوند می‌توان از روش های محاسبه ارتباط سند استفاده کرد و یا برای هر بخش، یک تابع ارتباط جداگانه در نظر گرفت. حتی در بعضی از خزنده ها، به ازای هر یک از اطلاعات موجود در مورد پیوند (یعنی متن پیرامون، متن لنگر و آدرس پیوند)، نیز از یک تابع ارتباط جداگانه استفاده می‌شود.

برای مثال، در مقاله[Hat10]، با فرض اینکه پیوندهای در مورد یک موضوع، از نظر آدرس باهم مشابه هستند، یک تابع ارتباط برای بررسی میزان مشابهت آدرس ها و محاسبه فاصله آنها نسبت به همارائه شده است.در این خزنده، مجموعه مفاهیم مرتبط شامل آدرس اسناد مرتبط با موضوع است که پس از دریافت کلمات کلیدی از کاربر، با استفاده از روش فراجستجو بدست می‌آیند. در طول فرآیند خزش، به ازای هر پیوند بدست آمده، در صورتی که فاصله آدرس پیوند از آدرس اسناد مرتبط با موضوع که در مجموعه مفاهیم مرتبط قرار دارند، کمتر یا مساوی از یک باشد، آن پیوند به عنوان پیوند مرتبط در نظر گرفته شده و در صف خزش قرار می‌گیرد. در این راستا، برای محاسبه فاصله پیوند **i**با پیوند **j**، از فرمول۲-۱ استفاده می شود.

$$dURL(i, j) = (dHost(i, j) +1) * (dPATH(i, j) +1) * (dQUERY(i, j)+1)-1 \qquad (۲-۱)$$

در این فرمول، مقادیر متغیر های dHost، dPATH و dQUERy به ترتیب برابر با میزان فاصله نام سایت میزبان، مسیرو پرس‌وجودر آدرس دو پیوند i و j است.

اگر نام سایت میزبان پیوند i با پیوند jبرابر باشد،dHostبرابر با صفر و در غیر این صورت برابر با ۳۲ است. عدد ۳۲ به صورت تجربی بدست آمده است. اگرمولفه پرس‌وجو دو پیوند با هم برابر باشد، مقدارdQUERY برابر با صفر و در غیر این صورت برابر با ۱ است.عدد ۱ با این دیدگاه انتخاب شده است که در بیشتر سایت‌ها، آدرس هایی که تنها در بخش پرس و جو با هم متفاوت هستند، توسط یک قالب یکسان ایجاد شده و دارای موضوع یکسان هستند. برای محاسبه فاصله مسیر در آدرس دو پیوند یعنی dPATHنیز با فرض n>=mاز فرمول ۲-۲ استفاده می شود:

$$dPATH(i, j) = (k * 2) + (n-m * 4) \quad (۲-۲)$$

- n: طول مسیر (تعداد پوشه ها به همراه نام فایل) پیوند i
- m: طول مسیر پیوند j
- k: تعداد مکان‌های متفاوت در مسیر دو پیوند(m مکان اول)

۳-۳-۲- پویایی عمق خزش

همانطور که بیان شد، یک خزنده متمرکز، میزان ارتباط پیوندها را با موضوع مورد نظر تخمین می زند و بر مبنای میزان ارتباط محاسبه شده، آنها را برای خزش اولویت بندی می کند. به این نوع خزش، خزش متمرکز نرم گفته می‌شود. اما چنانچه میزان ارتباط سند واکشی شده، از یک مقدار مشخص پایینتر باشد، خزندهمتمرکزمی‌تواند از پیوندهای داخل آن سندصرف‌نظر کند و در مسیر جاری از گراف خزش متوقف شود. در اصطلاح به این نوع خزش، خزش متمرکز سخت گفته می‌شود[Che99].

بعضی از خزنده‌های‌متمرکزنیز به صورت پویا عمل می‌کنند.در واقع،خزنده در صورت دستیابی

به یک سند نامرتبط، بلافاصله در آن مسیر متوقف نمی‌شود و از سوی دیگر، همه مسیرهای موجود را نیز مورد خزش قرار نمی دهد. در این حالت، به صورت پیش فرض، خزنده در هر مسیر از گراف خزش حداکثر تا عمق d می تواند پیش برود و در صورت دستیابی به یک سند مرتبط،پیش‌روی بیشتری در مسیر جاری خواهد داشت. این کار به خزنده کمک می کند تا در مسیرهایی که اسناد مرتبط وجود دارد با عمق بیشتر و در مسیرهایی که تعداد اسناد مرتبط در آنها کم است، با عمق کمتری (حداکثر تا عمق d) به خزش بپردازد[Her98,Bat12].

۴-۳-۲- آموزش خزنده متمرکز همزمان با فرآیند خزش

با توجه به وجود سایت های گوناگون با محتوا و ساختار پیوندی متفاوت بر روی وب، استفاده از یک مجموعه مفاهیم مرتبط از پیش تعریف شده و یا تابع ارتباطی که بر اساس یک مجموعه آموزش تعریف شده باشد، می‌تواند عملکرد خزنده متمرکز را با مشکل مواجه کند؛ چرا که ممکن است دید اولیه خزنده از محیط با واقعیت آن متفاوت باشد. اما استفاده از اسناد مرتبط بدست آمده در طول فرآیند خزش، می‌تواند دید خوبی از محیط خزش به خزنده بدهد.

برای مثال،پس از واکشی هر سند یا تعداد مشخصی از اسناد مرتبط ،می‌توان آنها رابه عنوان مجموعه آموزش به تابع ارتباط خزنده دادتا خزنده بر اساس آن دوباره آموزش ببیند[Umb09,Cha02]. سپس از تابع ارتباط بدست آمده می‌توان در ادامه فرآیند خزش استفاده کرد. بعضی از خزنده‌های متمرکز نیز، پیوندهای داخل صف را با کمک تابع ارتباط جدید، مجددا اولویت بندی می‌کنند[Agg01,Cha02]. همچنین می‌توان از مفاهیم داخل اسناد مرتبط واکشی شده، برای بسط موضوع و یا بروز رسانی مفاهیم مرتبط استفاده کرد[Agg01].

آموزش خزنده در طول فرآیند خزش، هر چند که به بهبود عملکرد خزنده می‌تواند کمک کند اما هزینه‌های زمانی و مکانی ناشی از آن وهمچنین هزینه زمانی اولویت بندی مجدد اسناد، نیز لازم

است تا مورد توجه قرار بگیرد.

۵-۳-۲- استفاده از خزش متمرکز در حوزه وب معنایی

در زمینه استفاده از خزش متمرکز در حوزه وب معنایی نیز، کارهایی انجام شده است که در آنهاهدف خزنده متمرکز، جمع‌آوری اسناد وب معناییدر مورد یک یا چند موضوع خاص است. برای این منظور، موضوع موردنظر در قالب یک آنتولوژی به خزنده داده می‌شود و سپس با نگاشت بین آنتولوژی موضوع و آنتولوژی سند و یا محاسبه فاصله بین مفاهیم بر اساس ساختار آنتولوژی، میزان ارتباط سند با موضوع محاسبه می‌شود [Mae08,Ehr03].

از سوی دیگر خزنده BioCrawler متعلق به موتور جستجوی WebOWL[Bat12] به جای خزش موضوعی اسناد وب معنایی، از ایده پویایی عمق در خزنده‌های متمرکز، برای دستیابی به اسناد وب معنایی استفاده می کند. به عبارت دیگر، در صورت دستیابی خزنده به یک سندوب معنایی، خزنده می‌تواند به خزش در مسیر جاری ادامه دهد. اما اگر به ازای n سند واکشی شده، هیچ یک از آنها معنایی نباشند، خزنده در مسیر جاری متوقف شده و از مسیر دیگری شروع به خزش می‌کند.

۶-۳-۲- استفاده از خزش متمرکز برای پیش‌بینی نوع اسناد

بعضی از خزنده‌های وب، به دنبال جمع آوری نوع خاصی از اسناد مانند عکس، ویدئو و غیره هستند. در این حالت، خزنده باید ضمن تلاش برای دستیابی سریع‌تر به اسناد از نوع مورد نظر، از واکشی سایر اسناد جلوگیری کند و هزینه ناشی از واکشی آنها را به حداقل برساند.

برای رسیدن به این هدفمی‌توان از ایده خزش متمرکز وب برای پیش‌بینی نوع محتویاسناد قبل از مرحله واکشی استفاده می‌شود. برای مثال، مقاله [Umb08] با انجام آزمایشاتی بر روی انواع مختلف اسناد وب، نشان می‌دهد کهمی‌توان از وجود کلمات خاص در مولفه مسیر آدرس پیوند و همچنین مکان پیوند در سند، برای تشخیص نوع محتوای پیوند استفاده کرد. برای مثال اگر

کلمه"image" در مولفه مسیر آدرس یک پیوند وجود داشته باشد و یا پیوند در تگ <img> قرار داشته باشد، می‌توان آن را از نوع عکس در نظر گرفت.

در مقاله [Umb09] نیز، یک خزنده متمرکز ارائه شده است که در مرحله استخراج پیوند، نوع محتواپیوندها را پیش‌بینی کرده و به پیوندهای از نوع مورد نظر اولویت بالاتری برای خزش می-دهد.ایده این روش این است که با توجه به پویایی وب و گسترده و متنوع بودن آدرس پیوندها، نمی-توان تنها با استفاده از آموزش یک دسته‌بند قبل از فرآیند خزش، از آن برای پیش‌بینی نوع رسانه پیوندها استفاده کرد. به همین دلیل، لازم است تا خزنده در طول فرآیند خزش و بر اساس اسناد مرتبط واکشی شده (اسناد از نوع مورد نظر)، به تدریج آموزش داده شود. برای این منظور از روشی مشابه به تولید قوانین همبستگی در طول فرآیند خزش استفاده می‌شود.

به بیان دقیق‌تر، پس از واکشی هر سند اطلاعات زیر در مورد آن استخراج می‌شود: ۱) پسوند فایل، ۲) کلمات موجود درمسیر فایل به صورت مجزا و ۳)ترکیب نام سایت میزبان و هر کلمه در مسیر آدرس سند. سپس این اطلاعات به همراه نوع واقعی سند، به صورت مجموعه‌ای از قوانین در یک پایگاه دانش قرار می‌گیرند. برای مثال اگر پسوند فایل در آدرس سند برابر با ".jpg" و نوع واقعی آن برابر "image/jpeg" باشد، آنگاه قانون ".jpg ⇒ image/jpeg" به پایگاه دانش افزوده می‌شود. ضریب اطمینان هر قانون نیز در طول فرآیند خزش بروز می‌شود که از فرمول ۲-۳برای محاسبه آن استفاده می‌شود.

$$conf(a \longrightarrow b) = \frac{\sup(a \cap b)}{\sup(a)} , \sup(x) = \frac{\#x}{\#total} \qquad (۲-۳)$$

به ازای هر سند واکشی شده، پیوندهای داخل آن استخراج می‌شوند و خزنده با کمک قوانین موجود در پایگاه دانش، میزان ارتباط هر پیوند را محاسبه می‌کند. برای محاسبه میزان ارتباط هر پیوند، قوانین بر روی پسوند فایل، کلمات موجود در مسیر فایل و در آخر ترکیب نام سایت میزبان و با کلمات مسیر فایل در آدرس پیوند، به ترتیب اعمال می شود و در صورت برقرار بودن یکی از قوانین،

ضریب اطمینان آن قانون به عنوان میزان ارتباط پیوند درنظر گرفته می‌شود. اگر میزان ارتباط پیوند از 0.5 بیشتر باشد، پیوند به عنوان یک پیوند مرتبط (از نوع رسانه مورد نظر) پیش‌بینی شده و بر مبنای میزان ارتباطش در صف اولویت‌بندی می‌شود. پیوندهای از نوع HTMLنیز در صف قرار می-گیرند تا خزنده با استفاده از آنها بتواند به اسناد دیگر دست پیدا کند. از سایر پیوند ها نیز صرف‌نظر می‌شود.

۷-۳-۲- خلاصه کارهای گذشته

تاکنون خزنده‌های متمرکز زیادی در حوزه وب پیاده‌سازی شده‌اند. همانطور که بیان شد،این خزنده‌ها با استفاده از یک تابع ارتباط و بر پایه‌ی مجموعه‌ای از مفاهیم مرتبط با موضوع مورد نظر، تشابه متنی (یا معنایی) سند (یا پیوند)را با موضوع بررسیکرده و میزان مرتبط بودن آن را پیش‌بینی می‌کنند. این پیش‌بینی بر مبنای یک یا چند فرض زیر انجام می‌شود:

۱) یک سند مرتبط می‌تواند بهاسناد مرتبط پیوند دهد.

۲) پیوند اشارهکننده به یک سند می‌تواند توصیف‌کننده ارتباط آن سند با موضوع باشد.

۳) یک پیوند در صورت داشتن همزادهای مرتبط، می‌تواند مرتبط باشد.

۴) اسناد بدست آمده از خروجی موتورهای جستجوی معروف مانند گوگل (روش فراجستجو)، مرتبط‌ترین اسناد برای موضوع مورد نظر هستند.

۵) موضوع مورد نظر برای خزنده از پیش تعریف شده است.

۶) اسناد مرتبط دارای ویژگی های (محتوا،آدرس، متن پیرامون ومتن لنگر) مشابه به هم هستند.

در جدول ۲-۲ خلاصه‌ای از بعضی کارهای بررسی شده در حوزه خزش متمرکز آورده شده است که از نظر شیوه تعیین و بسط موضوع، نوع تشابه، نوع تابع ارتباط، ورودی‌های تابع ارتباط، فرض‌های در نظر گرفته شده برای محاسبه ارتباط و نوع تمرکز با هم مقایسه شده اند.

جدول(۲-۲):خلاصه کارهای گذشته درحوزه خزش متمرکز وب

مرجع	تعیین مفاهیم مرتبط با موضع		تحلیل سند		تحلیل پیوند		تمرکز	آموزش همزمان	اولویت بندی مجدد	مفروضات
	تعیین موضوع	بسط موضوع	تابع ارتباط	تشابه	تابع ارتباط	تشابه				
[Yuv06]	کلمه کلیدی	آنتالوژی	-	-	مبتنی بر آنتالوژی	معنایی	سخت	-	-	۲و۶
[Jal11]	کلمه کلیدی	آنتالوژی	استنتاج فازی	معنایی	-	-	نرم	-	-	۱و۶
[Zhe08]	کلمه کلیدی	آنتالوژی	شبکه عصبی	معنایی	-	-	نرم	-	-	۱و۶
[Mae08]	آنتالوژی	-	مبتنی بر آنتالوژی	معنایی	-	-	نرم	-	-	۱و۶
[Hat10]	کلمه کلیدی	فراجستجو	-	-	فاصله آدرس پیوندها	متنی	سخت	-	-	۲و۴و۶
[Hat10]	کلمه کلیدی	فراجستجو	تشابه کسینوسی	متنی	تشابه کسینوسی	متنی	نرم	-	-	۱و۲و۴و۶
[Cha99]	فهرست موضوعی	فراجستجو	دسته بند بیز	متنی	-	-	سخت-نرم	-	-	۱و۵و۶
[Dil00]	اسناد هدف	گراف زمینه	دسته بند بیز	متنی	-	-	نرم	-	-	۱و۵و۶
[Pan04]	کلمه کلیدی	-	تشابه کسینوسی	متنی	شبکه عصبی	متنی	نرم	-	-	۲و۵و۶
[Hat10-2]	کلمه کلیدی	فراجستجو	تشابه کسینوسی	متنی	تشابه کسینوسی	متنی	سخت	-	-	۲و۴و۶
[Sun08]	دانشنامه	-	-	-	مبتنی بر آنتالوژی	معنایی	سخت	-	-	۲و۶
[Li05]	کلمه کلیدی	-	-	-	درخت تصمیم	متنی	نرم	-	-	۲و۵و۶
[Cha02]	فهرست موضوعی	-	دسته بند بیز	متنی	دسته بند بیز	متنی	سخت	تابع ارتباط	به ازای N سند	۱و۲و۵و۶
[Agg01]	کلمه کلیدی	-	دسته بند بیز	متنی	دسته بند بیز	متنی	نرم	مفاهیم مرتبط	به ازای هرسند	۱و۲و۳و۵و۶
[Umb09]	-	-	-	-	قوانین همبستگی	متنی	نرم	تابع ارتباط	-	۲و۶
[Xu07]	-	-	-	-	قوانین همبستگی	متنی	سخت	-	-	۲و۵و۶

۴-۲- خلاصه فصل

در این فصل ابتدا، فرآیند خزش وب و معماری کلی خزنده‌های وب توضیح داده شد. رفتار هر خزنده برگفته از سیاست انتخاب و بروز رسانی اسناد، مودبانه بودن و موازی سازی آن در طول فرآیند خزش است. در بخش دوم، خزنده های وب معنایی معرفی شدند که هدف آنها جمع‌آوری اسناد وب معنایی از سطح وب است. همچنین کارهای مرتبط در این حوزه و چالش های موجود مورد بررسی قرار گرفت. بخش پایانی نیز به خزنده های متمرکز وب و کارهای مرتبط در این حوزه اختصاص داشت. یک خزنده متمرکز، به جمع آوری اسناد وب در مورد یک یا چند موضوع خاص می‌پردازد.

Formatted: Font: +Body (Calibri)

فصل ۳- روش پیشنهادی

در این فصل، یک خزنده متمرکز وب پیشنهاد شده است کهبه جای خزش موضوعی اسناد، با تحلیل پیوندهای داخلاسنادHTML، تلاش می‌کند تا دسترسی سریعتری به اسناد وب معنایی احاطه شده توسط اسناد HTMLداشته باشد. برای این منظور از دو تابع ارتباط برای پیش‌بینی نوع پیوندها استفاده شده است. سپس بر مبنای خزنده متمرکز پیشنهادی،دوچارچوب برای خزش وب معناییارائهشده است.

در ادامه، بخش های مختلف خزنده متمرکز پیشنهادی شرح داده می شود. در بخشپایانی نیز به توصیف چارچوب‌های پیشنهادی برای خزش وب معنایی پرداخته می‌شود.

۳-۱- خزنده متمرکز HTML با هدف دستیابی به اسناد وب معنایی

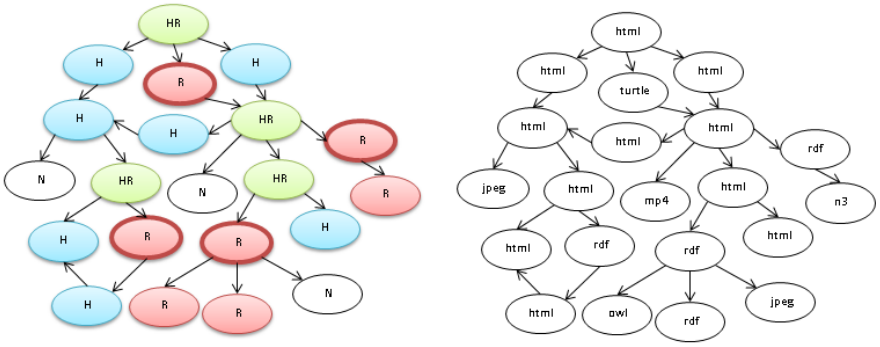
خزنده متمرکز پیشنهادی وظیفه دارد تا اسناد وب معنایی احاطه شده توسط اسناد HTML را از سطح وب جمع‌آوری کند. از این رو، خزنده در بین پیوندهای استخراج شده از اسناد HTML به دنبال پیوندهای اشاره کننده به اسناد وب معنایی می‌گردد (پیوند های مرتبط) تا آنها را زودتر از پیوندهای دیگر مورد واکشی قرار داده و بدین ترتیب دستیابی سریعتری به اسناد وب معنایی داشته باشد.

در اینجا،اسناد وب را به چهار دسته تقسیم می‌کنیم و به هر دسته یک برچسب اختصاص می-دهیم.

- اسناد RDF (R): اسناد وب معنایی. از آنجایی که اسنادوب معنایی از مدل دادهیRDF استفاده می‌کنند، در اینجا به اختصار آنها رااسناد RDFمی‌نامیم.
- اسناد HTMLوالد RDF (HR): اسناد HTML که شامل پیوندهایی به اسناد RDF هستند.در اینجا به اختصار این اسناد را اسناد والدRDFمی‌نامیم.

- اسناد HTML(H): سایر اسناد HTML.
- سایر اسناد (N).

در شکل ۳-۱، یک نمونه فرضی از گراف وب نشان داده شده است. در سمت راست شکل، نوع محتوای واقعی هر سند (گره) و در سمت چپ شکل، برچسب هر سند مشخص شده است.



شکل (۳-۱): دسته بندی اسناد وب از دیدگاه خزنده متمرکز پیشنهادی

وقتی یک سند HTML تجزیه‌شده[1] و پیوندهای داخل آن استخراج می‌شود، خزنده متمرکز باید پیش‌بینی کند که هر پیوند به چه نوع سندی اشاره می‌کند (تعیین برچسب پیوند). سپس خزنده بر اساس برچسب پیش‌بینی شده برای هر پیوند، در مورد آن تصمیم‌گیری می‌کند.

در صورت درست بودن پیش‌بینی درباره پیوندهای والد RDF و دادن اولویت بالاتر به آنها نسبت به سایر پیوندهای HTML، خزنده متمرکز می‌تواند دستیابی سریع‌تری به این اسناد و در نتیجه به پیوندهای RDF داخل آنها داشته باشد. همچنین، در صورت درست بودن پیش‌بینی‌ها در مورد پیوندهای اشاره‌کننده به اسناد RDF و تشخیص آنها قبل از مرحله واکشی، خزنده می‌تواند با دادن اولویت بالاتر به آنها نسبت به سایر پیوندها، آنها را زودتر از سایر اسناد مورد واکشی قرار دهد. در صورت شناسایی صحیح اسناد غیر معنایی قبل از مرحله واکشی نیز، خزنده می‌تواند از واکشی آنها

---

[1] Parse

جلوگیری کرده و بدین ترتیب حجم واکشی اسناد غیر معنایی را کاهش دهد.

نکته مهمی که باید به آن توجه داشت این است که خزنده متمرکز پیشنهادی، پس از دستیابی به اسناد وب معنایی، آنها را مورد خزش قرار نمی‌دهد و پیوند های داخل آنها را دنبال نمی‌کند. چرا که هدف خزنده تنها کشف اسناد وب معنایی احاطه شده توسط اسناد HTML است. به عبارت دیگر هدف خزنده پیشنهادی دستیابی سریع‌تر به اسناد RDF ای است که در شکل ۳-۱ دور آنها با رنگ قرمز برجسته شده است.

همانطور که در بخش ۲-۳ بیان شد، طراحی هر خزنده متمرکز شامل دو مرحله اصلی است:

۱) تعیین مجموعه مفاهیم مرتبط که به صورت $C=\{c1,c2,c3,\ldots,cn\}$ نمایش داده می‌شود و هر ci نماینده یک مفهوم مرتبط است.

۲) تعیین تابع ارتباط که به صورت $Relevancy(\prod,C)$ نمایش داده می‌شود و در آن، $\prod$، مجموعه اطلاعات جمع‌آوری شده در مورد یک سند یا پیوند و C، مجموعه مفاهیم مرتبط است.

با توجه به اهداف بیان‌شده برای خزنده متمرکز پیشنهادی، فضای مسئله را به دو بخش تقسیم می‌کنیم:

۱) شناسایی پیوندهای RDF

۲) شناسایی پیوندهای والد RDF.

از آنجا که شیوه‌ی انتخاب مفاهیم مرتبط و تابع ارتباط برای هر یک از این بخش‌ها متفاوت است، در ادامه، راه‌حل پیشنهادی برای هر بخش رابه صورت جداگانه شرح خواهیم داد.

۱-۱-۳- تابع ارتباط برای شناسایی پیوندهای RDF

در این بخش یک تابع ارتباط پیشنهاد داده می‌شود که بر اساس اطلاعات موجود در مورد یک

پیوند، پیش‌بینی می‌کند که آن پیونداز نوع RDF هست یا خیر.

برای تشخیصRDF بودن یک پیوند می‌توان از پسوند فایل موجود در آدرس پیوند مانند ".rdf" و یا ".owl" استفاده کرد. اما پسوند فایل در آدرس بسیاری از پیوندها مشخص نیست. همچنین اسنادی با انواع محتوای مختلف وجود دارند که دارای پسوند فایل یکسان هستند.

همانطور که در بخش ۲-۳-۶ بیان شد، مقاله [Umb09] یک تابع ارتباط مبتنی بر قوانین همبستگی برای پیش‌بینی نوع پیوندها ارائه داده است که در آن علاوه بر پسوند فایل،از کلمات موجود در مولفه مسیر و نام سایت میزبان در آدرس پیوندها، برای پیش‌بینی نوع آنها استفاده می‌شود.در این تابع ارتباط، کلمات موجود در بخش آدرس هر پیوند با کلمات موجود در پیوندهای مرتبط(پیوندهای از نوع مورد نظر) که در طول فرآیند خزش بدست آمده اند، با هم مقایسه شده و بر مبنای آن نوع پیوندها پیش‌بینی می‌شود.

همچنین، در بعضی از خزنده‌های متمرکز موضوعی [Hat10,Bat09]، به ازای هر پیوند استخراج شده از اسناد، متن لنگر و متن پیرامون پیوند بررسی می‌شود و در صورتی که شامل کلماتی در مورد موضوع موردنظر باشد، خزنده، پیوند را مرتبط با موضوع در نظر گرفته و به آن اولویت بالاتری برای خزش می‌دهد.

این در حالی است کهآدرس، متن لنگر و متن پیرامون پیوندهای RDFمی‌تواند شامل هر مفهومی باشد و این مفاهیم با پیوندهای دیگر اشتراکات زیادی دارند. بنابراین تنها به علت آنکه مجموعه ای از کلماتدر متن لنگر، متن پیرامون و یا آدرس یک پیوند RDF وجود دارد، نمی‌توان آنها را به عنوان مفاهیم مرتبط در نظر گرفت.

اما در گرامر اسناد HTML، پیوندها در تگ‌های مختلفی مانند"anchor"، "link" و غیره قرار می‌گیرند. هر تگ پیوند، شامل مجموعه‌ای از صفات و مقادیر آنها است. برخی از این صفات عبارتند از:

نوع محتوای پیوند، عنوان پیوند و متن لنگر[1] معمولا کاربران زمانی که می‌خواهند در یک سند HTML، یک پیوند از نوع RDF را قرار دهند، در متن پیرامون، متن لنگر، صفت نوع محتوا یا صفت عنوان پیوند، به نوع محتوای آن اشاره می‌کنند.

در ادامه، با توجه به مطالب بیان شده، تابع ارتباط $Relevancy_R$را برای شناسایی اسناد RDFمعرفی می‌کنیم. این تابعبه فرم ریاضی $label=Relevancy_{HR}(\prod_L, C_{HR})$ نمایش داده می‌شود. $\prod_L$، شامل مقدار صفت نوع محتوا، مقدار صفت عنوان، متن لنگر، متن پیرامون، مولفه مسیر و پسوند فایل برای پیوند L، C_R برابر با مجموعه مفاهیم مرتبط و label، برچسب پیش‌بینی شده برای پیوند L است که برابر با یکی از مقادیرH،Rو یا Nخواهد بود.

مجموعه مفاهیم مرتبط C_R، شامل انواع محتوای خاص اسناد وب معنایی مانند 'text/n3' ($S_{RDFContentType}$)، مفاهیم خاص اسناد وب معنایی مانند rdf، foaf، triple($S_{RDFConcept}$) وپسوندهای فایل خاص اسنادوب معنایی مانند '.owl'($S_{RDFExtension}$) است. مجموعه C_Rرا می‌توان به فرم ریاضی نمایش داد:

$$C_R = S_{RDFContentType} \cup S_{RdfConcept} \cup S_{RdfExtension}$$

تابع ارتباط $Relevancy_R$ با استفاده از این مفاهیم و به کارگیری مجموعه‌ای از قوانین (۱۱ قانون)، RDF بودن یک پیوند را تشخیص می‌دهد. قوانین پیشنهادی به تفکیک نوع عملکرد، در سه دسته قرار می‌گیرند (برچسب هر قانون در مقابل آن نوشته شده است):

- تطبیق پسوند فایل پیوند (ex1) با یکی از پسوندهای خاص اسنادوب معنایی
- تطبیق صفت نوع محتوا (ty1)، صفت عنوان (ti1)، متن لنگر (an1) و یا متن پیرامون (co1) پیوند با یکی از انواع محتوای اسناد وب معنایی

[1] Anchor Text

- وجود حداقل یک مفهوم از مفاهیم خاص وب معنایییدر صفت نوع محتوا (ty2)، صفت عنوان (ti2)، متن لنگر (an2)، متن پیرامون(co2) و یا بخش مسیر (pa2) در آدرس پیوند.

در صورت برقرار بودن یکی از این قوانین، خزنده پیوند را از نوع RDF پیش‌بینی کرده وبرچسب Rرا به آن اختصاص می‌دهد.اما از میان پیوندهایی که از نوع RDF پیش بینی نمی‌شوند، پیوندهایی که دارای پسوند فایل خاص اسناد HTML هستند و یا پسوند فایل ندارند، در دسته پیوندهای HTML قرار می‌گیرند و برچسب H به آنها داده می‌شود. به پیوند های باقی مانده نیز برچسب N تعلق می-گیرد.

۲-۱-۳- تابع ارتباط برای پیش‌بینی پیوندهای HTML والد RDF

خزندهمتمرکز پیشنهادی، در صورتی که شناساییخوبیازاسنادHTMLوالدRDFداشتهباشدمی-تواندددسترسیسریع‌تریبهپیوندهایRDFپیداکند؛ چرا که برای رسیدن به پیوندهای RDFلازم است تا خزنده اسناد شامل آنها را قبلا واکشی کرده باشد.

در متن لنگر و متن پیرامون پیوند های والد RDF، کلمات خاصی برای شناسایی آنها وجود ندارد.همچنین، کلمات موجود در این بخش ها با پیوند های غیر والد RDF اشتراکات زیادی دارند. دلیل آن هم این است که پیوندهای والد RDF می‌توانند در مورد هر موضوعی و در نتیجه شامل هر کلمه‌ای باشند.

اما نتایج حاصل از بررسی چند سایت نمونه نشان می دهد که آدرس پیوندهای والد RDF در یک سایت یا دامنه،در بخش مسیر شان (نام پوشه‌ها) تقریبا مشابه به هم هستند. این مشابهت زمانی قابل استفاده است که کلمات استخراج شده از آدرس یک پیوند، در کنار هم و با در نظر گرفتن ساختار آن (یعنی نام سایت میزبان، مولفه مسیر و مولفه پرس و جو) مورد بررسی قرار گیرد.(به عبارت دیگر از نظر حوزه پردازش زبان طبیعی، به جای بررسی جداگانه هر کلمه، مجموعه‌ای از

کلمات متوالی (n گرم ها) در آدرس پیوندها در نظر گرفته شود.)

بنابراین با این فرض که آدرس پیوندهای والد RDF در یک دامنه تقریبا مشابه به هم هستند، نیاز به یک تابع ارتباط داریم که میزان مشابهت آدرس‌ها را تعیین کند.

همانطور که در بخش ۲-۳-۳ بیان شد، در مقاله[Hat10]، یک تابع ارتباط برای محاسبه مشابهت آدرس‌ها بر اساس ساختار آنها ارائه شده است. اما تابع ارتباط معرفی‌شده، بر اساس دیدگاه خزش متمرکز موضوعی است.همچنین آدرس اسناد مرتبط با موضوع، باید قبل از شروع فرآیند خزش به خزنده داده شود که این امر باعث ایجاد محدودیت برای خزنده می‌شود.

در ادامه، با ایده مشابه به تابع ارتباط معرفی شده در مقاله[Hat10]، تابع ارتباط $Relevancy_{HR}$ را برای بررسی شباهت آدرس‌ها و پیش‌بینی والد RDF بودن یک پیوند، پیشنهاد می‌دهیم. تابع ارتباط$Relevancy_{HR}$ فرم ریاضی زیر نمایش داده می‌شود:

$label=Relevancy_{HR}(\prod_L, C_{HR}(t))$

$\prod_L$، شامل نام سایت میزبان، مولفه مسیرو مولفه پرس‌وجو در آدرس پیوندL، $C_{HR}(t)$ برابر با مجموعه مفاهیم مرتبط در لحظه t و label نیز برابر با برچسب پیش‌بینی شده برای پیوند L است.

در اینجا، مجموعه مفاهیم مرتبط که با $C_{HR}(t)$ نمایش داده می‌شود، شامل نام سایت میزبان (Hj) و مولفه مسیر (Pj) در آدرس اسناد والد RDFای است که تا لحظه t توسط خزنده شناسایی شده‌اند. به عبارت دیگر، یادگیری مفاهیم مرتبط در مورد اسناد والد RDF، همزمان با فرآیند خزش انجام می‌شود، بدین ترتیب که در مرحله واکشی، به ازای هر سند RDFبدست آمده، سند والد آن به عنوان یک سند والد RDF در نظر گرفته می‌شود ومفاهیم موجود در آدرس آن بهمجموعه$C_{HR}(t)$ افزوده می‌شود.

از سوی دیگر، در مرحله استخراج پیوند، زمانی که پیوند L از یک سند HTML استخراج شود،

آدرس آن با آدرس اسناد والد RDF موجود در مجموعه$C_{HR}(t)$ مقایسه می‌شود و در صورت وجود مشابهت با یکی از آنها، پیوندL به عنوان والد RDFپیش‌بینی شده و به آن برچسب 'HR' داده می-شود. روش تعیین مشابهت پیوند ها در ادامه توضیح داده شده است.

فرض کنید که می خواهیم مشابهت دو آدرس پیوند iبا آدرس پیوند j را بررسی می کنیم. برای این منظور، هر یک از بخش‌های مسیر و نام سایت میزبان در آدرساین پیوندها، به صورت دو بدو با هم مقایسه می‌شوند. برای مثال وضعیت بخشمسیر (sPath)به صورت زیر تعریف می‌شود:

- (=): اگر Pi=Pj,Pi!=null و Pj!=null
- (#): اگر PFi!=PFj,Pi!=null و Pj!=null
- (n): اگر Pi=null و Pj=null
- (#n): اگر Pi!=null و Pj=null،یا Pi=null و Pj!=null

وضعیت نام سایت‌های میزبان نیز به همین شکل تعیین‌می‌شود (sHost). همچنین، اگر بخش PFدر دو پیوند i و j با هم متفاوت باشد (sPath برابر با '#' باشد)، در این صورت مقادیر متغیرهایn، m، k و x به صورت زیر محاسبه می‌شود:

- n: طول مسیر در آدرس پیوند i
- m: طول مسیر در آدرسپیوند j
- k: تعداد مکانهای متفاوت در مسیر آدرس دو پیوند، به ازای m مکان اول (با فرض n≥m)
- x: مکان آخرین اختلاف در مسیر آدرس دو پیوند

سپس،بر مبنای قوانین زیر، در مورد مشابه بودن پیوند ها تصمیم‌گیری می‌شود. این قوانین بر اساس رفتار احتمالی طراحان سایت و منتشرکنندگان اسناد RDF قابل توجیه است.

- (r1): اگر دو پیوند i و j متعلق به یک سایت میزبان باشند (sHost برابر با '=' باشد) اما

مولفه مسیر آنها متفاوت باشد (sPath برابر با '#' باشد)، در صورتی این دو پیوند، مشابه در نظر گرفته می‌شوند که تنها در آخرین مکان در بخش PF با هم متفاوت باشند، یعنی$n>1$، $n=m$، $k=1$ و$x=n-1$.

برای مثال، فرض کنید یک سایت، اطلاعات مربوط به اعضای خود را در قالب فایل‌های RDF تولید کرده است. معمولا فایل RDF مربوط به هر عضو در صفحه شخصی همان عضوقرار داده می‌شود. فهرست اعضا نیز در فهرستاعضای سایت قرار دارد (برای مثال members). بنابراین از نظر ساختار سلسله مراتبی آدرس صفحات شخصی اعضا تنها در آخرین مکان در مولفه مسیر با هم تفاوت دارد. برای مثال:

آدرس ۱= http://example.com/members/member1/

آدرس ۲= http://example.com/members/member2/

اگر خزنده به صفحه شخصی عضو ۱ (آدرس ۱) مراجعه کند، از آنجایی که در این صفحه یک فایل RDF قرار دارد، خزنده آن صفحه را به عنوان سند والد RDF در نظر می‌گیرد. حال در صورت برخورد با پیوندی با آدرس ۲، به دلیل مشابهت آدرس ۲ با آدرس ۱، خزنده می‌تواند آن پیوند را از نوع والد RDF پیش‌بینی کند.

شرط $n>1$به این دلیل در نظر گرفته شده است که اگر n برابر با یک باشد، آنگاه بخش مسیر برابر با نام فایل و یا نام یکی پوشه‌های اصلی سایت است. برای مثال دو آدرس http://example.com/members/ و http://example.com/news/را در نظر بگیرید. با وجود اینکه مکان آخر در مسیر دو آدرس با هم متفاوت است، اما مسیر فایل آنها با هم مشابه نیست؛ بنابراین نمی‌توان این دو پیوند را مشابه در نظر گرفت.

- (r2): اگر بخش سایت میزبان و مسیر دو پیوند i و jبا هم یکسان باشد (sHost برابر با '=' و sPath برابر با '='باشد)، آنگاه دو پیوند مشابه در نظر گرفته می‌شوند.

در این حالت اختلاف دو پیوند تنها در بخش پرس‌وجوی آنها است، بنابراین خزنده می‌تواند در صورت یکسان بودن بخش سایت میزبان و مولفه مسیر دو پیوند، آنها را مشابه در نظر بگیرد.

۳-۱-۳- تابع ارتباط نهایی

تابع ارتباط نهایی، ترکیبی از توابع ارتباط پیشنهادی در دو بخش قبلی است که به فرم ریاضی priority=Relevancy($\prod_L$, C_R, $C_{HR}(t)$) بیان می‌شود. $\prod_L$، اطلاعات استخراج شده در مورد پیوند L، C_R، مجموعه مفاهیم مرتبط برای شناسایی پیوندهای RDF و $C_{HR}(t)$، مجموعه اطلاعات موردنظر درباره آدرس اسناد والد RDFای است که تا لحظه t توسط خزنده واکشی شده‌اند. خروجی تابع نیز برابر با اولویت پیوند برای خزش است (priority).

ابتدا خزنده با استفاده از تابع ارتباط $Relevancy_R$، پیش‌بینی می‌کند که پیوند L، یک پیوند RDF هست یا خیر. در صورتی که پیوند از نوع RDF نباشد ولی از نوع HTML پیش‌بینی شود، خزنده با انجام تحلیل بیشتر و با استفاده از تابع ارتباط $Relevancy_{HR}$، والد RDFبودن یا نبودن پیوند L را پیش‌بینی می‌کند. در نهایت، برچسب پیوند شامل یکی از مقادیر R، HR، H و Nخواهد بود که بر اساس آن به ترتیب، اولویت۱، ۲،۳ و ۱- به پیوند تعلق می‌گیرد. شبه کد تابع ارتباط پیشنهادی در شکل ۳-۲ نمایش داده شده است.

```
1:  fFunction Relevancy ( ∏L , CR , CHR(t) )
2:  {
3:      label=RelevancyR ( ∏L , CR )
4:      if  label is H
5:          label=RelevancyHR ( ∏L , CHR(t) )
6:      if label is R
7:          priority=1
8:      else if label is HR
9:          priority=2
10:     else if label is H
11:         priority=3
12:     else
```

Formatted: Font: (Default) Courier New, Complex Script Font: Courier New

Formatted: Font: (Default) Courier New, Not Bold, Complex Script Font: Courier New, Not Bold

Formatted: Font: (Default) Courier New, Complex Script Font: Courier New

Formatted: Font: (Default) Courier New, Not Bold, Complex Script Font: Courier New, Not Bold

Formatted: Font: (Default) Courier New, Complex Script Font: Courier New

Formatted: Font: (Default) Courier New, Not Bold, Complex Script Font: Courier New, Not Bold

Formatted: Font: (Default) Courier New, Complex Script Font: Courier New

Formatted: Font: (Default) Courier New, Not Bold, Complex Script Font: Courier New, Not Bold

Formatted: Font: (Default) Courier New, Complex Script Font: Courier New

Formatted: Font: (Default) Courier New, Not Bold, Complex Script Font: Courier New, Not Bold

Formatted: Font: (Default) Courier New, Complex Script Font: Courier New

Formatted: Font: (Default) Courier New, Not Bold, Complex Script Font: Courier New, Not Bold

Formatted: Font: (Default) Courier New, Complex Script Font: Courier New

Formatted: Font: (Default) Courier New, Not Bold, Complex Script Font: Courier New, Not Bold

Formatted: Font: (Default) Courier New, Complex Script Font: Courier New

Formatted: Font: (Default) Courier New, Not Bold, Complex Script Font: Courier New, Not Bold

Formatted: Font: (Default) Courier New, Complex Script Font: Courier New

Formatted: Font: (Default) Courier New, Not Bold, Complex Script Font: Courier New, Not Bold

Formatted: Font: (Default) Courier New, Complex Script Font: Courier New

Formatted: Font: (Default) Courier New, Not Bold, Complex Script Font: Courier New, Not Bold

```
13:         priority=-1
14:     return priority
15: }
```

شکل (۲-۳): شبه کد تابع ارتباط برای خزنده متمرکز پیشنهادی

Formatted: Font: (Default) Courier New, Not Bold, Complex Script Font: Courier New, Not Bold

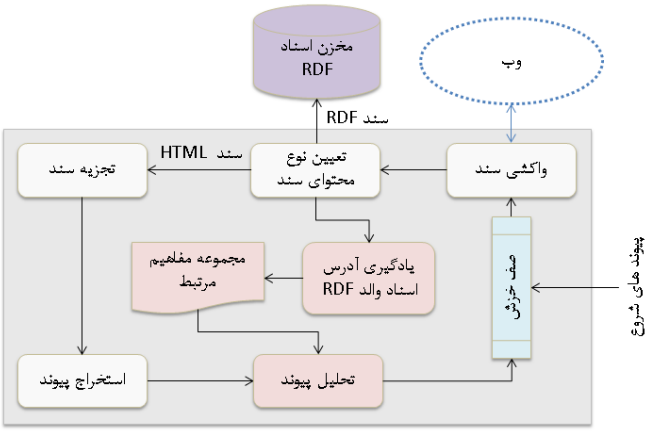
Formatted: Font: (Default) Courier New, Complex Script Font: Courier New

Formatted: Font: (Default) Courier New, Not Bold, Complex Script Font: Courier New, Not Bold

۴-۱-۳- الگوریتم خزش متمرکز

شکل ۳-۳معماری خزنده متمرکز پیشنهادی را نشان می‌دهد. مجموعه مفاهیم مرتبط شامل مجموعه های C_R و $C_{HR}(t)$ است. مجموعه $C_{HR}(t)$ در ابتدا خالی است و در طول فرآیند خزش، به تدریج، آدرس اسناد والد RDFبه آن افزوده می‌شود.

برای شروع فرآیند خزش، چند آدرس HTML به عنوان پیوندهای شروع در صف خزنده قرار داده می شود. سپس در هر چرخه از فرآیند خزش، خزنده متمرکز یک پیوند را از صف بر می‌دارد (به ترتیب اولویت) و آن را مورد پردازش قرار می‌دهد.برای این منظور،ابتدا درخواست واکشی سند به سایت میزبان آن ارسال می‌شود.سپس در صورت موفق بودن وضعیت واکشی،در صورتی که فیلد نوع محتوای سند برابر با یکی از انواع خاص اسناد وب معنایی یا اسناد HTML باشد و یا برچسب سند از نوع R باشد، خزنده محتوای سند را واکشی می‌کند و در غیر این صورت از آن صرف‌نظر می‌کند (صافی واکشی).



شکل (۳-۳): معماری خزنده متمرکز پیشنهادی

Commented [m6]: این با شکلش در یک صفحه باشند

بعد از مرحله واکشی، نوع محتوای واقعی سند تعیین می‌شود.در این حالت، خزنده متمرکز با یکی از سه وضعیت زیر روبرو می‌شود:

۱) اگر سند واکشی شده از نوع HTML یا RDF نباشد، از آن صرف نظر می‌شود.

۲) اگر سند واکشی شدهاز نوعRDFباشد، خزنده آن سند را در مخزن اسناد RDF ذخیره می کند. همچنین، آدرس سند والد آننیز به مجموعه مفاهیم مرتبط $C_{HR}(t)$ افزوده می‌شود (بخش یادگیری).

۳) اگر سند واکشی شده از نوعHTML باشد، سند تجزیه می‌شود و پیوندهای داخل آن استخراج می‌شود سپس هر پیوند مورد تحلیل قرار می گیرد. مراحل تحلیل پیوند به صورت زیر است:

۱. اگر پیوند دارای پسوند فایل غیر معنایی شناخته شده مانند '.jpg'، '.pdf'، '.txt' و غیره باشد، از آن صرف نظر می‌شود (صافی پیوند اول).

۲. در صورتی که پیوند از صافی عبور کند، با استفاده از اطلاعات جمع‌آوری شده در مورد آن ($\prod_L$)، مجموعه مفاهیم مرتبط (C_R و $C_{HR}(t)$) و بر مبنای تابع ارتباط معرفی‌شده در بخش ۳-۱-۳، برچسب پیوند پیش‌بینی شده و اولویت آن برای خزش تعیین می‌شود.

۳. اگر پیوند از نوع N پیش‌بینی شده باشد از آن صرف نظر می‌شود (صافی پیوند دوم).

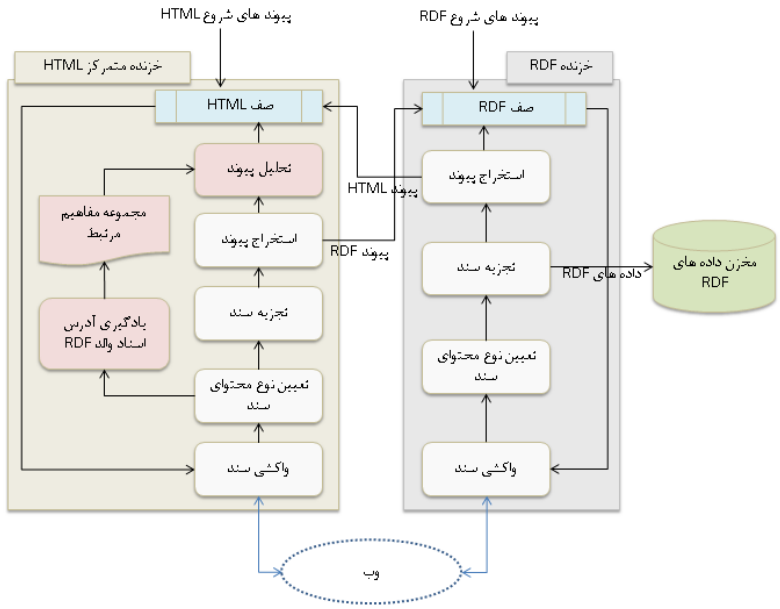
۳-۲- چارچوب‌هایپیشنهادی برای خزش وب معنایی

همانطور که در بخش ۲-۲-۲ بیان شد، با توجه به عدم اتصال کافی بین اسناد وب معنایی و احاطه شدن آنها توسط اسناد HTML، یک خزنده وب معنایی در صورتی که بخواهد یک پوشش

حداکثری از اسناد وب معنایی داشته باشد، لازم است تا در کنار اسناد وب معنایی، اسناد HTML را نیز مورد خزش قرار دهد. اما از آنجایی که هدف خزنده‌های وب معنایی، خزش اسناد وب معنایی است، خزنده باید سعی کند ضمن دستیابی سریعتر به اسناد وب معنایی، حجم واکشی اسناد غیر-معنایی را نیز کاهش دهد. در ادامه، برای رسیدن به این اهداف دو راه‌حل پیشنهاد داده می‌شود.

اولین راه‌حل پیشنهادی این است که به جای خزش اسناد HTML توسط خزنده وب معنایی، می‌توان از خزنده متمرکز HTMLپیشنهادی در بخش ۳-۱ در کنار خزنده وب معنایی استفاده شود؛ به گونه‌ای که این دو خزنده با هم تبادل پیوند داشته باشند. به بیان بهتر، خزنده‌وب معنایی تنها اسناد وب معنایی را خزش می‌کند و در صورت برخورد با یک سندHTML، پس از استخراج پیوندهای داخلسند، آنها را در اختیار خزنده متمرکز HTML قرار دهد. خزندهHTML نیز در هنگام برخورد با یک سندوب معنایی، پیوند های استخراج شده از آن را در اختیار خزنده وب معنایی قرار می‌دهد. در واقع دو خزنده نقش جمع آوری کننده seed را برای هم ایفا می‌کنند. چارچوب خزش پیشنهادی برای این راه‌حل در شکل ۳-۴ نشان داده شده است. در این چارچوب منظور از خزنده RDF یک خزنده وب معنایی است که تنها اسناد وب معنایی را مورد خزش قرار می دهد.

اگر تبادل پیوند بین دو خزنده خارج از فرآیند خزش و به صورت دستی انجام شود، چارچوب پیشنهادی مشابه به چارچوب خزش در موتور جستجوی Swoogle است[Din06]، با این تفاوت که خزنده HTML در چارچوب پیشنهادی، یک خزنده متمرکز است که تلاش می‌کند با اولویت بندی پیوندهای HTML، دستیابی سریع تری به اسناد وب معنایی احاطه شده توسط آنها داشته باشد و درنتیجه آنها را سریعتر در اختیار خزنده RDF قرار دهد. همچنین، خزنده RDF در چارچوب پیشنهادی، بر خلاف خزنده RDF در چارچوب خزش Swoogle، اسناد HTML را خزش نمی‌کند بلکه آنها را در اختیار خزنده متمرکز HTML قرار می دهد.



شکل (۴-۳): چارچوب پیشنهادی خزش برای وب معنایی شامل یک خزنده RDF و یک خزنده متمرکز HTML

اما در چارچوب پیشنهادی، تبادل دستی پیوندها موجب تاخیر در پیشرفت فرآیند خزش نیز می‌شود. از این رو، برای انجام خودکار تبادل پیوند بین دو خزنده RDF و HTML، می‌توان دو خزنده را به صورت یک خزنده در نظر گرفت. در واقع در این حالت خزنده وب معنایی تبدیل به یک خزنده وب معنایی متمرکز می‌شود.

بنابراین، راه حل پیشنهادی دوم این است که توابع ارتباط معرفی شده در بخش‌های ۳-۱-۱ و ۳-۱-۲ به عنوان دانش پیش‌زمینه به یک خزنده وب معنایی استفاده شود تا خزنده با کمک آن بتواند پیوندها را تحلیل کرده و نوع آنها را قبل از مرحله واکشی پیش‌بینی کند. در واقع در این حالت خزنده وب معنایی تبدیل به یک خزنده وب معنایی متمرکز می‌شود که با دادن اولویت بالاتر به پیوندهای پیش‌بینی شده از نوع RDF و HTMLوالد RDF، می‌تواند دسترسی سریعتری به اسناد RDF داشته باشد و با عدم واکشی پیوندهایی که از نوع غیر معنایی پیش‌بینی می‌شوند، می‌تواند حجم واکشی

اسناد غیر معنایی را کاهش دهد.

الگوریتم خزشبرای این خزنده مشابه به خزنده متمرکز پیشنهادی است با این تفاوت که علاوه بر اسناد HTML، اسناد RDF نیز مورد خزش قرار می‌گیرند. بنابراین پیوندهای شروع می‌توانند از نوع RDFوHTML باشند. در مرحله واکشی نیز، اگر سند از نوع RDF باشد، ابتدا آدرس سند والد آن به بخش یادگیری داده می‌شود. سپس سند تجزیه شده و داده‌های معنایی بدست آمده از آن در مخزن ذخیره می‌شود.URIهای داخل سند نیز استخراج می‌شوند. اما در بخش تحلیل پیوند، تنها بر اساس پسوند فایل در مورد نوع پیوندهای استخراج شده از اسناد RDF قضاوت می‌شود. بدین ترتیب که:

۱. اگر پیوند دارای پسوند فایل خاص اسناد وب‌معنایی باشد و یا پسوند فایل نداشته باشد از نوع R پیش‌بینی می‌شود.
۲. اگر پیوند دارای پسوند فایل خاص اسناد HTML باشد، از نوع H پیش‌بینی می‌شود.
۳. در غیر این صورت، پیوند از نوع N پیش‌بینی شده و از آن صرف نظر می‌شود.

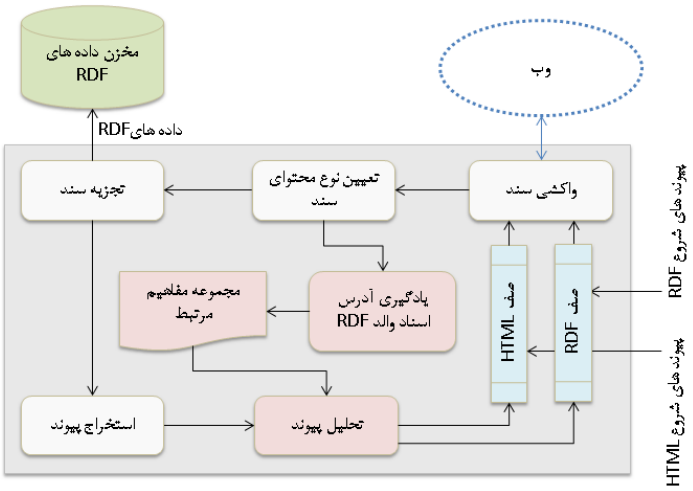
اما یکی از مزیت‌های راه‌حل اول این است که جداسازی خزندهRDF از خزنده HTML، باعث می‌شود تا صف پیوندهای RDF از پیوندهای HTMLمجزا باشد. جدا بودن صف پیوندهای RDF و HTML در طول فرآیند خزش وب معنایی، به افزایش انعطاف‌پذیری در پیاده‌سازی سیستم کمک می‌کند.در ادامه به چند مورد اشاره می‌شود:

- تفاوت در شیوه پیاده‌سازی صف‌ها: برای مثال، می‌توان صف RDF را مبتنی بر PLD اما صف HTML را مبتنی بر URI پیاده‌سازی کرد.
- تفاوت در معیارهای اولویت‌بندی: ملاک‌های اولویت خزش در بین پیوندهای RDF و پیوندهای های HTMLمی‌تواند با هم متفاوت باشد. پیوندهای هایHTML باید بر اساس میزان توانایی آنها در رسیدن به اسناد وب معنایی، در صف قرار گرفته و اولویت‌بندی شوند.

اما در مورد پیوندهای RDF، در صورت نیاز به اولویت‌بندی، این کار می‌تواند بر اساس میزان معنایی بودن آنها صورت بگیرد.

- امکان توزیع پویای منابع بین صف‌ها: از آنجایی که هدف خزنده، جمع‌آوری اسناد وب-معنایی است، می‌توان منابع بیشتری را به صف RDF اختصاص داد، برای مثال، تعداد نخ اجرایی و درحالت توزیع شده تعداد ماشین‌های بیشتر، صف حافظه بزرگ‌تر و غیره.

بر این اساس، معماری خزنده وب‌معنایی متمرکز با دو صف مجزا برای پیوندهای RDF و HTML، در شکل ۳-۵ نشان داده شده است. الگوریتم خزش برای این خزنده، مانند الگوریتم خزش در حالت تک صفه است با این تفاوت که پس از مرحله تحلیل پیوند، پیوندهای با بر چسب R در صف RDF و پیوندهای با برچسب H و HR در صف HTML قرار می‌گیرند.



شکل (۵-۳): معماری خزنده وب معنایی متمرکز با دوصف مجزا برای پیوندهای RDF و HTML

۳-۳- خلاصه فصل

در این فصل، یک خزنده متمرکزHTML پیشنهاد داده شد که از تحلیل پیوندهای استخراج شده از اسناد HTML برای دستیابی سریعتر به اسناد وب معنایی استفاده می‌کند. برای استفاده از

این خزنده پیشنهادی در فرآیند خزش وب معنایی دو راهکار ارائه شد. در راهکار اول از اسناد جمع-آوری شده توسط خزنده متمرکز به عنوان Seed برای یک خزنده وب معنایی استفاده می‌شود. در راهکار دوم توابع ارتباط ارائه شده برای خزنده متمرکز به عنوان دانش پیش‌زمینه در اختیار یک خزنده وب معنایی قرار داده می‌شود.

فصل ۴- پیاده‌سازی و ارزیابی

۴-۱- پیاده سازی

برای پیاده‌سازی سیستم پیشنهادی، از کتابخانه متن بازCrawler4j[۱]استفاده شده است.Crawler4j، یک خزنده وباست که به زبان برنامه نویسی Java نوشته شده است و امکان موازی-سازی فرآیند خزش به صورت چند نخی را نیز فراهم می‌کند. بخش‌هایی از کد خزندهCrawler4j،مطابق با نیازهای موردنظر تغییر داده شده است. افزودن بخش تحلیل پیوند، تغییر بخش‌های صافی واکشی و صافی پیوند،استفاده از پارسر Jena[۲]برای تجزیهاسناد وب معنایی و استخراج پیوندهای داخل آنها و استفاده از پارسر استنفورد[۳]برای استخراج کلمات و بررسی وجود کلمات مرتبط، از جمله این تغییرات هستند. همچنین، از کتابخانه تحلیل محتوای Apache Tika[۴]برای تعیین نوع واقعی اسناد استفاده شده است. فراداده‌های مربوط به نتایج خزش نیز،در پایگاه داده رابطه‌ای MySqlذخیره می‌شوند.

۴-۲- معیارهای ارزیابی

خزنده‌های متمرکز با کمک توابع ارتباط و بر پایه مجموعه‌ای از قوانین، نوع (برچسب) پیوند-ها را پیش‌بینی می‌کند که این پیش‌بینی‌هامی‌تواند درست و یا نادرست باشد. یکی از معیار ها برای ارزیابی پیش‌بینی‌های انجام شده توسط خزنده‌های متمرکز، معیار دقت است که از محاسبه نسبت

---

[۱] http://code.google.com/p/crawler4j/

[۲] http://www.apache.org/dist/jena/binaries/

[۳] http://nlp.stanford.edu/software/corenlps.html

[۴] http://tika.apache.org/

تعداد پیش‌بینی‌های درست به تعداد کل پیش‌بینی‌های انجام شده، بدست می آید. همچنین، برای نشان دادن عملکرد خزنده در دسترسی سریع‌تر به اسناد وب معنایی، از معیار نرخ دستیابی استفاده می‌شود که بیان کننده تعداد اسناد وب معنایی بدست‌آمده نسبتبه تعداد کل اسناد واکشی‌شده است.

البته باید توجه داشت که فقط اسنادی در محاسبه این دو معیار در نظر گرفته می‌شوند که با موفقیت از سطح وب واکشی شده باشند (یعنی کد وضعیت واکشی آنها برابر با ۲۰۰ باشد). چون در صورت عدم امکان واکشی (به دلیل فایل robot.txt)، وقوع خطا در مرحله واکشی و تغییر آدرس، نمی‌توان به محتوای سند دسترسی داشت و در نتیجه امکان تشخیص نوع واقعی آن وجود ندارد.

دقت (precision) و نرخ دستیابی (harvest) خزنده به ترتیب با کمک فرمول‌های ۴-۱ و ۴-۲ محاسبه می‌شود.

Commented [m7]: یک ارجاعی باید بدهید که اینها را کسان دیگر هم با همین تعاریف استفاده کرده اند.

$$\text{precision}_{\text{crawler}} = \frac{N_{LT}}{N_{LS}} \qquad (۴-۱)$$

$$\text{harvest}_{\text{crawler}} = \frac{N_{LT}}{N} \qquad (۴-۲)$$

- N: تعداد اسناد واکشی شده
- N_L: تعداد اسنادی که از نوع L پیش‌بینی شده‌اند. $L \in \{R, HR, H, N\}$
- N_{LS}: تعداد اسنادی کهاز نوع L پیش‌بینی شده و با موفقیت نیز واکشی شده‌اند.
- N_{LT}: تعداد اسنادی که به درستی از نوع L پیش‌بینی شده و با موفقیت نیز واکشی شده‌اند.

از سوی دیگر به دلیل آنکه توابع ارتباط پیشنهادی برمبنای مجموعه‌ای از قوانین عمل می‌کنند لازم است تا هر یک از قوانین نیز به طور جداگانه مورد ارزیابی قرار بگیرند [Xu07]. در این راستا، از معیارهای درصد اطمینان و درصد پشتیبانی استفاده می‌شود. درصد اطمینان یک قانون بیان کننده میزان پیش‌بینی‌های درستی است که با اعمال آن قانون انجام شده است. تعداد پیش‌بینی‌های درست بر مبنای یک قانون نسبت به کل تعداد پیش‌بینی‌هایی که توسط خزنده انجام شده است، نیز درصد

پشتیانی قانون را نشان می‌دهد. درصد اطمینان (confidence) و درصد پشتیبانی (support) هر قانون به ترتیب از فرمول‌های ۴-۳ و ۴-۴ محاسبه می‌شود.

$$\text{confidence}_{\text{rule}} = \frac{N_{RLT}}{N_{RLS}} \times 100 \quad (۴-۳)$$

$$\text{support}_{\text{rule}} = \frac{N_{RLT}}{N_{LT}} \times 100 \quad (۴-۴)$$

- N_{RLS}: تعداد اسنادی کهبا اعمال قانون R، از نوع L پیش‌بینی شده و با موفقیت نیز واکشی شده‌اند.
- N_{RLT}: تعداد اسنادی کهبا اعمال قانون R، به درستی از نوع L پیش‌بینی شده و با موفقیت نیز واکشی شده‌اند.
- N_{LT}: تعداد اسنادی که به درستی از نوع L پیش‌بینی شده و با موفقیت نیز واکشی شده‌اند.

از آنجایی که در مورد یک پیوند قوانین مختلفی قابل اعمال هستند، درصد اطمینان و پشتیبانی قوانین ممکن است با هم همپوشانی داشته باشد.

۴-۳- ارزیابی خزنده متمرکز پیشنهادی

همانطور که در فصل قبل بیان شد، در این پایان نامه یک خزنده متمرکز پیشنهاد داده شده است که از دو تابع ارتباط برای تحلیل پیوندهای استخراج شده از اسناد HTML و در نتیجه دستیابی سریعتر به اسناد وب معنایی احاطه شده توسط اسناد HTMLاستفاده می‌کند.سپس بر مبنای خزنده متمرکز پیشنهادی، دو چارچوب برای خزش وب معنایی ارائه شده است. بنابراین، موفقیت چارچوب-های پیشنهادی برای خزش وب معنایی، به موفقیت دو تابع ارتباط پیشنهادی در خزنده متمرکز بستگی دارد. از این رو در ادامه این بخش، عملکرد خزنده متمرکز پیشنهادی مورد ارزیابی عملی قرار گرفته است.

۴-۳-۱- محیط ارزیابی

برای ارزیابی خزنده متمرکز پیشنهادی لازم است تا سایت‌هایی برای خزش انتخاب شوند که علاوه بر اسناد HTML، شامل تعداد مناسبی اسناد وب معنایی نیز باشند. پس از بررسی دیتاست‌های منتشر شده بر روی ابر داده‌های پیوندی[۱] دو PLD برای ارزیابی انتخاب شده است که عبارتند از: openlylocal.com، dbtune.org. دو PLD انتخابی دیگر نیز عبارتند از: deri.org[Hog11] و rossettiarchive.org[Din06].

Commented [m8]: چرا نگفتید چهار تا. اگر این دو با اندو فرق می کنند باید بگویید

در انتخاب PLDها سعی شده است تا PLDهایی برای ارزیابی انتخاب شوند که از لحاظ ساختار گراف اسناد، ساختار آدرس‌ها و همچنین نوع محتوا با هم متفاوت باشند. این امر باعث می‌شود تا نتایج حاصل از ارزیابی دقیق‌تر و محدود به چندPLD خاص نباشد. در جدول ۴-۱ آدرس شروع برای خزش PLDهای انتخابینشان داده شده است.

جدول ۴-۱: PLD های انتخابی برای ارزیابی خزنده متمرکز پیشنهادی

شماره PLD	نام PLD	آدرس شروع
۱	dbtune.org	http://dbtune.org/
۲	openlylocal.com	http://openlylocal.com/
۳	deri.org	http://www.deri.ie/
۴	rossettiarchive.org	http://www.rossettiarchive.org/

۴-۳-۲- خزنده‌های مورد مقایسه

در آزمایش‌ها پنج خزنده با هم مقایسه شده‌اند که از نظر روش پیش‌بینی نوع پیوندها در مرحله استخراج پیوند و همچنین انتخاب سند در مرحله واکشی،با هم متفاوت هستند. هدف هر پنج خزنده، کشف اسناد وب معنایی احاطه شده توسط اسناد HTML است. هیچ یک از خزنده‌ها اسناد وب‌معنایی

[۱] http://datahub.io/group/lodcloud

را مورد خزش قرار نمی دهند.

خزنده ۱ (FC-R+HR): خزنده متمرکز پیشنهادی است که از تابع ارتباط $Relevancy_R$ (معرفی شده در بخش ۳-۱-۱) برای پیش‌بینی پیوندهای RDF و از تابع ارتباط $Relevancy_{HR}$ (معرفی شده در بخش ۳-۱-۲) برای پیش‌بینی پیوندهای والد RDF استفاده می‌کند.

خزنده ۲ (BC): یک خزنده اول-سطح است که سیاست انتخاب آن مشابه به خزنده‌های وب معنایی موجود است. این خزنده در مرحله استخراج پیوند، از پیوند ها با پسوند فایل غیر معنایی شناخته شده صرف نظر می کند. در مرحله واکشی نیز بر اساس پسوند فایل و فیلد نوع محتوای خاص اسناد وب معنایی و اسناد HTML، آنها را شناسایی کرده و از سایر اسناد صرف نظر می‌کند [Hog11, Din06 Cyg11,Ise10,Ore08, Che09 , Hae06]. هدف از مقایسه این خزنده با خزنده متمرکز پیشنهادی، نشان دادن تاثیر تحلیل پیوندهای استخراج شده از اسنادHTML و اولویت بندی آنها در میزان دستیابی به اسناد وب معنایی است.

خزنده ۳ (FC-R): این خزنده همان خزنده متمرکز پیشنهادی است با این تفاوت که پیوند-های والدRDF را پیش‌بینی نمی‌کند. مقایسه این خزنده با خزنده متمرکز پیشنهادی،میزان تاثیر پیش‌بینی پیوندهای والد RDFرا بر سرعت دستیابی خزنده به اسناد وب معنایی نشان می‌دهد.

خزنده ۴ ($FC\text{-}R_2$): همانطور که در بخش ۲-۳-۶ بیان شد، در مقاله [Umb09] یک تابع ارتباط ارائه شده است که نوع رسانه پیوندها را بر اساس کلمات داخل آدرس آنها پیش‌بینی می‌کند. خزنده ۴ از این تابع برای پیش‌بینی RDF بودن یک پیوند استفاده می‌کند. هدف از این ارزیابی، مقایسه عملکرد و میزان موفقیت تابع ارتباط پیشنهادی $Relevancy_R$ نسبت به تابع ارتباط مقاله [Umb09] در پیش‌بینی پیوندهای RDF است.

خزنده ۵ ($FC\text{-}R+HR_2$): خزنده ۵ از تابع ارتباط $Relevancy_R$برای پیش‌بینی RDF بودن

یک پیوند و از فرمول مشابهت آدرس پیوندها که در مقاله [Hat10] ارائه شده است برای پیش‌بینی والدهای RDF استفاده می‌کند (توضیحات بیشتر در مورد این تابع ارتباط در بخش ۲-۳-۲ بیان شده است). هدف از این ارزیابی، مقایسه عملکرد و میزان موفقیت تابع ارتباط پیشنهادی$Relevancy_{HR}$ نسبت به تابع ارتباط مقاله[Hat10] در پیش‌بینی پیوندهای والدRDF است.

در جدول ۴-۲ سیاست انتخاب پیوند و انتخاب سند برای هر یک از خزنده‌های مورد مقایسه به صورت خلاصه آورده شده است.

- CTF: صافی واکشی فیلد نوع محتوا در مرحله واکشی
- EXF: صافی واکشی پسوند فایل در مرحله واکشی
- NF: صافی پیوند برای حذف پیوندها با پسوند فایل غیر معنایی شناخته شده، مانند عکس، ویدئو و غیره

جدول (۲-۴): خزنده‌های مورد مقایسه برای ارزیابی خزنده متمرکز پیشنهادی

Commented [m9]: بناشد ترتیب را عوض کنید که دید بهتری بدهد

نام خزنده	نوع خزش	سیاست انتخاب پیوند	سیاست انتخاب سند
BC	اول سطح	صافی پیوندNF	صافی واکشی CTF و EXF
FC-R+HR (خزنده پیشنهادی)	اول بهترین	صافی پیوند NF- تابع ارتباط $Relevamcy_R$برای پیش‌بینی پیوندهای RDFو تابع ارتباط$Relevancy_{HR}$برای پیش‌بینی پیوندهای والد RDF	برچسب - صافی واکشی CTF
FC-R	اول بهترین	صافی پیوندNF- تابع ارتباط $Relevamcy_R$برای پیش-بینی پیوندهای RDF	صافی واکشیCTF
$FC\text{-}R_2$	اول بهترین	صافی پیوند NF- تابع ارتباط [Umb09] برای پیش-بینی پیوندهای RDF	برچسب - صافی واکشیCTF
$FC\text{-}R+HR_2$	اول بهترین	صافی پیوند NF–تابعارتباط$Relevancy_R$ برای پیش-بینی پیوندهای RDF و تابع ارتباط [Hat10] برای پیش‌بینی پیوندهای والد RDF	برچسب - صافی واکشیCTF

۳-۳-۴- روش ارزیابی

روش ارزیابی بدین صورت است که PLDهای معرفی شده در بخش ۴-۲، توسط هر خزنده

مورد خزش قرار می‌گیرد و سپس نتایج حاصل از خزش آنها با هم مقایسه می‌شود. برای بررسی دقیق‌تر نتایج حاصل از خزش، هر PLD به صورت جداگانه خزش می‌شود. همچنین، با توجه به نامحدود بودن فضای وب و محدودیت های پهنای باند و شبکه، برای محدود کردن محیط ارزیابی، فرض‌های زیر در نظر گرفته شده است:

- از هر PLD حداکثر ۲۰۰۰سند واکشی می‌شود.
- از هر سند حداکثر ۵۰۰ پیوند برای خزش انتخاب می‌شود.(۵۰۰ پیوند اول در صورت وجود)
- پیوندهای خارج از PLDتنها در صورتی خزش می‌شوند که والد آنها داخل PLD باشد.
- هر خزنده با ۵ نخ اجرا می‌شود.

۴-۳-۴- تحلیل نتایج

پس از خزش PLDهای انتخابی توسط پنج خزنده، برای بدست آوردن آمار کلی از فضای خزش شده، گراف خزش پنج خزنده به ازای هر PLD با ادغام شده است. جدول ۴-۳، نتایج حاصل از وضعیت واکشی اسناد را در گراف‌های خزش ادغام شدهنشان می‌دهد.

جدول (۳-۴): نتایج حاصل از وضعیت واکشی اسناد در PLDهای خزش شده

PLD	تعداد کل اسناد	واکشی موفق		تغییر مسیر		عدم امکان واکشی		خطا در واکشی	
		تعداد	درصد	تعداد	درصد	تعداد	درصد	تعداد	درصد
۱	۷۲۳۰	۱۰۶۳	٪۱۵	۵۴۵۸	٪۷۶	۲۵	٪۱	۶۸۴	٪۱۰
۲	۷۷۴۸	۶۸۴۰	٪۸۹	۱۹۷	٪۳	۱۱	٪۱	۷۰۰	٪۱۰
۳	۴۰۸۰	۳۴۴۸	٪۸۵	۳۶۶	٪۹	۳۷	٪۱	۲۲۹	٪۶
۴	۶۰۵۳	۵۸۷۶	٪۹۸	۳۲	٪۱	۵	٪۱	۱۴۴	٪۳
مجموع	۲۵۱۱۱	۱۷۲۲۷	٪۶۹	۶۰۵۳	٪۲۵	۷۴	٪۱	۱۷۵۷	٪۷

از خزش PLDهای انتخابی، در مجموع ۲۵۱۱۱ سند منحصر بفرد واکشی شده است که در

بین آنها، ۲۹۱۵ سند RDF و ۲۸۱۲ سند HTMLوالد RDFوجود دارد.

جدول۴-۴، نتایج بدست آمده از خزش فضای انتخابی توسطدو خزندهBC و خزنده پیشنهادی (FC-R+HR)را نشان می‌دهد. پارامترهای تعریف شده در این جدول عبارتند از:

- N:تعداد کل اسناد واکشی شده
- N_{RDF}: تعداد اسناد وب معنایی بدست آمده
- N_{op}: تعداد اسناد واکشی شده مشترک با اسناد واکشی شده توسط خزنده پیشنهادی
- N_{RDFop}: تعداد اسناد واکشی شدهمشترک با اسناد وب معنایی بدست آمده توسط خزنده پیشنهادی

جدول (۴-۴): آمار حاصل از خزش فضای انتخابی توسط خزنده BC و خزنده FC-R+HR

PLD	N	خزنده FC-R+HR	خزنده BC		
		N_{RDF}	N_{RDF}	N_{op}	N_{RDFop}
۱	۲۰۰۰	۸۹	۱۷	۳۷۸	۱۹
۲	۲۰۰۰	۷۲۶	۲	۹۵	۲
۳	۲۰۰۰	۳۲۶	۲	۱۴۴۰	۹۹
۴	۲۰۰۰	۹۷۵	۴	۱۰۲۷	۴
مجموع	۸۰۰۰	۲۱۱۶	۲۵	۲۹۴۰	۱۲۴
میانگین	۲۰۰۰	۵۲۹	۶	۷۳۵	۳۱

مطابق با جدول ۴-۳، در خزنده پیشنهادی (FC-R+HR) از مجموع هشت هزار سند واکشی شده، حدود ۲۶ درصد آنها از نوع وب معنایی هستند. اما خزنده BC که سیاست انتخاب اسناد آن مشابه به خزنده‌های وب معنایی موجود است، به تعداد بسیار کمی از اسناد وب معنایی دست پیدا کرده است. این در حالی است که از بین ۲۱۱۶ سند وب معنایی بدست آمده توسط خزنده پیشنهادی،۱۷۸۲ سند (۸۴ درصد) داری پسوند فایل و یا فیلد نوع محتوای خاص اسناد وب معنایی هستند. بنابراین این اسناد نیز می‌توانند توسط خزنده BC شناسایی شوند. اما خزنده پیشنهادی با پیش‌بینی این اسناددر مرحله استخراج پیوند و دادن اولویت بالاتر به آنها توانسته است دسترسی

سریع‌تری به آنها داشته باشد.

نکته قابل توجه دیگر این است که در PLD3، اسناد خزش شده توسط خزنده BC، درصد همپوشانیخوبی (حدود ۳۰ درصد) بااسناد وب معنایی بدست آمده توسط خزنده پیشنهادی دارند. با این وجود،خزنده BC تنها ۲ سند وب معنایی را شناسایی کرده است. علت این امر آن است کهاسناد وب معنایی در PLD3، دارای پسوند فایل ".xml" هستند ومقدار فیلد نوع محتوای آنها برابر با "application/xml" است. به همین دلیل خزنده BC حتی در صورت دسترسی به این اسناد، باز هم آنها را از نوع وب معنایی در نظر نمی‌گیرد.

در اینجا اگر صافی واکشی در خزنده BC به گونه‌ای تنظیم شود که اسناد نوع XML نیز از آن عبور کنند، مشکل خزنده می‌تواند برای شناسایی اسناد وب معنایی درPLD3 حل شود. اما این کارموجب افزایش حجم واکشی اسناد غیر معنایی و در نتیجه دستیابی دیرتر به اسناد وب معناییمی-شود؛ چرا کهدر گراف خزش ادغام شده،از مجموع ۲۵۱۱۱ سند واکشی شده، حدود ۲۹۷۸ سند دارای پسوند فایل یا فیلد نوع محتوای اسناد XML هستند و از میان آنها تنها ۳۲۶ سند (۱۰ درصد) از نوع واقعی RDF هستند.

این در حالی است که خزنده متمرکز پیشنهادی با این مشکل روبرو نمی شود؛ چونخزنده از بین پیوندهای با پسوند ".xml" تنها پیوندهایی را انتخاب می‌کند که از نوع RDF پیش‌بینی می‌شوند و از سایر آن ها صرف‌نظر می‌کند. برای مثال در PLD3، مسیر فایل در آدرس بیشتر پیوندهای از نوع XML که نوع واقعی آنها RDF است، دارای کلمهfoafاست؛ به همین دلیل خزنده پیشنهادی آنها را از نوع RDF پیش‌بینی می‌کند.

اما روش دیگر برای مقایسه خزش اول-سطح با خزش متمرکز (اول-بهترین)، بررسی چگونگی توزیع اسناد واکشی شده در سطوح مختلف گراف خزش است. برای این منظور در جدول ۴-۵، تعداد

اسناد واکشی شده (N) و همچنین تعداد اسناد RDF بدست آمده (N_{RDF}) در خزنده متمرکز پیشنهادی، خزنده BC و گراف خزش ادغام شده به ازای PLD4 مورد بررسی قرار گرفته است. خزنده BC، سطح به سطح، گراف اسناد در PLD4را مورد خزش قرار داده است و تا اسناد موجود در یک سطح تمام نشده است به سطح بعدی نرفته است. بنابراین خزنده BCدر پایان عمق ۳، فقط به ۳۵ سند وب معنایی دست پیدا خواهد کرد. اما خزنده متمرکز پیشنهادی با پیشروی همزمان در عمق ۳ و ۴ تلاش کرده است تا به ازای ۲۰۰۰ سند واکشی شده، به اسناد وب معنایی بیشتری دست پیدا کند.

جدول (۵-۴): تعداد اسناد واکشی شده در سطوع مختلف گراف خزش به ازای PLD4

	گراف خزش ادغام شده		خزنده BC		خزنده FC-R+HR	
عمق	N	N_{RDF}	N	N_{RDF}	N	N_{RDF}
۰	۱	۰	۱		۱	۰
۱	۲۰	۰	۲۰	۰	۲۰	۰
۲	۱۳۳	۰	۱۳۳	۰	۱۲	۰
۳	۲۰۴۳	۳۵	۱۸۴۶	۴	۹۸۹	۴
۴	۲۶۶۵	۱۱۹۱	۰	۰	۹۷۸	۹۷۱
۵	۶۰۵	۱	۰	۰	۰	۰
۶	۵۰۸	۰	۰	۰	۰	۰
۷	۴۷	۰	۰	۰	۰	۰
۸	۲۸	۰	۰	۰	۰	۰
۹	۳	۰	۰	۰	۰	۰
مجموع	۶۰۵۳	۱۲۲۷	۲۰۰۰	۴	۲۰۰۰	۹۷۵

جدول ۴-۶دقت خزنده متمرکز پیشنهادی را در پیش‌بینی نوع پیوندها به ازای هر PLD نشان می‌دهد. پارامترهای داخل جدول در بخش ۴-۲ توضیح داده شده‌اند.

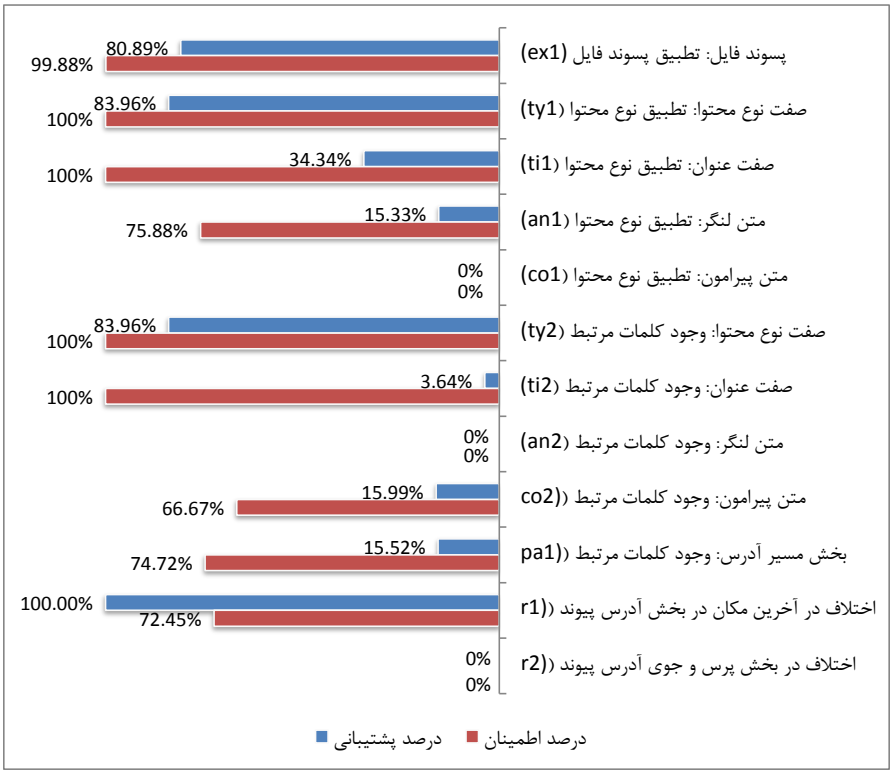
جدول (۶-۴): دقت خزنده متمرکز پیشنهادی به تفکیک نوع پیوندهای پیش‌بینی شده

	R				HR				N			
PLD	N_L	N_{LS}	N_{LT}	دقت	N_L	N_{LS}	N_{LT}	دقت	N_L	N_{LS}	N_{LT}	دقت
۱	۳۷۵	۱۵۷	۸۹	٪۵۶,۹۶	۲۷۸	۹۸	۸۹	٪۹۰,۸۲	۱۰	۶	۵	٪۸۳,۳۳

۲	۷۲۶	۷۲۶	۷۲۶	٪۱۰۰	۱۲۲۵	۱۲۲۵	٪۵۶,۵۷	۸۶۰	۶۵۰	۶۴۷	٪۹۹,۵۴
۳	۳۲۴	۴۳۰	۴۳۷	٪۷۵,۳۵	۱۸۲	۱۷۹	٪۹۸,۸۸	۶۶	۸	۵	٪۶۲,۵۰
۴	۹۷۵	۹۷۵	۹۸۲	٪۱۰۰	۹۸۵	۹۸۲	٪۹۸,۸۸	۹۸۶	۴۵۱	۴۵۱	٪۱۰۰
میانگین				٪۸۳,۰۸			٪۸۶,۲۹				٪۸۶,۳۴

بر اساس نتایج بدست آمده در جدول ۴-۶، خزنده متمرکز پیشنهادی به طور میانگین دقت بالایی در پیش‌بینی اسناد RDF (R)، والد RDF (HR) و همچنین اسناد غیر معنایی (N) دارد. این پیش‌بینی‌های درست به خزنده کمک می‌کند تا با دادن اولویت بالاتر به اسناد RDF و والد RDF، دسترسی سریعتری به آنها داشته باشد. همچنین خزنده با حذف اسناد غیر معنایی، تنها در مسیرهایی خزش می‌کند که احتمال رسیدن به اسناد وب معنایی در آنها وجود دارد. این امر، ضمن صرفه جویی در مصرف منابع، موجب دسترسی سریعتر خزنده به اسناد وب معنایی نیز می‌شود.

در شکل ۴-۱ میانگین درصد اطمینان و پشتیبانی قوانین اعمال شده توسط توابع ارتباط $Relevancy_R$ و $Relevancy_{HR}$ نشان داده شده است.

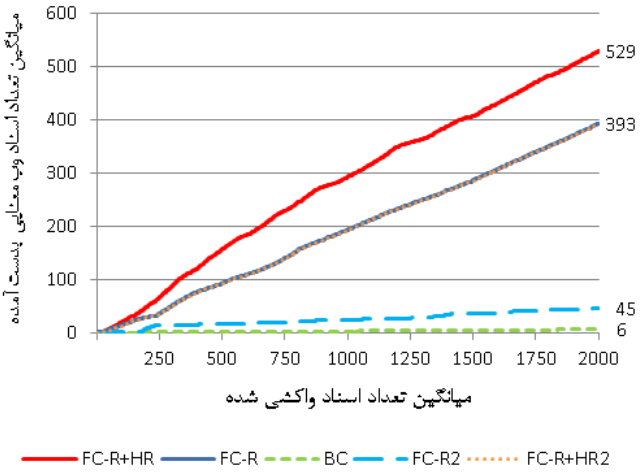


شکل (۱-۴): میانگین درصد پشتیبانی و اطمینان هریک از قوانین در تابع ارتباط خزنده متمرکزپیشنهادی

همه قوانین قابل اعمال در توابع ارتباط پیشنهادی دارای ضریب اطمینان بالای ۶۵درصد هستند. نکته قابل توجه، درصد پشتیبانی r1 در تابع ارتباط $Relevancy_{HR}$ است که توانسته است همه پیوند های والد RDF را تحت پوشش قرار دهد. این موضوع نشان می‌دهد که در PLDهای انتخابی، اسناد والد RDF در یک مسیر یکسان قرار دارند؛ به همین دلیل خزنده متمرکز پیشنهادی توانسته است با اعمال قانون r1 آنها را به درستی پیش‌بینی کند.

شکل ۴-۲ نمودار میانگین نرخ دستیابی به اسناد وب معنایی را به ازای PLDهای خزش شده نشان می‌دهد. خزنده پیشنهادی در مقایسه با سایر خزنده‌ها، نرخ دستیابی بالاتری به اسناد وب‌معنایی دارد. علت آن هم این است که استفاده از توابع ارتباط پیشنهادی به خزنده کمک می‌کند تا بتواند

اسناد غیرمعنایی را سریعتر، یعنی قبل از مرحله واکشی شناسایی کرده و از آنها صرف‌نظر کند. همچنین پیش‌بینی صحیح نوع محتوای اسناد وب معنایی در مرحله تحلیل پیوند و دادن اولویت بالاتر به آنها، باعث شده است تا خزنده آنها را سریعتر از اسناد دیگر واکشی کند.



شکل (۲-۴): میانگین نرخ دستیابی به اسناد وب معنایی درپنج خزنده مورد مقایسه برای ارزیابی خزنده متمرکز پیشنهادی

برای نشان دادن میزان اهمیت تابع ارتباط $Relevancy_{HR}$، نرخ دستیابی خزنده پیشنهادی با خزنده FC-R مقایسه شده است. خزنده متمرکز FC-R نسبت به خزنده متمرکز پیشنهادی، به دلیل عدم پیش بینی والدهای RDF نرخ دستیابی پایین‌تری به اسناد وب معنایی دارد.

برای ارزیابی میزان موفقیت تابع ارتباط پیشنهادی$Relevancy_{HR}$در پیش‌بینی پیوندهای والدRDF، خزنده پیشنهادی با خزنده $FC\text{-}R+HR_2$ مقایسه شده است (خزنده $FC\text{-}R+HR_2$ از تابع ارتباط مقاله[Hat10] برای پیش‌بینی والدهای RDF استفاده می‌کند).اما نرخ دستیابی به اسناد وب-معنایی در خزنده $FC\text{-}R+HR_2$ مشابه به نرخ دستیابی در خزنده FC-R است. چون خزنده $FC\text{-}R+HR_2$ هیچ از یک پیوندهای والد RDF را پیش‌بینی نکرده است. علت این موضوع در ادامه توضیح

داده شده است.

همانطور که در بخش ۲-۳-۲ بیان شد، بر اساس تابع ارتباط در مقاله [Hat10]، دو پیوند در صورتی مشابه در نظر گرفته می‌شوند که فاصله آنها (dURL) کمتر یا مساوی یک باشد. این حالت در صورتی اتفاق می افتد کهآدرس دو پیوند تنها در بخش پرس‌وجو با هم متفاوت باشد. اما هیچ یک از پیوندهای والد RDF در PLDهای انتخابی دارای چنین شرطی نیستند. این موضوع در شکل ۴-۱ نیز نشان داده شده استکه در آن درصد پشتیبانی قانون (r2) برای تابع ارتباط RelevancyHR برابر با صفر است؛ چون قانون r2 نیز دقیقا همین شرط را برای مشابه بودن دو پیوند بیان می‌کند.

از سوی دیگر، برای ارزیابی میزان موفقیت تابع ارتباط پیشنهادی$Relevancy_R$در پیش‌بینی پیوندهایRDF، خزنده متمرکز پیشنهادی با خزنده $FC\text{-}R_2$ مقایسه شده است. خزنده $FC\text{-}R_2$ از تابع ارتباط معرفی شده در مقاله [Umb09]برای پیش‌بینی RDF بودن یک پیوند استفاده می‌کند. اما خزنده $FC\text{-}R_2$ نتوانسته است هیچ یک از پیوندهای نوع RDF را در مرحله استخراج پیوند پیش‌بینی کند. این موضوع ناشی از دو دلیل زیر این است:

۱. در خزنده $FC\text{-}R_2$ به ازای هر سند RDF که در مرحله واکشی بدست می آید، مجموعه ای از قوانین بر مبنای آدرس سند به پایگاه دانش خزنده افزوده می‌شود. اما تا قبل از اینکه یک سند RDFتوسط خزنده واکشی شود، پایگاه دانش خزنده دارای هیچ قانونی نیست؛بنابراین خزنده مانند خزنده اول-سطح (خزنده BC) عمل می‌کند و در نتیجه دستیابی بسیار کندی به اسناد RDF دارد.

۲) پس از دستیابی به اسناد RDF، در صورتی یک قانون قابل اعمال خواهد بود که ضریب اطمینان آن بالای ۵۰ درصد باشد؛ یعنی آن قانون حداقل در مورد نیمی از اسناد RDF که توسط خزنده واکشی شده‌اند، برقرار باشد. اما به دلیل اشتراک زیاد بین کلمات موجود در آدرس اسناد RDF

با سایر آدرس ها، ضریب اطیمنان قوانین به بالای ۵۰ درصد نمی‌رسد.

اما خزنده $FC\text{-}R_2$نرخ دستیابی بالاتری به اسناد وب معنایی نسبت به خزنده اول سطح BCدارد، چوندر صورتی که هیچ یک از قوانین در مورد یک پیوند برقرار نباشد و پسوند فایل آن نیز خاص اسناد HTMLنباشد، خزندهFC-R آن پیوند را از نوع غیر معنایی (N) در نظر می‌گیرد و از خزش آن صرف‌نظر می‌کند. بنابراین خزنده با عدم پیشروی در بسیاری از مسیر ها دسترسی سریعتری به بعضی از اسناد RDFنسبت به خزنده اول سطح دارد. این در حالی است که دقت خزنده $FC\text{-}R_2$در تشخیص اسناد غیر معنایی ۴۰ درصد است و در بین ۲۲۳ سندیکه خزنده آنها را غیر معنایی تشخیص داده است و با موفقیت واکشی شده‌اند،۱۳۴ سند RDFوجود دارد.

۴-۴- ارزیابی چارچوب‌های خزش پیشنهادی برای وب معنایی

۴-۴-۱- محیط ارزیابی

۴-۴-۲- خزنده‌های مورد مقایسه

۴-۴-۳- روش ارزیابی

۴-۴-۴- تحلیل نتایج

۴-۵- خلاصه فصل

فصل ۵- نتیجه گیری و کارهای آینده

مراجع

Base–SemanticWeb

[Ber01] T. Berners-Lee, J. Hendler, O. Lassila, "The Semantic Web", Scientific American, 35-43, 2001.

Base–Crawling

[Dhe11] S. S. Dhenakaran, K. T. Sambanthan, “WEB CRAWLER - AN OVERVIEW.” International Journal of Computer Science and Communication, vol. 2, pp. 265-267, Jun 2011.

[Pan04]G. Pant,P. Srinivasan , F. Menczer, "crawling the web",2004

Swoogle

[Din06] L. Ding, *Enhancing Semantic Web Data Access*, Ph.D. dissertation,Faculty of the Graduate School, University of Maryland, USA, 2006.

[Han06] L. Han, L. Ding, R. Pan, T. Finin, Swoogle's Metadata about the Semantic Web, 2006.

BioCrawler

[Bat12]A. Batzios, P. A. Mitkas, "WebOWL: A Semantic Web search engine development experiment", *Journal of Expert Systems with Applications*, 39, 5052–5060, 2012.

[Bat07] A. Batzios, C. Dimou, A. L. Symeonidis, P. A. Mitkas,"BioCrawler: An intelligent crawler for the Semantic Web",*Journal of Expert Systems with Applications*, 35, 524-530, 2007.

SWSE

[Hog11]A. Hogan, A. Harth, J. Umbrich, S. Kinsella, A. Polleres, S. Decker ,"Searching and Browsing Linked Data with SWSE: the SemanticWeb Search Engine", *Journal of Web Semantics*, 9, 365-401, 2011.

Falcons

[Che09] G. Cheng, Y. Qu, “Searching Linked Objects with Falcons: Approach, Implementation and Evaluation.” International Journal on Semantic Web and

Information Systems, vol. 5, pp. 50-71, Sep. 2009.

Watson

[Sab07] M. Sabou, C. Baldassarre, L. Gridinoc, S. Angeletou, E. Motta, M. d'Aquin, M. Dzbor, "WATSON: A Gateway for the Semantic Web," in ESWC poster session, 2007.

Sindice

[Cyg11] R. Cyganiak, D1.1 Deployment of Crawler and Indexer Module, Linking Open Data Around The Clock (LATC) Project, 2011.

[Ore08] E. Oren, R. Delbru, M. Catasta, R. Cyganiak, H. Stenzhorn, G. Tummarello, "Sindice.com: A document-oriented lookup index for open linked data." International Journal Metadata Semant and Ontologies, vol. 3, pp. 37-52, 2008.

Slug

[Dod06] L. Dodds, Slug: A Semantic Web Crawler, 2006.

MultiCrawler

[Har06] A. Harth, J. Umbrich, S. Decker, "Multicrawler: A pipelined architecture for crawling and indexing semantic web data," In 5th International Semantic Web Conference, 2006, pp. 258–271.

LDSpider

[Ise10] R. Isele, J. Umbrich, C. Bizer, A. Harth, "LDSpider: An open-source crawling framework for the Web of Linked Data," In Poster. ISWC2010, Shanghai, Chinam, 2010.

Sewer

[Las12] Lasek I., Klimpera O., Vojtas P., 2012: Scalable Framework for Semantic Web Crawling, In Proceedings of 7th Workshop on Intelligent and Knowledge Oriented Technologies, Smolenice, Slovakia

Focsed for SemanticWeb

[Mae08]F. V. D. Maele, P.Spyns, R. Meersman, "An Ontology-based Crawler for the Semantic Web", *OTM Workshops*, Monterrey/Mexico, 1056–1065, 2008.

[Mae06] F. V. D. Maele. "Ontology-based Crawler for the Semantic Web." M.A. thesis, Department of Applied Computer Science, Brussel, 2006.

[Ehr03] M. Ehrig, A. Maedche, “Ontology-focused crawling of Web documents,” in Proc. of the 2003 ACM Symposium on Applied Computing, 2003, pp. 1174-1178.

Focsed for MimeType

[Umb08] J. Umbrich, A. Harth, A. Hogan, S. Decker, "Four heuristics to guide structured content crawling", *8st International Conference on Web Engineering*, Westchester/New York, 2008.

[Umb09]J. Umbrich, M. Karnstedt, A. Harth, "Fast and Scalable Pattern Mining for Media-Type Focused Crawling", *Workshop on Knowledge Discovery, Data Mining, and Machine Learning*,Darmstadt/Germany, 2009.

Focused - Url Distance

[Hat10]D. Hati, A. Kumar, "UDBFC-An effective focused crawling approach based on URI Distance calculation", *3rd IEEE International Conference onComputer Science and Information Technology*, Chengdu, 2010.

Focused

[Dil00] M. Diligenti, F. Coetzee, S. Lawrence, C. L. Giles, M. Gori, “Focused crawling using context graphs,” in Proc. of 26th International Conference on Very Large Databases, 2000, pp. 527–534.

[Bat09] S. Batsakis, E.G.M. Petrakis, E. Milios, "Improving the performance of focused web crawlers", Journal of Data & Knowledge Engineering, 68, 1001-1013, 2009.

[Cha99] S. Chakrabarti, M. V. D. Berg, B. Dom, "Focused crawling: a new approach to topic-specific web resource discovery", Journal of Computer Networks, 31, 1623-1640, 1999.

[Jal11] O. Jaliian, H. Khotanlou, “A New fuzzy-Based Method to Weigh the Related Concepts in Semantic Focused Web Crawlers,” IEEE Conference, 2011.

[Kum12] R. K. Rana, N. Tyagi, “A Novel Architecture of Ontology-based Semantic Web Crawler.” International Journal of Computer Applications, vol. 44, Apr. 2012.

[Yuv06] M. Yuvarani, N. Ch. S. N. Iyengar, A. Kannan, “LSCrawler: A Framework for an Enhanced Focused Web Crawler based on Link Semantics,” in Proc. of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence, 2006.

[Hat10] Hati, D., Sahoo, B., Kumar, A., "Adaptive Focused Crawling Based on Link Analysis", 2nd IEEE International Conference on Education Technology and Computer (ICETC), Shanghai, China, pp. 455-460, 2010.

[Rav09] Ravakhah M., Kamyar M., 2009: Semantic Similarity Based Focused Crawling, First IEEE International Conference on Computational Intelligence Communication Systems and Networks, Pages: 448 - 453

[Zhe08] Zheng H.T., Kang B.Y., Kim H.G., 2008: An ontology-based approach to

learnable focused crawling, International Journal of Information Sciences, Vol. 178, Pages 4512-4522

[Her98] Michael Hersovicia, Michal Jacovia Yoelle S. Maarek, Dan Pellegb Menachem Shtalhaima, and Sigalit Ur, The shark-search algorithm — An application: tailored Web site mapping, BM Haifa Research Laboratory, Technion, Haifa, Israel.1998

[Hat10-2]D. Hati, A. Kumar,Improved Focused Crawling Approach for Retrieving Relevant Pages Based on Block Partitioning, 2nd IEEE International Conference on Education Technology and Computer (ICETC) , Shanghai, 2010.

[Sun08] Y. Sun, P. Jin, L. Yue, A Framework of a Hybrid Focused Web Crawler, Second IEEE International Conference on Future Generation Communication and Networking Symposia,Sanya, 2008

[Li05] J. Li, Kazutaka Furuse, Kazunori Yamaguchi,Focused Crawling by Exploiting Anchor Text Using Decision Tree, WWW 2005, Chiba, Japan, 2005.

[Xu07] Qingyang Xu, Wanli Zuo,First-order Focused Crawling, ACM, WWW 2007, Banff, Alberta, Canada, Poster Paper, 2007

[Cha02]S. Chakrabarti , K. Punera , M. Subramanyam, Accelerated focused crawling through online relevance feedback,bProceeding WWW '02 Proceedings of the 11th international conference on World Wide Web, 2002

[Agg01] C. C. Aggarwal, F. Al-Garawi, P. S. Yu, intelligent Crawling on the World Wide Web with Arbitrary predicates, Proceeding WWW '01 Proceedings of the 10th international conference on World Wide Web,2001.