عنوان زیرپروژه:

مقدمه‌ای بر ذخیره و بازیابی اطلاعات متون زبان فارسی

فهرست مطالب

عنوان **شماره صفحه**

عنوان **شماره صفحه**

1. مقدمه

توسعه سیستم‌های رایانه‌ای و گسترش استفاده از فناوری اطلاعات در زندگی روزمره باعث شده تا اطلاعات از درجه‌ی اهمیتی والا برخوردار شوند؛ چنانکه عصر حاضر را «عصر اطلاعات» نامیده‌اند. میزان اطلاعات تولید شده و میزان استفاده از اطلاعات، دو معیار اساسی برای توسعه کشورها به شمار می‌آیند.

هر چه حجم اطلاعات افزایش می‌یابد کنترل و مدیریت آن مشکل‌تر می‌شود. لذا تولید و وجود اطلاعات به تنهایی کافی نیست بلکه باید ابزارهایی برای استفاده از این اطلاعات فراهم شوند. در واقع کاربران باید بدانند که چگونه باید به نیاز اطلاعاتی خود در این حجم عظیم منابع اطلاعاتی پاسخ دهند. درنتیجه روش‌های بازیابی اطلاعات در قالب پاسخ‌دهی به نیاز اطلاعاتی کاربران اهمیت ویژه‌ای پیدا می‌کند.

با وجود آنکه اطلاعاتی که امروزه عرضه می‌شوند صورت‌های مختلفی مثل صوت، انیمیشن، تصویر و غیره به خود گرفته‌اند، می‌توان گفت هنوز هم حجیم‌ترین و پراستفاده‌ترین اطلاعات، متون غیرساخت‌یافته هستند. به عبارت دیگر بازیابی اطلاعات عمدتا مرتبط است با بازیابی مستندات و مدارک متنی. کار معمول در بازیابی اطلاعات این است که بسته به نیاز مطرح شده از سوی کاربر، مرتبط‌ترین متون و مستندات را از میان انبوه مستندات بیرون بکشد.

هدف از ارایه این تحقیق این است تا مشکلات و موانع موجود بر سر طراحی و پیاده‌سازی یک موتور جستجوی فارسی شناسایی شوند و راه‌حل‌هایی برای آنان ارایه شوند. البته طراحی بخشهای عمده‌ای از موتور جستجو برای تمام زبان‌ها یکسان می‌باشد - که تا حد لزوم به بیان آن خواهیم پرداخت؛ اما تمرکز ما بر روی مواردی است که تحت تاثیر ساختار زبان هستند و باید با توجه به ساختار زبان و خط فارسی بومی‌سازی یا سفارشی شوند.

در این گزارش در بخش 1-1 برخی تعاریف در حوزه ذخیره و بازیابی اطلاعات را ارایه می‌دهیم. در ادامه در بخش 2 به طور کلی به بیان ساختار و معماری یک موتور جستجو می‌پردازیم و اجزای آنرا مشخص می‌کنیم. در بخش 3 به مسایل مربوط به زبان فارسی که در حوزه ذخیره و بازیابی اطلاعات مطرح است، می‌پردازیم. در بخش 4 سعی خواهیم کرد برای بومی‌سازی یا سفارسی کردن بخش‌هایی از موتور جستجو که متاثر از ساختار زبان است راه حل‌هایی را یافته و پیشنهاد دهیم. در بخش 5 به مبحث ریشه‌یابی در زبان فارسی خواهیم پرداخت؛ به دلیل اهمیت مبحث ریشه‌یابی یک بخش جداگانه برای آن

در نظر گرفته‌ایم. در نهایت در بخش 6 خلاصه‌ای از کارهایی که باید در پیاده‌سازی موتور جستجوی فارسی انجام شوند را برمی‌شماریم.

1-1 تعاریف

وقتی صحبت از بازیابی اطلاعات (Information Retrieval) می‌شود، آنچه در نظر می‌آید، استخراج اطلاعات از میان انبوه مستندات متنی است، اما حوزه بازیابی اطلاعات وسیع‌تر است و شامل تصویر و صوت نیز می‌گردد. همچنین است که تعاریف مختلفی برای بازیابی اطلاعات ارایه شده است. طبق [1] بازیابی اطلاعات به صورت زیر تعریف می‌شود:

> *اعمال، شیوه‌ها و رویه‌هایی برای بازیابی اطلاعات ذخیره شده در جهت تهیه اطلاعات حول موضوعی معین.*

همچنین تعریف بازیابی اطلاعات در [2] به صورت زیر آمده است:

> *بازیابی اطلاعات عبارت است از یافتن چیزهایی (معمولا مستندات) با ماهیت غیر ساخت‌یافته (معمولا متن) که پاسخگوی نیاز اطلاعاتی، از میان مجموعه‌ای عظیم باشد (معمولا بر روی سرویس‌دهنده‌های رایانه‌ای یا بر روی اینترنت).*

اطلاعات، یک زیرمجموعه از اسناد هستند که در ظاهر مرتبط با پرس و جو می‌باشد. تمام روش‌های جست و جو مبتنی بر مقایسه بین پرسش (query) و سند ذخیره شده می‌باشند. بطور معمول این مقایسه به صورت غیر مستقیم و با مقایسه پرس و جو با کلمات کلیدی انجام می‌گیرد.

باید توجه داشت که بین «بازیابی اطلاعات» و «بازیابی داده» تفاوت های زیادی وجود دارد. داده‌ها ابهام ندارند اما اطلاعات نیاز به تفسیر دارد و در نتیجه مبهم می‌شوند. سیستم بازیابی داده نیاز به رفع این ابهام ها را ندارد اما در سیستم بازیابی اطلاعات باید هر چه بهتر اطلاعات را مدل کنیم تا ابهام ها در درک اطلاعات توسط سیستم کمتر شوند. برای همین است که بر خلاف سیستم های بازیابی داده که کارایی سیستم از نظر سرعت و فضا به عنوان معیار ارزیابی در نظر گرفته می‌شود، در سیستم های

بازیابی اطلاعات، معیار دقت (precision) و بازخوانی (recall) و شبیه به آن، به عنوان معیار ارزیابی سیستم به کار می‌روند.

در این تحقیق ما به بازیابی مستندات می‌پردازیم؛ لذا از دید ما، بازیابی اطلاعات دانشی است که در آن، نحوه‌ی جستجو به دنبال یک سند (یا یک رکورد) در میان انبوهی از اسناد (یا رکوردها)، و بازیابی اطلاعات آنها، مورد مطالعه و بررسی قرار می‌گیرد.

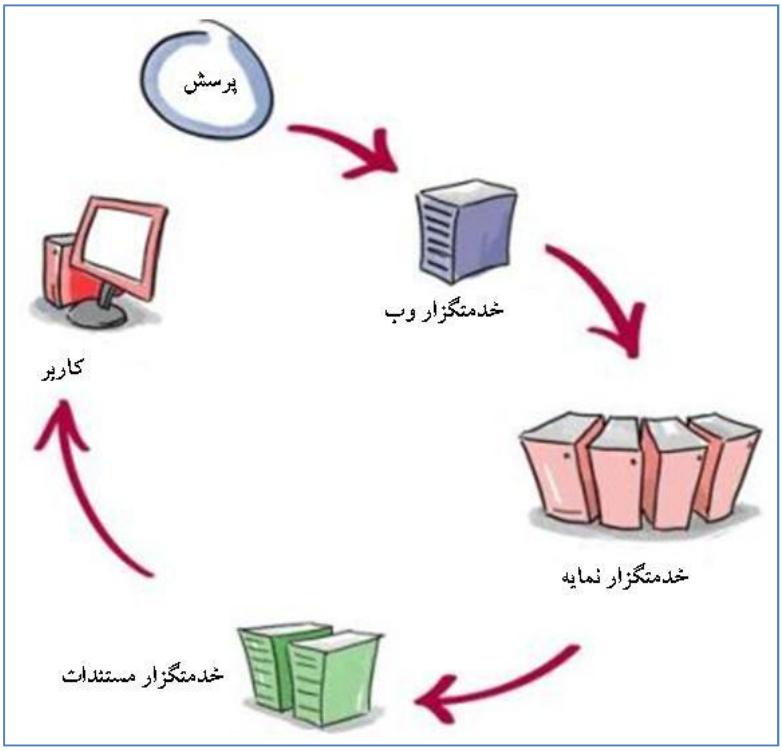
موتور جستجو

موتور جستجو نرم‌افزاری است که با گرفتن پرسشی از کاربر، مستنداتی را که می‌تواند پاسخی به نیاز اطلاعاتی وی باشد، به کاربر نمایش می‌دهد. منظور ما از پرسش (query)، کلمات کلیدی‌ای هستند که کاربر توسط آنها نیاز اطلاعاتی خود را بیان می‌کند. جمع‌آوری و نگهداری مستندات قابل دستیابی در یک ساختار بهینه، تشخیص مشابه‌ترین سند به پرسش کاربر و نمایش مستندات به ترتیب میزان مرتبط بودن، از نکات اصلی در معماری یک موتور جستجو است.

اصطلاح موتور جستجو معمولا به ابزارهایی اطلاق می‌شود که برای استخراج اطلاعات از وب طراحی شده‌اند. اما در اینجا این اصطلاح را به صورت عام به کار می‌بریم؛ یعنی فارغ از اینکه مستندات ما صفحات وب باشند یا فایل‌های متنی درون انباره یا رکوردهای پایگاه داده، تفاوتی ندارد.

2. آشنایی با معماری موتور جستجو

برای پاسخ به نیاز اطلاعاتیِ کاربر، موتور جستجو پرسش را از کاربر دریافت می‌کند. این پرسش شامل کلمات کلیدی - که کاربر به دنبال آنها است - و عملگرهای قابل استفاده مانند عملگرهای منطقی در پرسش است. همچنین ممکن است کلمات رزرو شده‌ای که توسط موتور جستجو به منظور بیان دقیق‌تر نیاز اطلاعاتی کاربر تعریف شده‌اند به پرسش ضمیمه شوند. چرخه‌ی ارائه‌ی پرسش تا دریافت مستندات مرتبط با آن توسط کاربر در شکل 1 آمده است.



شکل 1 - چرخه‌ی دریافت پرسش و ارائه‌ی نتایج به کاربر

ابتدا کاربر نیاز اطلاعاتی خود را بصورت پرسش مطرح می‌کند. ساختار این پرسش، ساختار استاندارد و تعریف شده‌ای برای همه‌ی موتورهای جستجو نیست و با توجه به پیاده‌سازی‌های مختلف ممکن است متفاوت باشد. اما استفاده از کلمات کلیدی متداول‌ترین نوع پرسش است.

بطور خلاصه وقتی موتور جستجو، پرسش را از کاربر دریافت می‌کند، یک پردازش اولیه بر روی آن انجام می‌دهد، این پردازش معمولا شامل مراحل زیر است:

- بررسی عملگرهای موجود در پرسش
- حذف کلمات عمومی (Stopword)
- ریشه‌یابی کلمات
- استخراج کلمات کلیدی

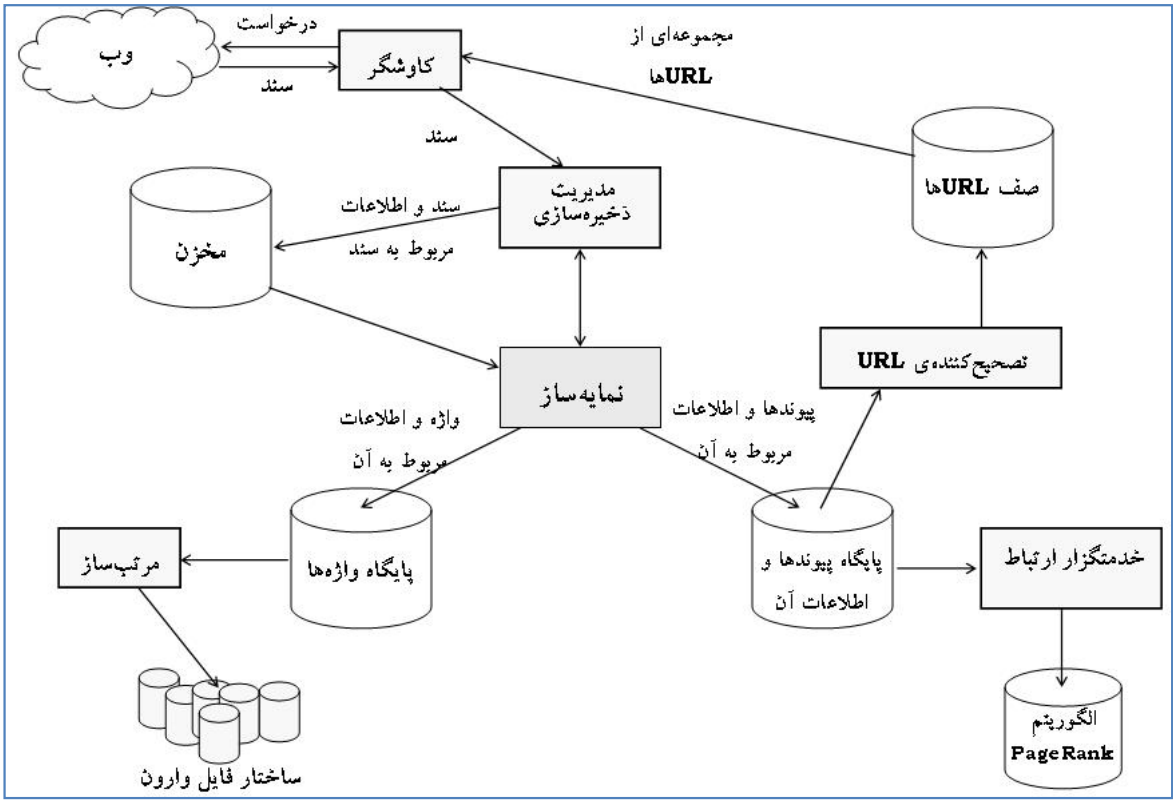
سپس پرسش به خدمتگزار نمایه (Index Server) به منظور مقایسه با مجموعه کلمات موجود در این خدمتگزار ارسال می‌شود. بعد از انجام عمل مقایسه در این خدمتگزار، مستندات مرتبط با پرسش کاربر مشخص می‌شوند. مشخصه‌ی این اسناد که شامل چکیده‌ای از متن، عنوان سند، آدرس سند و سایر ویژگی‌ها است، به خدمتگزار اسناد (Document Server) ارسال می‌شود تا به کاربر نمایش داده شود.

در شکل 2 معماری موتور جستجو و نیز چگونگی همکاری قسمت‌های مختلف آن برای پاسخ به پرسش کاربر نمایش داده شده است [3]. در این شکل ساختارهای داده‌ای مورد استفاده در موتور جستجو نیز مشخص شده‌اند.

موتور جستجو در اساس از سه بخش تشکیل شده است.

- بخش اول «گردآورنده اسناد» است که وظیفه‌ی گردآوری مجموعه اسناد موجود را به عهده دارد (در مورد وب به آن web crawler می‌گویند).
- بخش دوم «نمایه‌ساز» (Indexer) موتور جستجو است که مجموعه اسناد کاوش شده توسط کاوشگر را به نمایه‌های (Index) قابل استفاده تبدیل می‌نماید.
- بخش سوم شامل هسته‌ی اصلی موتور جستجو، «مدل‌های بازیابی اطلاعات و الگوریتم رتبه‌بندی» است. منظور از رتبه‌بندی، اولویت در نمایش مستندات و صفحات بازیابی شده است.

همانطور که از شکل 2 پیداست، موتور جستجو در ابتدا باید منابع اطلاعاتی و مستندات وب را از طریق نرم‌افزاری به نام کاوشگر وب جمع‌آوری کند. کاوشگر در مدل کلی، صفحات مربوط به سایت‌ها را درخواست می‌کند و در صورت مجاز بودن، صفحات را برای نمایه‌سازی به واحد مدیریت ذخیره‌سازی (Storage Manager) می‌دهد تا در مخزن (Repository) قرار گیرند.



شکل 2 - معماری موتور جستجو به همراه مولفه‌های اصلی آن

در این مرحله به هر سند یک DocID یکتا داده می‌شود. سپس همراه یکسری اطلاعات اضافی دیگر به صورت فشرده به مخزن فرستاده می‌شود. در واقع مخزن موتور جستجو را می‌توان آرشیو قسمتی از وب دانست که موتور جستجو آنها را واکشی نموده است. ساختار ذخیره اطلاعات در مخزن به صورت زیر است.

[DocID] [Download Date] [DocURL] [Last Update] [DocLen] [FullDoc]

با توجه به اصول مطرح شده در بازیابی اطلاعات [2, 4] برای بالا بردن کارایی جستجو و بازیابی مستندات، ابتدا باید مستندات موجود را به بردارهایی بر اساس واژه‌ها (term) و وزن آنها تبدیل کرد. این فرایند نمایه‌سازی نامیده می‌شود. نمایه‌ساز موجود در هسته‌ی موتورهای جستجو نیز به این منظور پیش‌بینی شده است. برای هر سندِ وب دو دسته اطلاعات اهمیت دارند.

- یکی واژه‌های بکار رفته در متن برای پیدا کردن مشابه‌ترین سند به پرسش کاربر و
- دیگری پیوندهایی که از این صفحه به سایر صفحات دیگر وب وجود دارد.

معمولا از ارتباط بین صفحات برای رتبه‌بندی سند استفاده می‌شود. به همین دلیل نمایه‌ساز خودکار، اطلاعات هر صفحه را حداقل در دو پایگاه، یکی پایگاه واژه‌ها به همراه مشخصات آنها و دیگری پایگاه پیوندها به همراه اطلاعات آنها نگهداری می‌کند. لازم به توضیح است که اصطلاح پایگاه داده‌ها در اینجا در معنای عام و کلی آن به کار رفته است.

سیستم‌های بازیابی اطلاعات برای نگهداری اطلاعات موجود در مستندات از ساختاری به نام *ساختار فایل وارون* (Inverted File Structure) استفاده می‌کنند [5]. در واقع ساختار داده‌ایی استفاده شده در پایگاه واژه‌ها شامل فیلدهای زیر است:

[DocID] [TermID] [Term Position] [Font Size] [Term type] [TF]

پایگاه واژه‌ها به منظور استفاده‌ی نهایی نیاز به مرتب‌سازی بر اساس واژه‌ها و نه بر اساس مستندات دارد. مرتب‌ساز (Soretr) این عمل را انجام می‌دهد و به این ترتیب صفحات وب در پایگاه داده‌ای نهایی‌شده قرار می‌گیرند تا مهم‌ترین قسمت برای تشخیص مستندات مرتبط به پرسش کاربر باشند. مرتب‌ساز ساختار واژه‌ها را از قسمت پایگاه‌واژه‌ها بازیابی می‌کند و ساختاری را که قبلاً بر اساس DocID مرتب شده بود، بر اساس TermID مرتب می‌کند. در واقع مرتب‌ساز یک پردازشگر ماتریس است که یک ماتریس Term×Document می‌سازد. در ضمن وزن Term را بر اساس معیارهایی از قبیل تعداد تکرار واژه در صفحه (Term Frequency)، تعداد تکرار واژه در مجموعه‌ی پایگاه و اندازه صفحه محاسبه می‌کند. ساختاری که در این قسمت ایجاد می‌شود و در ساختار فایل معکوس قرار می‌گیرد شامل فیلدهای زیر است [4, 5]:

[TermID] [DocID] [Term Position] [Weight]

فیلد Weight وزن واژه در سند را مشخص می‌کند. روش‌های مختلفی برای وزن‌دهی به واژه‌های متن وجود دارد که برخی از مهم‌ترین آنها در [6] بررسی شده‌اند. روش $tf.idf$ یکی از متداول‌ترین این الگوریتم‌ها است [4, 6]. در این رابطه t_f تعداد تکرار واژه در متن و idf وارون تعداد تکرار واژه در مجموعه‌ی اسناد است. به این ترتیب تاثیر یک واژه در متن در مقایسه با سایر واژه‌های موجود در اسناد بدست می‌آید.

از طرف دیگر نمایه‌ساز پیوندهای صفحات را در پایگاه پیوندها ذخیره می‌کند. اگر صفحات وب بصورت گرافی در نظر گرفته شوند که صفحات گره‌های گراف و پیوند بین آنها یال‌های گراف باشد، در این صورت از روی پیوندهای ذخیره شده در پایگاه پیوندها می‌توان تعداد یال‌های ورودی و خروجی به هر گره (صفحه وب) را بدست آورد. هر چه تعداد بیشتری صفحه به یک صفحه‌ی مورد نظر اشاره کنند یا به

بیان دیگر هر چه تعداد یال‌های ورودی به یک صفحه بیشتر باشد، اهمیت آن صفحه بیشتر است. این معیار الگوی مناسبی برای رتبه‌بندی اسناد بازیابی شده است و با آن می‌توان هنگام نمایش نتایج به کاربر، مجموعه‌ی اسناد بازیابی شده را بر اساس بیشترین ارتباط با پرسش کاربر نمایش داد. این معیار اولین بار در موتور جستجوی google به عنوان الگوریتم PageRank پیاده‌سازی شد [3].

وظیفه‌ی *خدمتگزار ارتباط* (Connectivity Server) بدست آوردن تعداد یال‌های ورودی و خروجی برای هر صفحه و ذخیره‌سازی نتایج حاصل از این الگوریتم در پایگاه داده‌ی PageRank است. ساختار داده‌ایی مورد استفاده در پایگاه پیوندها و اطلاعات آنها شامل فیلدهای زیر است:

[LinkID] [SourceDocID] [TargetDocID] [AnchoreText]

وقتی کاربر پرسشی را وارد سیستم می‌کند، موتور جستجو واژه‌های موجود در پرسش کاربر را با واژه‌های خود در ساختار فایل وارون مقایسه و مجموعه اسناد مرتبط با پرسش کاربر را استخراج می‌کند. اما قبل از نمایش، اسناد باید رتبه‌بندی شوند تا بر اساس اولویت به کاربر نمایش داده شوند. در این قسمت برای تعیین اولویتِ نمایش، موتور جستجو از پایگاه PageRank استفاده می‌نماید. بعد از این مرحله خلاصه‌ای از مستندات مرتبط با کاربر در یک رتبه‌بندی منطقی آماده‌ی نمایش به کاربر است.

در ادامه این تحقیق، جزییات و معماری درونی هر کدام از بخش‌ها را بررسی می‌کنیم.

2-1 گردآورنده اسناد

امروزه اسناد کامپیوتری با نرم‌افزارهای گوناگون نوشته می‌شوند. قبل از هر کاری باید متون درون این اسناد و قالب این پرونده‌ها خوانده شود. قالب این پرونده‌ها اغلب هم‌خوانی کمی با هم دارند. نرم‌افزارهای گوناگونی همچون pe2 ، زرنگار، کلک، نشر الف، word، pdf و latex برای نوشتن در رایانه به کار گرفته می‌شود که قالب پرونده نوشته شده در هر کدام ویژه خود آن نرم‌افزار است. آماده‌کردن یک برنامه رایانه‌ای که همه این قالبها را بخواند اگر ناممکن نباشد بسیار سخت خواهد بود.

برای گردآوری اسناد باید یک قالب یکسان انتخاب شود و تمام اسناد به آن قالب تبدیل شوند. صرفا استخراج متن از اسناد کافی نیست، بلکه باید ساختار متن مثل عنوان‌ها، زیرنویس‌ها، جداول، و برچسب‌ها جداگانه مشخص شوند، چرا که سیستم‌های پیشرفته وزن‌دهی به کلمات از جایگاه کلمات در

متن به عنوان یک پارامتر امتیازدهی استفاده می‌کنند. مثلا اگر کلمه در عنوان بیاید امتیاز بیشتری دارد. لذا قالبی لازم است که بتواند جامع و انعطاف‌پذیر باشد.

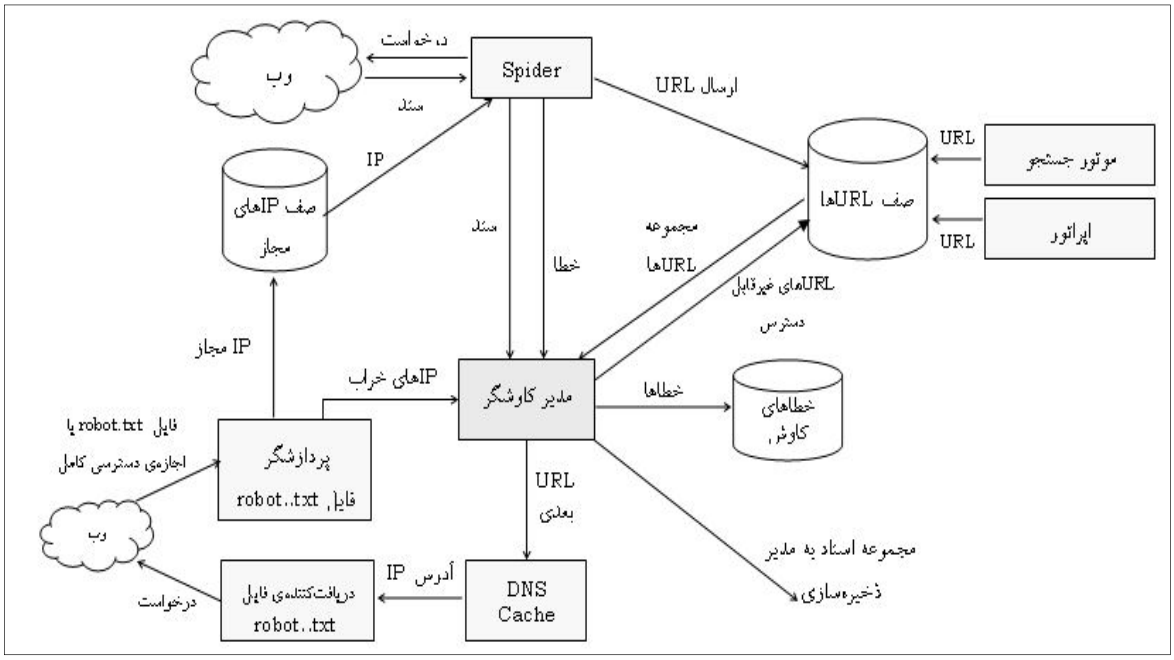
پیشنهاد می‌شود قالب xhtml به عنوان قالب مرجع انتخاب شوند و سایر قالب‌ها (مانند pdf و word) به آن تبدیل شوند. پرونده‌های با قالب xhtml از جنبه‌های گوناگون بهتر هستند. نخست آن که این پرونده‌ها قالب استاندارد دارند که به سادگی می‌توان کلمات درون آن‌ها را با برنامه خواند. دوم، به خوبی از سوی مجمع جهانی وب (W3C) پشتیبانی و به روز می‌شود. سوم، کاربرانی بسیاری از آن بهره می‌برند و روز به روز به دامنه آن‌ها افزوده می‌شود. چهارم، توانایی‌ها و امکانات xhtml روز به روز در حال گسترش است و هم‌زمان می‌توان هم برای نمایش و هم برای چاپ از آن کمک گرفت. البته باید به خوبی با قانون‌های آن و CSS آشنا بود؛ تا بتوان از همه توانایی‌های آن سود برد. پنجم، قابلیت حمل بالایی دارد و به خوبی بر روی رایانه‌های گوناگون و سیستم عامل‌های گوناگون از آن بهره برد.

البته در مورد مساله گردآوری اسناد در سازمان‌های مختلف ممکن است با مسایل متفاوتی مواجه شویم. ممکن است داده‌ها در قالب اسناد فایل‌های خاص سازمان یا پایگاه داده‌ی آن ذخیره شده باشند. در میان روش‌های گردآوری اسناد، اسناد وب متداول‌ترین و جذاب‌ترین آنها می‌باشند؛ بدین منظور در ادامه کاوشگر وب را بیشتر بررسی می‌کنیم.

2-1-1 کاوشگر وب

کاوشگر وب وظیفه‌ی انتقال صفحات از وب به موتور جستجو را به عهده دارد. برای این منظور از نرم‌افزاری موسوم به Spider استفاده می‌نماید. این قسمت، نرم‌افزاری است که به صفحات مختلف وب سر می‌زند و اطلاعات مورد نیاز موتور جستجوگر را جمع‌آوری می‌کند و آنرا در اختیار سایر بخش‌ها قرار می‌دهد. کار Spider شبیه به کار کاربران است. همانطور که کاربران صفحات مختلف را بازدید می‌کنند، Spider هم به منظور جمع‌آوری اطلاعات، این صفحات را پویش می‌کند، با این تفاوت که Spider کدهای اصلی صفحه را بررسی می‌کند. کاوشگر وب نرم‌افزاری است که به عنوان یک کنترل‌کننده‌ی Spider عمل می‌کند. کاوشگر وب مشخص می‌کند که Spider کدام صفحات را مورد بازدید قرار دهد. در شکل ۳، معماری و ساختار کاوشگر وب آمده است.

در واقع کاوشگر وب تصمیم می‌گیرد که کدام یک از پیوندهای صفحه‌ای، که Spider در حال حاضر در آن قرار دارد، دنبال شود. از نظر تئوری، کاوشگر وب می‌تواند از یک صفحه در وب شروع کند، تمامی پیوندهای آن صفحه را بگیرد و آنها را به نوبت کاوش نماید و این کار را آنقدر ادامه دهد تا تمامی صفحات اینترنت کاوش شوند. اما مشکل این ایده در عدم دستیابی به تمام صفحات وب از یک نقطه‌ی شروع است، زیرا بسیاری از صفحات، به صفحات دیگر پیوندی ندارند، بنابراین کاوشگر ممکن است قبلا توسط موتورهای جستجو برای آدرس‌های خاصی برنامه‌ریزی شده باشد.



شکل ۳ - معماری کاوشگر وب

همانطور که در شکل ۳ مشخص شده است، کاوشگر صفی از آدرس‌های صفحات را که در پایگاه URL هستند، تشکیل می‌دهد. کاوشگر به منظور بررسی مجوز دستیابی به صفحات یک سایت، که آدرس آن از صف آدرس‌ها واکشی شده است، فایل متنی robot را، که به عنوان یک مرجع برای مجوز دستیابی موتور جستجو است، بررسی می‌کند.

براساس اصول ارائه شده در طراحی سایت‌ها، طراحان سایت می‌توانند محدوده‌ی واکشی صفحات سایت توسط موتورهای جستجو را در این فایل مشخص کنند. در واقع در این فایل مدیر سایت صفحاتی را که موتور جستجو می‌تواند نمایه‌سازی کند، مشخص کرده است. بعد از بررسی این فایل توسط قسمتی از کاوشگر وب به نام robot.txt Processor، آن دسته از صفحاتی که کاوشگر اجازه‌ی واکشی آنها را

دارد مشخص می‌شود و آدرس این صفحات در پایگاه IPهای مجوزدار قرار می‌گیرد. Spider نزدیک‌ترین قسمت کاوشگر وب به خود صفحات است. در حقیقت Spider صفحات را بررسی می‌کند و یک نسخه از آنها را به مدیر ذخیره‌سازی موتور جستجو می‌فرستد.

آدرس سایت‌هایی که کاوشگر وب آنها را بررسی می‌کند معمولا به سه طریق در مجموعه آدرس‌های کاوشگر قرار می‌گیرد. موتور جستجو معمولا آدرس‌هایی را برای شروع تعیین می‌کند. بعد از بررسی هر صفحه پیوندهایی از صفحه استخراج می‌شود که آدرس صفحات دیگری از وب را به همراه دارند. این آدرس‌ها نیز به مجموعه آدرس‌ها اضافه می‌شود یا ممکن است خود کاربر آدرس سایت را به موتور جستجو ارسال کند.

2-2 نمایه‌ساز

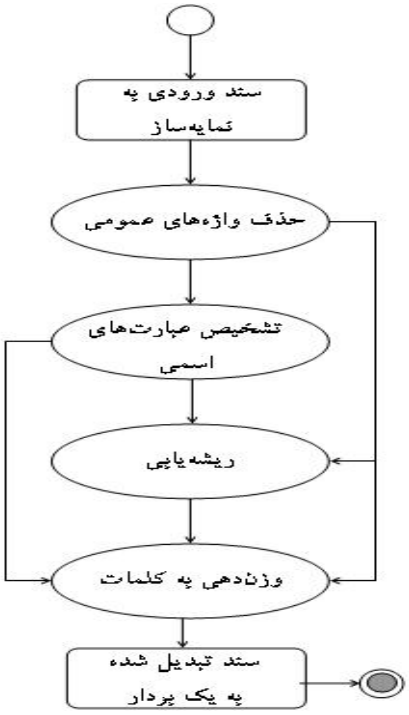
تمام اطلاعات جمع‌آوری شده توسط کاوشگر وب بعد از طی مراحل ذخیره‌سازی در مخزن، در اختیار نمایه‌ساز قرار می‌گیرد. در این بخش، اطلاعات ارسالی مورد تجزیه و تحلیل قرار می‌گیرد. تحلیل صفحات به این معنی است که اطلاعات از کدام صفحه هستند، کلمات کلیدی صفحه کدامند، وزن هر یک چقدر است، واژه‌ها در کدام قسمت صفحه به کار رفته‌اند و ... . در حقیقت نمایه‌ساز، صفحات را به پارامترهای آن تجزیه می‌کند و تمام این پارامترها را به یک مقیاس عددی تبدیل می‌کند تا سیستم رتبه‌بندی بتواند پارامترهای مختلف صفحات را با هم مقایسه کند.

نمایه‌سازی فرایند تحلیل محتوای اطلاعاتی سند به منظور استخراج کلید واژه‌ها به همراه ارزش آن با زبان ویژه نظام نمایه‌سازی است. از آنجا که حضور همه‌ی کلمات متن در نمایه‌سازی سربار زیادی برای سیستم دارد، به نظر می رسد یکی از مشکل‌ترین فعالیت ها در روند نمایه سازی انتخاب کلید واژه‌هایی است که نشان‌دهنده‌ی محتویات سند باشد . هنگام ذخیره اطلاعات به صورت الکترونیکی باید براساس ویژگی های هر زبان، از قواعد و دستور زبان خاصی پیروی کرد تا تشخیص محتوا به گونه‌ای صحیح انجام گیرد، لذا ضرورت تحقیق در مورد نمایه‌سازی خودکار متون فارسی به تفاوت نحوه‌ی ساخت و نیز بکارگیری واژگان در متون این زبان در مقایسه با سایر زبانهای طبیعی برمی‌گردد.

کلمات کلیدی، عناصر بسیار مهمی در جست و جو و دسترسی به اطلاعات هستند. آن‌ها می‌توانند به عنوان مجموعه‌ی کلمات (یک کلمه یا مجموعه‌ای از کلمات) تشریح‌کننده‌ی سند در طی عملیات جست

و جو مد نظر قرار گیرند. به عبارت دیگر، هر عبارت مهمی که محتویات داخل سند را تشریح کند، کلمه کلیدی گفته می‌شود.

برای استخراج کلمات کلیدی یک سری پیش‌پردازش‌هایی باید روی متن باید انجام بگیرد. یکی از این پیش‌پردازش‌ها، تعیین کلمات است. معمولا برای تعیین‌کردن کلمات از فضای خالی، علامات آخر جمله استفاده می‌کنند. در زبان فارسی استفاده از فضای خالی می‌تواند مشکل‌ساز شود، چون بعضی از کلمات فارسی چندبخشی هستند و ممکن است با این مکانیزم یک‌کلمه، چندین کلمه متمایز تشخیص داده شود. از کارهای دیگری که انجام می‌گیرد، می‌توانیم حذف نقطه گذاری، حذف کلمات کوچکتر از یک آستانه را نام ببریم.



شکل 4 - نمودار فعالیت برای نمایه‌سازی متن

برای ایجاد نمایه مناسب از متن معمولا مجموعه فعالیت‌ها مطرح شده در نمودار فعالیت شکل 4 انجام می‌گیرد [4]. البته لازم به ذکر است که برخی از فعالیت‌های مطرح شده در این شکل در برخی نمایه‌سازی‌ها انجام نمی‌شود. عمده فعالیت‌هایی که در نمایه‌سازی انجام می‌شود عبارتند از:

- حذف واژه‌های عمومی (stopword)

- استخراج عبارت‌های اسمی (noun phrase)
- ریشه‌یابی (stemming)
- وزن‌دهی به واژه‌ها و عبارت‌ها
- استخراج کلمات

در ادامه هر کدام از موارد فوق توضیح داده شده‌اند.

2-2-1 حذف واژه‌های عمومی

واژه‌های عمومی، کلماتی هستند که با تکرار بالایی در متون وجود دارند و در ارزیابی متن تاثیر مثبت ندارند یا به اصطلاح در تفکیک (discrimination) نقش ندارند. به منظور کاهش حجم پردازش در اولین فعالیت، این واژه‌ها را از مجموعه کلمات سند، حذف می‌کنیم.

2-2-2 استخراج عبارت‌های اسمی

عبارت‌های اسمی، واژه‌های ترکیبی از دو یا چند اسم هستند که در کنار هم به کار می‌روند و به صورت یک عبارت معرفی می‌شوند. استخراج عبارت‌های اسمی باعث بالا بردن دقت بازیابی (recall) می‌شود.

2-2-3 ریشه‌یابی

بسیاری از کلمات به کار رفته در متن، حالت‌های دستوری متفاوتی از یک ریشه هستند. به منظور کاهش حجم نمایه و بالا بردن معیار بازیابی، معمولا از ریشه کلمات به جای حالت‌های دستوری متفاوت آنها استفاده می‌شود.

2-2-4 وزن‌دهی به واژه‌ها و عبارت‌ها

یکی از موارد مهم در نمایه‌سازی که نقش کلمات را از نظر میزان تاثیر آنها به عنوان کلمات کلیدی متن مشخص می‌کند، وزن کلمه است. در این مرحله با استفاده از الگوهای مختلف وزن‌دهی، به هر کلمه یا عبارت استخراج شده وزنی نسبت داده می‌شود. این وزن بیانگر میزان تاثیر کلمه در موضوع اصلی متن در مقایسه با سایر کلمات به کار رفته در متن است.

2-2-5 استخراج کلمات

در نهایت کلمات و عبارت‌های استخراج شده به همراه وزن آنها به صورت نمایه، معرفی می‌شود.

پس از تعیین کلمات، کلمات عمومی را حذف کرده و بقیه متن را ریشه‌یابی می‌کنیم و سپس کلمات را وزن‌دهی کرده و تبدیل به بردار می‌کنیم و با اعمال آستانه، لیست کلمات کلیدی استخراج می‌شود. در ادامه، مراحل استخراج کلمات کلیدی را تشریح می‌کنیم.

نمایه‌ساز یکی از قسمت‌های اساسی و مهم در موتورهای جستجو است بطوریکه نمایه‌سازی مناسب می‌توانند تاثیر قابل توجهی در بالا بردن کارایی موتور جستجو داشته باشد. متاسفانه پیشرفت واحدهای تحقیقاتی در نمایه‌سازی خودکار متون فارسی کند بوده است. به این معنی که برخلاف متون غیرفارسی، نمایه‌ساز خودکاری که با هزینه‌ی مناسب قابل دستیابی باشد و بتواند یک متن فارسی را به نمایه‌های آن تجزیه کند در دسترس نیست. در [7] یک نمایه‌ساز خودکار فارسی با قابلیت ریشه‌یابی کلمات، طراحی و پیاده‌سازی شده است. در [8] نیز یک پیاده‌سازی برای نمایه‌ساز متون فارسی انجام شده است.

2-3 مدل‌های بازیابی و الگوریتم‌های رتبه‌بندی

اولین گام جهت طراحی سیستم بازیابی اطلاعات این است که مدلی برای توصیف و تعیین مشابهت‌های موجود میان اطلاعاتی که در اختیار دارد با نیازهای اطلاعاتی کاربر تعریف کند. در این بخش مدل یا مدل‌های مورد استفاده‌ی موتور جستجوگر، برای بازیابی اطلاعات و رتبه‌بندی آنها بیان می‌شود.

یکی از نکات اصلی که برای کاربر اهمیت زیادی دارد نحوه‌ی رتبه‌بندی نتایج بدست آمده توسط موتور جستجوگر است. تفاوت در کارایی موتورهای جستجو ناشی از الگوریتم‌ها و مدل‌های مختلفی است که در این قسمت از موتور جستجو پیاده‌سازی شده‌اند. یکی دیگر از نکات این مدل‌ها رفتار متفاوت آنها در زبان‌های مختلف و مجموعه اسناد مختلف است. به این معنی که مدل‌های بازیابی اطلاعات که در موتورهای جستجو به منظور یافتن مشابه‌ترین سند به پرسش کاربر از میان اسناد موجود استفاده می‌شود، باید برای زبان‌های متفاوت (انگلیسی، فارسی و ...) پیاده‌سازی وارزیابی شوند تا بتوان برای زبان مقصد بهترین مدل را انتخاب و استفاده کرد.

حاصل تحقیقات گسترده در بازیابی اطلاعات، طراحی و معرفی مدل‌های مختلفی برای سیستم‌های بازیابی اطلاعات است. برخی از مهم‌ترین آنها، مدل فضای برداری (Vector-Space)، دودویی (Binary)، احتمالی-آماری، شبکه عصبی، فازی، N-gram و شبکه‌های استنتاجی هستند. این مدل‌ها با توجه به مجموعه داده‌های مورد استفاده و زبان مقصد کارایی متفاوتی دارند. مدل‌های فوق را می‌توان در سه کلاس زیر طبقه‌بندی کرد:

- مدل‌های جبری: مانند مدل دودویی (Boolean)،
- مدل‌های تئوری مجموعه‌ای: مانند مدل فضای برداری (Vector Space)،
- مدل‌های احتمالی-آماری (Probabilistic Models)

این مدل‌ها با توجه به مجموعه داده‌های مورد استفاده و زبان مقصد کارایی متفاوتی دارند.

2-3-1 مدل دودویی

در مدل دودویی، نیاز اطلاعاتی کاربر به صورت عبارتی منطقی با عملگر های AND، OR و NOT بیان می‌شود و هر سندی که این عبارت در مورد آن صحیح باشد بازیابی می‌شود. مثلاً اگر نیاز اطلاعاتی به صورت Iran AND Oil بیان شود، تمامی اسنادی که کلمه‌ی Iran و Oil را با هم دارند به کاربر نمایش داده می‌شوند. متاسفانه در مدل دودویی سند یا باربط است یا نیست، و هیچ معیاری برای سنجش میزان ربط وجود ندارد. مثلا دو سندی که یکی تماما در باره ایران و نفت بحث می‌کند، و دیگری در مورد اقتصاد جهانی صحبت می‌کند و فقط از نام ایران و نفت به عنوان مثالی در یک جمله استفاده کرده است، از نظر سیستم تفاوتی نیست. در صورتیکه در واقع سند اول بیشتر به نیاز کاربر مربوط است.

استراتژی جست و جوی دوارزشی، اسنادی را بازیابی می‌کند که برای پرس وجو مقدار True را داشته باشند. این فرموله سازی زمانی قابل توجیه است که پرس وجو به صورت کلمات شاخص (کلمات کلیدی) و ترکیب این کلمات با استفاده از عملگرهای منطقی معمول مثل AND, OR, NOT نمایش داده شود.

برای مثال اگر پرس وجو $Q = (K_1 ANd\ k_2) OR (K_3 ANd (Not\ K_4))$ باشد، جست و جوی دوارزشی تمام اسنادی را بازیابی خواهد کرد که با استفاده از K_1 و K_2 شاخص شده باشند و هم‌چنین اسنادی که با استفاده از K_3 شاخص شده و با K_4 شاخص نشده باشند را نیز بازیابی خواهد کرد.

2-3-2 مدل برداری

در مدل برداری، هر مستند را به صورت برداری از کلمات در نظر می گیریم و فضایی چند بعدی که ابعاد آنرا کلمات تشکیل می‌دهند ایجاد می کنیم. سپس هر سند در این فضا به صورت یک بردار نمایش داده می‌شود. مولفه‌های این بردار سند، در واقع وزن هایی هستند که نشان می‌دهند هر یک از کلمات چقدر در متمایز کردن آن سند دخیل هستند. در مدل احتمالاتی ، به هر سند احتمالی اختصاص داده می‌شود که مربوط بودن آن مستند را به نیاز کاربر به صورت احتمال بین صفر و یک بیان می‌کند.

در مدل برداری، برای سنجش میزان ربط اسناد و نیاز اطلاعاتی کاربر، سیستم دقیقا به مانند قبل نیاز اطلاعاتی کاربر را هم به فضای چندبعدی از کلمات می‌برد و در نتیجه برای سنجش میزان شباهت میان این دو بردار می‌توان از زاویه‌ای که این دو بردار با هم می سازند استفاده کرد. اسنادی که با نیاز اطلاعاتی کاربر دقیقا هم جهت هستند مسلما نسبت کلماتشان به همان نسبت کلمات نیاز اطلاعاتی است و در نتیجه مرتبط‌تر خواهند بود. برتری این مدل این است که به ما درجه‌ای از ربط را می‌دهد.

مدل فضای برداری پایه‌ای‌ترین مدل در سیستم‌های بازیابی اطلاعات است. در این مدل ابتدا سند به برداری تبدیل می‌شود که حاوی کلمات مهم متن سند، به همراه وزن هر کلمه بر اساس میزان تاثیرگذاری کلمه بر محتوی متن در مقایسه با سایر کلمات است. تهیه بردار برای هر سند بر اساس تکنیکی به نام نمایه‌سازی صورت می‌گیرد. در نمایه‌سازی ابتدا کلمات عمومی از متن حذف می‌گردند و کلمات باقی مانده ریشه‌یابی می‌شوند. سپس بر اساس پارامترهای مختلفی مانند تعداد تکرار کلمه در متن، تعداد تکرار کلمه در متن، تعداد تکرار کلمه در اسناد مجموعه و مولفه‌های نرمال‌سازی وزنی به هر کلمه نسبت داده می‌شود. همین فعالیت‌ها برای پرسش کاربر نیز تکرار می‌شود. به این ترتیب هر سند از

مجموعه‌ای از کلمات به برداری تبدیل می‌شود که در فضای جدیدی به نام فضای بـرداری قـرار دارد. در این فضا که بسته به تعداد کلمات مجموعه یک فضای n بعدی است، بـردار هـر سـند ترسـیم مـی‌شـود. پرسش کاربر نیز بعد از اعمال فعالیت‌های نمایه‌سازی به برداری تبـدیل مـی‌شـود کـه در فضـای جدیـد ترسیم می‌گردد. در این فضا هر سندی که به پرسش کاربر نزدیک‌تر باشد سند مرتبط شناخته می‌شود و بازیابی می‌گردد. معیار نزدیکی در این فضا زاویه‌ای است که بردار پرسش با هـر یـک از بردارهـای سـند می‌سازد. این میزان نزدیکی، معمولا با رابطه زیر که به نام مشابهت کسینوسی شناخته می‌شود، محاسبه می‌گردد:

$$sim(q_i, d_j) = \frac{\vec{q_i}.\vec{d_j}}{|\vec{q_i}| \times |\vec{d_j}|} = \frac{\sum_{k=1}^{t} w_{ki} \times w_{kj}}{\sqrt{\sum_{k=1}^{t} w_{ki}^2} \cdot \sqrt{\sum_{k=1}^{t} w_{kj}^2}}$$

در این رابطه q_i بردار پرسش کاربر، d_j بردار سند jام، w_{ki} وزن کلمه k ام در پرسش کاربر و w_{kj} وزن کلمه kام در سند d_j است.

2-3-3 مدل احتمالاتی

در مدل احتمالاتی هم به ازای هر نیاز اطلاعاتی، تمامی اسناد بر اساس احتمال این که این سند بـا نیـاز اطلاعاتی مرتبط باشد، مرتب می‌شوند و لیست اسناد در نهایت به صورت درجه بندی شده (ماننـد مـدل برداری) به کاربر نمایش داده می‌شود به نحوی که اولین سندی که کاربر می بیند از همه بیشتر احتمال دارد که به نیاز او ربط داشته باشد.

بعد از تعریف این مدل، سیستم اکنون آماده است که نیاز اطلاعاتی کاربر را دریافت کند. معمولا کاربران نیاز اطلاعاتی خود را در قالب چندین کلمه یا عبارات معمولی به سیستم بیان می‌کنند. سیستم سپس بر اساس مدلی که اطلاعات را در آن مدل کرده است، میزان ربط هر سند را با نیاز اطلاعاتی کاربر محاسـبه می‌کند و آن سندهایی را که از همه باربط تر تشخیص داده شده اند به عنوان خروجی باز می گرداند.

2-3-4 معیارهای ارزیابی مدل

معیارهایی برای ارزیابی مدل‌های بازیابی و الگوریتم‌ها وجود دارند که در ادامه آمده‌اند.

2-3-4-1 دقت (Precision)

بیانگر قابلیت سیستم برای ارائه فقط موارد مربوط به درخواست کاربر است. این مقدار نسبت مستقیمی با میزان تعیین‌کنندگی کلمات در متن دارد. برای محاسبه پارامتر دقت بر اساس رابطه، نسبت تعداد اسناد مرتبط بازیابی شده بر کل اسناد بازیابی شده برای پرس و جو، محاسبه می‌شود.

$$\text{precision} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|}$$

2-3-4-2 بازخوانی (recall)

بیانگر قابلیت سیستم برای ارائه موارد مربوط به درخواست کاربر است. این مقدار نسبت مستقیمی با میزان دربرگیری کلمات در متن دارد. برای محاسبه بازخوانی نسبت تعداد اسناد مرتبط بازیابی شده بر کل اسناد مرتبط با پرس و جو محاسبه می‌شود. رابطه‌ی زیر برای محاسبه پارامتر بازخوانی به کار می‌رود.

$$\text{recall} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{relevant documents}\}|}$$

مقدار این دو پارامتر غالباً نسبت معکوس با هم دارند و بهبود یکی باعث افت دیگری می‌شود. با توجه به این که عملیات ارزیابی با استفاده از مجموعه‌ای از پرس و جوها انجام می‌شود، روشی به عنوان روش برتر در نظر گرفته می‌شود که برای مجموعه پرس و جوها میانگین بهتری داشته باشد. به عنوان مثال، اگر مقدار هر دو پارامتر در پرس و جوی A بیش‌تر از پرس و جوی B باشد، نتایج پرس و جوی A بهتر خواهد بود.

2-3-4-3 پارامتر Fall-out

این پارامتر بیانگر نسبت میزان خطا می‌باشد. و با محاسبه نسبت تعداد اسناد نامرتبط بازیابی شده بر کل اسناد نامرتبط با پرس و جو محاسبه می‌شود. رابطه زیر محاسبه پارامتر Fall-out را نشان می‌دهد.

$$\text{fall-out} = \frac{|\{\text{non-relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{non-relevant documents}\}|}$$

2-3-4-4 پارامتر $F_{measure}$

در حقیقت این پارامتر میانگین هارمونیک پارامترهای بازخوانی و دقت می‌باشد. هدف در سیستم‌های بازیابی اطلاعات بیشینه کردن این معیار می‌باشد. $F_{measure}$ بر اساس رابطه زیر محاسبه می‌شود.

$$F = \frac{2 \cdot \text{precision} \cdot \text{recall}}{(\text{precision} + \text{recall})}.$$

3. مسایل خط و زبان فارسی در بازیابی اطلاعات

خط و زبان فارسی بطور اساسی مشکلاتی را برای سیستم‌های ذخیره و بازیابی اطلاعات ایجاد می‌کننـد. در [9] می‌خوانیم:

> *بانکهای اطلاعاتی فارسی، پیش از آن که فرهنگستان زبان معیارهای لازم را برای کاربرد اصطلاحات علمی و رسم‌الخط فارسی تعیین کند شکل گرفتند. مجریان بانکهای اطلاعاتی و نمایه‌سازان، خواسته یا ناخواسته با مسائل واژه گزینی و جنبه‌هایی از زبانشناسی درگیر شدند. در کار واژه‌گزینی، اطلاع‌رسانان به لحاظ ماهیت حرفه خود واژه‌های رایج در جامعه تولیدکنندگان و استفاده‌کنندگان از اطلاعات را مد نظر دارند و خود را مجاز به واژه‌سازی و اعمال سلیقه نمی‌دانند. واژه‌های تازه‌ساخت نیز تا زمانی که در جامعه مقبولیت لازم را به دست نیاورده باشند و در مدارک به کرات دیده نشوند، در نظا مهای ذخیره و بازیابی اطلاعات یا ظاهر نمی‌شوند و یا میهمان چندروزه‌اند. بخش قابل توجهی از مشکلات نمایه‌سازان از رواج و کاربرد واژه ناشی می‌شود. متخصصان برای یک مفهوم واحد اصطلاحات متفاوت به کار می‌برند. حتی متخصصانی که در یک رشته و در یک جامعه کوچک کار می‌کنند خود را ملزم به هماهنگی در کاربرد واژه‌های تخصصی نمی‌بینند. به علاوه برای بسیاری از اصطلاح‌های وارداتی معادلهای متفاوت در زبان فارسی وجود دارد که در مواردی همه، کم و بیش، به یک اندازه کاربرد دارند. این گونه مطالب به علاوه مسائل رسم‌الخط فارسی، آوانویسی اسامی عناصر وترکیبات شیمیائی، سرواژه‌ها و کوته نوشته‌ها سبب شده است تا ذخیره اطلاعات به زبان فارسی با کندی صورت گیرد و جستجو و بازیابی کارایی مطلوب را نداشته باشد.*

در [9] به برخی از مشکلات زبان فارسی اشاره شده‌اند که عبارتند از:

- گوناگونی معادل‌های علمی
- ضبط اسامی
- تعیین مرز کلمات: سرهم‌نویسی، جدانویسی و بی‌فاصله نویسی

- انواع جمع‌ها
- صورت‌های مختلف نوشتاری

توضیح موارد فوق بطور اجمالی در ذیل می‌آید.

3-1 گوناگونی معادل‌های علمی

متخصصان در بیان و انتقال یک مفهوم از اصطلاحات متفاوت استفاده می‌کنند و هیچ وحدت رویه‌ای در تولید و بکارگیری واژگان جدید وجود ندارد. به عنوان مثال طبق [10] برای کلمه «Online»، 12 معادل و برای کلمه «Manual»، 9 معادل، بکار رفته است.

3-2 ضبط اسامی

در برگردان اسامی افراد، سازمان‌ها، عناصر و ترکیبات شیمیایی، ابزار و تجهیزات، محل‌های جغرافیایی و مانند آن‌ها از زبا نهای بیگانه به فارسی، قاعده خاصی وجود ندارد. هر متخصص، نویسنده و مترجمی بنا به ذوق و سلیقه، میزان آشنایی با زبان مبدأ و دانش و تخصص خود، آنها را به فارسی برگردانده و در متون بکار برده است. به عنوان مثال «رابینسون، روبینسون، ربینسون، روبنسن»، یا «پتاسیم، پتاسیوم، پوتاسیوم، پوتاسیم».

3-3 تعیین مرز کلمات: سرهم‌نویسی، جدانویسی و بی‌فاصله نویسی

شیوه خط فارسی چنان است که بسیاری از واژه‌ها را می توان به چند صورت نوشت. این چندگونگی شکل واژه‌ها، برای رایانه قابل درک نیست. چرا که واژه‌ها را تنها به همان صورتی که ذخیره کرده است

می‌شناسد و بازیابی می‌کند. لذا در مقابل سایر شکلهای نوشتاری یک اصطلاح ناآگاه است و در هنگام جستجوی اطلاعات پاسخ منفی می‌دهد. رابط‌های اطلاعات برای پرهیز از این مشکل، عموما از فهرست کلید واژه‌ها استفاده می‌کنند که این امر سبب شده است تا از امکانات منطق بول در بازیابی اطلاعات به خوبی بهره گرفته نشود. در مواقعی که بازیابی از محدوده فیلد کلید واژه‌ها، که اصطلاحات مهارشده‌اند، فراتر می‌رود و فیلدهای عنوان، پدیدآورنده و ناشر را شامل می‌شود، این ناهماهنگی کاملا به چشم می‌خورد. گاه یک واژه مرکب براساس شکل نگارش آن در چند محل الفبایی مختلف، جدا از هم قرار می‌گیرد. علامت جمع «ها» که به صورت سرهم یا جدا نوشته شود نیز، همین وضع را در فهرست‌های رایانه‌ای ایجاد می‌کند. برای مثال: «آب گرم کن، آ ب گرمکن، آبگرم کن، آبگرمکن» یا «علی‌رضا، علی رضا».

3-4 انواع جمع‌ها

تعدّد علائم جمع (ها؛ ان؛ ات ؛ ین ؛ ون) و وجود جمع بیقاعده در زبان فارسی سبب گردیده است در پایگاه‌هایی که کلیدواژ ه‌ها را به صورت جمع به کار می برند، مشکلی بر مشکلات بالا افزوده شود. نمایه‌ساز در هنگام نمایه‌سازی در انتخاب بین مدارس/مدرسه‌ها، اساتید/استادان/ استادها، محققان/محققین و مانند آن ها، مردد است. رابط اطلاعات در موقع بازیابی باید شکل‌های مختلف جمع کلیدواژ ه‌ها را در نظر داشته باشد و یا، با استفاده از علائم قراردادی، واژه را برش بزند. در هر دو صورت، باز هم احتمال پوشش ندادن بعضی از جمع‌های بی‌قاعده وجود دارد.

3-5 صورت‌های مختلف نوشتاری

همزه، الف مقصوره، تشدید و دوگانگی شکل نوشتاری واژه‌ها و اسامی، سبب ناهماهنگی‌هایی در ورود داده‌ها و پراکندگی اطلاعات پردازش شده می‌گردد. مانند «هیات، هیئت» و «اسمعیل و اسماعیل».

مورد 3-6 را می‌توان به موارد ذکر شده در [9] افزود.

3-6 استفاده از زبان محاوره‌ای در نوشتار

اخیرا با رواج وبلاگ‌نویسی در میان فارسی زبانان استفاده از واژه‌های محاوره‌ای در متون نوشتاری بسیار باب شده است. متاسفانه این گونه از متون محاوره‌ای به سایت‌های رسمی‌تر نیز کشیده شده است. با این وجود تشخیص کلمات معادل برای بازیابی اطلاعات کار دشواری است.

4. استاندارد خط فارسی در رایانه

روند فارسی‌سازی و استاندارد نمودن خط فارسی برای رایانه فراز و نشیب‌های زیادی داشته است. کوچک‌ترین واحد نوشته نویسه (character) نامیده می‌شود. نویسه یک حرف، اعراب، علامت نقطه‌گذاری، نشانه بریل یا نماد ریاضی می‌تواند باشد. هر حرف دارای یک یا چند شکل نمایش است که شکل (glyph) نامیده می‌شود. برای نمونه نویسه‌ی «ی» دارای شکل‌های نمایشی«یـی»، «ی» ، «ی»، «پیر » است. مجموعه کد به دو گونه تعریف شده است [11]:

- نگاشت میان هر شکل با یک بایت (یا چند بایت پیاپی)
- نگاشت میان هرنویسه با یک بایت (یا چند بایت پیاپی)

موسسه‌ی استاندارد و تحقیقات صنعتی ایران در استاندارد 2900 روش اول را برگزید و برای هر شکل یک نویسه یک کد یک بایتی قرار داد. این روش را روش تک نمادی نیز می‌نامند. شکل نمایش یک نویسه در کلمه بستگی به جای آن نویسه در کلمه و پیوندناپذیر بودن حرف دارد. برای نمونه «ی» پیوندناپذیر است. شکل‌های گوناگون «ی» در کلمات «یک»، «میان»، «یکی»، «برای» بستگی به جای آن دارد. بنابراین می‌توان با دسته‌بندی حروف فارسی و به کارگیری الگوریتم با توجه به جای حرف در کلمه شکل نمایش آن را شناسایی کرد. ولی به کلمه «خانه ها» دقت کنید که در آن می‌خواهیم «ها» در کنار و بدون فاصله با «خانه» باشد و «ه» در پایان خانه به شکل «خانهها» تبدیل نگردد. بنابراین نویسه‌ی فاصله مجازی (Zero-Width Non-Joiner) پیشنهاد شد. این نویسه پس از «خانه» و پیش از «ها» گذاشته شده است. هم‌چنین در «ه‌ش» می‌خواهیم که «ه» به شکل «ه. ش» نوشته نشود. بنابراین نویسه‌ی اتصال مجازی (Zero-Width Joiner) پیشنهاد گردید. این نویسه پس از«ه» در «ه‌ش»گذاشته شده است تا شکل دلخواه ما به دست آید. امروزه بیش‌تر روش تک نمادی به کار گرفته می‌شود. شرکت‌های بزرگ دنیا به جای پذیرش استاندارد ایران مجموعه کد دیگری را به کار گرفتند که بزرگ‌ترین تفاوت آن با استاندارد 3342 موسسه استاندارد ایران رعایت‌نکردن ترتیب چهار حرف «پ»، «چ» ، «ژ»، «گ» در این مجموعه کد است. البته با توجه به همه‌گیرشدن این کد به کمک نرم‌افزارهای خارجی در ایران، استاندارد ایران (حتی در درون کشور) به فراموشی سپرده شد. به همین ترتیب استاندارد 2901 برای صفحه کلید نیز تا اندازه‌ای به دست فراموشی سپرده شد. البته در برخی از سیستم‌های عامل (مانند linux) و برخی نرم‌افزارها (مانند unipad) استاندارد صفحه کلید ایران رعایت شده است.

چون یک بایت گنجایش همه نویسه‌های زبان‌های گوناگون را ندارد به هر مجموعه کد برای یک زبان نامی داده شد. مجموعه کد عربی (و فارسی) را «cp1256» یا «windows1256» یا « Arabic windows» نام نهادند. یکی از دردسرهای دیگر این مجموعه کد، گذاشتن حرف «ی» با دو کد 237 و 236 در آن است؛ استاندارد به روشنی میان این دو تفاوت گذاشته است. «ی» برای عربی و «ی» برای فارسی درنظر گرفته شده است. با این همه سیستم‌های عامل گوناگون و نرم‌افزارهای گوناگون بدون توجه به زبان، یکی از این دو را به کار می‌برند و در پردازش نوشته‌های رایانه‌ای فارسی باید دقت نمود. کدهای 152 و 233 نیز برای «ک» به کار رفته است ولی اغلب برای فارسی 152 به کار می‌رود.

با توجه به این که یک بایت برای همه زبان‌های دنیا بسنده نیست؛ پس به جای یک بایت پیشنهاد شد که دو بایت برای کد کردن نویسه‌ها به کار گرفته شود. این روش کدگذاری را یونی‌کد (Unicode) نامیدند. البته دراین کد نیز ترتیب چهار حرف فارسی رعایت نشده است. هم‌چنین مشکل حرف‌هایی با چندین کد (و رعایت نکردن فارسی یا عربی بودن آن در نرم‌افزارهای ویرایش‌گر) نیز وجود دارد. یونی‌کد با طول دو بایت «ucs2» نامیده شد. گسترش یونی‌کد به چهار بایت، «ucs4» نامیده شد.

اغلب سخت‌افزارها و نرم‌افزارها بر پایه یک بایت کار می‌کردند؛ هم‌چنین یونی‌کد یک استاندار دو بایتی (یا چهار بایتی ucs4) است؛ پس باید همه سیستم‌ها جایگزین سیستم‌هایی می‌شدند که بتوانند با دو یا چهار بایت کار کنند. تبدیل ناگهانی سیستم‌ها هزینه‌ی سنگینی را دربرداشت. بنابراین تصمیم گرفته شد به گونه‌ی کدگذاری شود که سخت‌افزارها و نرم‌افزارهای موجود هم بتوانند دست کم با حروف زبان انگلیسی (که برای آن هم ساخته شده بودند) کار کنند. پس باید مجموعه کدی ساخته می‌شد که برای نویسه‌های زبان انگلیسی (زیر 128) یک بایتی می‌بود. با توجه به این محدودیت، تنها راه چاره به کار بردن مجموعه کدی با طول متغیر بود. این روش کدگذاری با تعداد متغیر بایت، «UTF-8» نامیده شد. دو بایت (یا چهار بایت) یک نویسه در یونی‌کد در UTF-8 به کدی با تعداد بایت‌های متغیر (از یک تا حداکثر 6 بایت) نگاشته می‌شود. تعداد بایت در این نگاشت بستگی به نویسه دارد. اغلب برای پردازش پرونده‌ای با کد UTF-8 کد پرونده به یونی‌کد تبدیل می‌شود.

تعداد زیادی از سندهای رایانه‌ای کدهای ویژه خود را دارند. برای ریشه‌یابی (یا هر پردازش نوشتار) کدگذاری‌های گوناگونی باید به یک کد تبدیل شوند تا بتوان ریشه‌یابی را برروی کلمات آن انجام داد.

4-1 دستور خط فارسی

گرچه با کوشش فرهنگستان زبان و ادب فارسی استاندارد یکسانی برای دستور خط فارسی آماده شده است ولی هنوز به خوبی بسیاری از این دشواری‌ها در نوشته‌های درسی دیده می‌شود. هم‌چنین ناهماهنگی‌هایی در خود استاندارد دیده می‌شود. نمونه‌هایی که در این نوشتار آورده شده است بیش‌تر از کتاب‌های دیگر و یا از سایت‌های شبکه جهانی نمونه آورده شود؛ دامنه‌ی بسیار گسترده‌تری از این ناهماهنگی‌ها دیده می‌شد. برای برطرف شدن این ناهماهنگی‌ها در کار برنامه‌نویسی، باید همه حالت‌های ممکن پوشش داده شود. هم‌چنین برای کمک به بهبود این ناهماهنگی‌ها در نوشتار رایانه‌ای فارسی پیشنهادهایی داده شده است. در واقع باید یک رسم‌الخط یکسان به عنوان رسم‌الخط مرجع پذیرفته شده و تمامی مستندات قبل از پردازش‌های بعدی در صورت امکان به آن تبدیل شوند.

در [12] دستور خط زبان فارسی به طور کامل بیان شده است که باید توسط همگان رعایت شود؛ اما هیچ تضمینی بر آن نیست. از جمله مواردی که می‌توان به عنوان موارد اختلاف در متون مختلف اشاره کرد، موارد زیر هستند:

- اتصال «ی» پس از «ه» یا نویسه «ۀ» و دیگری با دو نویسه «ه» و«ء» است
- قوانین اتصال «ها»
- قوانین فاصله‌گذاری
- دگرگونی کلمه‌ها در هنگام پیوند

در [12] قوانین فوق و بسیاری قوانین دیگر آورده شده است که در اینجا به آن اشاره نمی‌کنیم. در ریشه‌یابی کلمات و تعیین کلمات کلیدی باید کلیه حالت‌ها در نظر گرفته شوند.

4-2 مشکل اعراب گذاری و نویسه‌های خاص

از دیگر مشکلاتی که در زبان فارسی وجود دارد مساله اعراب‌گذاری است. در اصل ما در نوشتار فارسی اعراب نمی‌گذاریم، اما ممکن است گاهی اوقات برای رفع ابهام از اعراب استفاده کنیم. در [12] قوانین مربوط به اعراب گذاری آمده است. بر اساس [12]:

در خط فارسی افزون بر حرف‌های الفبا 9 نشانه خطی دیگر نیز به کار می‌رود. این نشانه‌ها ـَ ـِ ـُ ـّ ـً ـٍ ـٌ ـْ ـٰ *هستند. کاربرد این نشانه‌ها کم است؛ زیرا در خط فارسی حرکت‌گذاری به کار برده نمی‌شود. در نوشتن کلمه‌ها از میان نشانه‌های نه گانه بالا مد، تشدید، تنوین نصب (اً ـّـً) بیش‌تر کاربرد دارند. تنوین رفع و جر (ـّـٍ) تنها در کلمات عربی رایج در فارسی به کار می‌رود و دیگر نشانه‌ها را در جاهایی به کار می‌بریم که رعایت نکردن آن‌ها ابهام و بدفهمی به وجود می‌آورد.*

این نشانه‌ها نیز دشواری دیگری در ناهماهنگی درنگارش فارسی به وجود آورده‌اند، مانند «رفت» و «رُفت» که تنها فرق آن‌ها در(ـَ) و(ـُ) است که برسر «ر» گذاشته شده است. ولی رُفتگر را بیش‌تر در نگارش بدون (ـُ) می‌گذارند و این کار ریشه‌یابی را سخت‌تر می‌کند. البته می‌دانیم که «رَفتگر» نداریم که باید به صورت استثنا به رایانه داده شود. بنابراین با توجه به زبان فارسی تعداد این استثناها بسیار زیاد خواهد بود.

خوشبختانه با گسترش رایانه‌ها و در دسترس بودن نرم‌افزارهای توانمند نگارش و ویرایش و دامنه بزرگی از نویسه‌ها که این نرم‌افزارها پشتیبانی می‌کنند از این دشواری کمی کاسته شده است و گذاشتن این نشانه‌ها نیز ساده‌تر گشته است. گرچه هنوز نمی‌توان به درستی گفت که کجا باید این نشانه‌ها رعایت شوند.

علاوه بر مشکلات اعراب گذاری، دسته‌ای دیگر از مشکلات به علت مسایل مربوط به کدگذاری نویسه‌های فارسی ایجاد می‌شود که در بخش 3-1 به آن اشاره شد. به عنوان مثال برخی از آنها در زیر اشاره شده‌اند:

- نویسه «ۀ» و دیگری با دو نویسه «ه» و«ء» است
- نویسه «ـ» برای مثال در «کـــــــمک» برای کشیده‌نویسی کلمات
- کاراکترهایی با دو کد مختلف مانند «ک»

البته به نظر می‌آید که بتوان مشکلات ناشی از اعراب‌گذاری و کاراکترهای خاص را به سادگی در یک مرحله پیش‌پردازشی مرتفع نمود؛ اما به هر حال این کار نیازمند دقتی خاص است تا نتیجه‌ی کار در بر گیرنده تمام حالت‌ها باشد.

5. سفارشی کردن موتور جستجو برای زبان فارسی

در بخش‌های پیشین به ساختار موتور جستجو و مسایل و مشکلات زبان فارسی اشاره شد. در این بخش ما به تشریح مسایل و راهکارهای مرتبط با زبان فارسی خواهیم پرداخت.

5-1 کارهای پیش‌پردازشی

با توجه به مشکلات بیان شده برای زبان و رسم‌الخط فارسی در بخش قبل، ما نیاز مند انجام یک مرحله‌ی پیش‌پردازشی برای یکسان‌سازی مستندات هستیم. این مرحله می‌تواند شامل اعمال زیر باشد:

- یکسان‌سازی کدگذاری نویسه‌ها
- یکسان‌سازی رسم‌الخط
- تشخیص مرز کلمات
- حذف یا یکسان‌سازی اعراب گذاری
- یکسان‌سازی املاهای مختلف کلمات

انجام مراحل فوق باعث یکسان‌سازی مستندات می‌شود و باعث می‌شود تا کیفیت بازیابی مستندات بسیار بهبود یابد.

5-2 کلمات عمومی

به عنوان اولین فعالیت در روند نمایه‌سازی، واژه‌های عمومی در متن ورودی حذف می‌گردند. با توجه به شیوه استفاده از واژگان زبان، بعضی واژه‌ها در همه متون با تکرار زیاد وجود دارند. این گونه واژه‌ها، واژه‌های عمومی زبان نامیده می‌شوند. در واقع این واژه‌ها، واژه‌هایی مثل ضمایر، قیود، حروف اضافه و ربط هستند که در بازیابی، تاثیری بر ارزش محتوایی سند ندارد. واژه‌های عمومی زبان یا توسط زبان شناسان معرفی می‌شود و یا بر اساس نرخ تکرار در هر سند بدست می‌آیند.

در [8]، برای واژه‌های عمومی زبان فارسی یک فهرست 180 واژه‌ای بر اساس تعداد تکرار واژه در سند و بالاتر بودن نرخ تکرار از آستانه تعریف شده، تهیه گردیده است. در اولین گام کلمات متن ورودی بعد از مقایسه با این واژه‌ها در صورت برابری حذف می‌شوند.

برخی از کلمات عمومی فارسی
امروز، گفتم، اکنون، خواهند، آر، آقا، آقای، آقایان، آمد، آمده، آن، آنان، آن‌جا، آنچه، آنکه، آن‌ها، آیا، اخیر، از، است، اسلامی، اش، افزود، اگر، اگرچه، الا، البته، الی، ام، اما، امروزه، اند، اندی، او، اولین، ای، ایران، ایشان، ایم، این، این‌جا، اینکه، این گونه، با، بااین، بااینکه، بار، باز، باشد، باشید، باشیم، بالاخره، باید، بجز، بدهید، بدون، بر، برای، براین، برخی، برلزوم، بسیار، بسیاری، به طور، بعد، بکنید، بگذاریم، بگوییم، بلکه، بماند، به، بود، بودند، بوده، بی، بیش، بین، پس، پی، پیش، تا، تر، تری، تمامی، تو، توسط، توی، جا، جز، ، چرا، چنان، چند، چنین، چه، چو، چون، چونکه، حال، حالی، حالی که، حتی، حدود، حقیقتا، خانم، خانمها، خواهد، خود، خودم، خودمان، خویش، داخل، داد، دادم، دادند، داده، دار، دارای، دارد، دارند، داریم، داشت، داشته، داند، دانند، در، درآن، دراین، درباره، دربر، دربعد، دربین، درپی، درجای، درحال، درحالی، درحالی که، دردو، درکل، درین، دور، دیگر، را، رسیده، رو، روی، زدند، ساعت، سر، سعی، سو، سوی، شامل، شد، شدن، شدند، شده، شما، شود، طی، علیرغم، علیه، غیر، فقط، کرد، کردم، کردن، کردند، کرده، کنار، کند، کنم، کنند، کنید، کنیم، که، گذاری، گرچه، گردند، گرفت، گرفته، گفت، گفتند، گفته، لزوم، ما، مانند، متوالی، مثلا، من، می، می‌شود، میان، میتواند، میخواهیم، میدانند، میرسد، میشود، میکنم، میکنند، ندارد، ندارم، ندارند، نداشته، نشدند، نظر، نماید، نموده، نمی، نمیکند، نیز، نیست، نیستند، ها، های، هایی، هر، هریک، هست، هستم، هستند، هستید، هستیم، هم، همان، همه، همین، هنوز، هیچ، و، وجود، ولی، وی، یا، یافت، یعنی، یکدیگر، یکم، یکی

بعضی از کلمات در همه‌ی متون با فراوانی زیاد وجود دارند که ارزش محتوایی ندارند، مثل ضمایر، قیود، حروف اضافه و ربط و بعضی از افعال پرتکرار. به این کلمات، کلمات عمومی گفته می‌شود. با حذف کلمات عمومی در متن کاوی آماری میزان محاسبات کم شده و کارآیی روش‌ها نیز بیش‌تر می‌شود. در جدول – برخی کلمات عمومی فارسی آمده‌اند.

3-5 بازیابی تحمل‌پذیر

مظور از بازیابی تحمل‌پذیر این است که اگر کاربر کلمات کلیدی را به اشتباه وارد کرد، سیستم توانایی تشخیص و اصلاح آن را داشته باشد. معمولا در مواقع زیر پیشنهاد اصلاح کلمه ورودی به کاربر داده می‌شود:

- وقتی کلمه وارد شده در موتورهای جستجو در میان کلمات نمایه شده در انباره نباشد یا تعداد رخداد آن کم باشد و نیز
- وقتی کاربر دو کلمه را در کنار هم وارد می‌کند که بسیار شبیه به یک عبارت متداول هستند

وقتی سیستم پیشنهاد اصلاح را به کاربر می‌دهد از عبارت «Did you mean?» استفاده می‌کند که باعث شده این روش مشابهت‌یابی، به همین نام نیز مصطلح شود.

الگوریتم‌هایی که برای مشابت یابی استفاده می‌شوند بر اساس فاصله ویرایشی و نمایه‌گذاری n-gram کار می‌کنند.

موتور جستجو با استفاده از این روش مشابهت یابی می‌تواند امکانات زیر را ارایه دهد:

- تصحیح املای کلمات
- تشخیص عبارات اسمی متداول

با توجه به هرج و مرج موجود در گذاشتن فاصله و نیم‌فاصله در میان کلمات و عبارات فارسی، یکی از طلایی‌ترین راه‌حل‌ها برای این مشکل، امکان تشخیص عبارت‌های متداول می‌باشد.

4-5 ریشه‌یابی

ریشه‌یابی یکی از پیچیده‌ترین مراحل کار در نمایه‌سازی متون است که مستلزم شناخت کافی از دستور زبان و تهیه ماشین حالت مناسب و متناسب با ساختار زبان است. اشتقاق یک واژه از ریشه اصلی موجب می‌شود تا برای ایفای نقش در جمله آماده شود. هدف از ریشه یابی، زدودن الحاقات و یافتن جوهره اصلی واژه است. هر چند در واقعیت گاهی الحاقات واژه معنای آن را چنان تغییر می دهند که حذف آنها

موجب از بین رفتن معنای اصلی می شود. ریشه یابی کلمات حجم ذخیره‌سازی نمایه را به مقدار قابل توجهی کاهش می‌دهد و از سوی دیگر بازخوانی اسناد مرتبط به پرسش کاربر را بهبود می‌بخشد.

بدلیل اهمیت و گستردگی موضوع ریشه‌یابی در زبان فارسی، در بخش 5 بطور جداگانه به آن خواهیم پرداخت.

5-5 وزن دهی

در این مرحله با استفاده از الگوهای مختلف، به هر یک از واژه‌های استخراج‌شده وزنی نسبت داده می‌شود. این وزن، بیانگر میزان تاثیر کلمه به موضوع متن در مقایسه با سایر کلمات به کار رفته است. وزن‌دهی به کلمات بر اساس اهمیت آن‌ها در متن انجام می‌گیرد. اهمیت کلمات را می‌توان بر پایه شرایط زیر مشخص کرد [2, 4]:

- وزن آماری کلمه: بر پایه‌ی تکرار کلمات در متن، بر پایه‌ی توزیع کلمات در متن
- مکان قرارگیری کلمه در متن: اهمیت کلماتی که در عنوان متن، زیر عنوان، بدنه متن و یا چکیده متن باشد متفاوت است. می‌توان از موقعیت کلمه برای ارزش دهی به کلمه استفاده کرد.
- مفهوم هر کلمه، که بیانگر ارتباط کلمه با کلمه‌های دیگر است (کلمات مترادف و متضاد).
- کاربرد خاص کلمه: مثلاً، اسامی در سیستمی که به دنبال اسامی خاص می‌گردد دارای اهمیت بیش‌تر است.

در بسیاری از روش‌های معمول، برای استخراج کلمات کلیدی از وزن‌دهی به کلمات بر اساس معیار فراوانی کلمات در متن استفاده می‌شود. فراوانی کلمات نیز به دو صورت زیر در اسناد بررسی می‌شود:

- فراوانی مطلق (Absolute Frequency)
- فراوانی نسبی (Relative Frtequency)

در فراوانی مطلق، فقط تعداد تکرار کلمه در یک سند سنجیده می‌شود ولی در فراوانی نسبی، تعداد تکرار کلمه در یک سند به همراه تکرار سایر کلمات در آن سند و تعداد تکرار کلمه در سایر اسناد مورد ارزیابی قرار می‌گیرد.

برای ارزیابی کلمات کلیدی استخراج شده از متن که از آستانه تعیین شده برای وزندهی عبور می‌کنند، باید معیارهای زیر را در نظر داشت:

جامعیت (Exhaustivity)

بیانگر میزانی است که همه کلمات متن در استخراج کلمات کلیدی ظاهر شده‌اند. در واقع هر چه کلمات بیش‌تری از متن در استخراج کلمات کلیدی به کار روند، میزان جامعیت کلمات کلیدی و نیز نسبت آیتم‌هایی که با آن می‌توانند بازیابی شوند زیاد خواهد بود.

تعیین کنندگی (Specifity)

یعنی هر کلمه‌ی کلیدی تا چه حد دقیق، متن‌های مربوط را مشخص می‌کند. کلمه کلیدی که دارای سطح بالایی از تعیین‌کنندگی است، موارد نامربوط را به کلمات به کار رفته در آن نگاشت نمی‌کند.

پارامترهای وزن‌دهی به کلمات زیاد می‌باشند که ما در زیر برخی از آن‌ها را برمی‌شماریم.

2-5-5 پارامتر *tf.idf*

یکی از پرکاربردترین روابط در حوزه بازیابی اطلاعات، پارامتر *tf.idf* می‌باشد [4]، که از حاصل‌ضرب فراوانی کلمه در فراوانی معکوس سند به دست می‌آید. این روش یک روش مبتنی بر چند سند می‌باشد، که در آن منظور از فراوانی کلمه، فقط تعداد تکرار کلمه در یک سند خاص است. هم‌چنین منظور از فراوانی معکوس سند، تعداد اسنادی است که این کلمه خاص در آن اسناد ظاهر شده است. دلیل مقبولیت این روش نسبت به سایر روش‌ها را می‌توان با توجه به سهولت در استفاده از این روش، محاسبات کم و نتایج قابل قبول دانست [7].

3-5-5 پارامتر سیگنال و نویز

در این روش از تئوری اطلاعات استفاده شده است [4]. در این تئوری، هر چه احتمال رخداد کلمه بیش‌تر باشد، بار اطلاعاتی کم‌تری برای آن در نظر می‌گیرند. کلمات با اهمیت که دارای توزیع متمرکز هستند، یعنی تنها در بعضی از اسناد متنی ظاهر شده اند میزان نویز کمی دارند.

4-5-5 پارامتر مقدار تمایز

در این روش، برای وزن‌دهی کلمات از قدرت تمییزدهندگی کلمات بین اسناد مختلف استفاده می‌شود [4] و [7]. مقدار تمایز (Discrimination Value) را با استفاده از معیارهای مشابهت محاسبه می‌کنند. استفاده از کلمه‌ای از سند به عنوان کلمه‌ی کلیدی که باعث کاهش مشابهت این سند با سایر اسناد می‌شود. هر چه مقدار تمایز بیش‌تر باشد، بیانگر تخصصی‌تر بودن این کلمه و اهمیت بیش‌تر آن در متمایز کردن سندی که در آن ظاهر شده، از سایر اسناد است. در واقع انتخاب کلمه‌ای از یک سند با مقدار تمایز زیاد به عنوان کلمه کلیدی، باعث کاهش شباهت این سند با سایر اسناد می‌شود. برای تعریف شباهت بین دو سند متنی از معیارهای مشابهت استفاده می‌شود.

5-5-5 وزن‌دهی در یک نمایه‌ساز فارسی

در [8] یک نمایه‌سازی متون فارسی معرفی شده است. در اولین گام کلمات عمومی بر اساس یک لیست از قبل آماده شده حذف می‌شوند. برای کلمات عمومی یک لیست 180 کلمه‌ای بر اساس تعداد تکرار کلمه در سند ایجاد شده است. برای ریشه‌یابی کلمات از یک روش مبتنی بر حذف پس‌وند و پیش‌وند استفاده کرده اند. نمایه‌ساز سینا از چهار روش وزن‌دهی tfidf , Lnu , ltn, ntc استفاده کرده است. برای پیکره از 450 متن شامل چکیده مقالات مرتبط با کامپیوتر استفاده شده است. الگوهایی که برای وزن‌دهی به واژگان در نمایه‌ساز *سینا* [8] پیاده‌سازی شده‌اند در جدول 1 آمده‌اند.

نام روش	الگوی وزن‌دهی
tf.idf	
Lnu	
ltn	
ntc	

جدول 1. الگوهای وزن‌دهی واژگان

پارامترهای به کار رفته در جدول به شرح زیر می‌باشند:

N: تعداد کل سندها می‌باشد.

idf: معکوس تعداد اسنادی است که کلمه در آنها به کار رفته است.

NUT: تعداد واژه‌های واحد در سند می‌باشد.

Slope: شیب منحنی در نرمالسازی اسناد مجموعه است.

Pivot: میانگین تعداد واژه‌های واحد در مجموعه مستندات می‌باشد.

با مقایسه‌ای که در استفاده از روشهای مختلف وزن‌دهی و نتایج از حاصل از آن در [8] انجام شد، مشخص گردید که دو روش *Lnu* و *tf.idf* سایر الگوها مناسب تر عمل می‌کند. دلیل بهینه‌تر بودن این دو روش تاثیر پارامترهای سایر واژه‌های سند در وزن‌دهی به یک واژه است. به عبارت دیگر وزن هر واژه علاوه بر پارامترهای مربوط به آن نسبت به سایر واژه های موجود در سند وزن‌دهی می‌شود. برای ارزیابی از معیارهای بازخوانی و دقت استفاده کرده‌اند. میانگین پارامتر دقت با ریشه‌یابی 66% و بدون ریشه‌یابی 54% بوده است.

6. ریشه‌یابی در فارسی

در این بخش به ریشه‌یابی یا «ریخت‌شناسی» کلمات در زبان فارسی می‌پردازیم. ریخت‌شناسی بخشی از علم پردازش زبان طبیعی است که به ساختارهای کلمات و ریشه‌یابی واژگان می‌پردازد. به عمل بیرون‌آوردن ریشه اصلی یک واژه؛ ریشه‌یابی (stemming) گویند. در واقع ریخت‌شناسی به علم شناختن اجزای معنی‌دار از یک واژه گویند که آن واژه را می‌سازد؛ به این اجزای معنی‌دار تکواژ (morpheme) گویند.

در ریخت‌شناسی، واژه‌ها به دو طریق بسط می‌یابند: تصریف (inflection) و اشتقاق (derivation). در تصریف، از ترکیب یک واژه با اجزای دستوری دیگر، واژه‌ای جدید در همان نوع و ردهٔ واژهٔ قبلی ایجاد می‌گردد. به عنوان مثال علامت جمع «ها» در فارسی که با اضافه‌کردنش به هر اسمی یک اسم جدید به وجود می‌آید؛ مثلاً واژهٔ «کتاب» با اضافه‌شدن «ها» به «کتاب‌ها» تبدیل می‌شود که در این صورت، هم کتاب از نوع دستوری اسم است و هم کتاب‌ها. روش دوم، روش اشتقاق است. در اشتقاق با افزودن یک جز دستوری به یک واژه، یک واژه در رده جدیدی به وجود می‌آید. به عنوان مثال اگر تکواژِ «-ِش» را به واژهٔ مصدری «کن» اضافه کنیم، واژهٔ کنش به وجود می‌آید که واژه جدید دیگر از نوع مصدر نیست و یک اسم است. در ریشه‌یابی موتورهای جستجو ما فقط به ریشه‌یابی تصریفی می‌پردازیم.

6-1 طبقه‌بندی روش‌های ریشه‌یابی

الگویتم‌های گوناگون زیادی برای ریشه‌یابی در زبان‌های مختلف از جمله زبان فارسی برای ریشه‌یابی داده شده است. این الگوریتم‌ها را می‌توانیم با توجه به نحوه عملکرد و میزان دقت آنها در دسته‌های جداگانه طبقه‌بندی کنیم. این دسته‌ها را در ادامه بیان می‌کنیم.

6-1-1 ریشه‌یاب جدولی

ساده‌ترین روشی که برای ریشه‌یابی به نظر می‌رسد، نگهداری ریشه هر واژه در یک جدول است. در این روش با جستجوی واژه در این جدول، ریشه واژه مشخص می گردد. هر چند از این روش می توان نتایج خوبی گرفت، اما نگهداری این جدول سربار زیادی برای سیستم خواهد داشت و تنها محدود به کلمات از پیش تعیین شده هستیم.

ساده‌ترین روش از جنبه پیاده‌سازی در بین ریشه یا بن‌ها است. در این روش ریشه کلمات در یک جدول نگهداری می‌شود. و برای یافتن ریشه، کلمه مورد نظر را از جدول جست و جو کرده و ریشه متناظر را مشخص می‌کند. در واقع این روش بیش‌تر شبیه یک عمل جست و جو است تا ریشه‌یابی. ریشه‌یاب جدولی بهترین نتایج را در بین ریشه‌یاب‌ها دارند. ولی از معایب آن سربار زیاد برای نگهداری جدول و هم‌چنین در دسترس نبودن این جدول برای واژگان فارسی می‌باشد.

6-1-2 ریشه‌یابی بر اساس الگوریتم پورتر

روش پورتر (Porter) یک روش توانمند و در عین حال یکی از قدیمی‌ترین روش‌های ریشه‌یابی در زبان انگلیسی است. این روش بر پایه زبان‌شناسی و دسته‌بندی کلمه‌ها به کمک واج‌ها و هجاها بنا نهاده شده است. پس از آن وندهای کلمات درون گردایه به طور خودکار برداشته می‌شوند [13, 14].

دسته‌ای دیگر از کارها، شامل الگوریتم‌های ریشه‌یابی هستند که بر اساس قوانین ریخت‌شناسی زبان مربوطه کار می‌کنند [15-17]. در این الگوریتم‌ها، برنامه از درون ساختاری تصمیم‌گیرنده مانند یک فلوچارت عبور کرده و با افزودن و کاستن وندها با رعایت قواعد املایی و دستوری، سعی در یافتن ریشه کلمات یا بطور خاص افعال دارد. این کارها عمدتا مشابه الگوریتم پورتر هستند که برای زبان انگلیسی طراحی شده است. مشکل این دسته از الگوریتم‌ها برای کلمات جدا از هم است. با توجه به اینکه در فارسی مرز دقیق کلمات مشخص نیست، برای کلمات چندپاره این روش‌ها خوب عمل نمی‌کنند. در [18] یک سیستم ساده مبتنی بر قانون برای افعال فارسی ارایه شده است.

این نوع ریشه‌یاب‌ها با حذف پیشوند و یا پسوند به ریشه واژه می‌رسند. الگوریتم این ریشه‌یاب‌ها از تعدادی قوانین تشکیل می‌شود که با یافتن اولین قانون مناسب و ممکن برای حذف پیشوند و یا پسوند، آن قانون مورد استفاده قرار می‌گیرد. اکثر ریشه‌یاب‌های موجود از این نوع، اقدام به زدودن بزرگترین

دنباله ممکن از حروف واژه بر طبق قوانین می‌نماید. این فرایند آنقدر ادامه می یابد تا هیچ حرف دیگری نتواند زدوده شود.

زبان فارسی برای اشتقاق و ساخت کلمات از الحاق پسوندها و پیشوندها استفاده می‌کند. بنابراین ریشه‌یابی در زبان فارسی فرایند حذف این الحاقات است. از طرفی متأسفانه قانون مدون و کلی برای ساخت واژگان اشتقاقی زبان فارسی وجود ندارد و در مورد هر پسوند و پیشوند استثناهای زیادی یافت می‌شود که کار ریشه‌یابی را بس مشکل می‌کند. بنابراین در مورد هر قانون باید استثنائات آن را شناسایی و نگهداری کرد، تا دقت ریشه یاب خودکار بهبود یابد.

6-1-3 ریشه‌یابی بر اساس مدل حالت متناهی

ریخت‌شناسی بر اساس مدل حالت-متناهی، روش متداولی است که در [19] و [20] نمونه‌ای از آن را می‌توان دید. اساس کار آنها بر یک مدل زبان جهانی است که در [21] ارایه شده است. تعریف الگو بر اساس عبارات منظم انجام می‌شود و پیاده سازی آن بر اساس مدل ماشین حالت-متناهی است. طراحی پردازشگر ریخت‌شناسی حالت-متناهی را می‌توان به دو بخش مجزا تقسیم کرد: بخش مربوط به «طرح زبانی» و بخش مربوط به «طرح رایانه‌ای» [19]. منظور از طرح زبانی، ارایه توصیف نظری جامع و کامل و مانع از ریخت‌شناسی افعال در زبان فارسی است. ارایه توصیف جامع از ریخت‌شناسی در زبان فارسی به گونه‌ای که قابل کاربرد در برنامه‌های رایانه‌ای باشد، نخستین گام جهت طراحی برنامه‌های کاربردی است. این توصیف می‌بایست تمام صورت‌های تصریفی فعل در زبان فارسی را ارایه دهد. بخش دوم، طرح رایانه‌ای است. در این بخش نیازهای سخت‌افزاری و نرم‌افزاری پردازشگر تعریف می‌شود، طرح زبانی پیاده‌سازی شده و ویژگی‌ها و ساختار داخلی فایلهای برنامه تشریح می‌گردد.

6-1-4 ریشه‌یابی به کمک روش‌های آماری

در این دسته از روش‌ها یک گردایه‌ی بزرگ از کلمه‌ها با ساخت‌های گوناگون گردآوری می‌شود. هرچه این گردایه بزرگ‌تر و کامل‌تر باشد این ریشه‌یاب‌ها بهتر کار می‌کنند. در این روش تحلیل آماری به کار گرفته می‌شود. با روش آماری وندهایی که در کلمه‌ها تکرار شده اند، شناسایی می‌گردند. این روش به زبان بستگی ندارد و این بزرگ‌ترین برتری این روش می‌باشد. در بیش‌تر زبان‌های هند و اروپایی، اغلب

بر پایه‌ی وند اشتقاق انجام می‌شود. اگر این روش بتواند برای زبان انگلیسی پاسخ شایسته‌ای بدهد؛ گسترش آن به دیگران زبان‌های دسته‌ی هند و اروپایی ساده خواهد بود. این روش با سه مشکل بزرگ روبروست:

- در این روش به یک گردایه‌ی بزرگ از کلمه‌ها نیاز است. این گردایه باید کامل باشد و کلمات درون آن نیز درست باشند. وجود کلمات نادرست در گردایه بر کارایی این ریشه‌یاب اثر بسیار بد می‌گذارد و آن را گمراه می‌کند. گردآوری گردایه‌ی بزرگی از کلمات صد در صد درست فارسی نیز، ناممکن می‌نماید.
- هنوز این روش‌ها در حال آزمایش هستند و کارایی آن‌ها چشم‌گیر نیست.
- این روش‌ها نیاز به رایانه‌های با سرعت زیاد و حافظه بزرگ دارند و اجرای برنامه‌های نوشته شده بر پایه‌ی این روش‌ها بسیار زمانبر است. برای اجرای این روش‌ها با رایانه‌های در دسترس باید تعدادی از آن‌ها با هم موازی شوند و شاید برای یک بار اجرا، چند روز زمان گرفته شود. گرچه در پیاده‌سازی این روش‌ها بهتر می‌توان به نیازهای آن‌ها پی برد.

6-2 کارهای انجام‌شده در ریشه‌یابی فارسی

از جمله کارهای انجام شده در زمینه ریشه‌یابی کلمات فارسی می‌توان به پروژه بن [22]، ریشه‌یاب آماری [23] و [24] اشاره نمود.

در [22] یک ریشه‌یاب خاص زبان فارسی طراحی گشته است که به عنوان جزئی از یک موتور بازیابی مورد استفاده قرار می‌گیرد. الگوریتم این ریشه‌یاب شبیه ریشه‌یاب Porter است. اولین قدم الگوریتم پیدا کردن زیر رشته‌ای از لغت ورودی است که در لیست پس‌وندهای فارسی (که از روی گرامر فارسی تهیه شده است) وجود داشته باشد. اگر بیش‌تر از یک پس‌وند برای لغت پیدا شد، الگوریتم طولانی‌ترین پس‌وندی را انتخاب می‌کند که تعداد حروف ریشه (بخش اصلی لغت) را کم‌تر از حد مجاز نکند. (مثلاً در اینجا کم‌ترین تعداد حروف برای ریشه 3 کاراکتر است) مثلاً برای لغت «دستشان» می‌توان دو پس‌وند «ان» و «شان» را دید که «شان» طولانی‌تر است و چون حروف باقی مانده «دست»، 3 حرف یا بیش‌تر هستند،مشکلی برای انتخابش وجود ندارد. در این کار برای تعیین پس‌وند آخر لغت از یک DFA استفاده

شده است که ورودی آن وارون شده‌ی رشته‌ی (کلمه‌ی) ورودی است و همه‌ی حالت‌ها در آن حالت نهایی‌اند.

بن [22]، یک ریشه‌یاب «حذف وند» است. یعنی در هر قدم پس‌وندها یا پیش‌وندهایی را برمی دارد تا به لغت اصلی برسد. دیکشنری بن شامل مصدر و بن مضارع فعل‌هاست. الگوریتم بن به این صورت است که بیش‌ترین کاراکترهای ممکن را از لغت برمی دارد (برمبنای قواعدی) و این کار را آنقدر تکرار می‌کند تا دیگر امکان‌پذیر نباشد. ولی با این روش ریشه‌ی به دست آمده ممکن است صحیح نباشد. مثلاً با برداشتن پس‌وند «ی» از لغت «خانگی»، ریشه‌ی «خانگ» به دست می‌آید. برای حل این مشکل، بن از روش Recoding استفاده می‌کند که تبدیلی به شکل «$AXC \circledR AYC$» است و در آن A و C زمینه تبدیل را مشخص می‌کنند و X رشته‌ی ورودی و Y رشته تغییر یافته است.

ریشه‌یاب طراحی شده در نمایه‌ساز سینا مشابه ریشه‌یاب Porter برای زبان انگلیسی است [25]. هر دو ریشه یاب کلمه را با یک سری پیشوندها و پسوندها در چند مرحله تطابق می‌دهند تا پسوندها و پیشوندها حذف شوند و ریشه کلمه به دست آید. تفاوت این ریشه‌یابها به تفاوت زبان آنها برمی‌گردد. الگوریتم Porter الگوهایی از حروف صدادار و بی‌صدا برای تخمین محتوای اطلاعات مشخص می‌کند. در فارسی بسیاری از حروف صدادار نوشته نمی شوند. لذا ریشه یاب نمی تواند از آنها استفاده کند. در این ریشه‌یاب برای رفع این مشکل از روش تعریف حداقل طول ریشه استفاده کرده‌ایم. تفاوت دیگر این ریشه یاب با ریشه یاب Porter در تشخیص پیشوند است، ریشه‌یاب می‌تواند پیشوندها را مشخص کند در حالیکه ریشه‌یاب Porter الگوریتمی برای تشخیص پیشوند ارائه نداده است.

7. خلاصه

در این تحقیق به ارایه مشکلات و راهکارهای طراحی و ساخت یک موتور جستجوی فارسی پرداختیم. بخش‌هایی از موتور جستجو برای تمام زبان‌ها یکسان هستند و بخش‌هایی دیگر از ساختاز زبان و خط تاثیر می‌پذیرند. ما به بیان کلی از ساختار موتور جستجو پرداختیم. سه بخش اصلی موتور جستجو عبارتند از:

- گردآورنده اسنا
- نمایه گذار
- مدل‌ها و الگوریتم‌های بازیابی

در گردآوری اسناد، مستندات از منابع گوناگون جمع‌آوری شده و به یک قالب یکسان برای ذخیره‌سازی تبدیل می‌گردند. با توجه به مشکلات کدگذاری زبان فارسی باید در همین مرحله برخی از یکسان‌سیازی‌های مربوط به رسم‌الخط و کدگذاری نویسه‌ها انجام شوند.

نمایه‌گذاری شامل مراحل مبسوطی است که در بخش 2 گفته شد. مهمترین آنها ریشه‌یابی و وزن‌دهی کلمات هستند. کارهای متعددی در ریشه‌یابی زبان فارسی انجام شده‌اند.

مدل‌ها و الگوریتم‌های بازیابی برای زبان فارسی باید بررسی و انتخاب شوند. کارایی مدل‌ها برای زبان‌ها و حوزه‌های تکنیکی مختلف متفاوت است که این کار باید برای زبان فارسی انجام شود.

بطور خلاصه زمینه‌هایی که در زبان فارسی نیازمند پویش و بازنگری هستند عبارتند از:

- گدگذاری فارسی
- رسم‌الخط فارسی
- انتخاب مدل بازیابی
- وزن‌دهی
- ریشه‌یابی

مراجع

[1] ISO, "Data processing -- Vocabulary -- Part 1: Fundamental terms," *ISO 2382-1*, 1984.

[2] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*: Cambridge University Press, 2008.

[3] S. Brin, and L. Page, "The anatomy of a large-scale hypertextual Web search engine," *Comput. Netw. ISDN Syst.*, vol. 30, no. 1-7, pp. 107-117, 1998.

[4] R. Baeza-Yates, and B. Ribeiro-Neto, *Modern Information Retrieval*: Addison Wesley, 1999.

[5] I. H. Witten, A. Moffat, and T. C. Bell, *Managing Gigabytes: Compressing and Indexing Documents and Images, Second Edition*: Morgan Kaufmann, 1999.

[6] G. Salton, and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Inf. Process. Manage.*, vol. 24, no. 5, pp. 513-523, 1988.

[7] م. تشکری, "ساخت یک نمایه‌ساز خودکار برای متون فارسی," دانشکده مهندسی کامپیوتر, دانشگاه امیرکبیر, تهران, 1380.

[8] ح. بشیری, ف. کربلایی, و ش. موسوی, "طراحی و ارزیابی نمایه‌ساز خودکار متون فارسی," در یازدهمین کنفرانس بین‌المللی کامپیوتر انجمن کامپیوتر ایران, تهران, 1384.

[9] ل. مرتضایی, "مسایل خط و زبان فارسی در ذخیره‌سازی و بازیابی اطلاعات," فصلنامه اطلاع رسانی, دوره 17, 1380.

[10] ا. هاشمی, واژگان کتابداری و اطلاع رسانی, تهران: دبیرخانه هیئت امنای کتابخانه‌های کشور, 1376.

[11] یوسفیان, صالحی‌زارعی, and مینایی‌بیدگلی, "دشواری ریشه‌یابی فارسی و روشی برای ریشه‌یابی فعل‌های ساده فارسی," در دومین کارگاه پژوهش زبان فارسی و رایانه, تهران, 1385.

[12] دستورخط: فرهنگستان زبان و ادب فارسی, 1380.

[13] J. E. Hopcroft, R. Motwani, and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, 2nd ed.: Addison Wesley, 2000.

[14] T. Winograd, *Language As a Cognitive Process: Syntax*: Addison-Wesley, 1982.

[15] R. Hessami-Fard, and G. Ghasem-Sani, "Stemmer Algorithm Design for Persian Language," in 11th International CSI Computer Conference (CSICC'2006), Tehran, Iran, 2006.

[16] M. I. Mobarakeh, and B. Minaei-Bidgoli, "Verb Detection in Persian Corpus," *International Journal of Digital Content Technology and its Applications* vol. 3, pp. 58-65, 2009.

[17] A. Mokhtaripour, and S. Jahanpour, "Introduction to a new Farsi stemmer," in Proceedings of the 15th ACM international conference on Information and knowledge management, Arlington, Virginia, USA, 2006.

[18] Megerdoomian, and Karine, "Developing a Persian Part-of-Speech Tagger," in First Workshop on Persian Language and Computers, Tehran University, Iran, 2004.

[19] ا. دفتری‌نژاد, "ساختواژه حالت-متناهی: روشی مناسب برای طراحی پردازشگر ساختواژی," در هفتمین همایش زبانشناسی ایران, 1386.

[20] K. Megerdoomian, "Finite-State Morphological Analysis of Persian," in Workshop on Computational Approach to Arabic Script-Based Languages, 2004.

[21] K. Beesley, and L. Karttunen, *Finite State Morphology*: Stanford, CSLI Publications, 2003.

[22] M. Tashakori, M. Meybodi, and F. Oroumchian, "Bon: The Persian stemmer," *Lecture Notes in Computer Science (LNCS), Springer Verlag*, vol. 2510, pp. 487-494, 2002.

[23] م. نصیری, م. ش. اسماعیلی, و ک. ابولحسنی, "یک ریشه‌یاب آماری برای زبان فارسی," در مجموعه مقالات یازدهمین کنفرانس بین‌المللی کامپیوتر, 1384.

[24] K. Taghva, R. Beckley, and M. Sadeh, "A Stemming Algorithm for the Farsi Language," in Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05) - Volume I - Volume 01, 200۵.

[25] M. F. Porter, "An algorithm for suffix stripping," *Program 14(3)*, pp. 130-167, 1980.