دانشگاه صنعتی امیرکبیر

(پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر

پایان‌نامه کارشناسی

گرایش نرم افزار

طراحی و پیاده سازی سامانه ای جهت جستجوی سوالات مشابه از طریق خزش سایت های پرسش و پاسخ فارسی

نگارش

محمدمهدی سمیعی پاقلعه

استاد راهنما

دکتر سیدرسول موسوی

مرداد ۱۳۹۷

به نام خدا

تعهدنامه اصالت اثر

تاریخ:

دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

اینجانب محمدمهدی سمیعی پاقلعه متعهد می‌شوم که مطالب مندرج در این پایان نامه حاصل کار پژوهشی اینجانب تحت نظارت و راهنمایی اساتید دانشگاه صنعتی امیرکبیر بوده و به دستاوردهای دیگران که در این پژوهش از آنها ا ستفاده شده ا ست مطابق مقررات و روال متعارف ارجاع و در فهر ست منابع و مآخذ ذکر گردیده است. این پایان نامه قبلاً برای احراز هیچ مدرک هم‌سطح یا بالاتر ارائه نگردیده است.

در صورت اثبات تخلف در هر زمان، مدرک تحصیلی صادر شده توسط دانشگاه از درجه اعتبار ساقط بوده و دانشگاه حق پیگیری قانونی خواهد داشت.

کلیه نتایج و حقوق حاصل از این پایان نامه متعلق به دانشگاه صنعتی امیرکبیر می‌باشد. هرگونه استفاده از نتایج علمی و عملی، واگذاری اطلاعات به دیگران یا چاپ و تکثیر، نسخه‌برداری، ترجمه و اقتباس از این پایان نامه بدون موافقت کتبی دانشگاه صنعتی امیرکبیر ممنوع است. نقل مطالب با ذکر مآخذ بلامانع است.

امضا

صفحه تقدیر و تشکر

در ابتدا بایستی از جناب آقای دکتر موسوی جهت ایده‌پردازی اولیه این پروژه و راهنمایی‌های بی دریغشان در طول اجرای آن تشکر کنم.

چکیده

سیستم‌های پرسش و پاسخ انجمنی[1] سامانه‌های تحت وبی هستند که کاربران آن‌ها ضمن مطرح کردن پرسش‌های خود، می‌توانند به پرسش‌های مطرح شده توسط دیگر کاربران نیز پاسخ دهند. در سال‌های اخیر با افزایش پرشتاب تعداد کاربران سامانه‌های پرسش و پاسخ فارسی و همچنین محتواهای تولید شده توسط آن‌ها شاهد این هستیم که این سامانه‌ها فضای اطلاعاتی غنی و البته گسترده‌ای را فراهم کرده‌اند. گستردگی این فضا باعث شده است تا کاربران نتوانند برای پرسش خود از میان انبوه‌ها پرسش‌هایی که قبلا در سامانه‌های پرسش و پاسخ مطرح شده اند پرسش‌های مشابه را پیدا کنند. این امر می‌تواند موجب اتلاف وقت در سمت کاربران و همچنین به وجود آمدن محتوای تکراری در سمت سامانه‌های پرسش و پاسخ شود. به همین دلیل نیازمند قابلیتی هستیم که طی آن بتوانیم محتواهایی را که قبلا در این سامانه‌ها مطرح شده‌اند و مشابه با پرسش جدید کاربر هستند را به دست آوریم. ما در این پروژه به دنبال طراحی و پیاده‌سازی سامانه‌ای هستیم که در بازه‌های زمانی به صورت مداوم سوالات مطرح شده در سیستم‌های پرسش و پاسخ را استخراج کند و با دریافت یک پرسش از کاربر خود، لیست پرسش‌های مشابه با آن پرسش را به ترتیب میزان مشابهتشان نشان دهد. نتیجه حاصله سامانه‌ای خواهد بود که خود یا مستقلا می‌تواند به عنوان یک موتورجستجوگر پرسش و پاسخ مورد استفاده قرار گیرد و یا این که با قرار گرفتن به عنوان یک زیر سامانه جز یک سامانه پرسش و پاسخ انجمنی به کاربران آن جهت پیدا کردن پرسششان کمک کند و مانع از تولید محتوای تکراری شود.

واژه‌های کلیدی:

سیستم پرسش و پاسخ انجمنی، بازیابی اطلاعات، خزشگری، جستجوی سوالات مشابه

[1] Community Question Answering (CQA)

فهرست مطالب

صفحه

فهرست اشکال

صفحه

فهرست جداول

صفحه

فصل اول

مقدمه

مقدمه

۱-۱- بیان موضوع

تالار گفتگو یا فروم به برنامه‌های مبتنی بر وب گفته می‌شود که برای نگهداری بحث‌ها و نوشته‌های کاربرهای یک وبگاه به کار می‌روند. در تالار گفتگو معمولاً افراد سؤالات خود را مطرح می‌کنند و سایر کاربران به این سؤالات پاسخ می‌دهند. علاوه بر این تالارهای گفتگو به جز سؤال پرسیدن می‌توانند واسطه بحثی در هر زمینه باشند و افراد می‌توانند نظرات و عقیده‌های خود را برای دیگران بیان کنند. از معروف ترین تالارهای گفتگوی فارسی می‌توان به ninisite.com اشاره کرد. همچنین در کنار این تالارهای گفتگو، وبگاه‌هایی نیز موجودند که اغلب حالت انجمنی دارند که به کاربران اجازه می‌دهند در آن هم پرسش مطرح کنند و هم به پرسش‌های دیگر کاربران پاسخ دهند. نمونه معروف این وبگاه‌ها تارنمای stackexchange.com است که در حوزه تخصصی برنامه نویسی دارای دامنه stackoverflow.com است و یا در میان سامانه‌های فارسی، از javabyab.com و porsak.ir می‌توان نام برد.

یکی از کاربری های روزمره اینترنت جستجو کاربران آن برای یافتن اطلاعات مورد نیاز خود و یا یافتن پاسخ پرسش‌هایشان است. در این میان منابع متفاوتی نظیر سایت‌های خبری، بلاگ‌های شخصی، شبکه‌های اجتماعی در بستر اینترنت وجود دارند. اما در این بین منابع تعاملی بیشتر مورد توجه کاربران قرار دارند و اغلب کاربران تمایل دارند تا در فضاهای تعاملی نظیر انجمن‌های گفتگو یا سامانه‌های پرسش و پاسخ انجمنی سوالات خود را جستجو کنند یا این که در نهایت در آنجا آن‌ها را مطرح کنند. این سایت‌ها انجمن و اجتماعی به وجود می‌آورند که به کاربران خود این قابلیت را می‌دهند که سوالات خود را در زمینه‌های متفاوت مطرح کنند و از یکدیگر پاسخ دریافت کنند. نکته حائز اهمیتی که در این میان وجود دارد این است که در کشورهایی مانند ایران که محتوای غیرانگلیسی کمتری (نسبت به محتوای انگلیسی زبان) در وبسایت‌ها یا بلاگ‌ها وجود دارد، این انجمن‌های گفتگو و سامانه‌های پرسش و پاسخ هستند که منبع اصلی کاربران وب برای یافتن پاسخ پرسش‌هایشان است.

در سال‌های اخیرتعداد کاربران این سیستم‌ها به طور روزافزون در حال افزایش است که نتیجه آن افزایش چشمگیر تعداد پرسش‌ها و پاسخ های مطرح شده در آن‌ها نیز است.به عنوان مثال تعداد پرسش‌های

مطرح شده در Stack Overflow که از سیستم‌های معروف پرسش و پاسخ انجمنی در سطح جهان است، تا به امروز حدود ۱۶ میلیون بوده و این رقم روز به روز در حال افزایش است.[۱] از نمونه‌های فارسی نیز میتوان به مورد ninisite.com اشاره کرد که بنا بر استناد پایگاه alexa چهاردهمین سایت پربازدید در میان تمامی سایت‌های ایرانی است.[۲] که این پدیده خود حاکی از اقبال بالای کاربران به این تالار گفتگو است.

تالارهای گفتگو و سایت‌های پرسش و پاسخ هر کدام معمولا در موضوع خاصی به فعالیت می‌پردازند و در هر کدام از آنها بنابر کاربردشان دسته سوالات مربوط به آن موضوعات نظیر تحصیلات، سرگرمی، ورزشی، مذهبی، پزشکی و یا بقیه موارد مطرح می‌شوند. در این صورت کاربران جهت پرسیدن سوال خود بایستی به تالار گفتگوی متناظر به آن موضوع مراجعه کنند و سوال خود را در آنجا مطرح کنند. در اینگونه سایت‌ها، فراوان پیش می‌آید که پرسش کاربر از قبل و در میان پرسش‌های پیشین موجود باشد و صرفاً جستجو به دنبال پرسش و نه طرح پرسش جدید کاربر را به اطلاعات موردنیازش برساند. پرسش‌های مشابه قبلاً پاسخ داده شده‌اند و بنابراین کاربران اغلب با یافتن پرسش مورد نظر به پاسخ مطلوبشان خواهند رسید. در نتیجه نیازمند سیستمی هستیم که با دریافت یک پرسش از کاربر پرسش‌های قبلا مطرح شده مشابه آن را به او نمایش دهند. از آنجایی که تعداد پرسش‌های موجود در سایت‌های پرسش و پاسخ زیاد است در نتیجه برای یک پرسش تعداد زیادی پرسش مشابه وجود دارد. از این رو سیستم مذکور همچنین باید بتواند پرسش‌های مشابه را بر اساس میزان مشابهاتشان با پرسش اولیه، رتبه بندی کند و به ترتیب مشابهت آن‌ها را به کاربر نمایش دهد. در این راستا در این پروژه به دنبال این هستیم که تا برای کاربر این امکان را فراهم کنیم که بتواند سوال خود را در میان انبوه سوالات مطرح شده در چند تالار گفتگو و و وبگاه‌های پرسش و پاسخ جستجو کند.

[۱] آمار مربوط به آگوست سال ۲۰۱۸ میلادی و برگرفته شده از بخش آمار سایت stackexchange است.

[۲] آمار مربوط به آگوست سال ۲۰۱۸ میلادی است.

۲-۱- پیشینه

پس از تحقیقات انجام‌شده مشخص گردید که در فضای سایت‌های فارسی تنها سایت موتورجستجوگر امین با قابلیت‌های مشابه اهداف این پروژه موجود می‌باشد. پاراگراف زیر توصیفی است که این سایت از خود ارائه داده است:

موتور جستجوی پرسش و پاسخ دینی امین[1]، به‌عنوان یک سامانه اطلاعاتی برای دسترسی حداکثری مخاطبان به پرسش‌ها و پاسخ‌های موجود در سطح وب در حوزه علوم اسلامی است که با جمع‌آوری اطلاعات از سایت‌ها و منابع دیجیتال از مراکز معتبر پاسخ‌گویی به سؤالات دینی؛ و تجمیع بانک جامعی از سؤالات (۳۹۹۶۶مسئله شرعی و ۴۰۴۸۲ استفتاء از مراجع و ۵۴۴۶۶ پرسش و پاسخ) با امکان بازیابی اطلاعات به روش‌های مناسب، یکی از کامل‌ترین مراجع دسترسی به پرسش و پاسخ و مفاهیم دینی در وب محسوب می‌شود.

این سایت در واقع صرفا بر روی سایت‌های مراجع تقلید خزشگری[2] کرده و فتاوای آن‌ها را استخراج می‌کند و سپس به کاربر امکان جستجوی سوال بر روی این مجموعه سوالات را می‌دهد. تفاوت‌های اصلی بین این سایت و چالش‌های پروژه ما عبارتند از:

- مجموعه اسناد خزشگری شده توسط این سایت تنها صرفا در حوزه علوم دینی می‌باشد و بنابراین اندازه لغات موجود در این اسناد بسیار محدودتر از اندازه لغات موجود در پروژه ما است که در آن پرسش‌های سایت‌های پرسش و پاسخ با تنوع موضوعی گوناگونی را مورد استخراج قرار می‌دهیم.
- این موتور جستجو خزشگری خود را بر روی منابعی انجام می‌دهد که به کندی پرسش‌های جدید در آن‌ها تولید می‌شوند، حال در آن که در پروژه ما به علت تعامل با سایت‌های پرسش و پاسخ که حالت تعاملی دارند، در بازه‌های زمانی کوتاه‌مدت شاهد تولید پرسش‌های جدید از سمت کاربران آن سایت‌ها هستیم.

[1] aminsearch.com

[2] crawling

به جز موتور جستجوگر امین که در بالا به آن اشاره شد، متاسفانه سایتی که دارای ویژگی ذکر شده در این پروژه یعنی خزشگری و قابلیت جستجوی بین سوال‌های سایت پرسش و پاسخ باشد، وجود ندارد.

اما برخلاف عدم وجود نمونه مشابه این پروژه در حوزه سایت‌های فارسی، در حیطه‌ی زبان انگلیسی سایت‌های متعددی جهت انجام عمل جستجو روی تالارهای گفتگوی انگلیسی زبان وجود دارند. نشانی یکی از این سایت‌ها در ذیل آورده شده است:

Boardreader.com

این سامانه داده هایش را از طریق خزش روی رسانه‌های اجتماعی (نظیر سایت‌ها، بلاگ‌ها و فروم‌ها) در سطح اینترنت جمع آوری می‌کند. خزشگرهای این سامانه اطلاعات را گردآوری می‌کنند و سپس داده‌های جمع آوری شده مورد نمایه‌سازی، جستجو و در نهایت مورد تحلیل و آنالیز واقع می‌شوند. شعار این سایت این است که کاربران بتوانند بحث‌هایی که بین دو طرف انسانی در فضای وب وجود دارند را جستجو کنند. همچنین این سایت رابط کاربری استفاده از امکانات خود را با اخذ مبلغی به مشتریان می‌فروشد.

۳-۱- تعریف مسئله

هدف ما در این پروژه پیاده‌سازی یک سیستم جستجوگر برای یافتن پرسش‌های مشابه از میان پر سش‌های مطرح شده در میان منابع پر سش‌ها برای یک پر سش دریافتی ا ست. منابع پر سش‌های ما چندین تالار گفتگو و سیستم پرسش و پاسخ انجمنی هستند که عبارتند از:

- وبسایت ninisite.com
- وبسایت applyabroad.org
- وبسایت porsak.ir
- وبسایت javabyab.com
- بخش مشاوره سایت tebyan.net

برای نیل به هدف فوق باید چهار فاز عملیاتی زیر محقق شوند:

1 - پیاده سازی بخش خزشگر: در این فاز باید هر یک از وبگاه‌های هدف را به طرز خاصی که مخصوص آن وبگاه است (منظور از مخصوص این است که پرسش‌های آن سایت به درستی و به طور کامل استخراج شوند). از آنجایی که ما در این پروژه تنها پنج سایت پرسش و پاسخ را منبع پرسش‌های خود قرار داده ایم در نتیجه با پرهیز از پیاده‌سازی یک خزشگر عام منظوره با کارایی پایین، به دنبال پیاده سازی پنج خزشگر خاص منظوره برای هر یک از سایت‌ها با کارایی بالا هستیم تا سوالات آن‌ها را به طور دقیق استخراج کنیم. خزشگرهای ما باید این ویژگی را داشته باشند که به صورت مداوم و طی بازه‌های زمانی مشخص و قابل تغییر سایت‌های هدف خودشان را خزش کنند. به علاوه یکی از چالش‌های اساسی در این فاز بازیابی سوالات از سایت‌هایی است که سرور آن‌ها اطلاعات را به صورت static html نمایش نمی‌دهند. بلکه از تکنولوژی های ajax و تحویل کد js استفاده می‌کنند.

2 - انجام پیش پردازش و پالایش روی متن سوالات: حاصل نهایی فاز قبل برای ما متون استخراج شده خام سوالات به همراه لینک آن‌هاست. در این فاز ما عملیات‌های پیش‌پردازش‌های زبانی را روی متن خام سوالات انجام داده و سپس دوباره متن پالایش شده آن‌ها را ذخیره می‌کنیم. با توجه به مداوم بودن عملیات خزشگر در طی بازه‌های زمانی این نکته واضح است که این فاز (فاز پالایش) نیز باید به صورت مداوم و طی بازه‌های زمانی مشخص انجام گیرند.

3 - پیاده سازی جدول شاخص معکوس[1]: در این فاز با استخراج کلمات از متن پالایش شده سوالات آن‌ها را در یک جدول که شاخص‌های آن کلمات و درایه‌های آن زوج مرتب‌های اشاره‌گر به سوالات حاوی آن کلمات و تعداد باز ظاهرشدن کلمه در آن سوالات است ذخیره‌سازی می‌کنیم. مزایای پیاده‌سازی این داده‌ساختار در فصل خود مورد بررسی قرار خواهد گرفت.

4 - پیاده‌سازی الگوریتم bm25 جهت جستجو میان سوالات: در این فاز ما با دو عنصر مواجه هستیم، عنصر اول پرسشی است که کاربر در ورودی به ما داده و در پی یافتن پرسش‌های مشابه آن است؛ عنصر دوم نیز داده‌های حاصل از فازهای قبلی است. جهت جستجو و تعیین میزان شباهت بین پرسش کاربر و پایگاه داده پرسش‌ها از الگوریتم bm25 استفاده می‌کنیم. کاربر این الگوریتم این است که

[1] Inverted index

با گرفتن یک متن از ورودی در میان مجموعه‌ای از اســـناد ســـعی در امتیازدهی به این اســـناد از لحاظ شباهت به متن ورودی دارد.

در انتها با پیاده سازی یک رابط برنامه نویسی تحت وب که قادر به دریافت پر سش و بازگرداندن پرسش‌های مشابه آن به ترتیب میزان مشابهتشان است قسمت پیاده‌سازی پروژه پایان می‌پذیرد.

۴-۱- خلاصه فصل‌های دو الی شش

در ادامه در ف صل دو به چگونگی خزش سی ستم‌های پر سش و پا سخ و پالایش متون سوالات پرداخته شده است. سپس در فصل سه انواع مدل‌های بازیابی اطلاعات مورد برر سی قرار گرفته اند. پس از آن در فصل چهار شرح پیاده سازی این پروژه در بخش جدول شاخص معکوس و جستجو روی سوالات و رابط کاربری تحت وب آمده است. سپس در فصل پنج روش تست و ارزیابی این سیستم مورد بحث قرار گرفته است. در انتها در فصل شش نتیجه‌گیری و پیشنهادات برای کارهای آینده درج شده است.

فصل دوم

جمع‌آوری و پالایش داده‌ها

جمع‌آوری و پالایش‌داده‌ها

۲-۱- مقدمه

قدم اول پیش از پیاده‌سازی سیستم جستجو، جمع‌آوری پرسش‌ها از سایت‌های پرسش و پاسخ هدف و پالایش متن آن‌هاست. در این فصل ابتدا به نحوه خزش سایت‌های مختلف و معماری خزشگرهای پیاده‌سازی شده و سپس در انتها به نحوه پالایش متن سوالات می‌پردازیم.

۲-۲- خزش سایت‌های پرسش و پاسخ

۲-۲-۱- درباره خزشگری

امروزه اینترنت با داشتن بیش از ۴ میلیارد کاربر، بزرگترین شبکه موجود در جهان است.[1] بخش بزرگی از کاربران اینترنت خود تولید کننده محتوا هستند، بدین صورت که با تولید پست‌های جدید در بلاگ‌ها، اخبار جدید در سایت‌های خبری، مقالات جدید در سایت‌های علمی، مطالب جدید در شبکه‌های اجتماعی و نظرات ارائه شده برای سایر محتواهای موجود در سطح اینترنت، هر روز حجم عظیمی از محتوا به محتواهای موجود در بستر اینترنت اضافه می‌شود. در چنین شرایطی می‌توان اینترنت را به اقیانوسی عمیق و وسیع تشبیه کرد که بزرگترین منبع اطلاعات و محتوای تاریخ بشریت است. در چنین شرایطی برای استخراج اطلاعات از عمق این اقیانوس نیازمند ابزاری هستیم که این امر را برای ما ممکن سازد. خزشگرهای وب این خواسته ما را محقق می‌کنند. خزشگر وب یک برنامه رایانه‌ای (یا در واقع یک روبات اینترنتی) است که به صورت خودکار در فضای وب به کند و کاو پرداخته و توانایی ذخیره‌سازی محتوای موجود بر بستر وب را داراست. امروزه از خزشگرها در جمع‌آوری اطلاعات برای کاربردهای مختلفی نظیر موتورهای جستجوگر، داده‌کاوی، پردازش زبان، آمار و ... استفاده می‌شود. در این پروژه نیز جهت جمع آوری پرسش‌های موجود در سایت‌های پرسش و پاسخ انجمنی با ایجاد خزشگرهای تحت وب از آنها استفاده شده است.

۲-۲-۲- چالش‌ها و اهداف خزشگری پروژه

با توجه به تعریف مسئله ارائه شده در فصل مقدمه، در بخش پیاده‌سازی خزشگرها ما با چالش‌های زیر مواجه هستیم:

1 - خزشگرهای ما بایستی عمل خزش روی سایت‌ها را به صورت مداوم و دوره ای انجام دهند. به این شرح که هر خزشگر طی بازه‌های زمانی معین (مثلا هر ۴ ساعت یکبار) به خزش سایت هدف خود بپردازد. ضمنا این بازه‌های زمانی باید در طول دوره حیات سامانه قابل تغییر و تنظیم باشند.

2 - خزشگر بایستی از خزش مجدد لینک‌هایی که قبلا آن‌ها را مورد پردازش قرارداده پرهیز کند. در صورت عدم امکان دستیابی به این مهم، خزشگر باید خزش این لینک‌های تکراری را به حداقل برساند. در صورتی که در سایت‌های هدف پرسش‌ها به ترتیب ایجادشان نمایش‌داده شوند نیل به این هدف آسان است اما متاسفانه در بعضی از سایت‌های هدف این امر تحقق نیافته است. در نتیجه در پیاده سازی هر خزشگر برای هر سایت مجبوریم استراتژی جداگانه‌ای را برای غلبه بر چالش خزش مجدد لینک‌های بازدید شده اتخاذ کنیم. شرح این استراتژی ها جلوتر آمده است.

3 - صفحات اینترنتی از لحاظ نوع تحویل از سرویس‌دهنده به سرویس گیرنده به دو نوع ایستا[1] و پویا[2] تقسیم می‌شوند. صفحات ایستا صفحاتی هستند که به طور کامل در سمت سرویس‌دهنده[3] تولید شده و به همان شکل دقیقا به سرویس‌گیرنده[4] تحویل داده می شوند. در مقابل، صفحات پویا، صفحاتی هستند که شکل آن‌ها ثابت است اما محتوای آن‌ها متغیر می‌باشد و فرآیند پر کردن شکل آن‌ها با محتوا، از طریق ارسال اسکریپت از سمت سرویس گیرنده کنترل می‌شود. در واقع در این گونه صفحات، سرویس‌دهنده تنها یک قالب که در میان آن کدهای جاوااسکریپت قرار داده شده است را به سمت سرویس گیرنده می‌فرستد. سپس در سمت سرویس گیرنده با پردازش کدهای جاوا اسکریپت صفحه پر می‌شود. فرآیند خزشگری بر روی سایت‌های پویا چالش‌ها و دشواری‌های متفاوت‌تری نسبت به خزشگری سایت‌های ایستا دارد و در

[1] static

[2] dynamic

[3] Server

[4] Client

نتیجه برای خزش سایت‌های پویا نمی‌توانیم عینا همان روش‌ها و ابزارهای خزش روی سایت‌های ایستا را به کار ببندیم.

۳-۲-۲- معماری کلی خزشگرهای پیاده‌سازی شده

ایده کلی ما برای این بخش از پروژه پیاده‌سازی پنج خزشگر برای پنج سایت پرسش و پاسخ هدفمان است؛ به این صورت که هر خزشگر برای خود پایگاه داده کوچکی (دقت شود که این پایگاه داده برای هر خزشگر مجزا از پایگاه‌داده سایز خزشگرهاست) دارد که پس از استخراج هر پرسش، عنوان پرسش و متن آن را به همراه لینکش به پایگاه داده‌خود اضافه می‌کند. وجود این پایگاه داده دو علت دارد، اولا این که خزشگر در خزش‌های بعدی خود بتواند پرسش‌های خزش‌شده را از پرسش‌های خزش نشده تمییز دهد و دوما این که در مرحله بعدی عامل پالایشگر بتواند پرسش‌های خزش شده توسط هر خزشگر را دریافت کرده و عمل پالایش را روی آن‌ها انجام دهد.

زبان مورد استفاده ما برای توسعه خزشگرها پایتون می‌باشد. استفاده از این زبان به علت گستردگی کتابخانه‌های موجود برای کاربردهای موردنیاز پروژه و همچنین وجود منابع و مستندسازی‌های فراوان برای کتابخانه‌های این زبان در سطح اینترنت است. به علاوه برای مدیریت کتابخانه‌ها و وابستگی‌های افزوده‌شده به پروژه از سیستم مدیریت وابسته‌های پیپ انو[1] استفاده شده است. از ویژگی این سیستم مدیریت وابستگی‌ها می‌توان به ایجاد محیط‌های منزوی در هر پروژه اشاره کرد که در نتیجه خطاهای مربوط به توسعه را کاهش می‌دهد.

۱-۳-۲-۲- معرفی اسکرپی[2]

یکی از چارچوب‌های معروف پایتون برای خزشگری در وب، چارچوب اسکرپی است. اسکرپی یک کتابخانه کامل از قابلیت‌های مورد نیاز برای خزش در وب، با قابلیت دنبال کردن لینک‌های موجود در هر صفحه و استخراج تمامی لینک‌های یافته شده است. اسکرپی همچنین در روند خزش صفحات از مکانیزم

[1] Pipenv

[2] Scrapy

پردازش موازی و همزمان صفحات و استفاده از خط لوله بهره می‌برد که باعث افزایش سرعت خزشگری آن می شود. از مزیت‌های دیگر این چارچوب می‌توان به وجود آموزش‌ها و مستندات فراوان برای آن در بستر اینترنت اشاره کرد.

شکل ۱- لوگوی اسکرپی

۲-۳-۲-۲- معرفی سلنیوم

با وجود قدرتمند بودن اسکرپی در امر خزش صفحات اینترنتی ایستا، اما این چارچوب خزشگری نمیتواند بخش‌های اسکریپتی صفحات پویا را خزش کند. برای نیل به این مهم، از ابزار سلنیوم جهت کمک به اسکرپی یاری می‌گیریم. سلنیوم در واقع ابزاری است که به طور خاص منظوره به جهت مرورگری خودکار صفحات وب به کار می‌رود. در واقع سلنیوم فرآیند یک مرورگر هنگام بازدید از یک صفحه اینترنتی را شبیه‌سازی می‌کند. کاربرد اصلی این ابزار در تست برنامه‌های تحت وب است اما ما در اینجا از آن برای پردازش اسکریپت‌های جاوا اسکریپت صفحات داینامیک استفاده می‌کنیم. سلنیوم برای زبان‌های مختلفی نظیر جاوا و روبی و پایتون رابط کاربری ارائه داده است که ما در اینجا طبعا از رابط کاربری مخصوص زبان پایتون آن استفاده می‌کنیم. این ابزار جهت پردازش نیاز به درایور یک مرورگر نظیر اینترنت اکسپلورر، فایرفاکس، اپرا یا کروم را دارد که ما در این پروژه از درایور مخصوص فایرفاکس برای سلنیوم استفاده کرده‌ایم.

شکل ۲- لوگوی سلنیوم

۳-۳-۲-۲- پایگاه‌داده

همان‌طور که در قبل گفته شد هر خزشگر به صورت جداگانه برای خود یک پایگاه داده دارد و پس از خزش هر صفحه متعلق به یک پرسش خزش‌نشده، سندی را که شامل فیلدهایی از قبیل لینک پرسش، عنوان پرسش، متن پرسش و یک فیلد بولی با نام checked (با مقدار پیش فرض False) است، به پایگاه داده خود اضافه می‌کند. ما برای نگهداری پایگاه داده این اسناد از پایگاه داده مانگو استفاده می‌کنیم. مانگو یک پایگاه‌داده رایگان و متن‌باز است که از نوع پایگاه‌داده‌های nosql محسوب می‌شود. از ویژگی‌های مانگودی‌بی می‌توان به عدم سخت‌گیری آن بر روی ساختار و مدل ا سناد ذخیره شده ا شاره کرد. در زیر می‌توانید یک نمونه از اسناد ذخیره شده توسط یکی از خزشگرها در پایگاه داده متعلق به اسناد خزش شده‌اش را مشاهده کنید.

```
{
    "_id" : ObjectId("5b720d3c75c60c90083b5f4c"),
    "title" : "خارج شدن بار ازاد از زمین"
    "body" : "چرا بار ازاد (لیبرو) با رسیدن به منطقه ۴ از زمین خارج میشود ؟"
    "url" :
"http://porsak.ir/?qa=12467/%D8%AE%D8%A7%D8%B1%D8%AC-%D8%B4%D8%AF%D9%86-%DB%8C%D8%A7%D8%B1-%D8%A7%D8%B2%D8%A7%D8%AF-%D8%A7%D8%B2-%D8%B2%D9%85%DB%8C%D9%86",
    "checked" : false
}
```

شکل ۳- نمونه یک سند ذخیره شده در پایگاه‌داده یک خزشگر پروژه

۴-۲-۲- نحوه خزش سایت‌های مختلف

همانطور که در بخش‌ها چالش‌ها گفته شد، به منظور نیل به اهداف خود مجبور هستیم تا برای هر سایت پرسش و پاسخ هدف خزشگری متفاوت طراحی کنیم. طراحی خزشگر برای هر سایت نیز خود مستلزم بررسی ساختار آن سایت و چالش‌ها و فرصت‌های ناشی از آن ساختار است. در ذیل به مختصر ساختار هر یک از سایت‌های هدف مورد بررسی قرار داده‌ایم و راهکار خود برای نیل به اهداف اولیه خزشگری را نیز شرح داده‌ایم.

۱-۴-۲-۲- خزش سایت ninisite

همانطور که در بخش مقدمه گفته شد این سایت چهاردهمین سایت پربازدید در میان تمامی سایت‌های ایرانی است. از این رو به نوعی مهم‌ترین سایت هدف ما در این پروژه است. این سایت اغلب در موضوعات مربوط به بانوان نظیر خانه‌داری، بچه‌داری و مسائل پزشکی بانوان به فعالیت می‌پردازد و از این رو اکثر مخاطبان این سایت را بانوان تشکیل داده‌اند. این سایت در کنار مطالبی که خود نشرشان می‌دهد دارای یک تالار گفتگو نیز هست که در آن کاربران سایت به یکدیگر با گفتگو پرداخته و پرسش‌هایشان را مطرح کرده و به پرسش‌های یکدیگر پاسخ می‌دهند. نکته جالب توجه در مورد این سایت این است که با بررسی انجام گرفته به این نتیجه دست پیدا کردیم که به صورت متوسط در هر ساعت در این بیش از صد پرسش جدید مطرح می‌شوند. این نکته خود در امر تنظیم بازه‌های زمانی جهت خزش این سایت بسیار مهم است. یکی از چالش‌های اساسی که در خزش این سایت با آن مواجه هستیم این است که پرسش‌ها در این سایت نه به ترتیب تاریخ پرسیده شدن که به ترتیب پاسخ‌های داده شده به آن‌ها به کاربر نمایش داده می‌شوند. البته این سایت دارای صفحه‌ای است که در آن لیست تاپیک‌های مطرح شده در ۴۸ ساعت اخیر به طور ترتیب مطرح شدن آن تاپیک به نمایش درآمده‌اند. از قابلیت مرتب‌سازی زمانی همین صفحه استفاده کرده و به خزشگر دستور خزش بر روی این صفحه و لینک‌های آن را می دهیم. خزشگر نیز لینک‌های این صفحه را خزش می‌کند و به ازای هر لینکی مشاهده کرد که لینک دیده شده قبلا خزش شده است ادامه خزش را متوقف می‌سازد.

۲-۴-۲-۲- خزش سایت applyabroad

این سایت نیز مانند ninisite یک تالار گفتگو است که در آن کاربران با یکدیگر درباره مباحثی همچون تحصیلات در خارج از کشور به بحث می پردازند. این سایت نیز مانند ninisite در صفحه اصلی تالارگفتگوی خود قابلیتی برای نمایش دادن تاپیک‌ها بر اساس تاریخ ایجاد شدنشان ندارد و از طرفی بر خلاف ninisite هم صفحه ای در آن جهت دیدن تاپیک‌های جدید ایجاد شده در طی دو روز گذشته نیز مشاهده نمی‌شود. اما پس از بررسی‌های انجام شده روی این سایت مشخص گردید که این سایت صفحه‌ای در خود دارد که در آن به صورت آرشیو شده تاپیک‌های کاربران را نگه می‌دارد. حالت آرشیو شده حالتی است که در آن عکس‌ها و استایل‌های css به نمایش درآورده نمی شوند و محتوای سایت صرفا به صورت متن و استایل‌های محدود نمایش داده می شود. در صحفه آرشیو با تعدادی لینک برای هر دسته از تاپیک مواجه هستیم که در لینک هر دسته، موضوعات متعلق به آن دسته به صورت تاریخ ایجاد شدن به نمایش در می‌آیند. از این رو با خزش روی این صفحات می‌توانیم هر گاه به لینک خزش شده‌ای رسیدیم ادامه خزش روی صفحه آن دسته را متوقف سازیم و به سراغ دسته بعدی برویم. این امر در کاهش زمان خزش نقش به سزایی را ایفا می‌کند.

۲-۴-۲-۳- خزش سایت javabyab

این سایت یک سایت پرسش و پاسخ است که در آن کاربران می‌توانند در مورد هر موضوعی پرسش‌های خود را مطرح کنند و یا به پرسش سایر کاربران پاسخ دهند. خوشبختانه در صفحه اصلی این سایت پرسش‌ها به ترتیب تاریخ ایجاد شدن (از جدید به قدیم) به نمایش درآمده‌اند و کار خزش ما در این سایت آسان تر از خزش در دو سایت قبلی است. در این جا نیز به علت همین قابلیت می‌توانیم هر گاه لینک خزش شده‌ای را مشاهده کردیم از خزش روی لینک‌های بعدی صرف نظر کنیم.

۲-۴-۲-۴- خزش سایت porsak

این سایت نیز مانند javabyab یک سایت پرسش و پاسخ عام منظوره است و از لحاظ ساختار نمایش پرسش‌ها نیز به همان شیوه javabyab عمل کرده است، در نتیجه شیوه خزشگری ما در این جا تفاوتی با مورد قبل ندارد.

۵-۴-۲-۲- خزش بخش مشاوره سایت تبیان

تبیان یک سایت عام منظوره ا ست که در خود بخ شی به نام م شاوره دارد. در این بخش کاربران پرسش‌های خود را مطرح می‌کنند و پس از مدتی کارشناسان تبیان به این پرسش‌ها پاسخ می‌دهند و پرسش کاربران را همراه پاسخ در بخش مشاوره قرار می‌دهند. دلیل انتخاب بخش مشاوره تبیان برای خزش از سوی ما، پویا بودن صفحات این سایت است. به این صورت که محتوای این صفحات به صورت پویا و با جاوا سکریپت پر می شوند و ا سکرپی به تنهایی قادر نی ست تا این محتواهای پویا را خزش کند. در نهایت پس از بررسی‌های انجام شده با ابزار سلنیوم آشنا شدیم که با گرفتن درایور یک مرورگر (نظیر فایرفاکس، کروم، اپرا یا اینترنت اکسپولرر) عملگر آن مرورگر را به صورت خودکار شبیه‌سازی می‌کند. ما در این پروژه از سلنیوم برای پردازش کدهای جاوااسکریپت سایت تبیان استفاده نموده‌ایم. به این صورت که هر بار کدهای جاوا سکریپت صفحه پر سش‌های بخش م شاوره سایت تبیان را با سلنیوم به محتوای ایستا تبدیل می‌کنیم و سپس به کمک اسکرپی آن محتواها را خزش می‌کنیم.

۳-۲-انجام پیش‌پردازش و پالایش روی پرسش‌ها

۱-۳-۲- معماری کلی بخش پالایش

پس از انجام خزش روی سایت‌های هدف اکنون با پنج پایگاه داده متعلق به پنج سایت پرسش و پاسخ مواجه هستیم . که قالب هر سند در هر پایگاه‌داده تشکیل شده از فیلدهای عنوان پرسش، متن پرسش، لینک پرسش و یک فیلد بولی با مقدار False است. متن‌هایی که در فیلدهای عنوان پرسش و متن پرسش قرار دارند نیاز به پیش‌پردازش و انجام پالایش دارند. بخش پالایش در پروژه ما وظیفه این امور را برعهده دارد. به این صورت که به صورت دوره‌ای و طی بازه‌های زمانی فعال می‌شود و از پنج پایگاه‌داده اسنادی را که فیلد بولی checked آن‌ها Fasle است را مکش می‌کند. سپس عملیات پالایش را روی متن و عنوان پرسش انجام می‌دهد. پس از این مقدار بولی checked هر سند پالایش شده را در سمت پایگاه‌داده خزشگر آن، به True تغییر می‌دهد تا در بازدیدهای بعدی دوباره آن را پالاش نکند. بعد از این سند جدیدی شامل عنوان پالایش‌شده، متن پالایش شده، لینک پرسش و یک فیلد بولی به نام checked (که محتوای آن به صورت پیش فرض False) است را به پایگاه‌داده‌ای به نام پالایش‌شده اضافه می‌کند. پایگاه‌داده پالایش شده

اکنون حاوی تمامی پرسش‌های پالایش شده است که در بخش‌های بعدی و در هنگام پیاده‌سازی الگوریتم جستجو مورد استفاده قرار می‌گیرد.

۲-۳-۲- عادی سازی متن

در این بخش هدف ما تمیز کردن متن از بی‌نظمی‌ها و ناهنجاری‌های رخداده در آن است. هدف کلی این مرحله شامل موارد زیر است:

- پاک کردن متن از ناهنجاری استفاده از رسم‌الخط مختلف برای بعضی از حروف مانند «ک» یا «ی»
- پاک کردن علائم نگارشی مانند علامت تعجب یا علامت سوال و ... از متن
- اصلاح نیم‌فاصله‌ها و فاصله‌ها در متن
- تبدیل ارقام انگلیسی و عربی به معادل فارسی
- اصلاح کردن ناهنجاری های رخداده برای کاراکترها نظیر تبدیل «زیـــــــــبا» به «زیبا»
- اصلاح کردن پرانتزها در متن
- اصلاح فاصله میان پیشوندها و پسوندها برای مثال تبدیل «می روم» به «می‌روم»
- حذف کردن کاراکترهای زائد از متن برای مثال تبدیل «سُوال» به «سوال»
- پاک‌کردن و جایگزین کردن / تنها با فاصله خالی، برای مثال تبدیل «زن/مرد» به «زن مرد»

با بررسی‌های انجام شده به این نتیجه رسیدیم که ابزار ارائه شده «هضم» برای تحقق اهداف این مرحله ما مناسب می‌باشد.[2] اما از آنجایی که تمامی اهداف ما را در حوزه عملیاتی خود پوشش نمی‌دهد در کنار هضم خود اقدام به اضافه کردن توابعی جهت اهداف پوشش نداده شده کردیم. کدهای نوشته‌شده توسط ما در این مرحله، بر پایه استفاده از عبارات باقاعده بناشده اند. جهت استفاده از عبارات باقاعده نیز از کتابخانه استاندارد RE پایتون استفاده‌کرده‌ایم. برای مثال عبارت باقاعده (/\!?>)/\(/\!?) باعث پیداشدن / های تنها در متن و در نتیجه پاک کردن آنها می‌شود.

۳-۳-۲- استفاده از ریشه‌یاب

پس از عادی‌سازی متن، متن را به کلمات جدا از هم تقسیم می‌کنیم. و سپس بر روی هر کلمه عمل stemming را انجام می‌دهیم. در ابتدا توضیحی مختصر از عمل stemming را ارائه می‌دهیم. در واقع stemming عملیاتی است که در آن ریشه کلمه از طریق حذف پیشوندها و پسوندها آن به دست می‌آیند. برای مثال سه واژه «دانشجویی»، «دانشجوها» و «دانشجویان» پس از اعمال stemming باید به واژه «دانشجو» تبدیل شوند. به علت وقت‌گیر بودن و احتمال به صرفه‌نبودن stemmer پیاده‌سازی شده از لحاظ کارایی زمانی از پیاده سازی شخصی آن پرهیز کرده و پس از بررسی کتابخانه PersianStemmer را برای این پروژه برگزیدیم.[3]

۴-۳-۲- حذف ایست واژه‌ها

ایست واژه ها لغاتی هستند که علی رغم تکرار فراوان در متن، از لحاظ معنایی دارای اهمیت کمی می‌باشند مانند «بسیار»، «لطفا»، «اما»، «اگر» و «ولی». به دلیل این که کلمات از بار معنایی کمی برخوردار هستند عموما در فعالیت‌های مربوط به حوزه پردازش زبان طبیعی که با تعداد فراوانی متن و واژه روبه‌رو هستیم، در فاز پیش پردازش حذف می‌شوند. حذف این کلمات از متن سبب کاهش بار محاسبات و افزایش سرعت خواهد شد. برای حذف این کلمات عموما لیستی از این کلمات از پیش تهیه می‌شود و سپس در صورت رخداد این کلمات در متن، حذف می‌شوند. ما در این پروژه نیز با بررسی چند لیست ارائه شده برای ایست واژگان فارسی، پس از بررسی و تلفیق آن‌ها، یک فهرست ایست واژه تهیه و تدوین نمودیم.

فصل سوم

مدل‌های بازیابی اطلاعات

مدل‌های بازیابی اطلاعات

۳-۱- مقدمه‌ای بر علم بازیابی اطلاعات

بازیابی اطلاعات دانشی است که با نمایش، جستجو و دستکاری متون دیجیتال و سایر منابع اطلاعاتی نظیر تصویر و صوت و ویدیو گره‌خورده است. به بیانی واضح‌تر می‌توان گفت که بازیابی اطلاعات فعالیتی جهت دستیابی به سیستم اطلاعاتی است که هدف آن استخراج اسناد مرتبط به یک سند درخواستی از میان انبوهی از مجموعه اسناد است. بازیابی اطلاعات از لحاظ علمی و رسمی نیز به شکل زیر تعریف می‌شود: بازیابی اطلاعات در واقع یافتن اطلاعات مورد نیاز از میان مجموعه‌ای از داده‌های بدون ساختار (معمولا متن) است که این اطلاعات استخراج شده یک نیاز اطلاعاتی را برطرف می سازند.[4] امروزه صدها میلیون انسان هر موقع که از یک موتورجستجوگر وب استفاده می‌کنند با بازیابی اطلاعات درگیر هستند. برای مثال جستجوگر گوگل معروف‌ترین و پراستفاده‌ترین سیستم بازیابی اطلاعاتی است که به کاربران برای یافتن و بازیابی اطلاعات دلخواه‌شان شامل اطلاعات متنی، تصویری و ویدیویی کمک می‌کند.

۳-۲- توضیحی بر TFIDF

در این جا قبل از شروع بحث درباره انواع مدل‌های بازیابی اطلاعات لازم است تا درباره مفهومی به نام tf-idf توضیح دهیم. Tf-idf در واقع یک مقدار عددی است که میزان اهمیت یک کلمه در یک سند از مجموعه اسناد را نشان می‌دهد. برای روشن‌تر شدن این موضوع در ادامه ابتدا tf و سپس idf را توضیح می‌دهیم. شایان ذکر است که در ادامه برای توضیح مطالب از علائمی استفاده می‌کنیم که توضیحشان در جدول ۱ آمده است.

جدول ۱- توضیح علائم اختصاری استفاده شده

علامت	توضیح
d	بیانگر یک سند است.
t	بیانگر یک واژه است.
D	بیانگر مجموعه اسناد است.
N	تعداد کل اسناد مجموعه
q	بیانگر متنی است که دنبال پیدا کردن متون مشابه آن هستیم.

۳-۲-۱- توضیحی بر TF

فرض کنید که ما یک مجموعه از متون انگلیسی در اختیار داریم و قصد ما بر این است تا آن ها را بر میزان مشابهت و ارتباط با متن درخواستیمان که "the red wall" می‌باشد رتبه‌بندی کنیم. یک راه ساده این است که تمامی متونی را که حاوی هیچ کدام از کلمات "the" و "red" و "wall" نمی‌باشند حذف کنیم. اما پرسشی که مطرح است این می‌باشد که چگونه متون باقی‌مانده را رتبه بندی کنیم؟ یکی از راهکاری‌های ارائه شده می‌تواند این با شد که تعداد بار ظاهر شدن هر یک از واژگان "the red wall" در بقیه اسناد را بشماریم و آنگاه اسناد را به ترتیب این مقدار عددی مرتب سازی کنیم. در واقع tf تعداد دفعاتی است که یک واژه در یک متن تکرار می‌شود.

۳-۲-۲- توضیحی بر IDF

این معیار در واقع میزان اهمیت یک کلمه در کل مجموعه ا سناد را ن شان می‌دهد. هر چه قدر که یک کلمه در مجموعه اسناد بیشتر ظاهر شده باشد مقدار IDF آن کمتر و هر چه قدر که کمتر ظاهر شده باشد مقدار IDF بیشتر خواهد بود. برای توضیح IDF روابط ریاضیاتی متعددی مطرح شده است که ما در این جا از رابطه لگاریتمی زیر استفاده کرده‌ایم[5]

$$IDF(t, D) = \log \frac{N}{1 + |\{d \in D: t \in d\}|}$$

۳-۲-۳- تفسیر TF-IDF

همانگونه که در بخش tf توضیح دادیم، یکی از معیارهای رتبه‌بندی اسناد می‌تواند tf واژگان سند مورد نظر ما در آن ها با شد. فرض کنید در مجموعه خود هفت سند داریم که در چهار سند آن‌ها یکی از واژگان "The" یا "red" یا "wall" ظاهر شده‌اند و tf واژگان در چهار سند مطابق با جدول زیر باشد.

واژه	سند اول	سند دوم	سند سوم	سند چهارم
The	۲	۵	۴	۴
Red	۲	۱	۳	۰
Wall	۲	۱	۰	۰

در صورتی که بخواهیم اسناد را تنها بر حسب tf واژگان آن‌ها رتبه‌بندی کنیم سندهای دوم و سوم در رتبه‌بالاتری نسبت به سند اول قرار می‌گیرند چرا که در سند دوم و سوم هفت بار مجموعا واژه‌های "The" و "Red" و "Wall" رخ داده‌اند حال آن‌که این عدد برای سند اول شش است. اما واقعیت امر این است که واژه The به علت تکرار زیاد در متن‌ها برای ما ارزش کمتری دارد تا واژه Wall. در ذیل مقدار idf هر از یک واژگان محاسبه شده است:

$$idf('the', D) = \log\frac{7}{5} = 0.14$$

$$idf('red', D) = \log\frac{7}{4} = 0.24$$

$$idf('wall', D) = \log\frac{7}{3} = 0.36$$

همانطور که مشـــاهده می‌کنید بر طبق معیار IDF ارزش اطلاعاتی wall برای ما بیشـــتر از the است، در نتیجه قاعدتا یکبار ظاهر شدن واژه wall نباید ارزشی یکسان با یکبار ظاهر شدن واژه the داشته باشد. معیار tf-idf این مشکل را برای ما حل می‌کند:

$$tfidf(q, d, D) = tf(q, d) * idf\ (q, D)$$

در واقع معیار tfidf به ما نشان می‌دهد که هر واژه با تکرارهایش در یک سند چه ارزش اطلاعاتی به آن سند می‌بخشد. اکنون مجموع tf-idf را برای مثال ذکر شده خودمان محاسبه می‌کنیم

tfidf(doc1) = 0.14*2 + 0.24*2 + 0.36*2 = 1.48

tfidf(doc2) = 0.14*5+0.24+0.36 = 1.3

tfidf(doc3) = 0.14*4+0.24*3 + 0 = 1.28

tfidf(doc4) = 0.14*4 = 0.56

همانطور که مشاهده می‌کنید معیار *tfidf* به خوبی توانست اهمیت بیشتر سند اول نسبت به سند دوم را تشخیص دهد. از معیار جلوتر در بعضی از مدل‌های بازیابی اطلاعاتی استفاده خواهیم نمود.

۳-۳-انواع مدل‌های بازیابی اطلاعات

سیستم‌های بازیابی اطلاعات جهت نیل به هدف خود (یعنی بازگرداندن اطلاعات مرتبط با اطلاعات درخواستی کاربر به او) از استراتژی‌های مختلفی استفاده می‌کنند. به کارگیری هر یک از این استراتژی‌ها نیازمند مدل‌سازی اسناد موجود به شکل خاص موردنیاز آن استراتژی است. این مدل‌ها بر اساس پایه ریاضیاتی خود به سه دسته مدل های مجموعه‌ای، مدل‌های جبری و مدل‌های احتمالاتی تقسیم می‌شوند. در ادامه شرح هر یک از مدل‌ها را آورده‌ایم.[5]

۱-۳-۳- مدل‌های مجموعه‌ای

در این دسته مدل‌ها، اسناد هر کدام به صورت مجموعه‌ای از کلمات نمایش داده می‌شوند. همچنین برای پیدا کردن ا سناد م شابه یک سند نیز از عملیات‌های مخ صوص نظریه مجموعه‌ها ا ستفاده می‌شود.

۲-۳-۳- مدل‌های جبری

این دسته‌مدل‌ها هر یک از اسناد را به صورت یک بردار یا ماتریس تصویر می‌کنند. به علاوه اعمال جبری روی این بردارها یا ماتریس‌ها اعمال می‌شوند که در نهایت حاصل این عملیات‌ها مقداری عددی

اســت که بیانگر میزان مشــابهت بین دو ســند اســت. در ادامه به توضــیح مدل فضــای برداری که یکی از مدل‌های جبری است می‌پردازیم.

۱-۲-۳-۳- مدل فضای برداری

در این مدل ا سناد و سند درخوا ست شده از سمت کاربر به بردار تبدیل می شوند. هر بعد از این بردارها اختصــاص به یک واژه دارد و مقدار آن بعد برای هر ســند نیز بر اســاس روابط مختلفی می‌تواند به دست آید. برای مثال هر بعد می‌تواند مقادیر صفر یا یک را داشته باشد که به حضور یا عدم حضور واژه در آن سند ا شاره کند یا این که مقدار هر بعد میزان تکرار آن واژه در سند یا مقدار tfidf واژه در سند با شد. سپس میزان مشابهت بین هر دو سند از طریق محاسبه میزان زاویه بین دو بردار متناظر با آن دو سند به دست می‌آید. میزان زاویه بین دو بردار را نیز می‌توان از طریق محاسبه کسینوس بین آن دو بردار به دست آورد یعنی:

$$\cos\theta = \frac{d.q}{||d||\,||q||}$$

که در این رابطه d ناظر به بردار یک سند از مجموعه سندها، q بردار سند مورد جستجوواقع شده و $||d||\,||q||$ ناظر بر مقدار نرم بردارهای q و d هستند که داریم:

$$||d|| = \sqrt{\sum_{i=1}^{n} d_i^2}$$

از مزایای این روش می‌توان به موارد زیر اشاره کرد:

- باینری نبودن میزان مشابهت (در مقایسه با روش های مجموعه‌ای)
- امکان تعیین حد آستانه بر روی میزان درجه کافی برای مشابه بودن دو سند (به این علت که میزان درجه بین دو بردار هر دو سند همیشه بین ۰ تا ۱۸۰ درجه است و در نتیجه میزان مشابهت نیازی به نرمال‌سازی ندارد)

از معایب این روش نیز می‌توان موارد زیر را برشمرد:

- این روش در مواجهه با متن‌های طولانی عملکرد ضعیفی از خود نشان می‌دهد.

- در صورتی که سند مورد مقایسه علاوه بر دارا بودن واژه‌های سند مورد جستجو واژه‌های بسیار دیگری را نیز داشته‌باشد در آن صورت ممکن است زاویه بین دو بردار این دو سند مقدار زیادی شده و دو سند به اشتباه با یکدیگر غیرمرتبط تشخیص داده شوند.[6]

۳-۳-۳- مدل‌های احتمالاتی

در مدل‌های احتمالاتی برای پیدا کردن میزان مشابهت بین دو سند از استنتاج‌های احتمالاتی استفاده می‌شود. در واقع نمونه‌های مشابه یک سند از طریق محاسبه احتمال مشابه بودن دو سند به دست می‌آیند. در ادامه ابتدا مدل Probabilistic relevance را توضیح داده و سپس BM25 را که یک تابع رتبه‌بندی بنا شده بر پایه این مدل است را بررسی خواهیم کرد.

۱-۳-۳-۳- مدل Probabilistic relevance:

این مدل در سال ۱۹۷۶ به عنوان چارچوبی برای مدل‌های احتمالاتی بازیابی اطلاعات توسط رابرتسون و جونز طراحی شد. کاربرد این مدل در پایه‌ریزی یک روش فرمال برای استفاده توابع رتبه‌بندی از آن است. به طوری که توابع رتبه‌بندی نظیر BM25 می‌توانند روی آن سوار شده و اسناد را از لحاظ میزان شباهت به یک سند درخواستی رتبه‌بندی کنند. فرض اساسی این مدل در این است که رخداد هر واژه در یک متن مستقل از رخداد باقی واژه‌ها در آن متن است. این مدل در نهایت از طریق روش‌های فرمال به رابطه زیر جهت محاسبه میزان شباهت بین دو سند دست پیدا می‌کند:

$$P(rel|d,q) = \sum_{q,tf_i>0} w_i$$

که در این رابطه برای پیدا کردن میزان مشابهت بین سند d و سندموردجستجوی q، ابتدا تک تک واژگان سند q جدا شده و به ازای هر کدامشان که tfشان در سند d بیشتر از صفر است (یعنی آن واژگانی از q که در سند d) رخ داده‌اند، وزن آن واژه بر طبق تابعی در سند d حساب شده و با گرفتن مجموع این اوزان مقدار مشابهت دو سند d و q به دست می‌آیند. [7]

۲-۳-۳-۳- تابع رتبه‌بندی bm25:

Bm25 (با نام تجاری Okapi bm25) یک تابع رتبه‌بندی استفاده شده توسط موتورهای جستجوگر برای رتبه‌بندی مجموعه‌ای از اسناد بر طبق شباهتشان با یک سند است. این تابع رتبه‌بندی بر پایه چارچوب مدل Probabilistic relevance بنا شده است. این تابع برای اولین بار در دهه ۸۰ میلادی دانشگاه لندن طراحی و پیاده‌سازی شد. این تابع و سایر گونه‌های جدیدترش[۱] در تخصیص وزن به هر واژه تاکید زیادی بر tfidf دارند.

در صورتی که سند Q را داشته باشیم که از واژگان $q_0, q_1, \dots, q_n$ تشکیل شده‌باشد، امتیازی که تابع رتبه‌بندی bm25 از جهت میزان شباهت سند d با سند Q اختصاص می‌دهد برابر است با

$$score(d,Q) = \sum_{i=1}^{n} IDF(q_i).\frac{f(q_i,d).(k_1+1)}{f(q_i,d)+k_1.(1-b+b.\frac{|d|}{avgdl})}$$

که در این رابطه $f(q_i,d)$ بیانگر تعداد رخداد q_i در سند d، $|d|$ بیانگر طول سند d بر حسب تعداد کلمه و $avgdl$ میانگین طول سندهای حاضر در مجموعه‌سندهای D بر حسب تعداد کلمه است. در این رابطه k_1 و b دو پارامتر آزاد هستند که معمولا از طریق روش‌های بهینه‌سازی روی داده‌های آموزش انتخاب می‌شوند. همچنین IDF حاضر در این رابطه نیز به شکل زیر محاسبه می‌شود:

$$IDF(q_i) = \log\frac{N - n(q_i) + 0.5}{n(q_i) + 0.5}$$

در توضیح رابطه فوق ارائه شده برای IDF نیز باید گفت که N نمایانگر تعداد کل اسناد در مجموعه اسناد و $n(q_i)$ نیز تعداد اسنادی است که شامل کلمه q_i هستند. تابع ارائه شده فوق برای محاسبه IDF یک مشکل اساسی دارد و آن این است که در صورتی که کلمه‌ای در بیش از نیمی از اسناد ظاهر شود در آن صورت IDF آن مقداری منفی خواهد بود و حضور این کلمه در متن یک سند باعث کاسته‌شدن از امتیاز آن متن خواهد بود. از این رو برای IDF حد پایین در نظر می‌گیرند و در صورتی که مقدار آن برای کلمه‌ای منفی باشد مقدارش را صفر در نظر می‌گیرند.[7]

[۱] در اصل bm25 یک خانواده از توابع رتبه‌بندی است.

فصل چهارم

پیاده‌سازی بخش جستجوگر

پیاده‌سازی بخش جستجوگر

در فصل دو چگونگی پیاده سازی بخش خز شگر و پالایش‌کننده را تو ضیح دادیم. سپس در فصل ســـه به توضـــیح انواع مدل‌های بازیابی اطلاعات و تابع رتبه‌بندی bm25 پرداختیم. همچنین بعد از اتمام پیاده‌ســـازی بخش پالایش‌کننده اکنون با یک پایگاه داده مواجه هســـتیم که عناصـــر آن هر کدام شـــامل فیلدهای عنوان پرســـش، متن پرســـش، لینک پرســـش و یک فیلد بولین با مقدار پیشـــفرض False به نام checked هســـتند. فعالیت بعدی ما چگونگی پیاده‌ســـازی بخش جســـتجوگر می‌باشـــد. به این منظور ابتدا پایگاه داده حاصل از بخش پالایش را به داده‌ساختار مناسب bm25 تبدیل می‌کنیم و سپس به پیاده‌سازی bm25 می‌پردازیم. در انتها نیز با پیاده‌سازی رابط کاربری تحت وب سامانه خود را عملیاتی می‌کنیم.

۱-۴- پیاده‌سازی جدول شاخص معکوس

همانطور که در فصل قبل گفتیم، bm25 در ابتدا یک سند Q شامل کلمات $q_1, q_2, q_3, \dots, q_n$ را دریافت می‌کند و سپس آن را به کلماتش می شکند و برای هر q_i در بین ا سناد موجود در مجموعه سندها وزن را با توجه به رابطه ارائه شـــده حســـاب می‌کند. بدیهی اســـت که این رابطه تنها بر روی اســـنادی اجرا می‌شود که حداقل حاوی یکی از q_i ها در خود باشند. پس در نتیجه در مرحله اول نیازمند این هستیم که این ا سناد را از میان کلیه ا سناد به د ست آوریم. در مرحله بعدی برای محا سبه امتیازی که هر واژه به هر سند می‌بخشد نیازمند اطلاعاتی از قبیل تعداد تکرار آن واژه در آن سند و همچنین idf آن واژه هستیم که خود نیازمند داشتن اطلاعاتی راجع به این است که آن واژه در چند سند تکرار شده است. پرواضح است که در صـــورتی که بخواهیم این اطلاعات را از طریق پرس و جوی روی پایگاه‌داده حاصـــل از مرحله پالایش به د ست آوریم زمان زیادی را در این بین از د ست می‌دهیم، چرا که مجبور به بررسی همه عنا صر موجود در آن پایگاه داده هستیم. برای مثال در نظر بگیرید که پایگاه‌داده ما حاوی ده هزار پرسش است و می‌خواهیم بفهمیم که واژه «منابع» در چه پرســـش‌هایی و هر کدام چند بار ظاهر شـــده اســـت. برای نیل به این هدف مجبوریم کلیه ده هزار پرســـش را بررســـی کنیم حال آن که ممکن اســـت واژه «منابع» تنها در ده پرســـش ظاهر شده با شد. در نتیجه به خاطر این م سئله حدود هزاربرابر زمان لازم را صرف ج ستجوی این واژه در کلیه متون کرده‌ایم.

برای غلبه بر این چالش ما یک پایگاه داده جدید به نام جدول شاخص معکوس را در کنار پایگاه داده حاصل از مرحله پالایش ایجاد می‌کنیم. کلیدهای این پایگاه‌داده باید کلمات باشند و محتوی هر عنصر این پایگاه داده نیز از آرایه ای زوج‌های مرتب است که اولین عنصر زوج مرتب کلید یک سند واقع در پایگاه‌داده پالایش‌شده‌ها و دومین عنصر زوج مرتب نیز نمایانگر تعداد بار ظاهر شدن کلمه واقع شده به عنوان کلید پایگاه‌داده جدول شاخص معکوس در سند ذکر شده است. برای مثال در شکل ۴ یکی از یکی از عناصر پایگاه‌داده شاخص معکوس آمده است. این عنصر نشان می‌دهد که واژه «منابع» در چه پرسش‌هایی و به چند بار در هر یک از آن پرسش‌ها ظاهر شده است.


```
{
    "_id" : ObjectId("5b720d9ae464bd1bd48dc7f3"),
    "word" : "منابع"
    "documents" : [
        {
            "id" : ObjectId("5b720d6fe464bd1bd48dc7eb"),
            "num" : 1
        },
        {
            "id" : ObjectId("5b729ed2e464bd172f5f8cf5"),
            "num" : 1
        },
        {
            "id" : ObjectId("5b72a057e464bd18eb24326e"),
            "num" : 1
        },
        {
            "id" : ObjectId("5b72a14de464bd1a20efe12c"),
            "num" : 1
        },
        {
            "id" : ObjectId("5b72afa4e464bd2d10199426"),
            "num" : 1
        },
        {
            "id" : ObjectId("5b72c997e464bd1c51d0b36b"),
            "num" : 5
        },
        {
            "id" : ObjectId("5b72cf07e464bd1c51d0b435"),
            "num" : 1
        },
        {
            "id" : ObjectId("5b72cfade464bd1c51d0b452"),
            "num" : 2
        }
    ]
}
```


شکل ۴- اطلاعات مربوط به واژه منابع در جدول شاخص معکوس

برای پیاده‌ســازی این جدول باید به این نکته توجه کنیم که از آنجانی که بخش خزشــگر و بخش پالایش به صورت مداوم و طی بازه‌های زمانی اقدام به فعالیت می‌کنند در نتیجه برنامه سازنده این جدول نیز بایســتی به صــورت متناوب اجرا شــود. کلیت فعالیت این برنامه به این شــکل اســت که هر بار که فعال می‌شود از جدول پالایش تمامی پرسش‌هایی را که فیلد بولین checked آن‌ها مقدار False است استخراج کرده و برای تک تک کلمات حا ضر در متن و عنوان آن پر سش‌ها اقدام به برر سی می‌کند. در صورتی که کلمه ای در جدول شاخص معکوس ح ضور ندا شته با شد این برنامه برای آن در جدول ردیف جدید ایجاد می‌کند. همچنین در صــورت وجود داشــتن آن کلمه در جدول عنصــری جدید در قســمت documents متعلق به آن کلمه در جدول شاخص معکوس ایجاد می شود و کلید پر سشی (کلید پر سش در پایگاه داده پالایش شده‌ها) که کلمه در آن ظاهر شده به همراه تعداد بار ظاهر شدن کلمه در آن پر سش در آن عنصر قرار می‌گیرند.

۲-۴- پیاده‌سازی bm25:

اکنون در این مرحله با سه عنصر اساسی روبرو هستیم:

۱. پایگاه داده حاوی اسناد پالایش شده
۲. پایگاه داده جدول شاخص معکوس
۳. سند مورد جستجوی کاربر

در این جا ابتدا تمامی پیش پردازش‌های زبانی را که در بخش پالایش بر روی محتواهای خزش شده خود انجام دادیم، بر روی متن دریافتی سند مورد جستجوی کاربر نیز انجام می‌دهیم تا واژگاه دو طرف (هم سمت متن مورد جستجو و هم سمت متن‌های پالایش شده) انسجام و هماهنگی داشته باشند. سپس با استفاده از جدول شاخص معکوس اسنادی را که حداقل شامل یکی از لغات موجود در متن سند مورد جستجوی کاربر هستند، پیدا می‌کنیم. پس از این با استفاده از جدول شاخص معکوس و جدول اسناد پالایش‌شده مقدار امتیاز هر سند بنابر مشابهتش با سند مورد جستجوی کاربر را به دست می‌آوریم و اسناد را بر حسب امتیاز کسب شده به صورت نزولی مرتب کرده و به عنوان خروجی تحویل می‌دهیم.

شایان ذکر ا ست که جهت پیاده سازی این ق سمت و ق سمت جدول شاخص معکوس از زبان پایتون و کتابخانه pymongo استفاده نمودیم.

۴-۳-پیاده‌سازی رابط کاربری تحت وب

پس از پایان پیاده سازی پروژه با بر رسی قسمت‌های مختلف آن تصمیم گرفتیم که برای قسمت جستجوی اسناد (bm25) و قسمت محاسبه idf واسط کاربری تدارک ببینیم تا در فعالیت‌های بعدی یا کاربرد سایر افراد مورد استفاده قرار گیرد. ما برای پیاده‌سازی رابط کاربری تصمیم گرفتیم تا از ریزچارچوب فلسک[1] استفاده کنیم. فلسک یک ریزچارچوب برای ایجاد نقاط پایانی http می‌باشد. از مزایای استفاده از این ریزچارچوب می‌توان به هسته‌ی بسیار ساده ولی قابل گسترش آن اشاره کرد. [8]

[1] flask

فصل پنجم

تست و ارزیابی

تست و ارزیابی

در این بخش که شامل سه زیربخش است ابتدا به چگونگی جمع‌آوری داده برای فرآیند تست و ارزیابی می‌پردازیم. سپس در بخش بعدی معیارهای ارزیابی یک تابع رتبه‌بندی را معرفی می‌کنیم و در آخر نتایج آزمایش تست داده‌های جمع آوری شده را روی سیستم خودمان نشان می‌دهیم.

۵-۱- داده‌ها

از آنجایی که هدف غایی ما در این پروژه رتبه‌بندی بهینه یک مجموعه سند بر اساس میزان مشابهت‌شان با یک سند است، در نتیجه به عنوان مجموعه داده ارزیابی به یک مجموعه اسناد و یک مجموعه سند مورد جستجو دیگر لازم داریم که برای هر کدام از این سندهای مورد جستجو بقیه اسناد رتبه‌بندی شده با شند. متا سفانه پس از برر سی‌های فراوان مجموعه داده‌ای را که در زبان فار سی برای ارزیابی سیستم‌های بازیابی اطلاعاتی به کار رود نیافتیم در نتیجه مجبور به تولید یک مجموعه داده ارزیابی شدیم. برای تولید این مجموعه، از پنج عامل انسانی کمک گرفتیم. به این صورت که هر فرد هشت سوال مشابه طراحی کرد. سپس از میان هر هشت سوال مشابه یک سوال را به تصادف انتخاب کردیم و از تعدادی عامل انسانی دیگر خوا ستار این شدیم که به هر سند مجموعه کل ا سناد به میزان شباهت با سند تصادفی انتخاب شده، یک عدد از صفر تا ده را نسبت دهد. اکنون به هدف اولیه خود یعنی داشتن یک مجموعه از سندهای مورد جستجو و یک مجموعه از اسناد که بنابر میزان شباهتشان با سندهای مورد جستجو امتیازدهی شده‌باشند رسیده‌ایم.

۵-۲-معیار ارزیابی

برای محاسبه میزان موثر بودن الگوریتم‌های رتبه‌بندی از معیارهای متفاوتی استفاده می‌شود. معیاری که ما در این پروژه مورد استفاده قرار داده‌ایم معیار NDCG می‌باشد. این معیار در واقع کیفیت عملکرد یک الگوریتم رتبه‌بندی را در قبال جستجوی یک سند در میان مجموعه از اسناد نشان می‌دهد. [9]

وقتی از یک الگوریتم رتبه‌بندی می‌خواهیم تا یک مجموعه ا سناد را بر ح سب میزان شباهت با سند درخوا ستی ما رتبه بندی کند، در واقع به دنبال این ه ستیم تا ا سنادی که میزان م شابهت شان با سنددرخوا ستی ما بی شتر ا ست در بالای رتبه‌بندی قرار بگیرند. این الگوریتم‌ها پس از اجرا یک لی ست مرتب به ما بر می‌گردانند. معیار DGC تا رتبه p برای یک لیست مرتب به این صورت تعریف می‌شود:

$$DCG_p = \sum_{i=1}^{p} \frac{rel_i}{\log_2(i+1)}$$

در رابطه بالا rel_i بیانگر میزان مشابهت سند قرار گرفته در رتبه i ام نسبت به سند مورد جستجو است. همچنین p نیز یک پارامتر است که میزان عملکرد الگوریتم رتبه‌بندی ما تا رتبه p ام برگردانده شده را نشان می‌دهد.

معیار DCG به تنهایی نمی‌تواند میزان عملکرد الگوریتم رتبه‌بندی ما را نشان دهد چرا که در قبال هر سند درخوا ستی ممکن ا ست تعداد و میزان م شابهت سندهای م شابه واقع در مجموعه ا سناد متفاوت باشد. از این جهت ما بایستی از این زاویه به مسئله بنگریم که الگوریتم رتبه‌بندی ما تا چه حد به رتبه‌بندی بهینه نزدیک شده است. در واقع ابتدا بایستی معیار DCG را برای بهترین رتبه‌بندی محاسبه کنیم. این معیار IDCG نام دارد و به شکل زیر محاسبه می‌شود:

$$IDCG_p = \sum_{i=1}^{|REL|} \frac{rel_i}{\log_2(i+1)}$$

در رابطه بالا REL بیانگر مجموعه مرتب شده ا سناد بر ا ساس میزان م شابهت شان با سند مورد جستجو است. لازم به ذکر است که وجود پارامتر p باعث می‌شود تا p عضو بالای مجموعه REL در آن باقی مانند و باقی اسناد کم اهمیت‌تر از آن حذف شوند.

اکنون که هم DCG و IDCG را تعریف کردیم می‌توانیم معیار NDCG را نیز تعریف کنیم که عبارت است از:

$$nDCG_p = \frac{DCG_p}{IDCG_p}$$

در واقع $nDCG$ نشان می‌دهد که الگوریتم رتبه‌بندی ما به چه میزان به رتبه‌بندی بهینه نزدیک شده است.

۳-۵-نتایج

در این بخش ابتدا داده‌هایی را که قبلا عوامل انسـانی تولید کرده بودیم را به عنوان مجموعه کل داده‌ها به سیـ ستم وارد کردیم و سپس نتایج جـ ستوی پنج سند تدارک شده را به عنوان ا سناد مورد جستجو در سیستم مورد ارزیابی قرار دادیم. جدول رتبه‌بندی این اسناد مطابق زیر است.[۱]

جدول ۲- نتایج حاصل از اجرای الگوریتم رتبه‌بندی bm25 در کنار بهترین نتایج ممکن

رتبه	سند اول		سند دوم		سند سوم		سند چهارم		سند پنجم	
	نتیجه الگوریتم	نتیجه بهینه	نتیجه الگوریتم	نتیجه بهینه	نتیجه الگوریتم	نتیجه بهینه	نتیجه الگوریتم	نتیجه بهینه	نتیجه الگوریتم	نتیجه بهینه
۱	۱۰	۱۰	۸	۸	۷	۸	۱۰	۱۰	۹	۹
۲	۷	۷	۲	۵	۸	۸	۲	۱۰	۸	۸
۳	۵	۵	۲	۴	۳	۷	۷	۹	۱	۸
۴	۰	۴	۵	۳	۲	۳	۱۰	۷	۷	۷
۵	۰	۴	۳	۳	۲	۲	۲	۳	۸	۷
۶	۰	۳	۳	۲		۲		۲	۶	۶
۷	۳	۲	۱	۲				۲		۱
۸	۲		۴	۱						
۹	۴									
۱۰	۴									

[۱] در این جدول برای برخی از اسناد شاهد این هستیم که تعداد عناصر نتیجه حاصل از الگوریتم کمتر از تعداد عناصر نتیجه بهینه است. این به معنی است که الگوریتم ما بعضا نتوانسته است سندی را که در حقیقت با سندمورد جستجو مشابه است به عنوان سند مشابه بپذیرد.

سپس در گام‌بعدی مقادیر DCG و NDCG و IDCG را برای هر سند مورد جستجو حساب میکنیم.

جدول ۳- نتایج DCG و NDCG و IDCG متعلق به آزمون انجام شده

	سند اول	سند دوم	سند سوم	سند چهارم	سند پنجم
DCG	20.92	16.25	15.20	19.85	22.82
IDCG	21.94	17.31	19.34	26.39	26.26
NDCG	0.95	0.93	0.78	0.75	0.86

اکنون میانگین NDCG پنج سند مورد جستجو شده را به دست می‌آوریم که برابر است با ۰,۸۵

فصل ششم

نتیجه‌گیری و پیشنهادات

نتیجه‌گیری و پیشنهادات

۶-۱- نتیجه‌گیری

این پروژه توانست به عنوان یکی از اولین تلاش‌ها در جستجوی پرسش‌های مشابه در بستر پرسش و پاسخ انجمنی فارسی ابزاری قابل استفاده ارائه کند که البته نیاز به بهبود چشمگیری نیز خواهد داشت. در این پروژه ابتدا جهت جمع آوری پرسش‌ها به صورت دوره‌ای از سایت‌های پرسش و پاسخ خزشگرهایی طراحی شدند. در فاز بعدی با انجام پیش‌پردازش روی محتواهای خزش شده آن ها را پالایش کردیم. سپس الگوریتم رتبه‌بندی bm25 را جهت جستجوی اسناد روی محتواهای پالایش شده پیاده سازی کردیم و در نهایت با ارائه یک رابط کاربری تحت وب آن را به صورت یک ابزار همگانی ارائه کردیم.

۶-۲- پیشنهادات

پیشنهادات و کارهای قابل انجام بر روی این پروژه در آینده در چند بخش مختلف قابل ارائه هستند:

- از آنجایی که محتوای ثبت شده توسط کاربران در بسیاری از موارد از گونه محاوره‌ای زبان هستند و نه نوشتاری، پیشنهاد می شود تا با ارائه الگوریتمی امکان تبدیل یک متن از گونه محاوره‌ای به گونه نوشتاری فراهم شود.
- بسیاری از لغات با وجود داشتن معنی نزدیک اما از لحاظ هم ریختی به یکدیگر شباهتی ندارند مانند «ماشین» و «اتومبیل» و در نتیجه این پروژه قادر به تشخیص شباهت میان محتواهای حاوی این دو واژه نیست. پیشنهاد می‌شود تا در کارهای بعدی در کنار استفاده از الگوریتم رتبه‌بندی bm25 از مدل فضای برداری و الگوریتم w2c نیز کمک گرفته شود تا مشابهت‌های معنایی نیز پوشش داده شوند.
- در الگوریتم bm25 با دو پارامتر آزاد مواجه هستیم که به اجبار و از روی نبود داده مجبور به تعیین آن‌ها به صورت پیش‌فرض ذکر شده در مقاله گشتیم. برای این که بتوان بهترین حالت را برای این

دو پارامتر تعیین کرد نیاز به داده‌های آموزش برچسب خورده داریم. به این صورت که با انجام روش‌هایی نظیر یادگیری ماشین، حالت بهینه این پارامترها را به دست بیاوریم. در صورتی که شرایط و امکانات تهیه و تدوین مجموعه داده برچسب خورده مناسب برای بازیابی اطلاعات فارسی فراهم باشد، انجام این مهم می‌تواند سهم به سزایی در بهبود الگوریتم‌های رتبه‌بندی داشته باشد.

منابع و مراجع

[1] [Online]. Available: https://www.internetworldstats.com/stats.htm.

[2] Sobhe, [Online]. Available: http://www.sobhe.ir/hazm/.

[3] [Online]. Available: https://github.com/htaghizadeh/PersianStemmer-Python.

[4] H. S. a. P. R. Christopher D. Manning, "Introduction to information retrieval," Cambridge University Press, 2008, pp. 1-3.

[5] 8 12 2018. [Online]. Available: https://en.wikipedia.org/wiki/Information_retrieval.

[6] [Online]. Available: https://en.wikipedia.org/wiki/Vector_space_model.

[7] H. Z. Stephen Robertson, "The probabilistic relevance framework: BM25 and beyond," *Foundations and Trends® in Information Retrieval*, vol. 3.4, pp. 333-389, 2009.

[8] [Online]. Available: http://flask.pocoo.org/docs/1.0/.

[9] [Online]. Available: https://en.wikipedia.org/wiki/Discounted_cumulative_gain.

[10] [Online]. Available: https://en.wikipedia.org/wiki/Vector_space_model.

Abstract

The community question answering systems are web-based platforms helping users to ask new questions and answer the existing questions asked by the other users. The rapid growth of users in these platforms in recent years has established a valuable source of information specially in persian web content. However, finding the existing questions similar to a new one asked by a user is a complicated task due to the huge amount of information in these platforms. To avoid creating duplicate questions and reducing the response time, a mechanism should be designed to address these issues. The primary goal of this project is the design and implementation of a responsive search engine that suggests top similar questions in order of their proximity to a given question. The result would be a system that can be used independantly as a Q&A search engine or as a subsystem of a community question answering, helping users in finding relevant qurstions and avoid creating redundant data.

Key Words: community question answering system, information retrieval, serach relevant questions

Amirkabir University of Technology
(Tehran Polytechnic)

Computer Engineering Department

BSc Thesis

Design and implementation a system for searching relevant question using crawling on Persian community question answering websites

Mohammadmahdi Samiei Paqaleh

Supervisor

Dr. Seyed Rasoul Mousavi

August 2018