

بسمه تعالی

گزارش سمینار کارشناسی ارشد

بازیابی اطلاعات برای وب معنایی Information retrieval for Semantic web

مهدی طالبیان، Talebian73@yahoo.com

استاد راهنما: دکتر شمس، f_shams@sbu.ac.ir

چکیده

با ظهور وب، داده ها و اطلاعات زیادی در موضوعات مختلف در اختیار همگان قرار گرفته است. این داده ها که اکثراً در قالب فایل های HTML ارائه شده اند، فرصت مناسبی را در اختیار کاربران وب قرار داده تا بتوانند با جستجوی کلمات کلیدی مناسب به داده های مورد نیاز خود دست یابند. اما امروزه با وجود انبوهی از اطلاعات گوناگون در وب، جستجو بر اساس کلمات کلیدی، ما را به تعداد زیادی صفحه وب می رساند که یافتن اطلاعات مورد نیاز در آن دشوار و زمانبر است. وب معنایی جهت رفع مشکلاتی از این قبیل به عنوان نسل جدید وب جاری، توسط آقای تیم برنرز لی [۱] معرفی شد.

وب معنایی به عنوان شاخه جدیدی از وب جاری لازم دارد تا به اطلاعات موجود در وب دسترسی داشته باشد. اما اطلاعات موجود در وب جهت نمایش آن به انسان ها سازماندهی شده اند. لذا برای استخراج اقلام اطلاعاتی از ابزاری بنام Wrapper استفاده می شود. در این نوشته بر استخراج از منابع اطلاعاتی^۱ تمرکز می کنیم و استخراج اطلاعات مبتنی بر یک Crawler که با استفاده از یک آنتالوژی دامنه به شناسایی اقلام اطلاعاتی می پردازد را پیشنهاد می کنیم.

کلمات کلیدی: وب معنایی، حاشیه گذاری معنایی، آنتالوژی، RDF، بازیابی اطلاعات، Wrapper.

فهرست مطالب

۱ - مقدمه	۴
۲ - معرفی وب معنایی	۸
۲-۱ - مشکلات وب موجود	۹
۲-۲ - نیاز به بازنمایی دانش	۱۰
۲-۳ - آشنایی با اجزای اساسی وب معنایی	۱۲
۲-۴ - معماری وب معنایی	۱۴
۲-۵ - آشنایی با خانواده استاندارد های XML	۱۷
۲-۶ - آشنایی با RDF (Resource Description Framework)	۲۱
۲-۷ - آشنایی با RDF Schema (RDFS)	۲۴
۲-۸ - معرفی Taxonomy ها	۲۶
۲-۹ - معرفی طیف آنتالوژی	۳۰
۲-۱۰ - معرفی آنتالوژی	۳۴
۲-۱۱ - توصیف منطقی آنتالوژی ها	۳۵
۲-۱۲ - سطوح بازنمایی دانش	۳۷
۲-۱۳ - مشکل نگاشت معنایی	۳۸
۲-۱۴ - خلاصه مطالب	۳۹
۳ - مدیریت دانش و بازیابی اطلاعات	۴۱
۳-۱ - بررسی مشکل	۴۱
۳-۲ - نقش آنتالوژی ها	۴۲
۳-۳ - معرفی معماری Sesame	۴۳

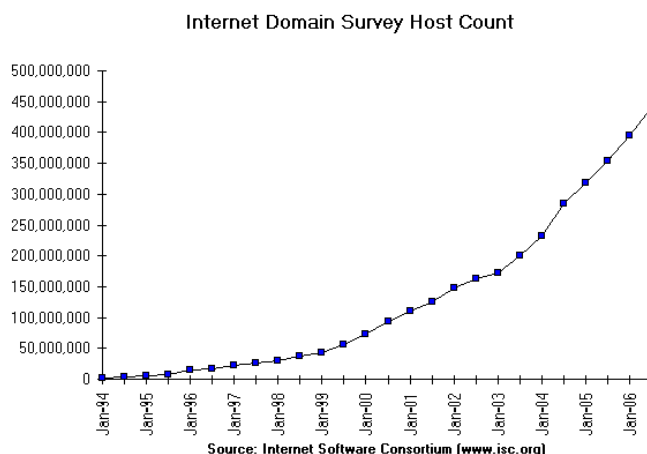
۴۵.....	۳-۴- بازیابی اطلاعات برای وب معنایی.....
۴۸.....	۳-۵- جمع بندی مطالب.....
۴۹.....	۴- بررسی فعالیت های انجام شده در زمینه استخراج اطلاعات.....
۵۰.....	۴-۱- WEBOQL.....
۵۲.....	۴-۲- (World Wide Web Wrapper Factory) W4F.....
۵۳.....	۴-۳- OntoKnowledge.....
۵۶.....	۴-۴- OntoPrise.....
۵۷.....	۴-۵- KIM.....
۵۹.....	۴-۶- OntoGather.....
۶۰.....	۴-۷- BYU-Ontos.....
۶۱.....	۴-۸- جمع بندی مطالب.....
۶۲.....	۵- ارائه یک چارچوب جهت بازیابی اطلاعات.....
۶۲.....	۵-۱- تحلیل مساله بازیابی اطلاعات.....
۶۴.....	۵-۲- استخراج اطلاعات صریح و اطلاعات ضمنی.....
۶۵.....	۵-۳- مقاومت در برابر تغییرات منابع.....
۶۶.....	۵-۴- معرفی چارچوب پیشنهادی.....
۷۰.....	۵-۵- فعالیت های بعدی.....
۷۲.....	مراجع.....
۷۲.....	[Intoduction to Semantic Web].....
۷۲.....	[Semantic Web Languages References].....
۷۳.....	[Knowledge Management & Information Retrieval].....
۷۲.....	[Others].....

۱- مقدمه

در این بخش به بررسی مشکل وب جاری پرداخته و وب معنایی را به عنوان یک راه حل معرفی می کنیم. سپس یکی از چالش های موجود در وب معنایی را تشریح کرده و محدوده بحث خود را مشخص می کنیم.

مشکل وب کنونی

با توجه به رشد روز افزون وب و با وجود ۴۰۰ میلیون وب سایت [۱۶] و چندین میلیارد منابع اطلاعاتی (شامل صفحات HTML، فایل های صوتی و ویدئویی، تصاویر فایل های XML، RSS و غیره)، عملاً وب جاری به یک کتابخانه بزرگ از منابع و اطلاعات مختلف تبدیل شده است.



شکل ۱: نمودار رشد سایت های اینترنتی [۱۶]

موتور های جستجوگر مانند گوگل و یاهو و غیره، به کاربران خود امکان جستجو بر اساس کلمات کلیدی مشخص را می دهند. اما بزرگترین مشکل در ارتباط با این شیوه جستجو، که به آن جستجوی کور هم گفته می شود، یافتن کلمات کلیدی مناسب می باشد. البته نباید فراموش کنیم که وب با همین وضعیت موجود نیز یک موفقیت بزرگ به حساب می آید. تا ده سال پیش، هرگز کسی فکر آن را نمی کرد که روزی اینترنت بتواند تا این حد در شیوه زندگی انسان ها، اخبار و اطلاع رسانی، ارتباطات و غیره تاثیر بگذارد.

اما از طرفی دیگر، وب موجود، دارای یک مشکل اساسی می باشد. می توان گفت که چالش اساسی در ارتباط با وب آن است که منابع موجود تنها برای انسان ها قبل فهم و استفاده می باشد. بنابراین ماشین ها

در جستجو، اشتراک و یکپارچه سازی انبوهی از اطلاعات، محدودیت های زیادی دارند. در حال حاضر ماشین ها جهت شناخت محتوی و اطلاعات موجود در وب، نیاز به دانش انسانی دارند و بدون آن تنها می توانند کورکورانه، وجود یا عدم وجود یک کلمه کلیدی را مشخص نمایند.

وب معنایی

ایده توسعه وب جاری با تزریق مقداری اطلاعات تکمیلی، بگونه ای که اطلاعات موجود قابل فهم برای ماشین ها باشد، برای اولین بار توسط آقای تیم برنرز لی^۲ مطرح گردید [۲۰۱]. ایشان نسل جدیدی از وب جاری را با نام وب معنایی معرفی کردند و هدف از وب معنایی را قابل فهم کردن اطلاعات موجود در وب و نیز افزایش قابلیت همکاری^۳ بین افراد و عوامل درگیر با داده های مشترک عنوان نمودند. بنابراین وب معنایی قصد ندارد تا وب موجود را از بین ببرد یا جایگزین آن شود، بلکه می خواهد آن را توسعه دهد و در کنار آن (بصورت یک لایه بر روی وب جاری) قرار گیرد.

در حوزه وب معنایی، منظور از قابل فهم نمودن اطلاعات برای ماشین ها آن است که ماشین ها بتوانند، از اطلاعات موجود استنتاج کرده و اطلاعات جدیدی را تولید نمایند. این اطلاعات جدید می توانند در پاسخگویی به پرس و جوی های کاربران مورد استفاده قرار گیرند. از این رو یکی از اجزای مهم در وب معنایی موتور استنتاج می باشد که باید مبتنی بر یک منطق باشد؛ مانند منطق مرتبه اول / دوم، منطق تشریحی و غیره.

در وب معنایی تمرکز بر روی داده ها است و سعی می شود تا از وابستگی داده ها به برنامه یا برنامه های خاص بکاهد. بدین منظور لازم است تا سمانتیک و ساختار نیز به درون داده ها تزریق گردد [۳]. در این صورت یک داده هوشمند خواهیم داشت که می تواند در یکی از چارچوب های وب معنایی، بدون نیاز به برنامه ای خاص، استفاده شده و در فرایند استنتاج منطقی شرکت نماید.

جایگاه آنتالوژی در وب معنایی

هر چند بهبود در جستجوی اطلاعات یکی از مهمترین اهداف وب معنایی می باشد ولی وب معنایی کاربرد های دیگری نیز دارد، مانند؛ اشتراک اطلاعات و یکپارچه سازی اطلاعات. برای رسیدن به این اهداف باید یک مدل معنایی از دامنه مورد بحث خود تعریف نماییم. در وب معنایی از آنتالوژی ها جهت مدل سازی معنایی استفاده می شود. در واقع، آنتالوژی ها ابزاری جهت توصیف مفاهیم ذهنی بشر می باشد و نقشی کلیدی را در وب معنایی و مدیریت دانش ایفا می کند. آنتالوژی ها را می توان با زبان های منطقی مانند منطق مرتبه اول و دوم، منطق تشریحی، RDF، OWL و یا حتی با یک مدل گرافیکی مانند UML نمایش داد.

Tim Berner's Lee²
Interoperability³

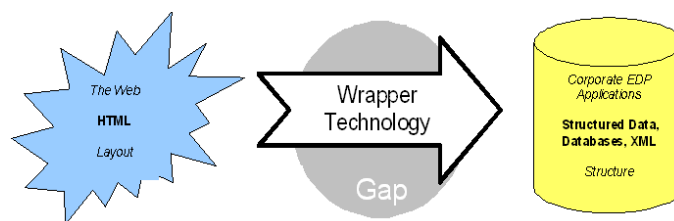
وب معنایی، هوش مصنوعی نیست

همانطور که گفته شد، وب معنایی سعی دارد تا اطلاعات موجود در وب را برای ماشین ها قابل درک کند. ولی هرگز قصد آن را ندارد تا به ماشین ها قابلیت های جادویی ببخشد تا با انسان ها رقابت کنند. بلکه تنها سعی دارد از سرعت و قدرت ماشین ها در پردازش اطلاعات موجود، جهت حل برخی مشکلات انسان ها استفاده نماید.

وب معنایی سعی دارد، تعامل بین انسان ها و ماشین ها، در سطح انسان ها صورت گیرد [۳]. یعنی بجای آنکه انسان ها اطلاعات، دستورات و یا نیازمندی های خود را به زبانی که به زبان سطح ماشین نزدیک تر است اعلام نمایند، آن ها را در سطحی نزدیک تر به سطح زبان مادری و ادراکات ذهنی خود وارد نمایند، بطوریکه ماشین آن موارد را درک کرده و اطلاعات داده شده را پردازش نموده و نتایج را در همان سطح، به انسان ها منتقل نماید. از این رو وب معنایی، هوش مصنوعی نیست ولی از روش ها و تکنیک های مطرح شده در هوش مصنوعی (مانند پردازش زبان طبیعی^۴) استفاده می کند.

بازیابی و استخراج اطلاعات برای وب معنایی

در وب معنایی، اطلاعات باید با فرمتی سازگار با زبان منطقی مورد استفاده، ثبت و نگهداری گردد. از این رو اطلاعات موجود را باید بگونه ای خاص استخراج کرده و به شکل قابل استفاده در وب معنایی تبدیل نمود. ایده کلی آن است که یک نرم افزار واسط، بنام Wrapper، نوشته شود تا با اجرای آن داده های موجود را به زبان قابل فهم برای وب معنایی تبدیل نماید (شکل زیر). تا کنون Wrapper ها و چارچوب های مختلفی برای بازیابی اطلاعات موجود تهیه شده است که از نظر میزان مکانیزه بودن (عدم نیاز به دخالت انسان) و دقت در بازیابی اطلاعات و نیز انعطاف پذیری نسبت به تغییرات ساختار نمایش در منبع اطلاعاتی اولیه، با هم متفاوت می باشند.



شکل ۲: عملکرد Wrapper ها

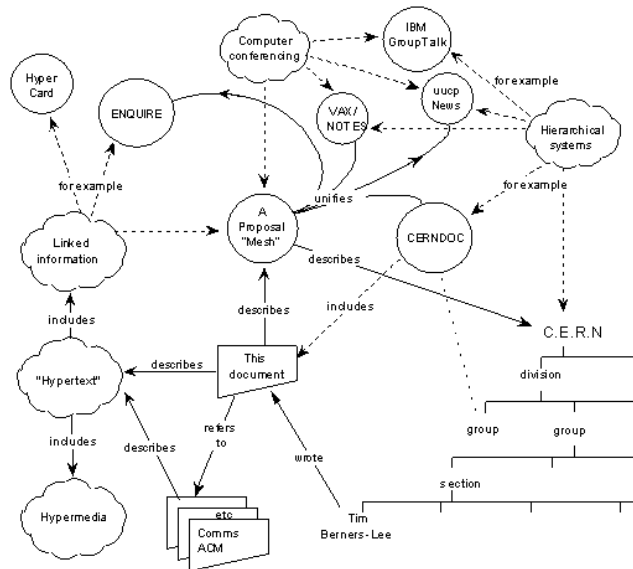
^۴ Natural Language Processing (NLP)

مساله بازیابی و استخراج اطلاعات برای استفاده در وب معنایی، موضوع اصلی این نوشته را تشکیل خواهد داد. در این نوشته، هدف ما از بازیابی اطلاعات، یافتن اقلام اطلاعاتی از مستندات موجود در وب و نگهداری آنها در قالب هایی مشخص و مناسب (مانند جداول بانک های اطلاعاتی رابطه ای و یا آنتالوژی ها و غیره) می باشد. تمرکز اصلی ما بر استخراج اطلاعات از منابع ساخت یافته و نیمه ساخت یافته خواهد بود و وارد مباحث پردازش زبان طبیعی (NLP – Natural Language Processing) و استخراج اطلاعات موجود در متون زبان طبیعی نخواهیم شد.

سایر بخش های این مقاله بصورت زیر سازماندهی شده است: در فصل دوم به اختصار وب معنایی را معرفی کرده و با ادبیات موضوع مورد بحث آشنا خواهیم شد. در فصل سوم اجمالا به معرفی مدیریت دانش در وب معنایی پرداخته و انواع روش های موجود جهت بازیابی اطلاعات برای وب معنایی را معرفی خواهیم کرد. در فصل چهارم، به بررسی اجمالی فعالیت های انجام شده در رابطه با بازیابی اطلاعات می پردازیم. در نهایت در فصل پنجم به معرفی چارچوب و روش پیشنهادی خود برای استخراج اطلاعات جهت استفاده در وب معنایی می پردازیم.

۲ - معرفی وب معنایی

ایده اولیه آقای Tim Berners-Lee درباره اینترنت با آنچه که ما امروزه از اینترنت می شناسیم، متفاوت است. از نظر ایشان وب آینده شامل دو بخش مهم است: یکی امکان همکاری بیشتر گروه ها و رسانه ها و دیگری ایجاد یک شبکه وب قابل فهم و قابل پردازش توسط ماشین [۱].

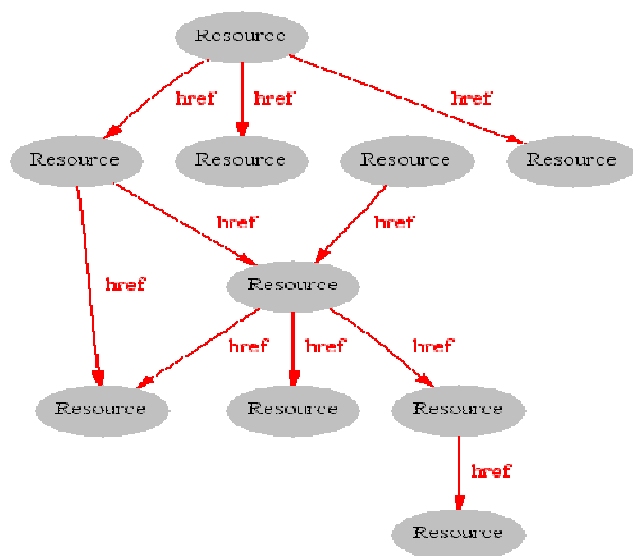


شکل ۳: ایده اولیه آقای برنرز-لی درباره وب

شکل فوق ایده اولیه آقای Tim Berners-Lee را نشان می دهد [۱]. همانطور که دیده می شود، از نظر ایشان، وب چیزی فراتر از مجموعه ای از صفحات HTML می باشد و باید امکان تعریف روابط بین منابع مختلف موجود در وب، وجود داشته باشد. مانند روابط *wrote*، *includes*، *describes* و غیره. متاسفانه در وب کنونی، این تعاریف دیده نمی شوند. در واقع در ایده اولیه، علاوه بر اطلاعاتی که برای استفاده انسان ها بر روی وب قرار می گرفت، مقداری متادیتا^۵ نیز بر روی آنها قرار داده می شد تا این اطلاعات توسط ماشین ها قابل پردازش باشند.

اما وبی که امروزه در اختیار داریم و اصطلاحاً به عنوان وب نحوی^۶ شناخته می شود، با ایده اولیه، تفاوت هایی دارد. اگر چه در وب موجود هم همانطوریکه در شکل زیر دیده می شود منابع توسط HyperLink ها به یکدیگر مرتبط می شوند ولی به دلیل استفاده گسترده از زبان طبیعی، عملاً پردازش ماشینی اطلاعات تحت الشعاع قرار گرفته است. در واقع، در وب فعلی تنها خود را محدود به یک رابطه ی خاص (HyperLink) بین منابع کرده ایم.

^۵ Metadata
^۶ Syntactical Web



شکل ۴: ساختار وب کنونی

بنابراین در وب موجود ما از کامپیوترها فقط به عنوان ابزاری جهت جمع آوری اطلاعات و نمایش آن ها در قالب صفحات وب استفاده کرده و وظیفه تفسیر آنها را بر عهده انسان ها قرار می دهیم. با توجه به گسترش روز افزون اطلاعات در وب، نیاز به استفاده بهتر از ماشین ها برای تفسیر و انجام اعمال بر اساس اطلاعات روزانه، بیشتر می شود. سوال اساسی این است که چگونه می توان یک وب از داده هایی بسازیم که قابل پردازش توسط ماشین باشد؟

پاسخ به سوال فوق را باید در روش نگرش خود به داده ها جستجو کنیم. تاریخ نشان می دهد که در ابتدا، هوشمندی، تماما در درون برنامه ها قرار داشت و (از نظر ساختاری) برنامه ها کاملا وابسته به داده های خود بودند. در این نسل از برنامه ها، داده ها نقش ثانویه را در پردازش اطلاعات، بر عهده داشتند. با ظهور مدل شی گرایی در زبان های برنامه نویسی، داده ها استقلال بیشتری یافتند. ولی هنوز سازندگان برنامه ها، بدلائل رقابتی و بازار، سعی می کردند تا همچنان داده های داخلی را وابسته به برنامه های خود نگه دارند. در وب، به لطف وجود XML و ظهور وب معنایی، داده ها نقش اول را ایفا می کنند. از این رو تلاش های زیادی صورت می گیرد تا هوشمندی را از درون برنامه ها به سمت داده ها منتقل نمایند. به این ترتیب می توان گفت که هدف اصلی در وب معنایی، ایجاد بستری جهت قابل درک کردن داده ها برای ماشین (کامپیوتر) ها و ایجاد داده های هوشمندتر^۷ می باشد [۲].

۲-۱ - مشکلات وب موجود

^۷ منظور از هوشمندی داده ها، آن است که داده ها به تنهایی و بدون نیاز به نرم افزار مربوطه قابل استفاده بوده و بتوان دانش لازم را از اطلاعات موجود استنتاج نمود

در حال حاضر وب موجود، به عنوان ابزاری برای جمع آوری اطلاعات و نمایش آنها در قالب صفحات HTML، استفاده می شود و همانطور که گفته شد، این صفحات از طریق ابر لینک ها^۸ با یکدیگر ارتباط دارند. رشد سریع و پیوسته اطلاعات موجود در اینترنت و اینترنت، عملیات جستجوی صحیح اطلاعات، سازماندهی اطلاعات، دسترسی و نگهداری اطلاعات مورد نیاز کاربران را دچار مشکل کرده است.

به عنوان نمونه، یکی از مسائلی که همگی ما آن را در وب تجربه کرده ایم جستجوی اطلاعات است. به دلیل اینکه موتورهای جستجو مانند گوگل بر روی کلمات کلیدی جستجو انجام می دهند تعداد زیادی نتیجه بر می گردانند تا کاربر خود تصمیم گرفته و مناسب ترین گزینه را انتخاب نماید. حتی در بیشتر موارد، نتایجی ارائه می کنند که از دید یک انسان خیلی غیر عادی به نظر رسد (مانند اکثر جستجو های تصاویر). در اینجا به برخی از مواردیکه انجام آن ها بوسیله وب موجود امکانپذیر نیستند اشاره شده است:

- سوالات پیچیده که نیاز به اطلاعات زمینه^۹ دارد: مثلا اطلاعاتی در مورد حیواناتی که از رادار استفاده می کنند ولی خفاش یا دلفین نیستند پیدا کن.
- پیدا کردن اطلاعات در مخازن داده: اطلاعات سفر، قیمت اجناس و سرویس ها، نتایج آزمایشات ژنی انسان
- محول کردن وظایف پیچیده به عامل^{۱۰} ها: به عنوان مثال انتصاب وظیفه زیر به یک عامل: برای تعطیلات آخر هفته آینده ام یک جایی که گرم بوده، خیلی از اینجا دور نبوده و انگلیسی یا فرانسه صحبت کنند را رزرو کن.

۲ - ۲ - نیاز به بازنمایی دانش

مسئله اصلی در وب جاری، این است که از دید ماشین ها صفحات وب عمدتا شامل اطلاعات و قواعدی برای نمایش اطلاعات هستند و وظیفه آنها پیروی از این قواعد نمایشی (اندازه فونت ها، رنگ ها و غیره) و نشان دادن صفحه به استفاده کننده است. معنای اطلاعات تنها بوسیله انسان ها قابل فهم بوده و ماشین ها تنها می توانند تقریبی برای آن معانی داشته باشند.

بعنوان مثال، جمله "Ali lives in tehran" تنها برای یک انسان قابل فهم است و از نظر کامپیوتر رشته ای از حروف می باشد. انسان با دیدن این جمله می فهمد که علی یک انسان است که در شهر تهران زندگی می کند ولی از دید کامپیوتر Ali می تواند معادل یک شناسه مانند A123 و تهران معادل شناسه T06 و ... باشد. بنابراین جمله فوق می تواند در ماشین، بصورت زیر دیده شود.

"A123 L5 I2 T06"

⁸ HyperLink
⁹ Contextual information
¹⁰ Agent

یکی از روش هایی که برای تفهیم جملات تشریحی زیان طبیعی به ماشین، پیشنهاد شده استفاده از xml و قراردادن اقلام اطلاعاتی در تگ^{۱۱} های معنی دار است. بعنوان مثال، ذیلا، رشته فوق با تگ های name و verb و city تفکیک شده است:

```
<name>Ali</name>
<verb>lives</verb>
<city>tehran</city>
```

ماشین با دیدن متن فوق متوجه می شود که متن اولیه شامل سه بخش (Name, Verb, City) است و هر بخش نشان دهنده یک قلم اطلاعاتی می باشد، به این ترتیب گفته می شود که مقداری سمانتیک به متن اولیه تزریق شده است. به این عمل، یعنی جدا کردن بخش های مختلف اقلام اطلاعاتی و ارائه یک تعریف مشخص برای هر قلم اطلاعاتی، حاشیه نویسی (Annotation) گفته می شود^{۱۲}. اما باید توجه داشت که اطلاعات فوق را می توان بصورت های مختلف با استفاده از تگ های متفاوت بیان نمود. به عنوان نمونه در مثال فوق می توانیم بجای تگ name از تگ person استفاده کنیم:

```
<person> Ali </person>
<verb>lives</verb>
<city>tehran</city>
```

بنابراین برنامه ای که از داده های فوق استفاده می کند باید از نام تگ های بکار رفته، نیز اطلاع داشته باشد. از طرفی دیگر تا کنون از دید ماشین، تنها بخش هایی از متن (با توجه به تفسیر انسان از آن متن)، از هم تفکیک شده است. ولی باز هم هنوز متن فوق، نامفهوم می باشد. به عنوان مثال ماشین می تواند متن فوق را بصورت زیر مشاهده نماید:

```
<p_06> A123 </p_06>
<v_123> L5 </v_123>
<c_753> T06 </c_753>
```

بنابراین لازم است از روش هایی برای افزودن معنی اطلاعات، به نحوی که برای ماشین ها قابل فهم باشد استفاده کرد. برای این منظور دو روش کلی وجود دارد:

- توافق اولیه بر روی معنی حاشیه گذاری^{۱۳} هایی که بر روی اطلاعات قرار می گیرند. به عنوان مثال همه قبول کنند که بجای تگ name از تگ person استفاده شود. اما مشکل این روش آن است که بسیار غیرقابل انعطاف بوده و در ابعاد وب عملا قابل استفاده نیست. در ضمن در این روش تنها تعداد محدودی از مفاهیم را می توان در قالب یک قرارداد (توافق اولیه) بیان کرد.

¹¹ tag

¹² در حال حاضر اتوماتیک کردن نشانه گذاری (بر روی منابع وب جاری) یکی از چالش های وب معنایی است.

¹³ Annotation: منظور توضیحات حاشیه ای است که غالبا در قالب تگ ها به اطلاعات موجود اضافه می گردد.

- استفاده از آنتالوژی^{۱۴} ها: آنتالوژی توصیف رسمی از واژگانی است که در یک دامنه خاص با آنها برخورد می کنیم. این امکان نیز وجود دارد که ارتباط بین مفاهیم مختلف را در آنتالوژی های گوناگون مشخص کنیم. بنابراین این روش نسبت به روش قبل از انعطاف بسیار بالایی برخوردار بوده و عملاً محدودیتی را بر روی مفاهیم قابل تعریف قرار نمی دهد. به همین دلیل آنتالوژی ها مورد استقبال بسیار گسترده ای قرار گرفته اند.

۲-۳- آشنایی با اجزای اساسی وب معنایی

وب معنایی برای تبدیل شدن به واقعیت بر پایه استانداردها و تکنولوژی های متنوعی استوار است که مهمترین آنها عبارتند از: Agent, Ontology, RDF, XML, URI.

(Universal Resource Identifier) URI

در وب با منابع بسیار متنوعی برخورد می کنیم که لازم است روشی استاندارد برای نامگذاری آن ها بیابیم بطوریکه هر نامی که به یک منبع تخصیص داده می شود یکتا بوده و تنها برای همان یک منبع استفاده گردد. برای این منظور از URI استفاده می کنیم. URI در واقع یک رشته فرمت شده است که معمولاً برای شناسایی یک منبع در اینترنت استفاده می شود. به عنوان مثال URI زیر به یک صفحه وب اشاره دارد (به URI هایی که به یک صفحه وب اشاره دارند، URL^{۱۵} نیز گفته می شود)

<http://www.MyHomePage.com/index.aspx>

URI زیر نیز به عنصر Actions در داخل منبع Books.OWL، اشاره دارد:

<http://www.MyHomePage.com/Books.OWL#Actions>

اما منظور از منبع چیست؟ یک منبع می تواند یک موجودیت الکترونیکی (مانند فایل) یا یک مفهوم (مانند انرژی) باشد. بطور کلی هر چیزی که قابل شناسایی (مادی یا معنوی) باشد را یک منبع گوییم.

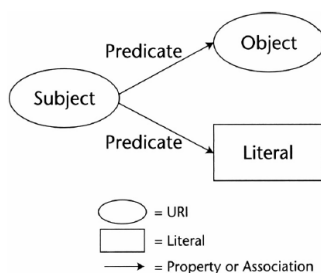
(eXtensible Markup Language) XML

¹⁴ Ontology
¹⁵ Universal Resource Locator

XML لایه بنیادی، در وب معنایی می باشد. هدف آن امکان همکاری نحوی^{۱۶} بین واحد های مختلف می باشد. سایر تکنولوژی های وب معنایی (RDF و Ontology و غیره) بر روی این لایه بنا شده اند. خود XML نیز از سایر تکنولوژی ها مانند، Unicode و URI استفاده می کند. استاندارد یونیکد، این امکان را فراهم می کند که فایل های XML با هر زبانی قابل کد سازی باشند. URI ها هم همانطور که گفته شد، جهت آدرس دهی منابع موجود در وب استفاده می شوند.

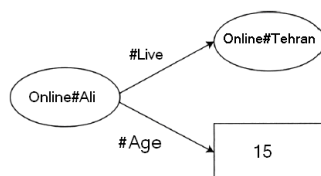
RDF

بطور خلاصه می توان گفت، RDF، یک زبان مبتنی بر XML جهت توصیف منابع می باشد. هر جمله RDF دارای سه بخش می باشد: Subject, Predicate, Object. به عنوان مثال جمله Ali lives in Tehran بصورت سه تایی (Ali, Live, Tehran) نشان داده می شود. جملات RDF را می توان بصورت یک گراف نیز نشان داد، که در این صورت Subject و Object بصورت یک گره و Predicate بصورت یال ارتباطی نشان داده می شود.



شکل ۵: گراف RDF

در شکل فوق، URI ها با بیضی و داده های تحت اللفظی^{۱۷} با مستطیل نشان داده شده است. همانطور که دیده می شود جملات مختلف که بخش Subject (و یا Object) آن یکسان است را می توان با هم ترکیب نمود. به عنوان مثال جملات "Ali Lives in Tehran" و "Ali is 15 years Old" بصورت گراف زیر نمایش داده می شوند.



شکل ۶: گراف RDF نمونه

¹⁶ Syntactic Interoperability
¹⁷ Literal

در RDF، هر یک از بخش‌های Subject و Predicate باید نشان دهنده یک URI باشند ولی بخش Object می‌تواند علاوه بر URI نشان دهنده یک Literal نیز باشد. به عنوان مثال در شکل بالا سن Ali که در داخل یک مستطیل نشان داده شده است یک عدد Literal است ولی محل زندگی Ali که بصورت بیضی نشان داده شده است یک منبع است و با یک URI نشان داده شده است. در ادامه، RDF و کاربرد های آن را بیشتر معرفی خواهیم کرد.

Ontology

همانطور که گفته شد، آنتالوژی توصیفی رسمی از واژگانی است که در یک دامنه خاص با آنها برخورد می‌کنیم. معمولاً در یک آنتالوژی کلاس‌های مهم دامنه مورد بحث، مشخص شده و در قالب سلسله مراتب ارث‌بری، سازماندهی می‌شوند. هر کلاس نیز با خصوصیتی^{۱۸} که بین تمام نمونه‌های آن کلاس مشترک می‌باشد، توصیف می‌گردد. در ضمن روابط و ضوابط مهم بین کلاس‌ها و یا بین اجزای کلاس‌ها نیز بخشی از آنتالوژی می‌باشد.

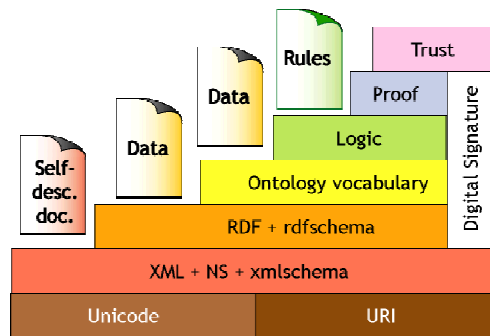
در واقع، آنتالوژی‌ها روشی جهت مدل‌سازی بخشی از جهان می‌باشند و از آن برای بازنمایی دانش استفاده می‌گردد. امروزه آنتالوژی‌ها، یکی از مفاهیم کلیدی در بسیاری از رشته‌ها (مانند پردازش زبان طبیعی، بازیابی اطلاعات، بانک‌های اطلاعاتی، مدیریت دانش، سیستم‌های چندعامله^{۱۹} و ...) می‌باشند.

Agent

پس از اینکه اطلاعات قابل فهم توسط ماشین‌ها در قالب وب معنایی در اختیار آنها قرار گرفت، لازم است عامل‌هایی برای پردازش و استفاده از این اطلاعات تولید شوند. این عامل‌ها اطلاعات مورد نیاز خود را از وب معنایی دریافت کرده و با پردازش آنها به نتایج مورد نظر خود می‌رسند.

۲-۴ - معماری وب معنایی

در شکل زیر معماری ارائه شده توسط آقای لی (در سال ۲۰۰۱) برای وب معنایی نمایش داده شده است. البته معماری‌های جدیدتری نیز در سالهای ۲۰۰۳ و ۲۰۰۵ برای وب معنایی مطرح شده‌اند که برای آشنایی با آنها می‌توانید به [۶ و ۷] مراجعه کنید:



شکل ۷: معماری پشته ای وب معنایی

معماری فوق، پشته ای از تکنولوژی ها و استاندارد های مورد استفاده در وب معنایی می باشد. البته این یک معماری قطعی و نهایی برای وب معنایی نیست و ما آن را به عنوان یک پروتوتایپ یا یک مدل ایده آل از چنین معماری ای می پذیریم [۴]. ذیلا به توصیف لایه های معماری فوق می پردازیم:

- تحتانی ترین لایه متشکل از استاندارد های **URI** و **Unicode** می باشد. با استفاده از Unicode هرکسی می تواند با زبان محلی خود داده ها را توصیف نماید، بدون اینکه کد زبان محلی اش با کد سایر زبان ها تداخل داشته باشد. به این ترتیب داده ها به هر زبانی که باشند بدون هیچ مشکلی قابل انتقال به هر مکان خواهند بود. از طرفی URI نیز به عنوان استاندارد برای آدرس دهی یکسان و منحصر به فرد منابع موجود در وب معنایی، برای انتقال اطلاعات الزامی می باشد.

- **لایه XML: XML** [۵ و ۸] به عنوان یک لایه جداگانه این امکان را فراهم می کند که داده ها را بتوان خارج از برنامه ها تعریف کرده و به این ترتیب به نوعی استقلال داده ای برسیم. XML مبتنی بر متن بوده و از قابلیت های URI و Unicode در تعریف داده ها استفاده می کند. به این معنی که می توانید به راحتی با استفاده از یک برنامه ویرایش متن (مانند NotePad) یک فایل XML را ایجاد کرده و یا آن را تغییر دهید. در این لایه برنامه ها می توانند به راحتی با استفاده از فایل های XML با یکدیگر ارتباط برقرار کنند. اما این ارتباط تنها یک ارتباط نحوی است. به عبارتی دیگر، هرچند XML با اضافه کردن تگ ها باعث خوانایی بیشتر داده ها (برای ماشین) می شود ولی تنها برنامه هایی می توانند از یک فایل XML خاص استفاده کنند که با نام تگ های موجود در آن و ساختار درختی آن فایل XML، آشنا باشند.

- **لایه RDF: XML** به تنهایی نمی تواند کافی باشد و باید اطلاعات بیشتری، شامل معنی و سمانتیک داده ها را اضافه کرد تا بتوان بر اساس آن استدلال کرده و روابط بین اجزای داده را بدست آورد. از این رو لایه RDF [۱۳] جهت توصیف منابع و روابط بین آن ها در بالای لایه XML قرار می گیرد. درحالیکه XML تگ هایی را جهت توصیف بیشتر به درون مستندات اضافه می کند، RDF به توصیف متا دیتای خارج از مستندات می پردازد. به عنوان مثال، در

یک سیستم آرشیو تصاویر، RDF به توصیف مالک تصویر، تاریخ ایجاد و نوع تصویر (یعنی متا دیتای تصویر) می پردازد. بنابراین می توان گفت که :

۱. RDF زبان توصیف متا دیتا است.

۲. RDF روشی جهت توصیف سمانتیک اطلاعات به شکل قابل فهم برای ماشین می باشد.

۳. RDF یک زبان اخباری^{۲۰} بوده و جهت نمایش دانش هایی که بصورت گزاره ای^{۲۱} عنوان می شوند، مناسب است و همانطور که گفته شد می توان آن را بصورت گراف نیز نمایش داد.

RDF یک روش رسمی جهت توصیف متا دیتا با استفاده از XML است ولی هیچ اطلاعاتی درباره فرهنگ لغات استفاده شده مانند مفاهیم نوع داده ای^{۲۲} و ارث بری (SubClassOf) نمی دهد. از این رو از RDF Schema استفاده می شود [۱۳].

• **لایه آنتالوژی :** اما RDF به تنهایی جهت نمایش سمانتیک داده ها و استنتاج از آن ها کافی نیست. به عنوان مثال با استفاده از RDF نمی توان نشان داد که یک شی، حاصل اجتماع یا اشتراک یا تفاضل اشیاء دیگر است. مفهوم آنتالوژی، مفهوم جدیدی نیست ولی با ظهور وب معنایی، و ارائه یک زبان رسمی برای تعریف آنتالوژی های تحت وب (OWL²³) [۱۵] از اهمیت ویژه ای برخوردار شد و کاربرد آن در سایر رشته ها نیز مورد تحقیق و بررسی قرار گرفت. آنتالوژی ها (۱)نسبت به RDF قدرت تشریحی بالاتری دارند (به عنوان مثال می توان در هنگام معرفی رابطه بین دو کلاس شرایط و محدودیت های مختلفی را عنوان کرد). (۲) از موتور های استنتاج بطور کامل پشتیبانی می کنند. یک آنتالوژی شامل مجموعه ای از کلاس ها (و سلسله مراتب ارث بری بین کلاس ها) و صفات^{۲۴} کلاس ها (یا Slot ها) و روابط بین کلاس ها (ارث بری، هم ارزی، گسستگی) و غیره می باشد. از جمله چالش های موجود در ارتباط با آنتالوژی ها، نگهداری و مدیریت آنتالوژی ها و نگاشت بین آنتالوژی های یک دامنه می باشد. لازم به ذکر است که آن چه باعث توجه سایر رشته ها به مبحث آنتالوژی (مطرح شده در وب معنایی) شده است، قدرت آن در بازنمایی دانش و استنتاج می باشد. لذا، زمانیکه هوش مصنوعی از ارائه یک تکنولوژی هوشمند و موثر، عاجز مانده است، آنتالوژی می تواند راه حل جایگزین مناسبی برای مسائل مرتبط با هوش مصنوعی باشد.

• **لایه Logic :** ما در فعالیت های روزمره خود، همواره از استدلال های منطقی استفاده می کنیم. به عنوان مثال اگر حرف N نشان دهنده یک پیام خوانده نشده در MailBox شخصی ما باشد. آنگاه اگر در مقابل یک نامه حرف N را مشاهده کنیم، پی می بریم که آن نامه جدید

²⁰ Assertional language

²¹ Propositional

²² Data Type

²³ Web Ontology Language

²⁴ Properties

است. لایه منطق در وب معنایی نیز بر همین اساس رفتار می نماید. در این لایه زبان استدلال منطقی (مانند منطق مرتبه اول^{۲۵} یا منطق تشریحی^{۲۶})، مشخص شده و توسط موتور استنتاج، جهت انجام استدلال های منطقی، مورد استفاده قرار می گیرد.

- **لایه های Trust و Proof^{۲۷}:** یکی از مسائل بسیار مهمی که در وب معنایی مطرح می شود مسئله اعتماد به اطلاعات گسترده ای است که در آن توسط افراد مختلف قرار داده می شوند. مسلماً استفاده از روش های متمرکز برای تصدیق درستی اطلاعات موجود در وب با توجه به ابعاد آن امکان پذیر نیست. بنابراین در وب معنایی از روشی دیگر استفاده می شود که در نهایت منجر به Web of Trust می شود. به این ترتیب که هر شخص می تواند قواعدی را بیان کند و سپس هنگامی که نیاز به بررسی درستی یک ادعا باشد، موتورهای استنتاج می توانند از قواعد تعریف شده توسط افراد مختلف استفاده کرده و میزانی که می توان به آن ادعا اعتماد کرد را مشخص کنند.

- **لایه امضای دیجیتالی (Disital Signature):** این لایه جهت تضمین Trust و امنیت در وب معنایی، بر لایه های RDF/S و Logic و Proof اعمال می گردد. در عمل، موتور استنتاج باید با سیستم اعتبار سنجی^{۲۸}، نزدیکی تنگاتنگی داشته باشد. علاوه بر اینکه اطلاعات موجود در مستندات باید در مراحل استنتاج شرکت نمایند، بلکه باید اعتبار و امنیت مستندات هم بررسی گردد.

اکثر فعالیت ها و تحقیقات در زمینه وب معنایی مربوط به چهار لایه اول می باشد. از این رو، با توجه به اهمیت آن، در ادامه به معرفی استانداردهای XML و زبان RDF(S) و مفهوم آنتالوژی می پردازیم.

۲- ۵- آشنایی با خانواده استانداردهای XML

XML و اهمیت آن در وب معنایی

XML [۵ و ۸] یک زبان نیست، بلکه مجموعه ای از قواعد نحوی است که برای تزریق سمانتیک به مستندات مورد استفاده قرار می گیرد. به عبارتی دیگر از XML جهت ساخت زبان های جدید، مانند MathML^{۲۹}، استفاده می شود. XML، چهار خصوصیت و مزیت بارز دارد، که باعث موفقیت چشم گیر آن در بین شرکت ها و جوامع کامپیوتری شده است. این خصوصیات بارز عبارتند از [3]:

²⁵ First Order Logic

²⁶ Description Logic

²⁷ Proof تعیین اعتبار اطلاعات بدست آمده از بکار گیری منطق می باشد و در مقابل، Trust، با جامعیت و یکپارچگی Proof مرتبط است.

²⁸ Validation

²⁹ Math Markup Language

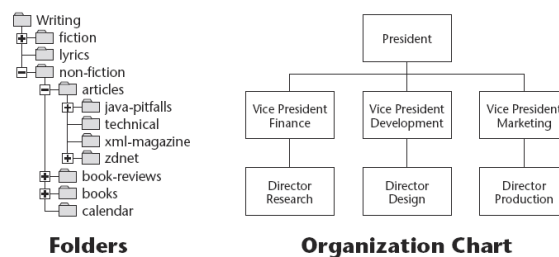
- ایجاد مستندات و داده های مستقل از برنامه های کاربردی: XML یک فایل متنی می باشد. از این رو بدون نیاز به برنامه کاربردی خاص، براحتی قابل درک توسط انسان می باشد. از طرفی دیگر برخلاف فرمت های دودویی^{۳۰} براحتی قابل استفاده در موتور های جستجوگر می باشد.

- XML یک استاندارد نحوی برای توصیف متا دیتا می باشد: یک تعریف عمومی برای متا دیتا عبارت است از: "داده درباره داده" به عنوان مثال در جدول زیر، FirstName داده ای است که مقدار Mahdi را توصیف می کند.

Meta Data	Data
First Name	Mahdi
Last Name	Talebian
Age	30
Telephone	021-12345678

جدول ۱: مقایسه داده و متادیتا

- XML یک استاندارد ساختاری برای توصیف مستندات و داده ها می باشد: XML استاندارد جهت توصیف ساختارهای درختی، مانند چارت سازمانی و یا ساختار Folder های یک دایرکتوری (شکل زیر)، ارائه می کند.



شکل ۸: یک مثال از چارت سازمانی و ساختار درختی Folder ها

- XML یک تکنولوژی جدید نمی باشد: XML زیر مجموعه ای از زبان SGML^{۳۱} می باشد و از این رو یک تکنولوژی جدید نیست. SGML در سال ۱۹۶۹ توسط Dr. Charles Ed Mosher, Goldfarb و Ray Lorie، ابداع شد. SGML یک استاندارد جهانی برای توصیف ساختار و محتوای یک سند متنی^{۳۲} است. در واقع می توان گفت که XML، SGML تنظیم شده برای وب می باشد.

³⁰ Binary
³¹ Standardized Generalized Markup Language
³² Text Document

مهمتر از همه، سرعت بالای کامپیوترها و فراوانی و ارزانی حافظه، در پذیرش فرمت XML (که قبل از استفاده نیاز به پردازش داشته و نیز حافظه زیادی را اشغال می نماید)، نقش مهمتری داشته اند. اما در وب معنایی، علاوه بر موارد فوق، اهمیت XML به آن دلیل است که سایر زبان های مدل سازی در وب معنایی از قواعد و استاندارد های XML، استفاده می کنند. در واقع می توان گفت که XML الفبای وب معنایی را تشکیل می دهد.

(XML Schema Definition) XSD

همانطور که گفته شد، XML یک استاندارد باز و مبتنی بر متن می باشد. از این رو هم محتوای داده و هم ساختار آن، به راحتی قابل تغییر می باشد. این ویژگی در عین آنکه یک مزیت است، یک عیب نیز می باشد. چرا که برنامه هایی که با فایل های XML کار می کنند، باید پیش از هر چیزی با ساختار فایل XML مورد استفاده آشنا باشند، مانند نام المان ها و محل قرار گیری آنها مشخصات صفات هریک از المان ها و غیره. از این رو نیاز است تا ساختار فایل های XML، بررسی شده و اعتبار آن ها ارزیابی گردد. بدین منظور XSD [۹ و ۱۲] جهت تعریف ساختار فایل های XML و ارزیابی آن، استفاده می شود.

XPATH

XPath [۱۱] استاندارد دی جهت آدرس دهی صریح بخش های مختلف یک سند XML می باشد. XPath در مجموعه استاندارد های خانواده XML دارای اهمیت بالایی است چرا که سمانتیک، نحو و عملیات اصلی مورد استفاده برای بسیاری از استانداردها (مانند XSLT, XPointer, XQuery,...) را فراهم می کند. با استفاده از عبارات XPath و یک چارچوب^{۳۳} مشخص نرم افزاری و API ها، می توان در ساختار درختی XML، به هر یک از المان ها دسترسی یافت. عبارات XPath از نظر ظاهری بسیار به روش نام گذاری مسیر ها در سیستم عامل یونیکس شباهت دارند:

poem/stanza/line

عبارت فوق، با توجه به ساختار سلسله مراتبی XML، تمام المان های line را نشان می دهد که پدر آنها المان stanza بوده و المان پدر stanza نیز poem باشد. عبارات XPath در هنگام پردازش، نسبت به گره جاری، سنجیده می شوند. گره جاری نیز با علامت نقطه "." نشان داده می شود.

به عنوان مثال به کد XML زیر توجه کنید:

```
<poem>
  <title>Roses</title>
  <author>Ima Poet</author>
  <stanza>
```

```

<line>Roses are red</line>
<line>violets are blue</line>
</stanza>
<stanza>
  <line>I'm a poet</line>
  <punch>and you're not!</punch>
</stanza>
</poem>

```

عبارت poem/stanza/line در XPath نتایج زیر را برمی گرداند:

```

<line>Roses are red</line>
<line>violets are blue</line>
<line>I'm a poet</line>

```

اگر از عبارت poem/stanza/* استفاده کنیم، خروجی شامل عبارت زیر نیز خواهد بود:

```

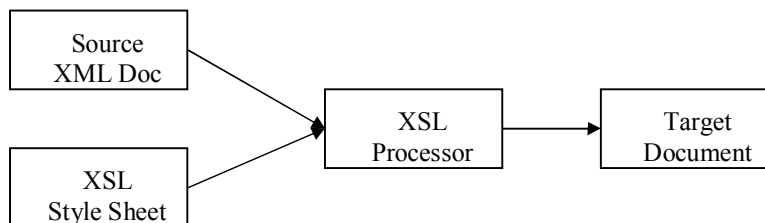
<punch>and you're not!</punch>

```

همچنین اگر بخواهیم تنها اولین المان را انتخاب نماییم، از عبارت poem/stanza/line[1] استفاده می کنیم.

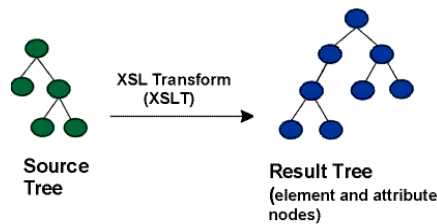
XSLT (Extensible Stylesheet Language Transformations)

XSLT [۱۰] اجازه می دهد تا مشخص کنیم، چگونه فایل های XML باید به مستندات دیگر منتقل شده و نیز چگونه یک فایل XML باید در فرمت ها و قالب های دیگر (مانند HTML، RTF و ...) نمایش داده شود. XSLT دارای دو ورودی می باشد: (۱) فایل XML که شامل داده های اصلی می باشد. (۲) فایل XSL که چارچوب درج داده ها و دستورات XSLT را شامل می شود.



شکل ۹: معماری XSLT

همانطور که در شکل فوق دیده می شود، XSLT نیاز به یک پردازشگر دارد. این پردازشگر داده های اصلی را از فایل XML ورودی می خوانند و با توجه به چارچوب و قالب تعریف شده در فایل XSL، خروجی مورد نظر را می سازد. البته همانطور که پیشتر گفته شد، در XSL جهت ارجاع به المان های موجود در XML، از روش آدرس دهی XPath، استفاده می شود. شکل زیر مدل داده ای XSLT را نشان می دهد



شکل 10 : مدل داده ای XSLT

۲-۶- آشنایی با RDF (Resource Description Framework)

RDF [۱۳] یک زبان مبتنی بر XML، جهت توصیف منابع می باشد. همانطور که می دانیم، XML امکان آن را فراهم می سازد تا متا دیتا را به بخش هایی از مستندات اضافه کنیم، در مقابل، یکی از کاربرد های RDF ایجاد متا دیتای مستقل از مستندات است؛ مانند ارتباط نویسندگان، تاریخ ایجاد و نوع کتاب. یکی از کاربردهای مناسب RDF در توصیف منابع غیر متنی (مانند تصاویر، صدا ها، فیلم ها و غیره) می باشد.

امروزه میلیاردها منبع در اینترنت وجود دارد. این منابع از نوع متن، گرافیک (تصویر)، صوت، فیلم، بانک های اطلاعاتی، برنامه های کاربردی و غیره می باشند. متأسفانه روشی عمومی برای توصیف یکنواخت این منابع وجود ندارد. ضمن آنکه بسیاری از این منابع موقتی و یا پویا (مدام در حال تغییر) می باشند.

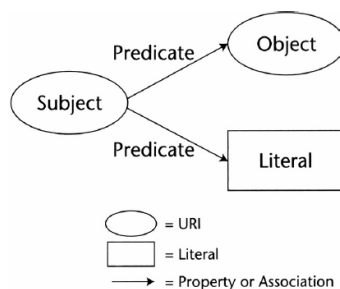
همانطور که گفته شد، کاربرد RDF در توصیف متا دیتا است (متا دیتا عبارت است از داده های ساخت یافته درباره داده). RDF دارای یک مدل گرافیکی برای نمایش متا دیتا می باشد، بطوریکه این اطلاعات قابل دسترسی توسط ماشین می باشند. در RDF لیست منابع تشکیل یک لغتنامه بزرگ را می دهند و از گزاره ها جهت توصیف اطلاعات (با استفاده از لغتنامه بزرگ منابع) استفاده می کند.

مدل RDF

در RDF مدل از مجموعه ای از جملات تشکیل شده است. هر جمله در قالب یک سه تایی مرتب به فرم زیر ارائه می شود:

(Predicate, Subject, Object)

در قالب فوق، Predicate و Subject معرف یک منبع می باشند و با استفاده از یک URI نشان داده می شوند. Object می تواند هم یک منبع و هم یک عبارت صریح (Literal) باشد. قالب فوق را می تواند بصورت گرافیکی شکل زیر نیز نمایش داد:



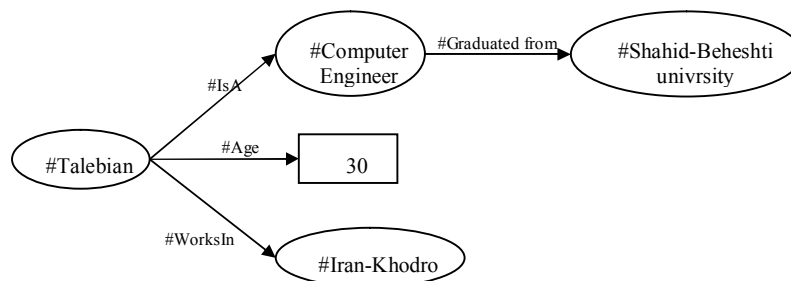
شکل 11: مدل جملات RDF

با توجه به طبیعت گزاره ای جملات RDF، این جملات قابل مقایسه با ساختار زبان طبیعی انسان ها می باشد [۳].

- Subject : در گرامر زبان طبیعی، این بخش یک اسم یا عبارت اسمی است که فاعل یک عمل می باشد. به عنوان مثال در جمله "The company sells batteries"، subject عبارت است از The Company.
- Predicate : در گرامر زبان طبیعی، Predicate شامل فعل می باشد. در مثال فوق، Predicate عبارت است از Sells.
- Object : در گرامر زبان طبیعی، این بخش یک اسم یا عبارت اسمی است که مفعول بوده و عمل مورد نظر بر روی آن انجام می گیرد. در مثال فوق، Object عبارت است از batteries.

در این مدل، Subject و Object های یکسان می توانند با هم ادغام شده و تشکیل یک گراف جهت دار را بدهند. به عنوان مثال گراف زیر نشان دهنده مدل سه جمله زیر می باشد:

Talebian is a computer engineer
Talebian is graduated from Shahid-Beheshti univrsty
Shahid-Beheshti univrsty is located in tehran
Talebian Works in Iran khodro
Talebian is 30 years old



شکل 12 : گراف RDF

نمایش دانش با استفاده از RDF

مدل RDF را می توان به چهار روش مختلف زیر نمایش داد:

- زبان طبیعی

جملات زیر با توجه به بخش های اصلی (Subject, Object, Predicate) RDF اطلاعات را در قالب زبان انگلیسی بیان می دارد:

Buddy Belden owns a business

Buddy is the father of Lynne

The business has a Web site accessible at <http://www.c2i2.com/~budstv>

در زندگی روزمره نیز می توان کلیه اطلاعات موجود را در قالب فوق استخراج نمود.

- (N3) Notation3

در این روش جهت نمایش جملات RDF از سه تایی مرتب استفاده می شود. شکل زیر جملات ارائه شده به زبان طبیعی فوق را به فرمت N3 نشان می دهد.

```
<#Buddy> <#owns> <#business>  
<#Buddy> <#father-of> <#Lynne>  
<#business> <#has-website> <http://www.c2i2.com/~budstv>
```

در این روش برای هر جمله زبان طبیعی، بخش های Subject, Object, Predicate مرتبط با آن را مشخص نموده و آنها را به عنوان یک منبع در وب در نظر گرفته و یک URI به آن نسبت می دهیم. علامت # نشان دهنده آن است که URI مربوطه در سند جاری قرار دارد. به عنوان مثال اگر URI سند جاری <http://www.c2i2.com/buddy/ontology> باشد، منبع #Buddy معادل با URI زیر است:

<http://www.c2i2.com/buddy/ontology#Buddy>

مشاهده می شود که به این ترتیب نمایش URI ها خیلی ساده می شود. البته می توان پیشنهادهایی را تعریف کرده و در نام منابع از آنها استفاده نمایید (مانند Namespace ها در XML)

@prefix bt: <<http://www.c2i2.com/buddy/ontology/>>

با استفاده از پیشوند فوق، جمله اول بصورت ساده تر زیر نمایش داده می شود:

```
<bt:Buddy> <bt:owns> <bt:business>
```

RDF/XML –

در این روش از XML جهت نمایش جملات RDF استفاده می شود. با استفاده از ابزارهایی مانند Jena می تواند مدل N3 را به راحتی به RDF/XML تبدیل کرد. شکل زیر نمایش جملات فوق را با استفاده از XML نشان می دهد:

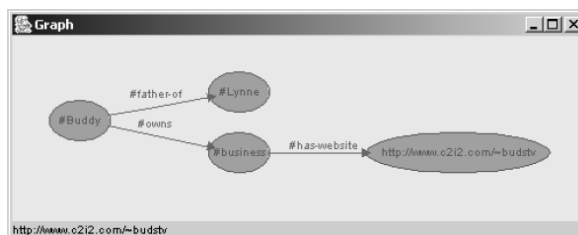
```
<rdf:RDF
  xmlns:RDFNsId1='#'
  xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'>
  <rdf:Description rdf:about='#Buddy'>
    <RDFNsId1:owns>
      <rdf:Description rdf:about='#business'>
        <RDFNsId1:has-website
          rdf:resource='http://www.c2i2.com/~budstv' />
        </rdf:Description>
      </RDFNsId1:owns>
      <RDFNsId1:father-of rdf:resource='#Lynne' />
    </rdf:Description>
  </rdf:RDF>
```

شکل 13: RDF/XML

در این شیوه هر یک از المان های description یک منبع (که در صفت rdf:about معرفی مشخص شده) را توصیف می کند.

– گراف

همانطور که قبلا هم نشان داده شد، یک روش دیگر برای نمایش جملات RDF استفاده از گراف ها می باشد. این شیوه نمایش برای انسان ها دارای خوانایی بیشتری است و ابزار هایی مانند IsaViz امکان نمایش و ویرایش گراف های RDF را ارائه می کند. شکل زیر نمایش گراف جملات فوق را با استفاده از ابزار IsaViz نشان می دهد:



شکل 14: گراف RDF

۲ – ۷ – آشنایی با RDF Schema (RDFS)

- RDF روشی رسمی جهت حاشیه نویسی متا دیتا در قالب XML را ارائه می کند. اما هیچ نوع اطلاعات جدیدی را به دایره لغات اضافه نمی کند (مانند Type و SubClassOf). از این رو RDF Schema [۱۳] امکان توصیف لغت نامه و ایجاد ارتباط معنایی بین لغات را فراهم می آورد. به این ترتیب اطلاعات معنایی بیشتری به گزاره ها و منابع، اضافه می گردد. بطوریکه با استفاده از آن می توان از اطلاعات موجود تفسیر و استنتاج نمود.

ذیلا مثال هایی از RDFS را ارائه می کنیم:

```
<Person,type,Class>
<hasColleague,type,Property>
<Professor,subClassOf,Person>
<Carole,type,Professor>
<hasColleague,range,Person>
<hasColleague,domain,Person>
```

مدل داده ای توصیف شده توسط RDS با مدل داده ای استفاده شده در زبان های برنامه سازی شی گرا (مانند جاوا) یکسان می باشد [۳]. با استفاده از RDFS می توان کلاس ها و صفات و روابط سلسله مراتبی و ارث بری بین کلاس ها را توصیف نمود. علیرغم قابلیت های RDFS در توصیف نوع های داده ای و ایجاد رابطه ارث بری بین کلاس ها، RDFS برای توصیف جزئیات منابع، بسیار ضعیف می باشد. برخی ضعف های RDFS عبارتند از:

- نمی توان بصورت محلی شرایط دامنه را محدود نمود. مثلا زمانی که hasChild در ارتباط با Person استفاده می شود باید فقط مقادیر از نوع Person را بپذیرد و زمانی که در ارتباط با Elephant استفاده می شود باید فقط مقادیر از نوع Elephant را بپذیرد.
- شرط وجود یا کمیت^{۳۴} را نمی توان در روابط تعیین نمود. به عنوان مثال نمی توان گفت که هر فرزند باید حداقل باید یک پدر و یک مادر داشته باشد.
- نمی توان صفاتی از نوع تراگذری (Transitive) و یا متضاد (Inverse) و یا هم ارزی را تعریف نمود. مثلا نمی توان گفت IsPartOf خاصیت تراگذری دارد و با hasPart متضاد است.

از این رو پشتیبانی از قواعد استنتاج در RDF(S) مشکل می باشد.

ابزارها

ابزارها و چارچوب^{۳۵} های مختلفی در ارتباط با RDF تولید شده است. که ذیلا جهت آشنایی اولیه برخی از آنها را فقط نام می بریم:

Cardinality³⁴
Framework³⁵

- **پارسر ها:** این ابزار ها فایل RDF ورودی را دریافت کرده و المان های آن را خوانده در حافظه نگهداری می کنند تا مورد استفاده در سایر برنامه ها قرار گیرد. مانند: Jena, SWI-Prolog, RedLand
- **Validators:** این دسته از ابزار ها فایل RDF ورودی را دریافت کرده و اعتبار آن را بررسی می کنند. مانند: W3C RDF validation service [۱۴]
- **ویرایش گر ها:** با استفاده از این ابزار ها می توان براحتی فایل های RDF را بصورت گرافیکی یا درختی ایجاد نمود. مانند: InferEd, RDFEd, RDF Author, IsaViz
- **ذخیره سازی و نگهداری:** این ابزار ها چارچوبی را جهت ذخیره سازی جملات RDF و پرس و جو از میان آنها فراهم می کنند. مانند: RDFLib, RSSDB, Sesame
- **پرس و جو (Query):** این بخش شامل زبان های پرس و جو جملات RDF می باشند. مانند: RDQL

۲- ۸- معرفی Taxonomy ها

اولین قدم برای حرکت به سمت وب معنایی توصیف Taxonomy ها به شکل قابل استفاده برای ماشین ها است. Taxonomy روشی جهت مقوله بندی^{۳۶} و طبقه بندی^{۳۷} مجموعه ای از اشیاء می باشد. اغلب طبقه بندی ها در Taxonomy به شکل سلسله مراتبی^{۳۸} می باشد. تعریف Taxonomy در فلسفه بصورت زیر است^{۳۹}:

- مطالعه اصول عمومی طبقه بندی؛ طبقه بندی سیستماتیک؛ خصوصاً طبقه بندی منظم و مرتب گیاهان و حیوانات مطابق با روابط طبیعی استنباط شده از آنها
 - رشد سریع تکنولوژی اطلاعات (IT) معنای Taxonomy را تاحدی تغییر داده است و این اصطلاح، برای مفاهیم دیگری، همچون محصولات و خدمات الکترونیکی بکار گرفته شد:
 - تعریف IT برای Taxonomy: طبقه بندی موجودیت های اطلاعاتی بشکل سلسله مراتبی. مطابق با روابط استنباط شده از موجودیت های جهان هستی ای که نمایش می دهند.
- شکل زیر یک Taxonomy ساده را نشان می دهد.

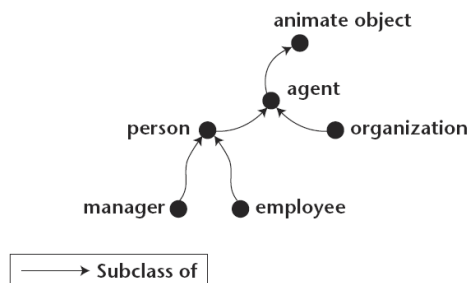
³⁶ Categorizing

³⁷ Classifying

³⁸ Hierarchy: منظور از سلسله مراتبی آن است که همانند یک درخت دارای یک ریشه و چند فرزند باشد (به هریک از فرزندان یک نود گفته می

شود).

³⁹ فرهنگ لغت Marrian-Webster, www.m-w.com



شکل ۱۵: یک Taxonomy ساده

هر نود Taxonomy یک موجودیت اطلاعاتی است که نشان دهنده یک موجودیت در جهان واقعی می باشد. هر یک از رابطه های بین نود ها نشان دهنده یک رابطه بخصوص (بنام رابطه SubClass و SuperClass) می باشد. شکل فوق نشان می دهد که کلاس Person دارای زیر کلاس ها manager و employee و دارای سوپر کلاس agent می باشد.

در ساختار سلسله مراتبی Taxonomy ها هر چه که به سمت ریشه نزدیکتر می شویم، کلاس ها عمومی تر می شوند. و هر چه که به سمت برگ ها نزدیکتر شویم، کلاس ها خاص تر می شوند. به این نوع از طبقه بندی **Generalization/Specialization Taxonomy** گویند.

البته باید به این نکته نیز توجه داشت که زیر کلاس ها دارای یک خصوصیت منحصر به فرد می باشند که آنها را از سایرین مجزا می سازد. به عنوان مثال در شکل فوق manager دارای این خصوصیت است که هر مدیر، چندین کارمند تحت امر خود دارد. حال سوال اینجاست که آیا واقعا manager و employee از هم جدا می باشند؟ مسلما نه و یک رابطه از نوع Aggregation بین آن دو برقرار است. که نشان می دهد که هر مدیر، چند کارمند را مدیریت می کند. این یک رابطه جدید بین موجودیت های اطلاعاتی است که منجر به تشکیل نوع جدیدی از طبقه بندی، بنام **Association Taxonomy** می شود.

همانطور که گفته شد، Taxonomy ها برای طبقه بندی موجودیت ها مناسب می باشند. بدین معنی که با استفاده از آنها می توان حداقل یک سمانتیک ساده را برای یک فضای اطلاعاتی^{۴۰} فراهم نمود. شکل زیر یک نوع دسته بندی معروف که در تجارت الکترونیک کاربرد دارد، بنام دسته بندی خدمات و محصولات استاندارد جهانی^{۴۱} یا بطور خلاصه UNSPSC را نشان می دهد:

Comment[i]:

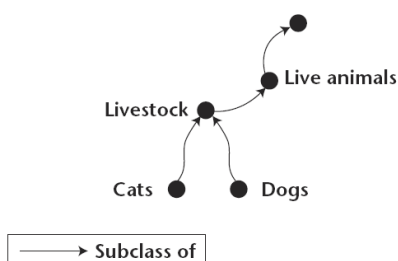
⁴⁰ information space

⁴¹ Universal Standard Product and Service Classification

SEGMENT	FAMILY	CLASS	COMMODITY	TITLE
10	00	00	00	Live Plant and Animal Material and Accessories and Supplies
10	10	00	00	Live Animals
10	10	15	00	Livestock
10	10	15	01	Cats
10	10	15	02	Dogs

شکل ۱۶: بخشی از طبقه بندی UNSPSC

هر چند Taxonomy فوق به شکل جدولی نشان داده شده است ولی می توان آن را بصورت درختی زیر نیز نشان داد (با توجه به روش کد گذاری):



شکل ۱۷: بخشی از طبقه بندی UNSPSC

Taxonomy جهت دسته بندی موجودیت های اطلاعاتی مناسب می باشند. آنها حداقل سمانتیک مورد نیاز برای مسائل مربوط به ارتباط بین اشیاء یک فضای اطلاعاتی را فراهم می آورند. آنها روشی جهت سازماندهی و توصیف متادیتا می باشند. و در واقع مدلی ساده از اجزاء قابل تفکیک (از دیدگاه ما) را ارائه می کنند. از طرفی دیگر Taxonomy ها باعث کاهش افزونگی اطلاعات می گردند چراکه بخاطر ساختار درختی آن، هنگام تعریف رابطه ها، اطلاعات گره پدر در رابطه ها تکرار نمی گردد.

چرا از Taxonomy ها استفاده می کنیم؟

در نظر بگیرید که می خواهید با استفاده از یک موتور جستجو، مانند گوگل، مطلبی را در وب جستجو کنید. برای این کار باید ابتدا چند کلمه کلیدی از موضوع مورد جستجو را وارد نمایید تا موتور جستجو بتواند آن را در صفحات وب جستجو نماید. حال اگر شما یک تصور کلی درباره موضوع مورد نظر خود داشته باشید و کلمات کلیدی مناسب را ندانید، چه کار باید کرد؟ از آنجا که تمرکز Taxonomy ها بر روی دسته بندی اطلاعات است، بنابراین می توانند به موتور های جستجو کمک کنند تا با سرعت و دقت بیشتر، موجودیت های اطلاعاتی را جستجو کنند.

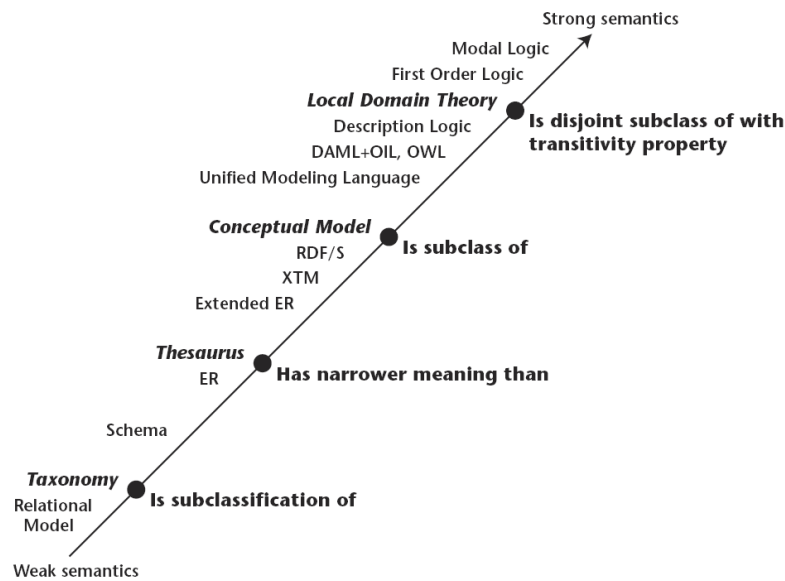
Taxonomy هایی مانند فرهنگ مفهومی^{۴۲} و آنتالوژی ها روشی جهت سازماندهی داده ها و موجودیت های اطلاعاتی و ایجاد حداقل سمانتیک می باشند. در وب Taxonomy ها می توانند جهت کمک به مشتریان برای یافتن محصولات و خدمات مورد نظرشان موثر باشند. همچنین می توانند از طریق دسته بندی علاقه مندی ها ما را در کنترل نیازمندی های اطلاعاتی مان کمک نمایند.

جهت دسته بندی محصولات و خدمات موجود در وب Taxonomy هایی مانند UNSPSC و NAICS (سیستم طبقه بندی صنعتی آمریکای شمالی^{۴۳}) وجود دارند. حال در نظر بگیرید که سیستمی که با UNSPSC کار می کند بخواهد داده های خود را با سیستم NAICS به اشتراک بگذارد. از آنجا که روش های طبقه بندی اطلاعات در این دو سیستم متفاوت می باشد، بنابراین باید بین آنها یک قاعده نگاشت موجودیت های اطلاعاتی را مشخص نمود. در حالت کلی، مساله اتوماتیک کردن نگاشت بین Taxonomy ها، یکی از چالش های موجود در این زمینه است (البته این چالش در رابطه با آنتالوژی ها نیز همچنان وجود دارد که بعدا آن را تشریح خواهیم نمود).

Thesaurus⁴²
North American Industry Classification System⁴³

۲-۹- معرفی طیف آنتالوژی^{۴۴}

مفاهیمی مانند Taxonomy، فرهنگ جامع مفهومی (Thesaurus) و مدل های مفهومی^{۴۵} و تئوری های منطقی همگی سعی می کنند تا تکنیک هایی را در راستای بازنمایی و طبقه بندی و شفاف سازی مفاهیم معنایی ارائه کنند. شکل زیر روابط میان مفاهیم فوق را نشان می دهد [۳].

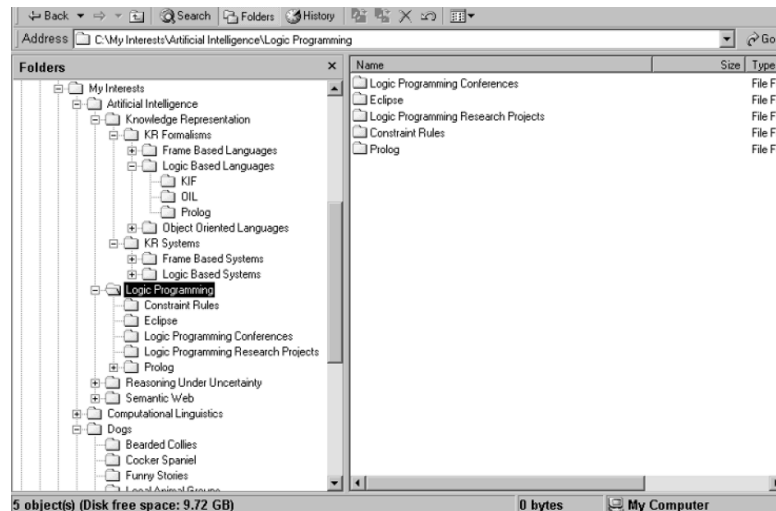


شکل ۱۸: طیف آنتالوژی: سمانتیک ضعیف تا سمانتیک قوی

شکل فوق جهت سنجش مقدار سمانتیک موجود در طبقه بندی و مدل های مبتنی بر دانش استفاده می شود. هر قدر در این طیف به سمت بالا حرکت می کنیم، میزان سمانتیک نیز افزایش پیدا می کند. بنابراین قطب پایین سمت چپ را سمانتیک ضعیف و قطب بالا سمت راست را سمانتیک قوی می نامیم و منظور آن است که میزان قابلیت توصیف و تشریح معانی از قطب ضعیف به قطب قوی افزایش پیدا می کند. در قطب ضعیف تنها مفاهیم ساده قابل توصیف هستند ولی در قطب قوی هر مفهوم پیچیده ای را می توان توصیف نمود.

- **Taxonomy**: در Taxonomy، همانطور که گفته شد، سمانتیک روابط پدر و فرزندی موجودیت های اطلاعاتی مشخص می گردد. در یک Taxonomy ضعیف روابط تعریف شده ممکن است زیر کلاسی از روابط اصلی و یا بخشی از آن باشد و یا حتی روابط اصلی در

Taxonomy تعریف نشده باشد. مانند ساختار درختی دایرکتوری های سیستم عامل ویندوز که شکل زیر یک نمونه از آن را نشان می دهد:



شکل ۱۹: ساختار درختی زیردایرکتوری ها در سیستم عامل ویندوز

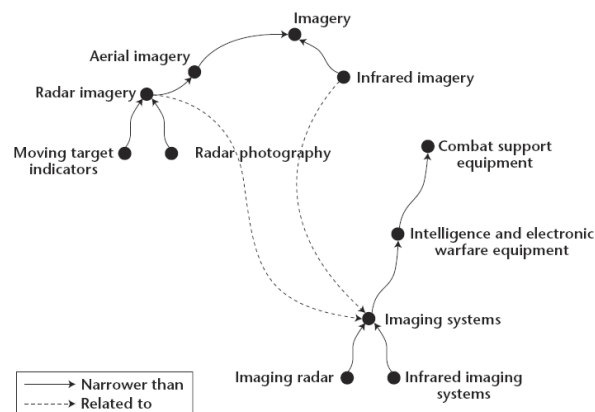
همانطور که دیده می شود شاخه Logic Programming Research زیر دایرکتوری Logic Programming تعریف شده است که ممکن است واقعا بطور کامل یک زیر کلاس از آن نباشد.

- **فرهنگ جامع مفهومی (Thesaurus):** یک فرهنگ لغت کنترل شده، که با یک ترتیب شناخته شده و ساخت یافته، مرتب شده است. بطوریکه روابط هم ارزی و هم شکلی (از نظر املاء) و سلسله مراتبی و شرکت پذیری، بطور شفاف نشان داده شده است. هدف اصلی در فرهنگ جامع مفهومی تسهیل در بازیابی مستندات و سازگاری در ایندکس گذاری مستندات می باشد. روابط موجود در این فرهنگ عبارتند از Equivalence و Homographic و Hierarchical و Associative. جدول زیر مفاهیم روابط فوق را به همراه یک مثال تشریح می کند:

SEMANTIC RELATION	DEFINITION	EXAMPLE
<i>Synonym</i> Similar to Equivalent Used for	A term <i>X</i> has nearly the same meaning as a term <i>Y</i> .	"Report" is a synonym for "document."
<i>Homonym</i> Spelled the same Homographic	A term <i>X</i> is spelled the same way as a term <i>Y</i> , which has a different meaning.	The "tank," which is a military vehicle, is a homonym for the "tank," which is a receptacle for holding liquids.
<i>Broader Than</i> (Hierarchic: parent of)	A term <i>X</i> is broader in meaning than a term <i>Y</i> .	"Organization" has a broader meaning than "financial institution."
<i>Narrower Than</i> (Hierarchic: child of)	A term <i>X</i> is narrower in meaning than a term <i>Y</i> .	"Financial institution" has a narrower meaning than "organization."
<i>Associated</i> Associative Related	A term <i>X</i> is associated with a term <i>Y</i> , i.e., there is some unspecified relationship between the two.	A "nail" is associated with a "hammer."

شکل ۲۰: انواع روابط معنایی در یک فرهنگ جامع مفهومی

فرهنگ جامع مفهومی نوعاً برای ایجاد ارتباط بین معانی پیچیده اصطلاحات مختلف استفاده می‌شود. شکل زیر نشان می‌دهد که Radar Imagery مفهومی محدود تر از Aerial Imagery بوده و نیز با Imaging System مرتبط است.



شکل ۲۱: نمونه ای از یک Thesaurus

فرهنگ جامع مفهومی اطمینان می‌دهد که: (۱) مفاهیم به یک شیوه سازگار تعریف شده باشند و (۲) کاربران با تجربه قادر خواهند بود تا روش جستجوی اطلاعات مورد نیازشان را بهبود بخشند. (۳) کاربران نیازی ندارند تا به اطلاعات تکنیکی و اصطلاحات محلی آشنا باشند. یکی از معروف ترین فرهنگ های مفهومی ها، WordNet می باشد. WordNet مطابق با تئوری های روان شناختی در ارتباط با حافظه واژگانی بشر، طراحی شده است. در WordNet یک کلمه ممکن است یک یا چند تعریف متمایز داشته باشد. به هریک از تعریف های متمایز کلمه یک Word Sense گفته می شود. در WordNet، به ازای هر کلمه اطلاعاتی مانند هم معنی

های آن (Synonyms) و کلمات با مفهوم عام تر (Hypernyms) و کلمات با مفهوم خاص تر (Hyponyms) وجود دارد.

- **مدل های مفهومی (Conceptual Models) :** قدم بعدی در جهت غنی سازی سمانتیک استفاده از مدل های مفهومی است. مدل مفهومی عبارت است از مدلی مجرد که روابط و قواعد معمول میان اجزاء اصلی یک فضای اطلاعاتی را ترسیم می کند. مدل مفهومی در واقع یک دید از خارج به درون است و نشان می دهد که اطلاعات موجود در جهان خارج چگونه دیده می شوند. این مدل یک مدل مجرد است، بنابراین شامل جزئیات نمی باشد تا بر روی مفاهیم اصلی سیستم تاکید گردد.

در مدل های مفهومی می توان براحتی رابطه زیر کلاس بودن را بین کلاس پدر و کلاس فرزند مشخص نمود. در این خصوص، زبان های مدل سازی برنامه نویسی شی گرا (مانند UML) جهت توصیف سمانتیک زیر کلاس بودن کلاس ها، به اندازه کافی قدرتمند هستند. همچنین مهم است که تعریف کلاس ها و سوپر کلاس ها و زیر کلاس ها در سطح متا مدل بخوبی انجام شده باشد تا کلاس های موجود در سطح مدل شی (مانند کلاس Person و زیر کلاس Manager)، از نظر سمانتیکی براحتی قابل توصیف باشند.

در این سطح از سمانتیک، مدل سازی در ۳ سطح مختلف انجام می شود:

۱. **سطح Meta-Model :** این همان سطحی است که در آن زبان های مدل سازی مفهومی (مانند UML) تعریف می شوند. در اینجا، محصولاتی مانند کلاس ها، روابط و صفت ها که از آنها در سطح object-level استفاده می کنیم، تعریف می گردند.

۲. **سطح Object-Model :** در این سطح ما مدل سیستم مورد علاقه خود و دامنه آن را مشخص می کنیم. در این مدل از کلاس ها و صفات تعریف شده در سطح Meta-Model استفاده می کنیم.

۳. **سطح Content-Model :** با توجه به مدل سیستم که در سطح Object-Level تعریف شده است، در این سطح، موجودیت های اطلاعاتی توصیف می گردند.

مدل ER (و نیز مدل EER) که جهت توصیف اسکیمای مفهومی یک DB استفاده می شود، به عنوان زبان مدل سازی مفهومی مورد توجه است. هنگام ایجاد یک DB ابتدا یک اسکیمای مفهومی (که در آن مفاهیم اولیه دامنه پایگاه داده مدل سازی شده است) می سازیم. سپس آن را به یک اسکیمای منطقی محدود می کنیم و در نهایت آن را به اسکیمای فیزیکی تبدیل می کنیم. این اسکیمای سطوح مختلفی از تجرید را نشان می دهند که عبارتند از : سطح مفهومی انسان، سطح جدول و ستون پایگاه داده، سطح پیاده سازی جدول ها و ستون ها و کلید ها

• **تئوری منطقی:** در وب معنایی هدف نهایی در ارتباط با آنتالوژی ها، آن است که بتوان آنها را بشکل یک تئوری منطقی نمایش داد. چرا که در این صورت، توسط نرم افزار ها (ماشین) قابل استنتاج خواهند بود. از این رو اهم فعالیت های جاری در مهندسی آنتالوژی و نمایش دانش، آن است که بتوانند آنتالوژی را به عنوان یک تئوری منطقی بنا نهند.

آنتالوژی: آنتالوژی توصیف کننده ی مجموعه ای از کلمات و مفاهیم است که جهت تشریح و نمایش یک بخش از دانش بکار می روند. آنتالوژی می تواند توسط انسان ها، بانک های اطلاعاتی و برنامه های کاربردی، جهت به اشتراک گذاری اطلاعات یک دامنه (Domain) مورد استفاده قرار گیرد.

بدلیل اهمیت و جایگاه ویژه آنتالوژی ها در وب معنایی، ادامه بحث را به معرفی بیشتر آنتالوژی و ارتباط آن با منطق می پردازیم.

۲-۱۰- معرفی آنتالوژی

تعریف آنتالوژی در تکنولوژی اطلاعات (IT):

• (۱) یک آنتالوژی کلمات و مفاهیم (مورد استفاده جهت توصیف و نمایش یک دامنه^{۴۶}) را تعریف می کند. (۲) آنتالوژی یک محصول مهندسی شامل یک لغتنامه مشخص جهت توصیف بخشی از واقعیات، به همراه مجموعه ای از فرضیه های واضح و شفاف (با توجه به معنای مرتبط در یک لغتنامه) می باشد. به عبارتی دیگر، تعیین و توصیف مفاهیم ذهنی^{۴۷}.

تعریف اول دارای دو بخش است: (۱) آنتالوژی یک حوزه از دانش (دامنه) را توصیف کرده و نمایش می دهد. (۲) کلمات و مفاهیم توصیفات را تعریف می کند.

منظور از توصیف یک حوزه ای از دانش، هنر تشریح و بیان نوشتاری یا گفتاری بخش های مهم آن حوزه از دانش می باشد. برای مثال هنگام توصیف حوزه تعمیرات اتومبیل، احتمالا درباره موارد زیر صحبت می کنیم. (۱) انواع ماشین های موجود (سدان، بنز، استیشن واگن و غیره)، (۲) انواع موتورها (گاز سوز، بنزینی، گازوئیلی دیزلی، برقی و مختلط) (۳) مشخصات موتور (V-6 Ford و غیره) (۴) سازنده (ولو، هیوندای و غیره) (۵) اجزای تشکیل دهنده ماشین (موتور ها، سیستم های الکتریکی، سیستم های سرما ساز، شاسی و بدنه و غیره)

⁴⁶ در اینجا منظور از دامنه، محتوای یا مضمون یک ناحیه ای از دانش می باشد. مانند پزشکی، تعمیرات اتومبیل، طرح ریزی مالی، مدیریت اقتصادی و غیره.

هنگام توصیف یک حوزه از دانش (دامنه) اشیاء مهم آن دامنه، خصوصیات آنها و روابط میان آنها را توصیف می کنیم. همچنین ممکن است قواعدی را درباره آن دامنه مطرح کنیم. به عنوان مثال؛ قاعده یافتن نقص های فنی یک - اگر ماشین استارت نمی زند، باتری و اتصالات آن را چک کنید و ...

بنابراین می توان گفت که یک توصیف می تواند یک آنتالوژی باشد. یک آنتالوژی شامل بخش های زیر است: (۱) کلاس ها (۲) نمونه ها (۳) روابط میان اشیاء (۴) خصوصیات اشیاء (۵) عملیات و فرایندهای در ارتباط با اشیاء (۶) شرایط و قواعد مرتبط با اشیاء

علاوه بر توصیف یک حوزه از دانش، نیاز است تا آن توصیف را نمایش داد. بدین معنی که یک توصیف را بگونه ای کد کنیم تا دیگران نیز بتوانند از آن استفاده نمایند. یک توصیف، شامل کلمات و عباراتی به زبان طبیعی، مانند؛ انگلیسی، چینی یا فارسی می باشد. لذا در صورت نیاز، ابتدا کلمات و اصطلاحات خود را تعریف می نماییم. سپس آنها را جهت توصیف حوزه دانش خود با هم ترکیب می کنیم.

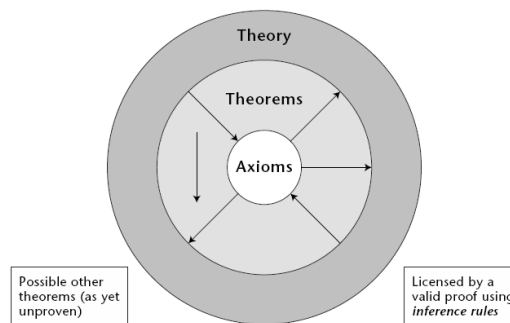
در حوزه IT، مفهوم نمایش کمی پیچیده تر می باشد. در واقع، در حوزه IT، باید مدلی را خلق کنیم که نرم افزار قادر به استفاده از آن باشد. بدین منظور، یک حوزه از دانش را با کلاس ها، نمونه ها، خصوصیات، روابط و قواعد آن نمایش می دهیم. ما از اصطلاحات توصیفی زبان طبیعی، به عنوان برچسب یا نام مفاهیم اساسی (کلاس ها، خصوصیات، روابط) استفاده می کنیم.

معمولا جهت نمایش آنتالوژی ها بجای زبان طبیعی از زبان منطقی بازنمایی دانش، استفاده می کنیم.. چرا که می خواهیم توصیفات خود را تا حد امکان، دقیق و شفاف و غیر مبهم ارائه کنیم در حالیکه زبان های طبیعی خیلی مبهم می باشند. همچنین می خواهیم که مفاهیم و توصیفات مورد نظر جهت استفاده در تکنولوژی اطلاعات آماده باشد.

۲- ۱۱- توصیف منطقی آنتالوژی ها

همانطور که قبلا گفته شده، آنتالوژی ها معمولا با یک زبان بازنمایی دانش مبتنی بر منطق، تشریح می گردند. برخی از ابزارهای آنتالوژی ها، می توانند بصورت اتوماتیک از آنتالوژی استفاده کرده و خدمات پیشرفته تری را به برنامه های هوشمند مانند؛ جستجو های معنایی / مفهومی، پشتیبانی در تصمیم گیری، فهم زبان مادری، مدیریت دانش بانک های هوشمند و تجارت الکترونیکی ارائه کنند.

یک آنتالوژی می تواند از یک طبقه بندی (Taxonomy) ساده تا یک تئوری منطقی پیچیده را شامل شود. از نظر تکنیکی یک آنتالوژی، هم یک فرهنگ لغات مورد استفاده در توصیف و نمایش یک ناحیه از دانش و نیز به معنای یک فرهنگ لغات می باشد.



شکل ۲۲: تئوری، فرضیه و اصول بدیهی

معمولا یک آنالوژی را به شکل یک سیستم بدیهی^{۴۸} یا به عبارتی یک مجموعه از اصول بدیهی و قواعد استنتاج، نشان می دهند. که این دو (اصول بدیهی و قواعد استنتاج) با هم یک مجموعه از فرضیه ها را مشخص می کند. که در مجموع یک تئوری را تشکیل می دهند (شکل قبل).

از دید تکنیکی، آنالوژی یک فرهنگ لغت برای توصیف موجودیت ها و روابط یک مدل مفهومی می باشد. هر دوی فرهنگ لغات و شرایط معنایی^{۴۹} برای ارتباط بین مدل اطلاعاتی و دامنه جهان حقیقی مرتبط با آن، لازم می باشد. شکل فوق از نظر معنایی می خواهد نشان دهد که، فرضیه ها با استفاده از اصول بدیهی و قواعد استنتاج قابل اثبات می باشد. اصول بدیهی، قواعد استنتاج و فرضیه ها، همگی باهم تشکیل یک تئوری را می دهند.

AXIOMS	INFERENCE RULES	THEOREMS
Class(Thing)	And-introduction: Given P, Q , it is valid to infer $P \wedge Q$.	If $P \wedge Q$ are true, then so is $P \vee Q$.
Class(Person)	Or-introduction: Given P , it is valid to infer $P \vee Q$.	If X is a member of Class(Person), then X is a member of Class(Person).
Class(Parent)	And-elimination: Given $P \wedge Q$, it is valid to infer P .	If X is a member of Class(Parent), then X is a member of Class(Person).
Class(Child)	Excluded middle: $P \vee \neg P$ (i.e., either something is true or its negation is true)	If X is a member of Class(Child), then NameOf(X, Y) and Y is a String.
If SubClass(X, Y), then X is a subset of Y . This also means that if A is a member of Class(X), then A is a member of Class(Y).		If Person (JohnSmith), then $\neg$ ParentOf(John Smith, JohnSmith).
SubClass(Person, Thing)		
SubClass(Parent, Person)		
SubClass(Child, Person)		
ParentOf(Parent, Child)		
NameOf(Person, String)		
AgeOf(Person, Integer)		
If X is a member of Class (Parent) and Y is a member of Class(Child), then $\neg (X=Y)$.		

شکل ۲۳: تئوری، فرضیه و اصول بدیهی

جدول فوق بخش از یک آنتالوژی ارائه شده به شکل اصول بدیهی و قابل استنتاج را نشان می دهد. ستون اول، اصول بدیهی می باشد. که همواره درست فرض می شوند. ستون دوم، نشان دهنده قواعد استنتاج و ستون سوم، نشان دهنده قضیه ها می باشد. قضیه، فرضیه ای است که درستی با نادرستی آن هنوز اثبات نشده است. در زمانیکه درستی آن اثبات شد، می توان آن را به مجموعه اصول بدیهی اضافه نمود. قضایا توسط فرایندی بنام برهان^{۵۰} اثبات می گردند.

اثبات یک قضیه به این معنی است که با فرض وجود مجموعه ای از اصول بدیهی، با دنبال کردن قواعد استنتاج در این مجموعه از اصول بدیهی بتوان به ادعا های اولیه رسید. که در این صورت قضیه اثبات شده و معتبر می شود. مجموعه ای از اصول بدیهی، قواعد استنتاج و قضایا با همدیگر تشکیل یک تئوری را می دهند. که به همین دلیل نیز در طیف آنتالوژی، به آن تئوری منطقی گفته می شود.

۲-۱۲- سطوح بازنمایی دانش

وقتی که از آنتالوژی ها صحبت می کنیم، لازم است تا با تفاوت های میان سطوح نمایش دانش آشنا باشیم. این شناخت لازم است، چرا که آنتالوژی ها می توانند در قالب زبان ها و یا لغتنامه نحوی به همراه سمانتیک، نشان داده شوند. از طرفی از آنجا که آنتالوژی ها محتوی هستند و محتوی تنها می تواند توسط زبان محتوی (که معمولا به آن زبان بازنمایی دانش گفته می شود) نمایش داده شوند. بنابراین حداقل دوسطح برای بازنمایی وجود دارد. یک سطح زبان بازنمایی دانش که با توجه به سطوح پایینی آن (سطح شی) به آن سطح متا (Meta) نیز گفته می شود. سطح دوم، سطح شیء (Object Level) است که در آن آنتالوژی ها توصیف می گردند. واضح است که یک سطح سوم که همان سطح نمونه^{۵۱} ها است نیاز داریم. در این سطح سوم، نمونه هایی از کلاس های آنتالوژی وجود دارند. از طرفی اگر در سطح نمونه ها تمرکز کنیم، می توان آن را سطح شیء در نظر گرفت و سطح متا سطح بالایی آن یعنی سطح دوم خواهد بود که در آن، آنتالوژی ها تشریح می گردند. همچنین اگر در سطح دوم تمرکز کنیم می توان آن را بشکل سطح اشیاء دید و سطح بالایی آن یعنی سطح یک که در آن زبان بازنمایی دانش تعریف می گردد، سطح متا خواهد بود. پس می توان گفت که سطوح متا و شیء نسبی می باشند. جدول زیر سطوح نمایش آنتالوژی ها را به همراه یک مثال نشان می دهد.

- سطح اول: سطح بازنمایی دانش است. این سطح، سازه های مورد استفاده در سطح مفاهیم آنتالوژی را تعریف می کند. این سازه ها، شامل مفاهیم؛ کلاس، رابطه، خصوصیات و غیره

است. مثال هایی از این سطح عبارتند از زبان های CYCL, ONE, Ontolingua, KIF, UML, OWL و RDF/S و DAML+OIL

LEVEL	EXAMPLE CONSTRUCTS
Knowledge representation (KR) language (Ontology Language) level:	Class, Relation, Instance, Function, Attribute, Property, Constraint, Axiom, Rule
Meta level to the ontology concept level	
Ontology concept (OC) level:	Person, Location, Event, Parent, Hammer, River, FinancialTransaction, BuyingAHouse, Automobile, TravelPlanning, etc.
Object level to the KR language level, meta level to the instance level	
Ontology instance (OI) level:	Harry X. Landsford III, Ralph Waldo Emerson, Person560234, PurchaseOrder TransactionEvent6117090, 1995-96 V-6 Ford Taurus 244/4.0 Aerostar Automatic with Block Casting # 95TM-AB and Head Casting 95TM
Object level to the ontology concept level	

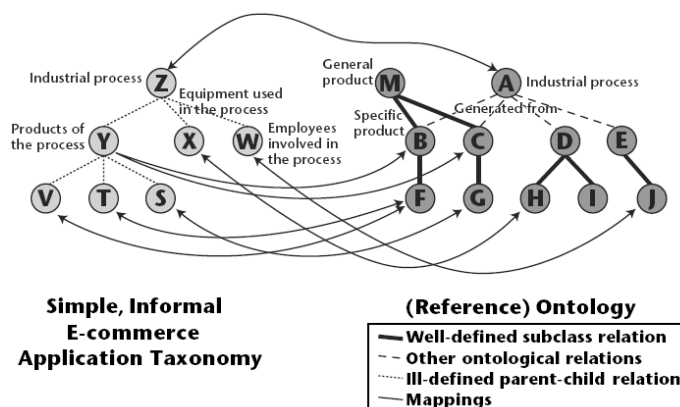
شکل ۲۴: سطوح بازنمایی آنتالوژی

- سطح دوم: سطح مفهوم آنتالوژی (Ontology Concept) است. در این سطح مفاهیم آنتالوژی ها با استفاده از سازه های سطح یک ایجاد می شوند. در این سطح باید مفاهیم کلی جهان را مدل سازی کنیم. دانش دامنه درباره Persons, Locations, Events و Parents و Hammers و تراکنش ها مالی و غیره
- سطح سوم: سطح نمونه های آنتالوژی (Ontology Instances). در این سطح، با توجه به سازه های تعریف شده در سطح فوق، نمونه های موجود در دامنه مورد بحث تعریف می گردد. به عبارتی می توان گفت، این سطح شامل پایگاه دانش یا پایگاه حقایق از ادعاها و گزاره هایی درباره نمونه ها می باشد. مانند؛ علی و رضا به عنوان نمونه هایی از کلاس Person.

۲- ۱۳- مشکل نگاشت معنایی

یکی از موضوعات مهم، در تفهیم و پیاده سازی آنتالوژی ها مشکل نگاشت معنایی آنتالوژی می باشد. این مشکل هر موضوعی در محدوده IT که با مسائل مربوط به سمانتیک سر و کار داشته باشد را تحت تاثیر قرار می دهد. مانند؛ سیستم ها، برنامه های کاربردی، بانک های اطلاعاتی و غیره.

گاهی لازم است دو آنتالوژی را که شبیه به هم هستند ولی چیزهایی که تفاوت های کمی با هم دارند را نمایش می دهند، نگاشت کنیم. و گاهی هم نیاز است تا یک آنتالوژی مرجع را با یک فرهنگ مفهومی، Taxonomy و یا آنتالوژی دیگر، بدون از دست دادن روابط معنایی، نگاشت کنیم. شکل زیر نگاشت از یک آنتالوژی به یک Taxonomy در تجارت الکترونیک را نشان می دهد.



شکل ۲۵: نگاشت بین مفاهیم آنالوژی ها

در سمت راست شکل، آنالوژی مرجع، به همراه روابط معنایی خوش تعریف قرار دارد. در سمت چپ هم Taxonomy ای که در تجارت الکترونیک استفاده می شود به همراه روابط معنایی خوش تعریف (ولی اندک) وجود دارد. در عمل، فقط اطلاعات نگاشت نگهداری می گردد، چرا که آنالوژی ها معمولاً توسط سازمان های مربوطه نگهداری می گردد. معمولاً نگاشت بین دو مفهوم آنالوژی و یا به عبارتی تعیین هم ارزی آنالوژی ها، عملی سخت بوده و نیاز به دانش انسانی دارد. حتی اگر نام دو مفهوم یکسان باشد، هیچ ضمانتی وجود ندارد که آن دو مفهوم یک معنا را داشته باشند. از این رو تعیین هم ارزی معنایی و ایجاد نگاشت بین دو مفهوم یک فرایند نیمه اتوماتیک است [۳].

۲- ۱۴- خلاصه مطالب

بطور خلاصه می توان گفته که وب معنایی عبارت است از ایجاد تسهیلاتی جهت جستجو، اشتراک و یکپارچه سازی اطلاعات، از طریق قابل فهم کردن اطلاعات موجود در وب کنونی برای ماشین ها. هر چند که چالش جاری جستجو کورکورانه اطلاعات، یک انگیزه قوی برای روی آوردن به وب معنایی می باشد ولی همانطور که در تعریف وب معنایی آمده است، جستجو تنها موضوعی نیست که در وب معنایی بر آن تاکید شده است. اشتراک و یکپارچه سازی اطلاعات نیز در لیست اهداف وب معنایی قرار دارد.

معماری پشته ای وب معنایی، پشته ای از تکنولوژی های مختلف را نشان می دهد که جهت پیاده سازی وب معنایی لازم است تا از آنها استفاده نمایید. XML شالوده و اساس این تکنولوژی ها را تشکیل می دهد و سایر این تکنولوژی ها بر مبنای استانداردهای معرفی شده در XML بنا شده اند. XML بخاطر قالب متنی و ساختار درختی آن، به سرعت مورد توجه اکثر شرکت ها و سازمان ها قرار گرفت. با استفاده از XML می توان براحتی داده ها را حاشیه گذاری کرده و به آن ساختار داد ولی نمی توان با آن معنای حاشیه گذاری ها و تگ ها را بیان نمود. بدین منظور زبانهای RDF-Schema جهت توصیف اشیاء موجود در یک دامنه و RDF جهت توصیف نمونه هایی از آن اشیاء، در وب معنایی مورد استفاده قرار می گیرند.

آنتالوژی روشی جهت مدل سازی معنایی یک دامنه بوده و یکی از تکنولوژی های مهم در وب معنایی می باشد. با استفاده از آنتالوژی می توانیم توصیفات خود از یک دامنه را مدل سازی نماییم. آنتالوژی ها توسط یک زبان بازنمایی دانش مانند OWL، نشان داده می شوند. آنها شرایط و قواعد منطقی موجود در یک دامنه را نشان می دهند. از این رو ارتباطی نزدیک با منطق دارند. با استفاده از منطق می توان از نمونه های موجود با استفاده از اصول و قواعد تعریف شده، به اطلاعات جدیدی رسید.

وب معنایی در موضوعاتی که به نوعی با سمانتیک داده ها و فرایندها در ارتباط هستند، کاربرد دارد. مانند:

- ارتباط هوشمند بین عامل ها^{۵۲}: اینکه دو عامل، بدون هیچ پروتکل گفتگو و توافق اولیه جهت برقراری ارتباط، بتوانند با هم ارتباط برقرار کنند.
- ترکیب وب سرویس های معنایی: اینکه دو یا چند وب سرویس را با هم ترکیب کرده تا نیاز معنایی تعریف شده توسط کابر را برآورده سازیم.
- جستجوی اخبار برای یافتن جدیدترین و مهمترین موضوعات
- برنامه نویسی معنایی: تسهیلات موجود در زبان های برنامه نویسی موجود، حاشیه گذاری های معنایی را امکان پذیر ساخته است. هدف آن است تا با استفاده از این حاشیه گذاری ها بتوان بین اجزای یک نرم افزار ارتباط معنای برقرار کرد (مثلا Pattern ها را یافته و اعمال نمود).
- تجارت هوشمند^{۵۳}: تجارت هوشمند سعی دارد تا با استفاده از منابع اطلاعاتی موجود؛ مدیران، کارمندان، خریداران و مشتریان را در تصمیم گیری های روزانه شان راهنمایی نماید. وب معنایی می تواند با اشتراک دانش و انجام استنتاج های معنایی باعث بهبود در تجارت هوشمند شود.
- یک پارچه سازی داده ها بر اساس آنتالوژی ها^{۵۴}: داده های موجود در یک دامنه که توسط افراد مختلف تولید می شوند دارای انواع ناهمگنی های لغوی، ساختاری و معنایی می باشند. هدف آن است تا بتوانیم آنها را بصورت یکپارچه ذخیره کرده یا جستجو نماییم.
- انطباق کاتالوگ ها^{۵۵}: کاتالوگ محصولاتی که توسط فروشندگان مختلف ارائه می شود حتی اگر درباره یک محصول (مانند تلفن همراه) باشد باز هم دارای ساختارهای متفاوت می باشد. هدف آن است که بتوان این کاتالوگ ها را یکپارچه نمود.

از آنجا که وب معنایی شاخه ای از وب جاری می باشد و از داده های موجود در وب جاری استفاده می کند، بنابراین استخراج داد های موجود و تبدیل آن به زبان قابل فهم در وب معنایی، یکی از چالش های موجود در وب معنایی می باشد که موضوع اصلی تحقیق ما را نیز تشکیل می دهد.

Agent Communication⁵²
Business Intelligence⁵³
Ontology Driven Data Integration⁵⁴
Cotolog Matching⁵⁵

۳- مدیریت دانش و بازیابی اطلاعات

۳-۱- بررسی مشکل

امروزه وب در بهره برداری موثر از دانش صریح و شفاف از یک سو و دانش ضمنی از سوی دیگر نقشی اساسی دارد. دانش صریح، بصورت منابع مختلف وب (مانند صفحات وب فایل های XML و RSS) ارائه می گردد. موتور های جستجوگر سنتی، با استفاده از کلمات کلیدی مرتبط، می توانند لیستی از مستندات مرتبط را به عنوان نتیجه جستجوی ارائه کنند. اما این نوع جستجو اغلب جوابگوی خواسته های کاربر جستجو کننده نمی باشد. در این حالت، کاربر می بایست زمان زیادی از وقت خود را صرف جستجو و خواندن مستندات نموده تا از آن طریق یا بوسیله لینک های موجود در سند به مستندات دیگر برسد و یا کلمات کلیدی مناسب تری را بیابد و جستجوی های دیگری را انجام دهد. دانش ضمنی از طریق محاوره با افراد خبره در یک دامنه بدست می آید. در این خصوص نیز وب می تواند با در اختیار قرار دادن ابزارهایی جهت ارتباط میان افراد با علاقه مندی ها و ارتباطات مشابه، بدون در نظر گرفتن موقعیت جغرافیایی، امکان این نوع محاوره را بوجود آورد.

جهت بهره برداری از دانش صریح و دانش ضمنی وجود در وب، به یک سیستم مدیریت دانش نیازمندیم. امروزه سیستم های مدیریت دانش، ضعف هایی اساسی دارند که عبارتند از [۱۷]:

- **جستجوی اطلاعات:** روش های موجود در بازیابی اطلاعات بر اساس کلمات کلیدی منتهی به بازیابی اطلاعات غیر مرتبط می گردد که محتوای آنها شامل کلمه کلیدی نظر بوده ولی در معنای دیگری استفاده شده است. همچنین در این روش برخی مستندات که شامل کلمه کلیدی نبوده ولی در ارتباط مستقیم با موضوع مورد نظر می باشند، در نتیجه جستجو نخواهند بود.
- **استخراج اطلاعات:** در حال حاضر، جهت یافتن اطلاعات مورد نظر باید تمام صفحاتی که ممکن است به نحوی با موضوع مورد نظر در ارتباط باشند را بررسی و مرور نمود. در حالیکه اگر عامل های نرم افزاری اتوماتیک، دارای ادراک انسانی در استخراج اطلاعات بودند می توانستند ما را سریعتر و راحت تر به منابع مورد نظر برسانند.
- **نگهداری ضعیف منابع متنی ساخت یافته:** وقتی که این منابع بزرگ و حجیم می شوند، مدیریت آنها کاری وقت گیر و مشکل می باشد. نگهداری و مدیریت این منابع حجیم بصورت پویا، سازگار و بروز شده، نیاز به بازنمایی معنایی مکانیزه ای دارد تا ما را در کشف آنومالی ها، کمک نماید.

- تولید اتوماتیک مستندات، امکان ایجاد وب سایت هایی را می دهد که قادرند خود را با توجه به پروفایل کاربران منطبق سازند. به عبارتی این گونه وب سایت ها با توجه به تنظیمات هر کاربر (زبان، رنگ و غیره) خروجی متفاوتی دارند. بازیابی داده های نیمه ساخت یافته از این صفحات وب، نیاز به سمانتیک بیشتری در نمایش صفحات وب دارد.

قدرت رقابت یک شرکت، به میزان استفاده آن از دانش و حافظه سازمانی، وابسته است. یکی از چالش های اساسی در ارتباط با این موضوع، دسترسی و بروز رسانی سریع به منابع اطلاعاتی می باشد. از این رو اکثر شرکت ها اطلاعات خود را دیجیتالی کرده تا بتوانند آن را در اینترنت به اشتراک پرسنل و یا مشتریان خود در آورند. لذا نیاز به ابزار هایی جهت مدیریت دانش می باشد تا بتوان اطلاعات مختلف توزیع شده در بین منابع گوناگون را یکپارچه ساخت.

اکثر تحقیقات گذشته در ارتباط با یکپارچه سازی اطلاعات مربوط به بانک های اطلاعاتی ناهمگن و پایگاه های دانش (که اطلاعات را بشکل کاملاً ساخت یافته ارائه می کرده اند) بوده است. اما در وب بسیاری از منابع بصورت نیمه ساخت یافته و یا کاملاً غیر ساخت یافته وجود دارند.

همانطور که گفته شد، وب معنایی شاخه ای از وب جاری است که همانند وب جاری قابل استفاده انسان ها از طریق مرورگر های وب می باشد. در وب معنایی، منابع دارای اطلاعات حاشیه گذاری شده می باشند. بطوریکه این متا دیتای تزریق شده در منابع اطلاعاتی، ماشین ها را قادر به پردازش اطلاعات موجود می سازد. لذا برای آنکه بتوانیم به چنین هدفی (پردازش اطلاعات توسط ماشین) برسیم، باید اولاً، اطلاعات موجود در وب جاری را به نحوی، بازیابی نماییم و ثانیاً، آنها را بگونه ای نگهداری و مدیریت نماییم که مورد استفاده جهت پردازش اطلاعات باشد. در این ارتباط، آنتالوژی ها نقش بسیار مهمی جهت حل مشکل بازنمایی های ناهمگن منابع وب ایفا می کنند. در ادامه به بررسی نقش و اهمیت آنتالوژی ها در مدیریت دانش می پردازیم.

۳-۲- نقش آنتالوژی ها

آنتالوژی ها مختص وب معنایی نبوده و از مدت ها قبل وجود داشته اند. آنها برای اولین بار در هوش مصنوعی، جهت بازنمایی، به اشتراک گذاری و استفاده مجدد دانش ابداع شده اند. در دهه ۹۰، آنتالوژی ها موضوع تحقیق عموم محققان شدند و کاربرد آن در بسیاری از رشته های هوش مصنوعی، مانند؛ مهندسی دانش، پردازش زبان طبیعی (NLP) و ... مورد بررسی قرار گرفت. امروزه از آنتالوژی ها در بسیاری از زمینه ها مانند، بازیابی اطلاعات، مدیریت دانش، تجارت الکترونیک، سیستم های اطلاعاتی شراکتی^{۵۶} و غیره بصورت وسیع استفاده می گردد.

⁵⁶ cooperative information systems

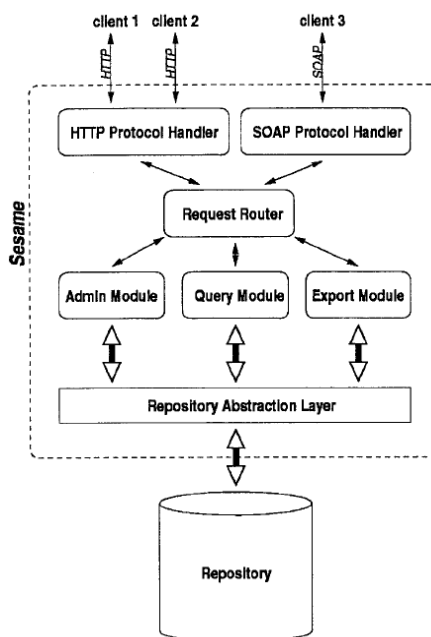
شعار اصلی آنتالوژی ها که توجه همه محققان را به خود جلب کرده، آن است که : اشتراک اطلاعات و فهم مشترک از یک دامنه، می تواند بین انسان ها و سیستم های کاربردی ارتباط برقرار سازد. بنابراین استفاده از آنتالوژی ها و ابزارهای آن، باعث بهبود قابل توجه در مدیریت دانش در سازمان های بزرگ خواهد شد.

ابزارهای مدیریت و نگهداری آنتالوژی ها باید شامل سه خصوصیت کلیدی زیر باشند [۱۷]:

- اکتساب آنتالوژی ها و ایجاد ارتباط بین آنها و انبوهی از داده ها. جهت افزایش میزان توسعه پذیری، این ویژگی باید مبتنی بر روش های اتوماتیک استخراج اطلاعات و پردازش زبان طبیعی (NLP) باشد.
- ذخیره سازی و نگهداری آنتالوژی ها و نمونه های آنها. همانطور که گفته شد، آنتالوژی ها را می توان با هر زبانی توصیف نمود. هر چه زبان بیان آنتالوژی ها تشریحی تر و دارای قدرت بیشتری باشد، به همان نسبت، سمانتیک بیشتری را می توان تعریف نمود. از طرفی پیاده سازی زبان های تشریحی تر دارای پیچیدگی بیشتری بوده و هزینه اجرایی بیشتری دارند.
- پرس و جو و پیمایش منابع اطلاعاتی دارای سمانتیک. موتور های جستجوی معنایی، نیاز مندی پرس و جوی کاربران را دریافت کرده و با اجرای آن توسط یک موتور استنتاج معنایی امکان مرور نتایج جستجو را برای کاربران فراهم می کند.

۳-۳ - معرفی معماری Sesame

Sesame [۱۷ و ۱۸ و ۳۱] یک معماری جهت ذخیره سازی و پرس و جوی اطلاعات ارائه شده توسط آنتالوژی ها می باشد. شکل زیر معماری Sesame را نشان می دهد [۱۸].



شکل ۲۶: معماری Sesame

SESAME می تواند به دو طریق مورد استفاده قرار گیرد. یکی بصورت سرور مستقل جهت ذخیره سازی RDF و RDFS و دیگری بصورت کتابخانه ای از توابع آماده و نوشته شده به زبان جاوا [۳۱].

مخزن، محل نگهداری داده های ماندگار و دائمی می باشد. مخزن، ناخواسته، تداعی گر سیستم مدیریت بانک اطلاعاتی (DBMS) می باشد. سیستم مدیریت بانک اطلاعاتی (DBMS) برای سال ها موضوع اصلی تحقیقات در ارتباط با ذخیره سازی داده های حجیم بود. در آن زمان ها انواع مختلفی از بانکهای اطلاعاتی پیاده سازی شد که هر کدام دارای نقاط قوت و ضعفی بوده و برای یک پلت فورم خاصی طراحی شده بودند. اما مخزن داده های Sesame مستقل از بانک اطلاعات رابطه ای می باشد و این یکی از مزیت های مهم Sesame می باشد. چرا که می توان بدون تغییر سایر کامپوننت ها، آن را به محیط های مختلف بانک های اطلاعاتی منتقل نمود.

Sesame درخواست های کاربران را از پروتکل های مختلف (مانند پروتکل HTTP و پروتکل SOAP) دریافت کرده و پس از تفسیر و تعیین نوع نیازمندی آن را به یکی از ماژول های اصلی خود منتقل می نماید.

Sesame دارای ۳ ماژول اصلی می باشد [۱۷]:

- ماژول Admin: این ماژول وظیفه ذخیره سازی اطلاعات داده و اسکیمای (آنتالوژی ها) را برعهده داد.

- **ماژول Query** : این ماژول، پرس و جوی کاربر را گرفته، تفسیر کرده و بر اساس آن درخت تجزیه پرس و جوی مورد نظر را ایجاد می نماید. سپس این درخت پرس و جو بهینه سازی شده و در نهایت اجرا شده و خروجی های مورد نیاز تشکیل می گردد.
- **ماژول Export**: این ماژول بسیار ساده می باشد. این ماژول اطلاعات مورد نیاز را از مخزن دریافت نموده و آن را با فرمت مورد نظر؛ به عنوان مثال به زبان RDF که با XML-Serilaized کد شده است، منتقل می نماید. به این ترتیب می توان داده های مخزن را در اختیار انواع مختلفی از ابزار ها قرار داد، بدون آنکه نیاز به تغییراتی در ساختار معماری Seesame باشد.
- برخی ابزار ها مانند ویرایش گر های آنتالوژی تنها نیاز به بخش اسکیمای داده های موجود در مخزن دارند. همچنین برخی دیگر از ابزار ها، احتمالاً تنها نیاز به بخش نمونه های (غیر اسکیمای) موجود در مخزن دارند. بنا براین، جهت استخراج داده ها، اسکیمای و یا هر دو به این ماژول (ماژول Export) نیاز است.

۳-۴- بازیابی اطلاعات برای وب معنایی

همانطور که قبلاً هم اشاره شد، با ظهور وب داده ها و اطلاعات زیادی جهت دسترسی آن لاین در اختیار عموم مردم قرار گرفت. این فرصتی را در اختیار کاربران وب قرار داد تا بتوانند با جستجوی کلمات کلیدی و یا مرور صفحات، به اطلاعات مورد نظر خود دست یابند. امروزه با وجود انبوهی از اطلاعات گوناگون در وب، روش های سنتی فوق ما را به تعداد کثیری صفحات می رساند که یافتن اطلاعات مورد نیاز را بسیار سخت و دشوار می سازد.

مستندات موجود در وب را می توان از نظر ساختار نمایش اطلاعات بصورت زیر دسته بندی نمود:

- **ساخت یافته**: مانند مستندات XML یا جداول بانک های اطلاعاتی رابطه ای، که در آن اقلام مختلف داده با استفاده از تگ ها یا قواعدی مشخص و ثابت، مشخص شده اند.
- **نیمه ساخت یافته**⁵⁷: مانند مستندات HTML که در آن اقلام داده ها با یک قالب نیمه مشخص نمایش داده شده اند. این شیوه نمایش اطلاعات ثابت نبوده و می تواند به راحتی تغییر نماید. در این ساختار ها یافتن اقلام اطلاعاتی بصورت قانونمند، بسختی امکان پذیر است.
- **غیر ساخت یافته** : در این نوع ساختارها اقلام اطلاعاتی در قالب زبان طبیعی و به شکل متن ارائه شده اند و یافتن اقلام اطلاعاتی بدون دخالت انسان بسیار مشکل می باشد.

حجم اطلاعات نیمه ساخت یافته بسیار زیاد می باشد و روز به روز نیز در حال افزایش می باشد. در این نوع مستندات، بدلیل عدم وجود یک ساختار مشخص، دسترسی و کنترل داده ها مشکل می باشد. از این رو، باید ابتدا داده های موجود را استخراج نمود و جهت کنترل (و یا احتمالا تغییرات) به صورت ساخت یافته در یک مخزن اطلاعاتی (مانند یک بانک اطلاعاتی) ذخیره نمود.

بازیابی اطلاعات شامل سازماندهی، ذخیره سازی، بازنمایی (نمایش) و دسترسی به اقلام اطلاعاتی می باشد. قلم اطلاعاتی، معمولا متن است اما می تواند تصویر صدا و ویدئو و غیره نیز باشد. اطلاعات غیر متنی بر اساس توضیحات آنها بازیابی می شوند.^{۵۸}

روش سنتی جهت استخراج داده ها از منابع وب نوشتن یک ابزار بنام Wrapper می باشد. این ابزار وظیفه دارد تا داده های مورد نظر را شناسایی کرده و آنها را در قالب های مناسب مانند، XML و یا بانک های اطلاعاتی رابطه ای منتقل نماید. در سال های اخیر ابزارها و چارچوب های زیادی جهت حل مشکل استخراج داده ها طراحی و پیاده سازی شده اند. مانند ViPER [۲۱]، KIM [۲۹ و ۳۲]، OntoGather [۳۹]، OntoWrapper [۱۷ و ۲۵].

هدف مشترک تمام ابزارهای Wrapper، تولید Wrapper هایی با دقت بسیار بالا و انعطاف پذیر (مقاوم نسبت به تغییرات داده ها) می باشد که تا حد امکان، نیاز کمتری به دخالت انسانی داشته باشد. دو خصلت اتومات سازی (عدم دخالت انسان) و انعطاف پذیری (نسبت به تغییر مستندات) در مقابل هم می باشند. هرچه درجه اتومات سازی زیاد می شود از انعطاف پذیری کاسته می شود و برعکس. لذا باید بین این دو یک مصالحه صورت گیرد.

بطور کلی روش های استخراج داده ها را می توان بصورت زیر تقسیم بندی کرد [۲۲ و ۲۳]:

- اولین روش شامل نشانه گذاری مستقیم در مستندات می باشد که باید توسط سازنده سند (به عنوان مثال یک فایل HTML) انجام گیرد. این روش شامل ابزار خاصی نمی شود و هرکسی ممکن است هنگام ایجاد مستندات، از نشانه گذاری های خاص خودش استفاده نماید. از طرفی دیگر، بعد از ایجاد مستندات نشانه گذاری شده، تعمیم آن برای استفاده در سایر دامنه ها مشکل می باشد.

این روش در مستندات استفاده می شود که دارای ساختاری دقیق و مشخصی می باشند. این روش بخاطر محدودیت هایی که عنوان شده، کمتر مورد استفاده قرار می گیرد.

- استفاده از زبان های تولید Wrapper: یکی از اولین گام ها جهت حل مشکل تولید Wrapper ها، ایجاد زبان هایی جهت تعریف و تولید یک Wrapper می باشد. این زبان ها نسبت به زبان های همه منظوره مانند جاوا و C# و VB و ... خیلی ساده تر بوده و به عنوان یک دستیار در

⁵⁸ ما برای سادگی، در این نوشته، تنها بر روی اقلام اطلاعاتی از نوع متن تمرکز می نماییم.

تولید Wrapper ها به تولید کنندگان آن، کمک می کردند. از جمله این زبان ها می توان FLORID [۳۵] و WebOQL [۳۷] را نام برد.

- ابزارهای آگاه از HTML^{۵۹} : این نوع ابزار ها از روش های مبتنی بر خصوصیات ذاتی و ساختاری مستندات HTML استفاده می کنند. در این روش قبل از انجام عمل استخراج داده ابتدا برای یک فایل HTML درخت تجزیه ایجاد می گردد و تگ های آن بصورت سلسله مراتبی (به شکل یک درخت) در حافظه ایجاد و نمایش داده می شود. سپس قواعد استخراج که مبتنی بر خصوصیات نمایشی داده ها است، اجرا می گردند (اتوماتیک یا نیمه اتوماتیک) از جمله ابزارهایی که از این روش استفاده می کنند می توان به W4F [۳۸]، XWRAP اشاره کرد.

- ابزارهای مبتنی بر پردازش زبان طبیعی (NLP) : این روش معمولاً جهت اعمال در متون بدون ساختار (متن های مسطح نوشته شده به زبان طبیعی) استفاده می شود. پردازش زبان طبیعی توسط برخی ابزار ها مانند OntoWrapper [۱۷ و ۲۵] و KIM [۳۲] و غیره، مورد استفاده قرار می گیرد. در این روش نرم افزار قواعد استخراج را فرا گرفته تا داده های مرتبط را بازیابی نماید. در این روش جهت یافتن روابط بین جملات، عملیاتی مانند فیلترگذاری^{۶۰} و نشانه گذاری بخشی هایی از صحبت^{۶۱} و نشانه گذاری سمانتیک لغوی^{۶۲} را اجرا می کنند. تا به این ترتیب قواعد استخراج کشف گردند. این نوع قواعد مبتنی بر محدودیت های سمانتیکی و نحوی می باشند که در یافتن اطلاعات مرتبط در یک سند کمک می کنند.

در حال حاضر ابزار ها و ایده های مطرح شده در زمینه پردازش زبان طبیعی قدرت و دقت مورد نیاز جهت تشخیص محتوای جملات و متن ها را ندارند [۲۲]. از این روش تنها در مواردی محدود استفاده می شود (به عنوان مثال در متن هایی که حالت تلگرافی دارند) مانند : لیست کارها و آگهی های اجاره آپارتمان ها و اطلاعیه های برگزاری سمینار ها و غیره.

- ابزارهای استنتاج Wrapper: این دسته از ابزار ها مجموعه ای از نمونه های آموزشی را دریافت کرده و قواعد استخراج را تولید می کنند. تفاوت اصلی این دسته از ابزارها با ابزار هایی از نوع NLP آن است که در اینجا به شرایط زبانی داده ها متکی نیستیم، بلکه به خصوصیات قالب بندی ای که بصورت غیر صریح خصوصیات ساختاری بخش هایی از داده های دریافت شده را طرح می کنند، متکی است.

⁵⁹ HTML Aware
⁶⁰ Filtering
⁶¹ Part of Speech Tagging
⁶² Lexical Semantic Tagging

- ابزار های مبتنی بر آنتالوژی ها. اکثر روش های قبلی مبتنی بر ساختار و نحوه نمایش اقلام اطلاعاتی بوده اند و بر اساس آن قواعد، الگو های استخراج داده ها مشخص می شوند. اما علاوه بر ساختار نمایش داد ها، می توان اقلام اطلاعاتی را با تکیه بر خود داده ها استخراج نمود. بدین منظور یک آنتالوژی برای یک دامنه مشخص ایجاد می شود. تا بوسیله آن ثابت های موجود در یک صفحه شناسایی شده و اشیاء موجود در آن ایجاد گردد.

از آنجا که داده های مورد نظر، به روش های مختلف سازماندهی می شود، بهتر است تا آنها را مطابق با یک مدل عمومی^{۶۳} و مستقل از منابع اطلاعاتی سازماندهی مجدد نماییم. به این ترتیب، استخراج و ترکیب داده ها از منابع اطلاعاتی مختلف ساده تر و قابل اعتماد تر خواهد بود.

۳-۵- جمع بندی مطالب

بزرگترین چالشی که در ارتباط با مدیریت اطلاعات مطرح می گردد، جستجوی اطلاعات می باشد. امروزه با وجود حجم زیاد اطلاعات، جستجوی کورکورانه تنها بر اساس تعدادی کلمه کلیدی پاسخ گونخواهد بود. برای مقابله با این مشکل لازم است تا از تکنیک های ارائه شده در وب معنایی استفاده نماییم. از آنجا که آنتالوژی ها نقشی کلیدی در مدل سازی توصیفات و ادراکات جهان مورد بحث دارد، لازم است تا بطریقی اطلاعات موجود در وب جاری با استفاده از آنتالوژی ها مدل سازی گردند.

اما بدلیل حجم زیاد اطلاعات موجود و عدم دانش اکثر کاربران وب از مفاهیم مطرح شده در وب معنایی^{۶۴} و نیز نیاز به تغییرات زیاد در تمامی برنامه های کاربردی تحت وب و هزینه زیاد آن، انجام چنین کاری امکان پذیر نمی باشد. از این رو، راه حل بازیابی اطلاعات موجود و بازنمایی مجدد آن در وب معنایی مورد توجه اکثر محققان قرار گرفته است. در زمینه بازیابی اطلاعات از مستندات نیمه ساخت یافته و غیر ساخت یافته، تحقیقات چندی صورت گرفته که در حالت کلی می توان آنها را به دو دسته تقسیم کرد.

دسته اول به ساختار نمایش اطلاعات توجه می کنند و اقلام اطلاعاتی را با توجه به نحوه نمایش آنها در فرم های HTML شناسایی می کنند. دسته دوم، به ماهیت اقلام اطلاعاتی توجه می کنند و سعی می نمایند در مستندات موجود اقلام اطلاعاتی مورد نظر را بیابند.

برای درک بیشتر روش ها موجود در بازیابی اطلاعات، لازم است تا با تعدادی از این ابزار ها آشنا شویم. فصل بعد به این مطلب می پردازد.

⁶³ Common Model

⁶⁴ وب معنایی نیاز به آموزش دارد

۴- بررسی فعالیت های انجام شده در زمینه استخراج اطلاعات

در بخش سوم، اهمیت و نقش مدیریت دانش در وب معنایی را نشان دادیم. در این بین آنتالوژی ها نقشی کلیدی را ایفا می کنند. تاکنون معماری های مختلفی برای مدیریت دانش (و به عبارتی مدیریت آنتالوژی ها) معرفی شده است که اکثرا جهت اثبات قابلیت های آن چه که ادعا نموده اند، یک نمونه پیاده سازی شده نیز ارائه کرده اند [۲۸]. این ابزار ها در مقالات مختلف بررسی و مقایسه شده اند [۲۳ و ۳۰].

ابزار های معرفی شده معمولا شامل ۴ بخش مهم زیر می باشند:

۱. ابزاری جهت ایجاد و ویرایش آنتالوژی ها و یا Wrapper ها
 ۲. استخراج داده ها از منابع اطلاعاتی موجود در وب جاری. این منابع ممکن است ساخت یافته یا نیمه ساخت یافته یا غیر ساخت یافته باشد.
 ۳. مخزن آنتالوژی و اطلاعات استخراج شده. این مخزن محلی برای ثبت و نگهداری آنتالوژی ها و نیز اطلاعات استخراج شده از منابع وب می باشد.
 ۴. موتور استنتاج که از یک منطق مشخص، مانند منطق تشریحی^{۶۵} یا منطق مرتبه اول (دوم)^{۶۶} استفاده می نماید. موتور استنتاج، پرس و جوی کاربران را دریافت کرده و بر اساس اطلاعات موجود در مخزن آنتالوژی، استنتاج کرده و نتیجه را بر می گرداند.
- هدف ما از بازیابی اطلاعات (در محدوده این نوشته)، یافتن اقلام اطلاعاتی از مستندات موجود در وب و نگهداری آنها در قالب هایی مشخص و مناسب (مانند جداول بانک های اطلاعاتی رابطه ای و یا آنتالوژی ها و غیره) می باشد.
- همانطور که گفته شد، روش کلی جهت بازیابی اطلاعات از منابع فوق تهیه یک ابزار بنام Wrapper می باشد. این ابزار وظیفه دارد تا داده های مورد نظر را شناسایی کرده و آنها را در قالب های مناسب مانند XML یا بانک های اطلاعاتی، منتقل نماید.
- در این فصل به بررسی نمونه های از فعالیت های انجام شده در زمینه تولید Wrapper ها و بازیابی اطلاعات می پردازیم.

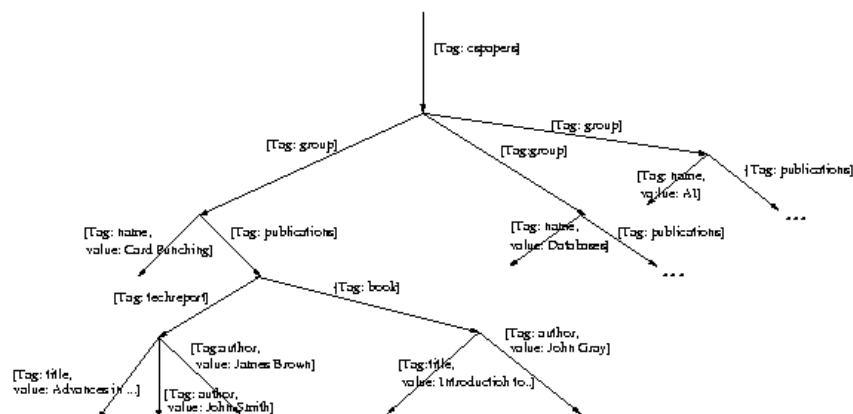
⁶⁵ Description Logic
⁶⁶ First (Second) Order Logic

WEBOQL – ۱ – ۴

WebOQL [۳۷] یک زبان پرس و جوی بیانی، جهت توصیف Wrapper ها می باشد. این زبان، دارای دو خصوصیت مهم زیر می باشد:

۱. دسترسی به ساختار داخلی اشیاء موجود در صفحات وب
۲. امکان ایجاد ساختارهای پیچیده جدید به عنوان خروجی پرس و جو

از آنجا که اکثر داده های موجود در وب نیمه ساخت یافته می باشد، این زبان نیز بر استخراج و بازیابی داده های نیمه ساخت یافته تاکید دارد. ساختار داده ای اصلی در این زبان HyperTree نام دارد. HyperTree، یک درخت برچسب دار و مرتب شده می باشد. در این درخت دو نوع کمان؛ داخلی و خارجی، وجود دارد. کمان های داخلی جهت نمایش اشیاء ساخت یافته و کمان خارجی جهت نمایش HyperLink ها استفاده می شود. شکل زیر یک نمونه از درخت HyperTree که می تواند توسط یک Wrapper تولید شود را نشان می دهد.



شکل 27: یک درخت HyperTree نمونه در WebOQL

این درخت می تواند در نتیجه تجزیه و تفسیر یک فایل، مانند فایل pubs.xml زیر، ایجاد گردد:

```
<?xml version="1.0"?>
<cspapers>
  <group>
    <name>Card Punching</name>
    <publications>
      <techreport>
        <title>Advances in Card Punching</title>
        <author>John Smith</author>
```

```

<author>James Brown</author>
</techreport>
<book>
  <title>Introduction to Card Punching</title>
  <author>John Gray</author>
</book>
</publications>
</group>
<group>
  <name>Databases</name>
  <publications>
    ...
  </publications>
</group>
...
</cspapers>

```

مجموعه ای از HyperTree های مرتبط در یک ساختار بنام web جمع آوری می گردد. هر دوی HyperTree و Web، می توانند توسط WebOQL مدیریت شده و به عنوان خروجی ایجاد گردند.

WebOQL یک زبان تابعی می باشد، اما پرس و جوها در قالب آشنای Select-From-Where نگهداری می گردد. برای مثال فرض کنید که فایل pubs.xml در شکل فوق، شامل اطلاعات نشریه ها باشد. پرس و جوی زیر تمام نشریه هایی که توسط Smith نوشته شده است را نشان می دهد:

```

[Tag:"result" /
  select y
  from x in browse("file:pubs.xml") via ^*[tag = "publications"],
    y in x',
    z in y'
  where z.tag = "author" and z.value ~ "Smith"
]

```

پرس و جوی فوق یک HyperTree را در HyperTree دیگر نگاشت می کند. به عبارتی، در واقع، پرس و جو یک تابعی است که یک وب را به وب دیگر نگاشت می کند. برای مثال، پرس و جوی زیر یک صفحه وب جدید برای هر گروه تحقیقاتی که شامل نشریه های آن گروه می باشد، ایجاد می کند.

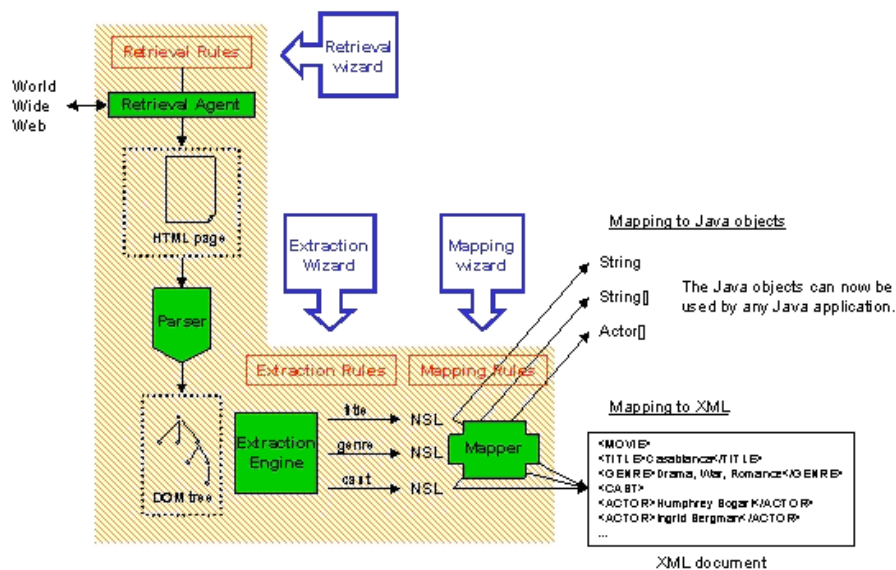
```

select x' as x'.value
from x in browse("file:pubs.xml")

```

۴ - ۲ - W4F (World Wide Web Wrapper Factory)

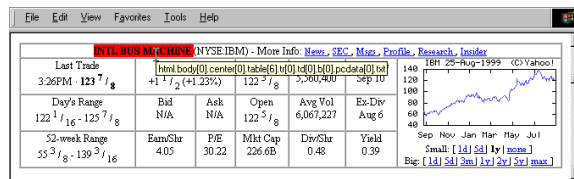
W4F [۳۸] یک جعبه ابزار^{۶۷} برای ساخت Wrapper ها می باشد. W4F فرایند تولید نرم افزار را به سه مرحله تقسیم می کند. در ابتدا کاربر مشخص می کند که چگونه باید به مستندات دست یافت. به عبارتی آدرس URL سند مورد نظر را می دهد. سپس کاربر اقلام اطلاعاتی که باید استخراج شود را توصیف می کند. در نهایت باید ساختار مقصد، جهت ذخیره سازی و نگهداری داده های استخراج شده، مشخص گردد. شکل زیر معماری این سیستم را نشان می دهد.



شکل 28: معماری W4F

ابتدا سند مورد نظر مطابق با یک یا چند قاعده استخراج، بازیابی می شود. سند مورد نظر به محض بازیابی، به بخش تجزیه کننده HTML داده می شود تا یک درخت تجزیه مطابق با مدل شیء سند^{۶۸} (DOM) ایجاد گردد. سپس کاربر می تواند قواعد استخراج را جهت یافتن اقلام اطلاعاتی در درخت تجزیه را مشخص نماید. این قواعد استخراج با فرمت داخلی W4F، با نام لیست رشته های تودرتو^{۶۹} (NSL)، ثبت می گردد. در نهایت سازه های NSL می توانند مطابق با تعیین قواعد نگاشت، به یک برنامه کاربردی سطح بالاتر منتقل گردند. زبان استفاده شده جهت تعریف قواعد استخراج HEL (HTML Extraction Language) نام دارد.

⁶⁷ Toolkit
⁶⁸ Document Object Model
⁶⁹ Nested String List



```

Action View Examples Options Help

METHOD: GET;
URL: "http://finance.yahoo.com/q?s=AOL+YHOO+IBM+CSOO+LU+EBAY+TKN+EGRP+NOR&d=t";

html.body.center.table[i:*]
# .tr[0].td[0].b[0].txt // name
# .tr[0].td[0].b[0].p[0].txt, match /([()(.*)?)/ // trading place
# .tr[0].td[0].b[0].p[0].txt, match /:(.*)?[/] // ticker
# .tr[1].td[0].b[0].p[0].txt // last trade
# .tr[1].td[3].p[0].txt // volume
# .tr[1].td[1].txt, match /([()(.*)?)/ // change %
# .tr[2].td[0].txt, match /Range(.*)/, split /-/ // Day range
# .tr[3].td[0].txt, match /Range(.*)/, split /-/ // Year range
)

where html.body.center.table[i].tr[0].td[0].getAttr(colspan) = "7";

.Portfolio*.Stock (
  .Full_Name^
  # .Market!Name
  # .Ticker^
  # .Last
  # .Volume
  # .Change
  # .Day_Range ( .Min # .Max )
  # .Year_Range ( .Min # .Max )
);

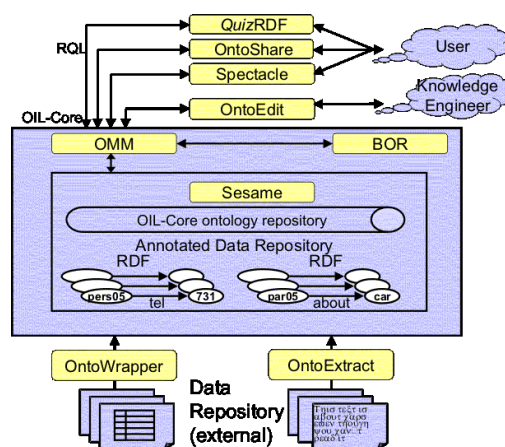
```

شکل 29: W4F Wizard

شکل فوق، بخشی از رابط کاربری مورد استفاده در W4F را نشان می دهد که جهت کمک به کاربران در نوشتن قواعد استخراج، تهیه شده است. کاربر بر روی اقلام اطلاعاتی مورد نظر خود، کلیک می کند و سیستم، قواعد استخراج مربوطه را بر می گرداند. سپس کاربر می تواند آن قاعده را جهت تعمیم به مجموعه ای از رکورد های مشابه، ویرایش نماید.

۴ - ۳ - OntoKnowledge

پروژه On-to-Knowledge، مجموعه ای از ابزار ها را جهت پردازش اطلاعات معنایی، طراحی و پیاده سازی کرده است [۲۶]. بخشی از این ابزار ها مائول های OntoWrapper [۲۵] برای استخراج اطلاعات از داده ها نیمه ساخت یافته و OntoExtract [۲۴] برای استخراج اطلاعات از داده های غیر ساخت یافته می باشد. شکل زیر معماری پیشنهادی جهت مدیریت دانش در وب معنایی را نشان می دهد [۱۷].



شکل 30: معماری OntoKnowledge

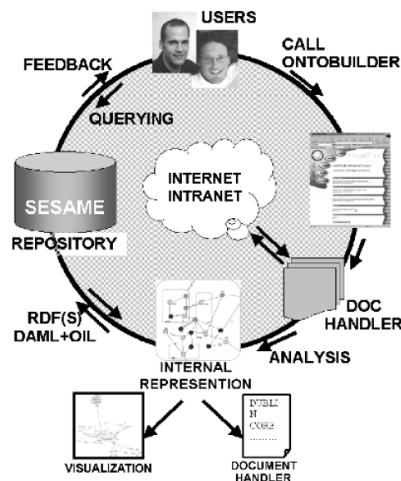
این معماری تمام مراحل چرخه مدیریت دانش را که ذیلاً به بررسی اجمالی آن می پردازیم، نشان می دهد [۱۷]:

- **اکتساب دانش:** بدلیل وجود مقدار بسیار زیادی از داده ها در منابع وب (بصورت نیمه ساخت یافته و یا غیر ساخت یافته)، استخراج این داده ها بصورت اتوماتیک یا حداقل نیمه اتوماتیک الزامی می باشد. این نیازمندی در بخش پایینی معماری فوق، توسط ابزار های OntoWrapper [۲۵] برای استخراج اطلاعات از منابع نیمه ساخت یافته و OntoExtract [۲۴] برای استخراج اطلاعات از منابع غیر ساخت یافته، پشتیبانی شده است.
- **نمایش دانش:** بعد از استخراج اتوماتیک اطلاعات از منابع موجود در وب، می بایست آن را توسط یک زبان تعریف آنتالوژی نمایش داد تا بتوان جهت استدلال و یا (احتمالاً) اعمال تغییرات در آینده، مورد استفاده قرار داد. همانطور که در شکل نیز نشان داده شده، در این پروژه جهت نمایش دانش از معماری Sesame و زبان RDF استفاده می گردد.
- **نگهداری دانش:** برای نگهداری، مدیریت و کار با آنتالوژی ها نیاز به یک میان افزار⁷⁰ می باشد. بدین منظور، در این پروژه از ماژول میان افزار آنتالوژی^{۷۱} (OMM) استفاده می شود.
- **استفاده از دانش:** در نهایت نیاز به ابزاری جهت دسترسی به اطلاعات می باشد تا به کاربران امکان استفاده از دانش ذخیره شده در سیستم را ارائه کرد. بدین منظور ابزارهایی مانند QuizRDF و OntoShare و Spectacle ایجاد شده اند. البته OntoEdit نیز برای استفاده توسط مهندس و خبره دانش، جهت تعریف آنتالوژی ها و ایجاد دانش اولیه، پیاده سازی شده است.

همانطور که گفته شد OntoKnowledge دارای دو ماژول مهم جهت بازیابی اطلاعات می باشد [۱۷].

⁷⁰ MiddleWare
⁷¹ Ontology Middleware Module

OntoExtract: OntoExtract مبتنی بر موتور تحلیل زبان طبیعی ساخته شده است. این موتور متون زبان طبیعی را تحلیل کرده و آنتالوژی های سبک وزن دامنه را تولید می کند و در این بین از دانش موجود در مخزن مرکزی نیز استفاده می کند. همانطور که گفته شد، پلتفرم SESAME جهت مدیریت مخزن استفاده می شود و توسط ابزار OntoEdit ویرایش و بروز رسانی می گردد. OntoExtract، جهت ارتباط با SESAME از زبان پرس و جوی RQL^{72} استفاده می کند. شکل زیر فرایند کلی استخراج اطلاعات را نشان می دهد.



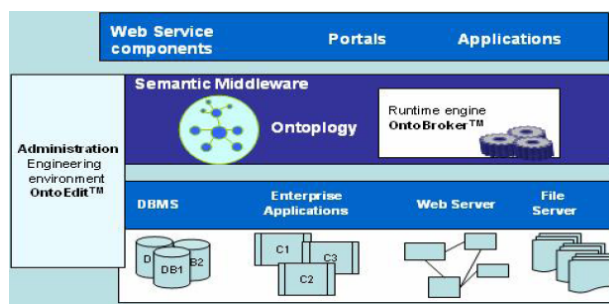
شکل 31: فرایند استخراج در OntoExtract

این فرایند تکراری می باشد. کاربر با تعیین دامنه اینترنتی مورد تحلیل (به عنوان مثال یک آدرس URL) ، فرایند را آغاز می نماید. به ازای هر صفحه موجود در دامنه تعیین شده که دارای متونی نوشته شده به زبان طبیعی می باشند، بصورت اتوماتیک، آنتالوژی های سبک وزن تولید می گردند و جهت ذخیره سازی به SESAME منتقل می گردند [۲۴].

TableAnalyzer و OntoWrapper: اغلب، مهندس آنتالوژی نیاز دارد تا نمونه های موجود در یک دامنه اینترنتی را که بصورت نیمه ساخت یافته یا تمام ساخت یافته ارائه شده اند، جمع آوری نماید. بدین منظور می توان از TableAnalyzer و OntoWrapper استفاده نمود. OntoWrapper، جهت استخراج موجودیت های موجود در یک صفحه از مجموعه ای از قواعد که بزبان RDF نوشته شده است. استفاده می نماید. بدین منظور مهندس آنتالوژی، مجموعه ای از الگوها، متغیر ها و قواعد را به زبان XML/RDF می نویسد. سپس این مازول با استفاده از آنها، صفحات موجود را مرور کرده و اطلاعات مورد نظر را بازیابی می نماید. اطلاعات استخراج شده به فرمت سه تایی های RDF در مخزن SESAME ثبت و نگهداری می گردد.

۴-۴- OntoPrise

نسخه تجاری پروژه On-to-Knowledge، سیستم OntoPrise [۲۷] می باشد که شامل ماژول های OntoWrapper و OntoExtract نمی باشد. محصولات اساسی این پروژه، OntoEdit و OntoBroker و Semantic Miner و OntoOffice می باشند. OntoEdit یک چارچوب مدل سازی و نظارت برای یکپارچه سازی ساختار های ناهمگن مبتنی بر آنتالوژی می باشد. OntoBroker موتور جستجوی مبتنی بر آنتالوژی می باشد. OntoPrise در موتور استنتاج خود از منطق F-Logic [۳۶] استفاده می کند. Semantic Miner، بازیابی اطلاعات از منابع ناهمگن را پشتیبانی می کند. OntoOffice ابزاری است که جهت حاشیه گذاری و بسط پایگاه دانش، امکان یکپارچه سازی با مجموعه نرم افزار های Microsoft Office را فراهم می کند.

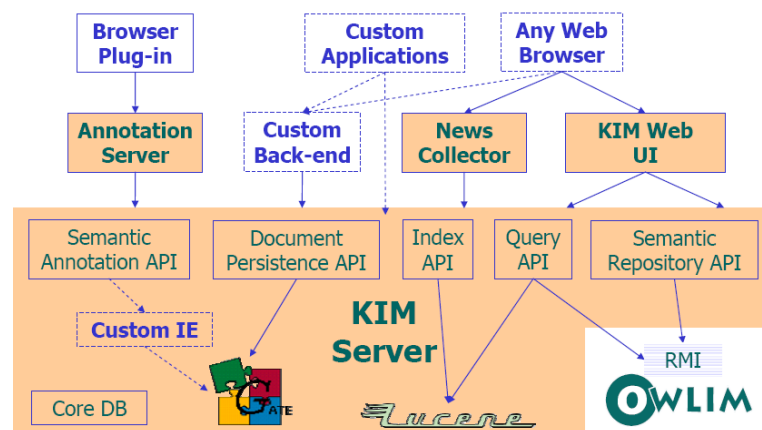


شکل 32: معماری OntoPrise

در OntoPrise جهت استخراج داده های موجود در منابع وب، باید به این ترتیب عمل کرد [۲۳]:

- ایجاد آنتالوژی دامنه، با استفاده از ابزار OntoEdit ایجاد می گردد. با این ابزار مفاهیم، خصوصیات، روابط، نمونه ها و اصل های متعارف مربوط به دامنه مورد نظر. این آنتالوژی باید توسط کارشناس خبره دامنه که به روابط معنایی میان داده های ذخیره شده در سازمان (مانند بانک های اطلاعاتی و وب سرویس ها) آگاهی کامل دارد، ایجاد گردد.
- سپس با استفاده از همان ابزار OntoEdit می توان اسکیمای یک بانک اطلاعاتی را به شکل آنتالوژی سبک وزن استخراج نمود. سپس از یک کامپوننت بنام OntoMap جهت نگاشت بین آنتالوژی دامنه و آنتالوژی بانک اطلاعاتی استفاده می شود. این کار نیز باید توسط کارشناس خبره دامنه انجام گیرد.
- سپس از ابزار OntoBroker جهت استنتاج منطقی از داده های موجود و از ابزار Semantic Miner جهت اجرای پرس و جو های پیچیده استفاده می گردد.

KIM مخفف Knowledge and Information Management بوده و پلت فورمی جهت حاشیه نویسی معنایی، جستجو و تحلیل می باشد [۲۹]. شکل زیر معماری KIM را نشان می دهد [۳۲]:



شکل 33: معماری KIM

از این پلت فورم در موارد زیر استفاده می شود:

۱. حاشیه نویسی معنایی متن
۲. شاخص گذاری و بازیابی اطلاعات
۳. پرس و جو و اکتشاف دانش رسمی
۴. پیگیری وقایع همزمان و رتبه بندی موجودیت ها
۵. تحلیل محدوده زمانی محبوبیت و شهرت موجودیت

KIM، ترکیبی از سه معماری منبع باز^{۷۳} زیر می باشد:

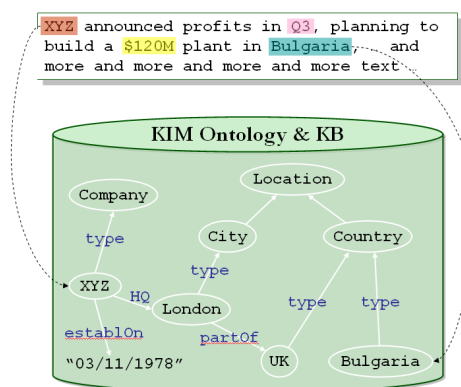
- GATE^{۷۴} (معماری عمومی جهت مهندسی متن) مشهورترین پلتفورم نرم افزاری جهت مهندسی زبان می باشد که توسط گروه پردازش زبان طبیعی (NLP) در دانشگاه شفیلد طراحی و پیاده سازی شد. GATE یک زیرساختی را جهت تولید نرم افزارهای پردازش زبان طبیعی ارائه می کند. کامپوننت های GATE بسیاری از فعالیت های سطح پایین، مانند ذخیره سازی و بازیابی داده ها و پردازش زبان طبیعی را انجام می دهند [۳۳].

⁷³ Open Source
⁷⁴ General Architecture for Text Engineering

- **SESAME**: یک چارچوب منبع باز [۳۱] برای ذخیره سازی و پرس و جو و استنتاج از اطلاعات ذخیره شده در قالب RDF و RDF Schema می باشد. SESAME در فصل سوم معرفی شد.
- **Apache Lucene**: موتور جستجوی متن با کارایی بسیار بالا می باشد که بصورت یک کتابخانه از توابع به زبان جاوا، ارائه شده است. تکنولوژی Lucene تقریباً برای هر برنامه کاربردی که نیاز به جستجوی کامل متن و خصوصاً نیاز به اجرا در چند پلتفرم مختلف، مانند ویندوز و لینوکس و غیره دارد، مناسب می باشد [۳۴].

پلتفرم KIM شامل آنتالوژی KIM، پایگاه دانش، سرور KIM (شامل API هایی جهت دسترسی و ادغام و یک پارچه سازی از راه دور)، نرم افزار های نهایی^{۷۵} (پلاگین برای مرورگر ها و رابط کاربری وب و مرورگر دانش). آنتالوژی KIM و پایگاه دانش آن در یک مخزن که مبتنی بر معماری SESAME [۱۷] می باشد، نگهداری می گردند. KIM یک زیر ساخت جدید برای استخراج داده ها که مبتنی بر پلتفرم GATE [۳۳] می باشد ارائه می کند. همچنین از پلتفرم Apache Lucene در شاخص گذاری مستندات بر اساس موجودیت های آنها و میزان ارتباط آنها با هم جهت بازیابی معنایی اطلاعات استفاده می کند.

برای آنکه برنامه های کاربردی کاملاً اتماتیک باشند، KIM دارای یک آنتالوژی سطح بالا و سبک وزن، بنام PROTON به همراه ۲۵۰ کلاس و ۱۰۰ خصوصیت می باشد. PROTON یک نمونه پیاده سازی شده از آنتالوژی KIMO می باشد. بعلاو پایگاه دانش KIM شامل ۲۰۰،۰۰۰ توصیف موجودیت به عنوان دانش اولیه می باشد.



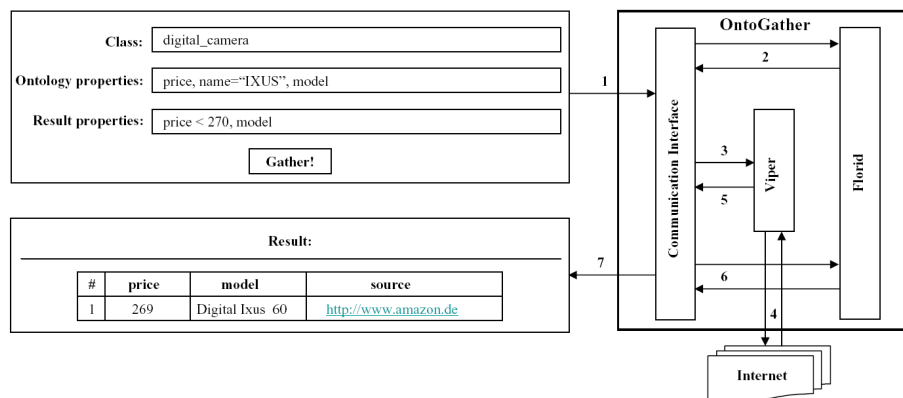
شکل 34: حاشیه گذاری در KIM

شکل فوق کاربرد آنتالوژی سطح بالای Proton و نیز دانش اولیه موجود در KIM را جهت پردازش یک جمله نشان می دهد.

ذکر این نکته مهم است که KIM یک پلتفرم است و می تواند توسط نرم افزار های دیگر مورد استفاده قرار گیرد تا در انجام حاشیه گذاری معنایی، بازیابی محتوی، پرس و جوی های معنایی و غیره، از قابلیت های آن استفاده نمایند.

۴-۶- OntoGather

OntoGather [۳۹] یک موتور پرس و جوی منابع وب مبتنی بر آنتالوژی می باشد. یکی از خصوصیات بارز این سیستم، بازیابی لحظه ای اطلاعات می باشد. به عبارتی، پرس و جو ها بر روی داده های آنلاین انجام می گیرد و کاربر می تواند آخرین اطلاعات را حداکثر با چند دقیقه تاخیر مشاهده نماید. OntoGather از یک سیستم کاملا اتوماتیک جهت بازیابی اطلاعات موجود در وب، بنام ViPER [۲۱] استفاده می کند و در کل دو فعالیت مهم را به موازات هم انجام می دهد. اول انتخاب منابع مرتبط با پرس و جوی مورد نظر و دوم استدلال بر روی داده های استخراج شده توسط سیستم ViPER. شکل زیر یک دید کلی از این سیستم را ارائه می کند.



شکل 35: مرور کلی سیستم OntoGather

OntoGather شامل دو کامپوننت اساسی است:

- **FLORID : FLORID** [۳۵] یک سیستم بانک اطلاعاتی شی گرا و استنتاجی است که از F-Logic [۳۶] جهت تعریف داده ها و زبان پرس و جو، استفاده می نماید. FLORID هسته مرکزی OntoGather را تشکیل می دهد. در واقع FLORID موتور استنتاج OntoGather جهت انتخاب منابع وب و ارزیابی پرس و جو های کاربر می باشد.

• **ViPER^{۷۶}: ViPER [۲۱]** ابزاری جهت استخراج اتوماتیک داده ها از منابع وب می باشد.

این ابزار جهت استخراج داد ها، در صفحات وب، به دنبال تکرار الگوی رکورد ها می گردد و برای پیدا کردن این الگو ها به شیوه نمایشی داده های موجود در صفحات وب تمرکز می کند. از این رو برای آنکه ViPER بتواند درست کار کند حداقل نیاز است تا دو رکورد وجود داشته باشد. این الگو ها همان قواعد استخراج را مشخص می کنند که در OntoGather جهت جمع آوری اطلاعات از منابع وب استفاده می گردد.

در OntoGather منابع اطلاعاتی باید مرتبط با یک آنتالوژی دامنه باشند. این آنتالوژی شامل توصیف اطلاعات موجود در منابع وب می باشد و با زبان F-Logic [۳۶] نشان داده شده و در سیستم FLORID ذخیره می گردند. با رجوع به شکل فوق، فرایند پرس و جو، در هفت مرحله انجام می گیرد:

۱. ابتدا کاربر از طریق یک رابط کاربری، مشخصات موضوع مورد جستجو را وارد می کند. در این مثال کاربر بدنبال دوربین های دیجیتالی است که قیمت آنها کمتر از ۲۷۰ باشد. دوربین در ضمن دوربین دیجیتال یکی از کلاس های موجود در آنتالوژی دامنه می باشد.

۲. سپس درخواست مورد نظر به واسطه ارتباطی ارسال می گردد. این واسطه از FLORID درخواست پرس و جوی مورد نظر را ارسال می کند. FLORID نیز لیستی از منابع موجود را که حاوی اطلاعات مورد نظر می باشند به همراه سایر مشخصات پرس و جو، به عنوان جواب ارسال می کند.

۳. این اطلاعات برای ViPER ارسال می گردد.

۴. ViPER منابع وب را کاوش کرده و اطلاعات درخواستی را بازیابی می کند.

۵. اطلاعات بازیابی شده برای واسطه ارتباطی ارسال می گردد.

۶. این نتایج به Fact های F-Logic تبدیل شده و نمونه های جدید در آنتالوژی ذخیره می شوند.

۷. نتایج نهایی برای ارائه به کاربر، برای واسطه کاربری ارسال می گردد.

۴ - ۷ - BYU-Ontos

این ابزار توسط گروه استخراج داده در دانشگاه Brigham Young ساخته شده است [۱۹]. در این ابزار، آنتالوژی ها جهت توصیف داده های مورد نظر شامل رابطه ها، واژگان مورد استفاده و کلمات کلیدی زمینه، استفاده می گردد. این ابزار می تواند با تجزیه و تحلیل این آنتالوژی، اقلام داده ای مورد نظر جهت

استخراج را تشخیص داده و یک بانک اطلاعاتی از آنها ایجاد نماید. قبل از استفاده از این آنتالوژی، باید توسط یک روتین اتوماتیک، متون شامل اقلام داده ای مورد نظر را از منابع وب (صفحات وب یا مستندات) استخراج نمود.

برای آن که این ابزار بدرستی کار کند، باید آنتالوژی اولیه بدرستی و با دقت زیاد ایجاد گردد. ساخت این آنتالوژی بر عهده فرد خبره در دامنه آن آنتالوژی می باشد. در صورتیکه آنتالوژی اولیه به اندازه کافی بیانگر و تشریحی باشد، روال استخراج می تواند کاملاً اتوماتیک انجام شود. شمن آن که Wrapper های تولید شده به این روش، ذاتاً نسبت به تغییرات ساختاری مستندات مقاوم می باشند. در واقع این مزیت ها به دلیل استفاده از آنتالوژی ها در تشخیص اقلام داده ای می باشد.

۴-۸- جمع بندی مطالب

در این فصل به بررسی چند فعالیت مرتبط با بازیابی اطلاعات از منابع وب پرداختیم:

- پروژه OntoKnowledge با هدف تحقیق در زمینه وب معنایی اقدام به ایجاد ابزارهایی جهت استخراج اطلاعات از وب نموده است. OntoKnowledge جهت ذخیره سازی اطلاعات بازیابی شده از چاقوب SESAME استفاده نموده است. OntoKnowledge دو ماژول مهم OntoExtract جت بازیابی داده های غیر ساخت یافته و OntoWrapper جهت بازیابی داده های ساخت یافته یا نیمه ساخت یافته می باشد.
- OntoPrise نسخه تجاری پروژه OntoKnowledge می باشد که برای استخراج داده های ساخت یافته مانند جداول بانک های اطلاعاتی و یا متون فرمت شده^{۷۷}، مناسب می باشد.
- KIM یک پلتفرم بسیار قوی می باشد که جهت استخراج داد ها از منابع غیر ساخت یافته از پلتفرم های بزرگ دیگر، مانند SESAME و GATE و Apache Lucene استفاده می نماید. KIM دارای یک آنتالوژی سطح بالا بوده که دانش اولیه آن را تشکیل می دهد و بر اساس تکنیک های پردازش زبان طبیعی (NLP) اقدام به بازیابی اطلاعات می کند.
- OntoGather با بهره گیری از FLORID به عنوان سیستم ذخیره سازی آنتالوژی ها و موتور جستجو و نیز بهره گیری از ابزار ViPER، توانسته است تا امکان جستجوی آن لاین از داده های موجود در وب را فراهم کند. ViPER با بهره گیری از روش های داده کلوی قادر است تا بصورت کاملاً اتوماتیک اقدام به بازیابی اطلاعات نماید.

۵- ارائه یک چارچوب جهت بازیابی اطلاعات

همانطور که در فصل سوم نیز عنوان شد، منابع موجود را می توان به سه دسته زیر تقسیم کرد:

- منابع ساخت یافته: مانند بانک های اطلاعاتی
- منابع نیمه ساخت یافته: مانند صفحاتی از وب که اطلاعات را با ساختار مشخص و تقریباً ثابتی به کاربران نشان می دهد
- منابع غیر ساخت یافته: مانند متون نوشته شده به زبان طبیعی

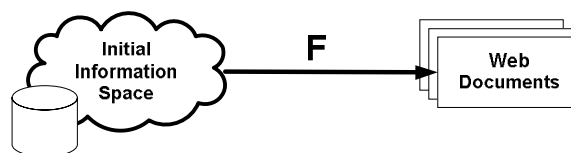
ما جهت بازیابی اطلاعات از آنتالوژی ها استفاده خواهیم کرد. این روش جدیدی نیست و قبلاً [۱۹ و ۳۲] نیز از آنتالوژی ها جهت بازیابی اطلاعات استفاده شده است. هدف ما آن است تا محدوده بیشتری از مسائل را با استفاده از آنتالوژی ها پوشش دهیم. از این رو لازم است تا ابتدا با برخی سوالات و مسائل موجود در این زمینه آشنا شویم.

در ادامه به تحلیل مساله بازیابی اطلاعات، باتوجه به انواع منابع فوق می پردازیم تا بتوانیم درک درستی از مساله بازیابی اطلاعات داشته و مسیر فعالیت های بعدی خود را مشخص نماییم.

۵- ۱- تحلیل مساله بازیابی اطلاعات

فرض اولیه

در اینجا فرض می کنیم که منابع موجود از نظر نحوی و ساختاری ناهمگن می باشند، به عبارتی هر یک از منابع موجود، می تواند دارای نحو و یا ساختار منحصر به خود باشند. بنابراین نمی توان یک الگوریتم ثابت برای استخراج اطلاعات ارائه نمود. همچنین فرض می کنیم که منابع موجود (مانند صفحات وب) از فضای داده ای اولیه^{۷۸} و بر اساس یک تابع تبدیل اولیه F ، ساخته شده اند.



شکل 36: ساخت مستندات براساس فضای اطلاعاتی اولیه

فضای داده ای اولیه می تواند، یک بانک اطلاعاتی، یک فایل XML و یا حتی دانش انسانی و ذهنی باشد. در شکل فوق Web Document می تواند هر نوع سند در وب باشد که رایج ترین نوع آن، صفحات HTML می باشد. تابع تبدیل F نمایانگر عملیات کد سازی باشد. زمانیکه یک وبلاگ نویس اطلاعات متن مورد نظر خود را می نویسد، همچنین زمانیکه اطلاعات موجود در یک بانک اطلاعاتی را در یک فایل HTML نمایش می دهیم، در واقع دانش موجود در فضای اطلاعاتی اولیه (دانش ذهنی و یا اطلاعات موجود در یک بانک اطلاعاتی) را در یک فایل HTML، کد (پیاده سازی) می نماییم.

در صورتیکه فضای اطلاعاتی اولیه را I و سند وب را D بنامیم. داریم:

$$I = \{ I_1, I_2, I_3, \dots, I_m \}$$

$$D = \{ d_1, d_2, d_3, \dots, d_n \}$$

که در آن I_k قلم اطلاعاتی اولیه و D_k قلم داده ای موجود در سند وب می باشد. با توجه به تعریف F داریم:

$$D = F(I)$$

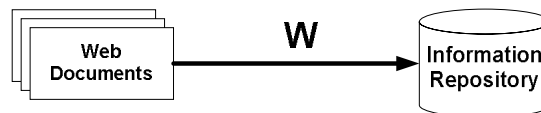
همانطور که گفته شد، تابع F عملیات کد سازی و ساخت مستندات وب را نشان می دهد. این تابع در منابع ساخت یافته، یک به یک و معمولاً پوشا می باشد. بطوریکه $m=n$ و به ازای هر k می توان نوشت: (در این حالت تابع F معمولاً وارون پذیر است)

$$d_k = F(I_k)$$

فایل های متنی فرمت دار و برخی فایل های XML، نمونه هایی از منابع ساخت یافته می باشند که می توانند جهت Export جداول بانک های اطلاعاتی مورد استفاده قرار گیرند.

هدف

در هنگام بازیابی اطلاعات، سیستم به مجموعه ای از مستندات وب دسترسی داشته و باید اقلام اطلاعاتی موجود در آن را تحت یک تابع W، به یک مخزن اطلاعاتی تبدیل نماید. اگر این مخزن اطلاعاتی برابر با فضای داده ای اولیه باشد، آنگاه تابع تبدیل W معکوس تابع F خواهد بود. $W=F^{-1}$



شکل 37: بازیابی اطلاعات از مستندات وب

در صورتیکه سند وب را D و مخزن اطلاعاتی را O بنامیم. داریم که :

$$D = \{ d_1, d_2, d_3, \dots, d_n \}$$

$$O = \{ O_1, O_2, O_3, \dots \}$$

که در آن d_k قلم اطلاعاتی موجود در سند وب و O_k قلم اطلاعاتی موجود در مخزن نهایی می باشد. با توجه به تعریف W داریم:

$$O = W(D)$$

تابع W نشان دهنده عملیاتی است که یک Wrapper جهت بازیابی اطلاعات انجام می دهد. Wrapper (تابع W) یک سند وب را به عنوان ورودی دریافت کرده و مجموعه اقلام اطلاعاتی موجود در آن را بازیابی می کند. تابع W موضوع اصلی تحقیق ما را تشکیل می دهد)

بنابراین می توان گفته که تابع W ، بین اقلام اطلاعاتی موجود در سند وب و اقلام اطلاعاتی مخزن یک نگاشت ایجاد می کند. لذا در یک Wrapper، دو فعالیت مهم زیر انجام می گیرد:

۱. استخراج قلم اطلاعاتی موجود در سند وب (یافتن d_i ها) : استخراج اقلام اطلاعاتی از منابع کاملاً ساخت یافته و نیمه ساخت یافته، با تمرکز بر روی ساختار نمایش اطلاعات، امکان پذیر است. اما در منابع غیر ساخت یافته، بدلیل آنکه اقلام اطلاعاتی در قالب جملات زبان طبیعی ارائه شده اند و نیز از آنجا که جملات زبان طبیعی دارای ابهام می باشند، یافتن اقلام اطلاعاتی مشکل و معمولاً دارای ابهام می باشد و نیاز به یک دانش اولیه دارد.

۲. نگاشت بین قلم اطلاعاتی سند وب با قلم اطلاعاتی موجود در مخزن (تابع W): جهت عملیات نگاشت، نیاز به آگاهی از سمانتیک اقلام اطلاعاتی اولیه (d_i)، می باشد. باید توجه داشت که عملیات نگاشت (تابع W) الزاماً تمام اقلام اطلاعاتی اولیه را نگاشت نمی کند. بنابراین جهت نگاشت، نیاز به دانش اولیه است تا سمانتیک اقلام اطلاعاتی را تعیین نمود.

همانطور که گفته شد، هم جهت استخراج و هم جهت نگاشت به دانش اولیه لازم است. این دانش اولیه را می توان با استفاده از آنتالوژی ها معرفی نمود. هر چه این دانش اولیه بیشتر و غنی تر باشد، در عملیات فوق، نیاز کمتری به دخالت انسان می باشد.

در استخراج اطلاعات، باید توجه داشت که همیشه تمام اطلاعات بصورت صریح و شفاف ارائه نمی شوند و برخی اطلاعات، بصورت ضمنی وجود دارند. همچنین وابستگی زیاد به ساختار نمایش اقلام اطلاعاتی، جهت استخراج داده، انعطاف پذیری و مقاومت فرایند استخراج اطلاعات را تا حد زیادی می کاهش دهد. در ادامه این دو موضوع را بیشتر تشریح می کنیم.

۵-۲- استخراج اطلاعات صریح و اطلاعات ضمنی

در مستندات وب، اطلاعات ممکن است به دو صورت صریح و یا ضمنی ارائه شوند.

- اطلاعات صریح: آن است که بصورت صریح در سند وب آمده است. به عنوان مثال مقادیر نام و نام خانوادگی در سلول های یک جدول از یک فایل HTML.

- اطلاعات ضمنی: این اطلاعات بصورت صریح در مستندات وب نیامده است. بلکه با توجه به سایر داده های موجود در سند مورد نظر قابل شناسایی می باشد. به عنوان مثال یک صفحه وب ممکن است تنها محصولات یک شرکت خاص باشد. بنابراین نام شرکت یک قلم اطلاعاتی است که در منبع اطلاعاتی نمی آید ولی باید در هنگام استخراج اطلاعات به آن توجه داشت.
- متاسفانه در حال حاضر اکثر روش های موجود تنها جهت استخراج اطلاعات صریح مناسب می باشند و نمی توان از آنها جهت استخراج اطلاعات ضمنی استفاده نمود. استفاده از آنتالوژی ها در استخراج اطلاعات ضمنی، یکی از موضوعات تحقیق آینده ما خواهد بود.

۵-۳- مقاومت در برابر تغییرات منابع

یکی از مشکلات موجود در بازیابی اطلاعات، تغییرات مکرر در منابع وب می باشد که باعث کاهش انعطاف پذیری در بازیابی اطلاعات می گردد. تغییرات در منابع شامل تغییرات ساختاری، تغییرات محتوایی و یا ترکیبی از آن دو می باشد.

- تغییرات ساختاری: آن دسته از تغییرات هستند که ساختار درختی نمایش اطلاعات را مستقیماً تحت تاثیر قرار می دهد. به عنوان مثال سازنده یک صفحه HTML تصمیم می گیرد که در لیست محصولات نام محصول را پررنگ (Bold) کرده و یا کاهش قیمت ها را با رنگ قرمز نشان دهد.

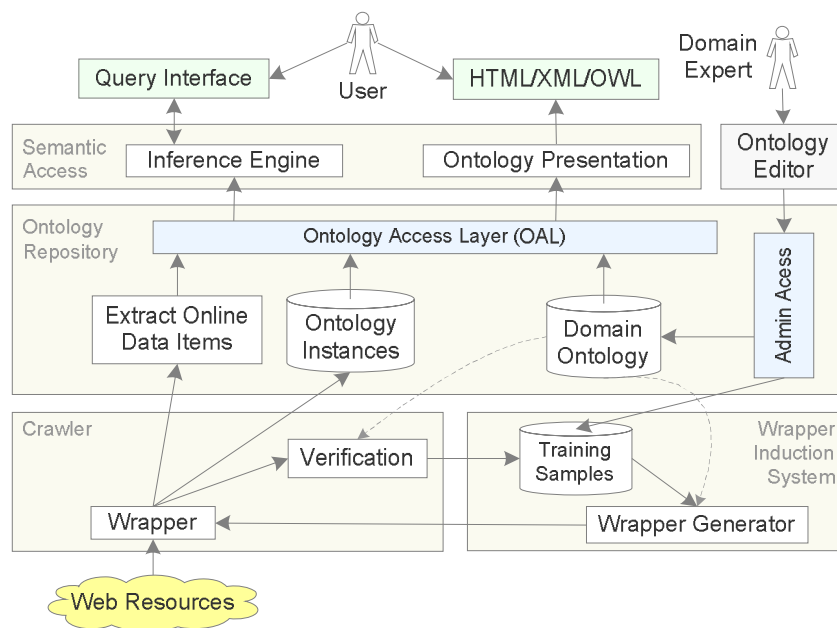
- تغییرات محتوایی: این تغییرات ساختار درختی نمایش اطلاعات را تغییر نمی دهد ولی اطلاعات موجود در هر گره از ساختار را تغییر می دهد. به عنوان مثال سازنده یک صفحه HTML تصمیم می گیرد که محل نمایش نام و نام خانوادگی را جابجا نماید و یا واحد پولی را از ریال به دلار تبدیل نماید.

در صورتیکه قواعد استخراج وابسته به ساختار اطلاعات نباشد (معمولاً در استخراج اطلاعات از منابع غیر ساخت یافته)، تغییرات ساختاری تاثیری در آن ایجاد نمی کند. در غیر این صورت اگر قواعد استخراج وابسته به ساختار اطلاعات باشد (مانند اکثر قواعد استخراج از منابع نیمه ساخت یافته و کاملاً ساخت یافته)، تغییرات ساختاری به راحتی قابل شناسایی می باشد. سوال اینجاست که آیا می توان با استفاده از آنتالوژی ها این نوع قواعد استخراج را نسبت به تغییرات ساختاری مقاوم ساخت؟

از طرفی دیگر، شناسایی تغییرات محتوایی در هنگام بازیابی اطلاعات بسیار مشکل می باشد. آیا می توان با استفاده از آنتالوژی ها این تغییرات را شناسایی کرد؟ و آیا می توان آن را اصلاح کرد؟ تا چه حد می توان به این روش بازیابی اطلاعات اعتماد نمود؟ این موضوع نیز یکی از موضوعات تحقیق آینده ما خواهد بود.

۵-۴- معرفی چارچوب پیشنهادی

با بررسی فعالیت های مرتبط و شناسایی نقاط قوت و ضعف آنها، معماری پیشنهادی خود را مطابق شکل زیر ارائه می کنیم.



شکل ۳۸: معماری پیشنهادی

این معماری، مبتنی بر یک Crawler می باشد که در بین منابع وب موجود جستجو کرده و بطور اتوماتیک اقدام به جمع آوری اطلاعات جهت تکمیل پایگاه دانش می نماید. داده های جمع آوری شده در قالب آنتالوژی در مخزن نمونه های آنتالوژی ذخیره و نگهداری می گردد. همانطور که دیده می شود، این معماری شامل چهار بخش اساسی می باشد:

۱. Semantic Access: این بخش امکان دسترسی معنایی به پایگاه دانش را فراهم می کند. این بخش شامل ماژول های موتور استنتاج (Inference Engine) و نمایش آنتالوژی (Ontology Presentation) می باشد. این ماژول ها اطلاعات مورد نیاز خود را از واحد مدیریت آنتالوژی ها دریافت کرده و به برنامه های کاربردی سرویس های مورد نظر را ارائه می کنند.

- Inference Engine : موتور استنتاج، بر اساس آنتالوژی های موجود در مخزن آنتالوژی های نمونه، پرس و جوهای وارده را با توجه به یک زبان منطق مانند F-Logic، یا Description Logic ارزیابی می نماید.
- Ontology Representation : این ماژول، با استفاده از یک آنتالوژی دامنه، اطلاعات جمع آوری شده را جهت انتقال به برنامه های کاربردی دیگر (Export) و یا جهت نمایش برای انسان ها (Visualizing)، آماده می نماید.

۲. Ontology Repository: واحد مخزن آنتالوژی ها مسئولیت حفظ و نگهداری آنتالوژی ها را در طول حیات یک آنتالوژی، بر عهده دارد. کلیه آنتالوژی ها در این واحد ذخیره می شوند و Semantic Access تنها می تواند از طریق لایه دسترسی به آنتالوژی، OAL^{۷۹}، به آنتالوژی های موجود دسترسی داشته باشد. این واحد شامل آنتالوژی و ماژول های زیر است:

- Domain Ontology : این آنتالوژی شامل مفاهیم و صفات بکار رفته در یک دامنه مشخص می باشد. کارشناس و خبره دامنه با استفاده از ابزار Ontology Editor، مفاهیم بکار رفته در دامنه مورد نظر را تعریف می نماید. این آنتالوژی، دانش اولیه سیستم را تشکیل داده و در هنگام بازیابی اطلاعات از منابع وب موجود، جهت شناسایی اقلام اطلاعاتی بکار می رود. از آنجا که ما بر روی دامنه خاصی تاکید نداریم می توان چندین آنتالوژی برای دامنه های مختلف تعریف نمود. از این آنتالوژی ها بعداً جهت ایجاد Wrapper ها استفاده می شود.
- Ontology Instances : همانطور که گفته شد، Wrapper ها با دریافت منابع وب موجود اقدام به استخراج اطلاعات می نمایند. سپس این اطلاعات استخراج شده با توجه به آنتالوژی دامنه اصلی، در مخزن Ontology Instances ذخیره و نگهداری می گردد.
- Ontology Access Layer: در این چارچوب دسترسی به آنتالوژی ها با استفاده از این لایه انجام می پذیرد. این لایه جزئیات ذخیره سازی آنتالوژی ها را برای برنامه های کاربردی، مخفی می نماید. همچنین در صورتیکه یکی از خصوصیات آنتالوژی دامنه (مانند قیمت یا مقدار) بصورت پویا تعریف شده باشد. زمان درخواست بازیابی آنتالوژی این خصوصیات در زمان اجرا بازیابی می گردند.
- Admin Access: این لایه جهت اجرای دستورات مدیریت و ساخت آنتالوژی ها در نظر گرفته شده است.

۳. Crawler: یک خزگر مبتنی بر آنتالوژی است که با استفاده از Wrapper های موجود و آنتالوژی های تعریف شده توسط کارشناس دامنه، اطلاعات موجود در منابع و صفحات وب را استخراج کرده و در مخزن آنتالوژی های نمونه نگهداری می نماید. از آنجا که قالب صفحات وب همواره در حال تغییر می باشند، لذا شناسایی این تغییرات و تولید مجدد Wrapper های مناسب بسیار حائز اهمیت می باشد. Crawler علاوه بر ماژول اجرای Wrapper شامل ماژولی بنام Verification نیز می باشد که می تواند به موازات اجرای Wrapper شروع به کار کرده و تغییرات احتمالی را شناسایی کند. خروجی ماژول Verification مجموعه ای از صفحات نمونه می باشد که جهت یادگیری و آموزش در Warpper Induction System مورد استفاده قرار می گیرد.

۴. Warpper Induction System: برای افزایش انعطاف پذیری سیستم و مقاومت در برابر تغییرات صفحات وب، در صورتیکه Crawler در هنگام اجرای ماژول Wrapper با مشکل مواجه شد، باید بتوانیم تا Wrapper جاری را ترمیم کرده و آن را با توجه به تغییرات انجام شده اصلاح نماییم. مهمترین ماژول این واحد Wrapper Generator نام دارد که با توجه به آنتالوژی دامنه و صفحات آموزشی نمونه (Training Samples) الگو های موجود در صفحات را شناسایی کرده و آنها را با توجه به آنتالوژی دامنه شناسایی می کند. سپس با توجه به الگوهای شناسایی شده، Wrapper های مورد نیاز را تولید می کند. نکته قابل توجه آن است که تمام این مراحل بصورت اتوماتیک و بدون نیاز به دخالت انسان صورت می گیرد.

همچنین ابزاری بنام Ontology Editor در نظر گرفته شده است تا کارشناس دامنه بتواند با استفاده از آن عملیات مدیریت آنتالوژی های دامنه را انجام دهد. کارشناس دامنه با استفاده از این ابزار، آنتالوژی های دامنه و Wrapper ها را معرفی می نماید. همچنین می تواند تعیین نماید که برخی اطلاعات، مانند قیمت، مقدار و غیره، در هنگام بازبایی آنتالوژی نمونه، بصورت آنلاین استخراج گردد. در ادامه برای آشنایی بیشتر با این معماری سه سناریو را مطرح می سازیم:

سناریوی یک: مدیریت و آماده سازی آنتالوژی های دامنه

کارشناس دامنه باید بتواند مجموعه سیستم را مدیریت کرده و آنتالوژی های لازم را جهت راه اندازی و عملیاتی نمودن سیستم فراهم نماید. بدین منظور ابزاری بنام Ontology Editor تعبیه شده است. این ابزار، قابلیت ایجاد و مدیریت آنتالوژی های دامنه و نیز تهیه نسخه های نمونه آموزشی جهت یادگیری سیستم تولید اتوماتیک Wrapper ها و همچنین کنترل نحوه اجرای Crawler در بازه های زمانی مشخص را فراهم می کند.

Ontology Editor جهت ارتباط با واحد Ontology Repository از ماژول Admin Access استفاده می کند. این ماژول امکان انجام فعالیت های زیر را فراهم می کند:

۱. ایجاد و بروزرسانی آنتالوژی دامنه
۲. ایجاد و بروزرسانی نسخه های آموزشی جهت یادگیری سیستم تولید اتوماتیک Wrapper ها
۳. کنترل نحوه اجرای Crawler

سناریوی دو: جستجوی معنایی

در این معماری کاربران تنها می توانند از طریق واحد Semantic Access با اطلاعات موجود در پایگاه دانش دسترسی داشته باشند. فرایند جستجوی معنایی بصورت زیر می باشد:

۱. کاربر درخواست پرس و جوی مورد نظر خود را از طریق یک واسط گرافیکی کاربر (GUI) ایجاد می کند. این پرس و جو مبتنی بر فرم بوده و توسط رابط گرافیکی کاربر ایجاد شده و برای ماژول استنتاج معنایی ارسال می گردد.
 ۲. ماژول استنتاج معنایی مبتنی بر یک منطق مانند F-Logic می باشد. این ماژول، پرس و جوی کاربر را دریافت کرده و با توجه به زبان پردازش منطق درخواست بازیابی بخش هایی از اطلاعات موجود در پایگاه دانش را برای لایه دسترسی به آنتالوژی ها ارسال می کند.
 ۳. لایه دسترسی به آنتالوژی ها درخواست رسیده را بررسی کرده و اطلاعات مورد نظر را از مخزن آنتالوژی نمونه بازیابی می کند.
 ۴. از آنجا که ممکن است دسترسی به برخی از خصوصیات آنتالوژی ها بصورت آنلاین تعریف شده باشد، لایه OAL این موضوع را نیز بررسی کرده و در صورت نیاز از ماژول Extract Online Data Items می خواهد تا اقلام اطلاعاتی مورد نظر را بصورت درجا بازیابی نماید:
- ماژول Extract Online Data Items درخواست مورد نظر را برای Wrapper مربوطه ارسال می کند.
 - با اجرای Wrapper مربوطه، اقلام اطلاعاتی مورد نظر بازیابی شده و به ماژول درخواست کننده ارسال می گردد.
۵. در نهایت موتور استنتاج نتیجه نهایی را محاسبه کرده و برای کاربر درخواست کننده ارسال می نماید.

سناریوی سه: استخراج اطلاعات بصورت اتوماتیک

Crawler در بازه های زمانی مناسب که توسط مدیریت سیستم تعیین می شود، بصورت زیر اجرا می گردد:

۱. برای هر آنتالوژی موجود در مخزن آنتالوژی های دامنه صفحات وب مرتبط شناسایی شده و بازیابی می گردند.
۲. Wrapper مربوط به هر صفحه اجرا شده و اقلام اطلاعاتی موجود بازیابی می گردند.
۳. در صورت موفقیت در اجرای Wrapper اقلام اطلاعاتی بازیابی شده در مخزن آنتالوژی نمونه ثبت می گردد.
۴. در صورت عدم موفقیت در اجرای Wrapper عملیات Verification انجام می گیرد تا در صورت امکان Wrapper جاری اصلاح گردد. نتیجه اجرای Verification مجموعه ای از صفحات جهت آموزش و یادگیری الگوهای موجود خواهد بود.
۵. صفحات نمونه آموزشی که توسط ماژول Verification تولید شده اند، در مخزن Training Samples ذخیره می گردند.
۶. Wrapper Generator با توجه به آنتالوژی دامنه و با استفاده از صفحات آموزشی موجود در مخزن Training Samples اقدام به تولید Wrapper های جدید می نماید.

۵ - ۵ - فعالیت های بعدی

تکمیل و پیاده سازی چارچوب پیشنهاد شده در این تحقیق، سبب ایجاد موضوعات جدیدتری در زمینه های مختلف می گردد که در ادامه پروژه قصد داریم تا آن ها را بررسی نماییم. ذیلا به معرفی برخی از این موضوعات می پردازیم.

اولین موضوع مربوط به نحوه جستجو در میان صفحات وب (جهت یافتن اقلام اطلاعاتی) می باشد. از آنجا که هدف نهایی استخراج اطلاعات موجود در قالب آنتالوژی های نمونه می باشد، بنابراین بنظر می رسد، در هنگام خزش در وب باید نقش آنتالوژی دامنه را هم در نظر داشت.

یکی دیگر از موضوعات جالب، یافتن صفحات مناسب جهت آموزش و استفاده در ماژول تولید Wrapper (البته بصورت اتوماتیک توسط ماشین) می باشد. احتمالا لازم است در این خصوص از برخی روش های یادگیری ماشین^{۸۱} نیز استفاده گردد. چرا که هرچقدر بخش تولید Wrapper کامل تر باشد، میزان استقلال عمل سیستم نیز بیشتر خواهد گشت.

از آنجا که چارچوب پیشنهاد شده در این نوشته، بر اساس آنتالوژی های دامنه اقدام به استخراج اتوماتیک اطلاعات می نماید، بنابراین هرچه آنتالوژی های دامنه کاملتر و غنی تر باشد، داده های استخراج

⁸¹ Machine Learning

شده نیز دقیق تر خواهد بود و نیاز احتمالی به نظارت انسان از بین خواهد رفت. از این رو ارائه الگویی جهت ساخت آنتالوژی های دامنه و یا شاید معرفی یک آنتالوژی سطح بالا^{۸۲} یکی دیگر از زمینه های تحقیق ما خواهد بود.

در نهایت، استخراج داده های ضمنی از منابع وب و همچنین ایجاد یک تراکنش هنگام استخراج همزمان اطلاعات مربوط به یک موضوع از چندین صفحه وب، از جمله موضوعاتی است که هنگام تحلیل و پیاده سازی سیستم آنها را بررسی خواهیم کرد.

[Introduction to Semantic Web]

- [1] Tim Berners-Lee, CERN, 1989. *"Information Management: A Proposal"*, <http://www.w3.org/History/1989/proposal.html>.
- [2] Berners-Lee, T, Hendler, J & Lassila, O, 2001. *"The semantic web"*, Scientific American.
- [3] Michael C. Daconta, Leo J. Obrst, Kevin T. Smith, 2003. *"The Semantic Web: A Guide to the Future of XML, Web Services and Knowledge Management"*, Wiley publishing.
- [4] Semantic Web Resources in W3C, Last Visited: 05/05/2007, <http://www.w3.org/2001/sw/>.

[Semantic Web Languages References]

- [5] N. Walsh. *"A Technical Introduction to XML"*. O'Reilly XML.com, October 3, 1998. At: <http://www.xml.com/pub/a/98/10/guide0.html>
- [6] New Architecture for Semantic Web, 2003, <http://www.w3.org/2003/Talks/0922-rsoc-tbl/>
- [7] New Architecture for Semantic Web, 2005, <http://www.w3.org/2005/Talks/0511-keynote-tbl/>
- [8] Extensible Markup Language (XML), Last Visited: 05/05/2007, <http://www.w3.org/XML>
- [9] XML Schema, Last Visited: 05/05/2007, <http://www.w3.org/XML/Schema>
- [10] XSL Transformations (XSLT) Reference, W3C Technical Reports, Last Visited: 05/05/2007, <http://www.w3.org/TR/xslt>
- [11] XML Path Language (XPath) Version 1.0, W3C Technical Reports, Last Visited: 05/05/2007, <http://www.w3.org/TR/xpath>
- [12] Extensible Stylesheet Language (XSL) Version 1.1, W3C Technical Reports, Last Visited: 05/05/2007, <http://www.w3.org/TR/xsl>
- [13] Resource Description Framework (RDF), Last Visited: 05/05/2007, <http://www.w3.org/RDF>
- [14] RDF Validation Service, Last Visited: 05/05/2007, <http://www.w3.org/RDF/Validator>
- [15] OWL Web Ontology Language Reference, W3C Technical Reports, Last Visited: 05/05/2007, <http://www.w3.org/TR/owl-ref/>

[Miscellaneous]

- [16] Internet Software Consortium, Last Visited: 05/05/2007, www.isc.org, January 2006

[Knowledge Management & Information Retrieval]

- [17] John Davies, Dieter Fensel, and Frank van Harmelen, editors. **"Towards the Semantic Web: Ontology-Driven Knowledge Management"**. John Wiley & Sons, 2003.
- [18] Broekstra, J. and Kampman, A., 2001. **"Sesame: A generic Architecture for Storing and Querying RDF and RDF Schema"**. Deliverable 10, On-To-Knowledge project, October. <http://www.ontoknowledge.org/download/del10.pdf>
- [19] Wessman, A., Liddle, S.W., Embley, D.W., **"A generalized framework for an ontology-based data-extraction system"**. In Proc. 4th Int. Conference on Information Systems Technology and its Applications, 239-253, 2005.
- [20] Shah, U., T. Finin, J. Mayfield. **"Information retrieval on the Semantic Web"** In Proceedings of the Eleventh International Conference on Information and Knowledge Management, McLean, Virginia, 4-9 November 2002, pp. 461-468.
- [21] Simon, K., and Lausen, G. 2005. **"ViPER: augmenting automatic information extraction with visual perceptions"**. In Proc. CIKM'05, 381--388. ACM.
- [22] Snoussi, H., Magnin, L. and Nie, J.-Y, **"Toward an Ontology-based Web Data Extraction"**, The AI-2002 Workshop on Business Agents and the Semantic Web (BASEWEB) held at the AI 2002 Conference (AI-2002), Calgary, Alberta, Canada, May 26, 2002.
- [23] Bartlett W., **"A Comparison of Techniques for Exposing Legacy Data to Semantic Web Technologies"**, In Proc. 21st Annual Computer Science Conference, 2005.
- [24] Robert Engels, 2002. **"CORPORUM-OntoExtract: Extraction of structured information from web based resources"**, OnToKnowledge Project Report, Delivery 6.
- [25] Robert Engels, 2002. **"CORPORUM-OntoWrapper: Extraction of structured information from web based resources"**, OnToKnowledge Project Report, Delivery 7.
- [26] The Onto-Knowledge Toolset, Last Visited: 05/05/2007, <http://www.ontoknowledge.org/tools/toolrep.shtml>.
- [27] The Ontoprise homepage, Last Visited: 05/05/2007, <http://www.ontoprise.com>.
- [28] Best known annotation Tools and Frameworks, Last Visited: 05/05/2007, <http://annotation.semanticweb.org/tools/>.
- [29] KIM Online reference, Last Visited: 05/05/2007, <http://www.ontotext.com>
- [30] Laender, A., Ribeiro-Neto, B., Silva, A. and Teixeira, J. **"A Brief Survey of Web Data Extraction Tools"**, in: SIGMOD Record, Volume 31, Number 2, June 2002.
- [31] SESAME Online Open Source, Last Visited: 05/05/2007, <http://www.openrdf.org/documentation.jsp>.
- [32] B. Popov, A. Kiryakov, D. Ognyanoff, D. Manov, A. Kirilov, M. Goranov, **"Towards Semantic Web Information Extraction"**, Human Language Technologies Workshop at the 2nd International Semantic Web Conference (ISWC2003), Florida, USA, 2003.
- [33] Cunningham H., Maynard D., Bontcheva K. and Tablan V., **"GATE: A Framework, Graphical Development Environment for Robust NLP Tools and Applications"**. In Proc. of the 40th Anniversary Meeting of the Association for Computational Linguistics, 2002.

- [34] Lucene - high performance full text search engine, Last Visited: 05/05/2007.
<http://jakarta.apache.org/lucene>.
- [35] Frohn, J., Himmeröder, R., Kandzia, P.T., Lausen, G., Schlepphorst, C., "**FLORID: A Prototype for F-Logic**", In: ICDE'97, IEEE Computer Society (1997) 583.
- [36] "**How to Write F-Logic Programs**," Online. Internet. Last Visited: 05/05/2007,
http://ontoprise.de/documents/tutorial_flogic.pdf
- [37] Arocena, G., Mendelzon, A., "**WebOQL: Restructuring Documents, Databases and Webs**", Proceedings of International Conference on Data Engineering, 1998.
- [38] A. Sahuguet and F. Azavant. "**Building Intelligent Web Applications Using Lightweight Wrappers**". to appear in: Data and Knowledge Engineering, 2000.
- [39] Thomas Hornung, Kai Simon, Georg Lausen, "**Information Gathering in a Dynamic World**". PPSWR 2006: 237-241
Principles and Practice of Semantic Web Reasoning, 4th International Workshop, PPSWR 2006, Budva, Montenegro, June 10-11, 2006, Revised Selected Papers.