عنوان زیرپروژه:

تحلیل سیستم یافتن خودکار کلمات کلیدی متون زبان فارسی

فهرست مطالب

عنوان **شماره صفحه**

عنوان **شماره صفحه**

عنوان **شماره صفحه**

1. مقدمه

جایگاه زبان فارسی در میان زبان‌های دیگر را سه جنبه‌ی وراثتی (تاریخی)، ناحیه‌ای و رده شناختی می‌توان بررسی کرد [1]. از دیدگاه زبان‌شناسی تاریخی فارسی همراه با زبان‌های هند – آریایی، زیر گروه هند – ایرانی را در گروه شرقی زبان‌های هند و اروپایی تشکیل می‌دهند. این زیر گروه شامل زبان‌هایی مانند فارسی، پشتو و کردی می‌باشد. از نظر ناحیه‌ای، به دلیل همسایگی با کشورهای عربی زبان، دارای بسیاری کلمات قرضی و حتی برخی قواعد مشابه با آن‌هاست. فارسی از دیدگاه ویژگی‌های زبانی (رده شناختی)، یک زبان پیوندی و ضمیرانداز است. فارسی از راست به چپ نوشته می‌شود و اگرچه در اصل دارای ترتیب فاعل – مفعول – فعل است ولی مملو از استثنائات مجاز در این ترتیب می‌باشد که حاصل فرایندهایی چون نامکانی، به‌هم ریختگی، حرکت جهت برجسته‌سازی، تاخیر، شکافت و شبه‌شکافت و غیره هستند و به دلیل استفاده‌ی فراوان، عملاً فارسی را به یک زبان بدون ترتیب تبدیل می‌کنند. در فارسی معمولاً فعل در انتها و هسته در ابتدا قرار می‌گیرد. مثلاً در عبارات وصفی موصوف قبل از صفت و در عبارات اضافی مضاف‌الیه قبل از مضاف واقع می‌شوند. البته این وضعیت در مورد صفات پیشین و حروف اضافه برقرار نیست. در این حالات هسته در انتها قرار می‌گیرد. به عبارت دیگر فارسی بهره‌گیر از حروف اضافه پیشین و صفات و اضافات پسین است.

1.1. مسائل و چالش‌های پردازش متن فارسی

در پردازش متون زبان طبیعی با زبان نوشتاری سروکار داریم. این مسأله باعث می‌شود گر چه به جهت از دست دادن اطلاعات گویشی مانند لحن گوینده، آهنگ صدا، تاکید و مکث، با مشکلات و ابهاماتی مواجه شویم، ولی در مقابل با شکل محدودتری از زبان کار می‌کنیم. بسیاری از بی ترتیبی‌های زبان متعلق به زبان گفتاری است و در زبان نوشتاری بیش‌تر قالب‌های دستوری رعایت می‌شوند و لذا تهیه دستور زبان پوشاننده‌ی تمام متن، ساده‌تر است.

در تلاش برای ساخت یک سیستم پردازش و درک متون فارسی با مسائل و مشکلاتی مواجه می‌شویم که بعضی در بیش‌تر زبان‌ها بروز کرده و برخی خاص زبان فارسی می‌باشند. هم‌چنین برخی از این پیچیدگی‌ها به طبیعت زبان و نارسایی‌های قواعد زبان‌شناسی مربوط و برخی دیگر برخاسته از مشکلات ایجاد سیستم‌های هوش مصنوعی است [2]. در این بخش به برخی از این مسائل اشاره می‌کنم.

با توجه به بحث اخیر می‌توان در کل اهم مشکلات فعلی پردازش متون فارسی را در چند دسته زیر خلاصه نمود:

1) عدم وجود منابع زبانی مناسب و کافی برای زبان فارسی مانند واژگان‌های تک زبانه و چند زبانه محاسباتی، واژگان‌های معنایی و متصل به هستان‌شناسی (هستان‌شناسی‌های لغوی)، هستان‌شناسی جامع عمومی و تخصصی، پیکره‌های عمومی و تخصصی ساده یا

برچسب‌خورده (با برچسب‌های اجزاء کلام، کسره‌ی اضافه، نقش‌های موضوعی، مفاهیم و روابط مفهومی و غیره)، مجموعه مدون قوانین ساخت‌واژی و دستوری پوشا، عدم وجود استاندارد(های) شیوه‌ی نگارش، فاصله‌گذاری و رمزگذاری حروف و علائم.

2) مشکل تشخیص مرز کلمات (مسأله شیوه‌های نگارش متفاوت)

3) مشکل تشخیص مرز گروه‌های اسمی (مسأله‌ی کسره‌ی اضافه نامرئی)

4) از دست دادن اطلاعات گویشی

5) مسأله‌ی ابهام

6) افعال مرکب و اصطلاحات

7) مسأله‌ی هم‌نگاره‌ها و تحت آن مسأله‌ی حذف مصوت‌های کوتاه (اعراب) از نوشتار

8) معناشناسی و مشکلات تحلیل معنایی.

1.2. مروری بر موضوع

1.2.1. پردازش لغوی

منظور از پردازش لغوی شناسایی مرز لغات و جملات در یک متن است. این مرز ممکن است به شکل ساده توسط جداکننده‌هایی مانند فاصله، کاما، نقطه، علامت سوال و ... تعیین شود و یا نیاز به پردازش‌های پیچیده‌تر داشته باشد مانند زمانی که میان بخش‌های یک کلمه از فاصله استفاده می‌کنیم (مثل کلمه «می توان») و یا وقتی که دو کلمه مجزا را بدون فاصله و پی در پی می‌نویسیم (مثل عبارت «در برابر باد»). تعیین مرز کلمات در زبان فارسی به دلیل گوناگونی رسم‌الخط و عدم وجود استانداردهای نگارشی و هم‌چنین به دلیل وجود شکل‌های مختلف حروف (اول – وسط – آخر و چسبان و غیرچسبان) بیش از زبان انگلیسی مشکل‌ساز است. این مشکل در زبان انگلیسی تنها برای کلمات مرکب ممکن است رخ دهد که آن‌ها را می‌توان در مراحل بعدی پردازش مثل پردازش نحوی تشخیص داد. اما در زبان فارسی علاوه بر کلمات مرکب که مشکلی مشابه با انگلیسی ایجاد می‌کنند، مرز کلمات غیرمرکب نیز ممکن است بدرستی تشخیص داده نشود. از فعالیت‌های انجام شده در این زمینه می‌توان به [3] اشاره نمود که به تشخیص انتهای کلمات و فاصله‌گذاری میان آن‌ها می‌پردازد. هم‌چنین [4] در مطالعه‌ای به بررسی نحوه‌ی تشخیص کسره‌ی اضافه در متن با استفاده از روش‌های آماری مبتنی بر پیکره‌های زبانی پرداخته است. تشخیص کسره اضافه محذوف کمک بسیاری به حل مشکل ابهام در شناسایی مرز گروه‌های اسمی

می‌نماید. از سوی دیگر در بعضی کارها این مرحله با مراحل دیگر ادغام و یا در طی مراحل دیگر انجام می‌شود. مثلاً در برخی تحلیل‌گر ساخت‌واژی معرفی‌شده در بخش بعد در حین تحلیل ساخت‌واژی، مرز کلمات نیز تعیین و فاصله‌های زائد درون کلمات حذف می‌شوند.

1.2.2. پردازش ساخت‌واژی

مرحله دیگر در پردازش متن تحلیل ساخت‌واژی می‌باشد. این مرحله به تجزیه و ترکیب اجزاء کلمات می‌پردازد. به عبارت دیگر در تحلیل ساخت‌واژی در هنگام درک یا تجزیه متن به تشخیص اجزاء کلمه (تک‌کلمه‌ها) و استخراج ریشه و وندهای متصل به آن و در هنگام تولید متن به ساخت کلمه از روی اجزاء سازنده‌اش توجه داریم. تحلیل ساخت‌واژی بر دو نوع است: تصریفی و اشتقاقی. تحلیل ساخت‌واژی تصریفی به تجزیه کلماتی می‌پردازد که با تصریف ساخته شده‌اند. تصریف افزودن وند به کلمه‌ای برای ساخت کلمه‌ی دیگر است به گونه‌ای که معمولاً منجر به تغییر مقوله (طبقه نحوی) و معنی کلمه نشود مانند صرف افعال برای شخص‌ها و زمان‌های مختلف، جمع بستن یا نکره کردن اسامی، افزودن ضمیر ملکی به اسم یا ضمیرمفعولی به فعل و امثالهم. نوع دوم تحلیل ساخت‌واژی مربوط به ساخت‌واژی اشتقاقی است. در اشتقاق، کلمه‌ی جدید حاصل از افزودن وند با کلمه قبل معمولاً از جهت مقوله نحوی و معنا متفاوت می‌شود. این بخش گستره‌ی وسیعی از ساخت‌واژی فارسی را می‌پوشاند و تاکنون نه مجموعه قواعد ساخت‌واژی کاملی برای آن تدوین شده و نه تحلیل‌گر جامعی برای پوشش این گستره ایجاد گشته است.

زبان فارسی از جهت ساخت‌واژی بخصوص از دیدگاه تصریفی زبانی قانون‌مند و ساخت یافته است. در تولید تحلیل‌گر برای یک زبان دو مسأله وجود دارد (1)؛ تهیه مجموعه مدونی از قواعد تحلیل (2) طراحی و ساخت تحلیل‌گری که با استفاده از این مجموعه قواعد قادر به تحلیل عناصر زبان باشد. در بسیاری موارد الگوریتم‌ها و روش‌های تحلیل، مستقل از زبان هستند و یا با تغییرات اندک قابل انطباق با زبان خاص می‌باشند. در این حالت مسأله‌ی اصلی، پیاده‌سازی این الگوریتم‌های شناخته شده، بهبود کارایی آن‌ها و افزودن قواعد وابسته به زبان به آن‌هاست. این نکته برای تحلیل ساخت‌واژی به تهیه قواعد ساخت‌واژی تصریفی و اشتقاقی زبان و ایجاد الگوریتم‌هایی که بتوانند بر اساس قواعد فوق کلمات زبان را تجزیه و تحلیل کنند تبدیل می‌شود. تاکنون تحلیل‌گرهای ساخت‌واژی تصریفی مختلفی برای زبان فارسی ساخته شده‌اند که عمدتاً در بخش تجزیه و برای استخراج اجزاء کلمات کار می‌کنند. از جمله می‌توان به تحلیل‌گر ساخت‌واژی ساخته شده در پروژه‌ی مترجم شیراز [5] و تحلیل‌گرهای معرفی شده در، اشاره نمود. در تحلیل‌گر شیراز مسأله‌ی تصریف در زبان فارسی تا حد زیادی حل شده است. اما نرم‌افزار آن به صورت آزاد در دسترس محققین برای استفاده قرار ندارد. از سوی دیگر برخی تحلیل‌گرهای عمومی برای زبان انگلیسی ایجاد و آزموده شده اند که قابل انطباق برای زبان فارسی نیز هستند. به عنوان نمونه [6] با جمع‌آوری مجموعه‌ی قواعد ساخت‌واژی تصریفی (و چند نمونه برای ساخت‌واژی اشتقاقی) تحلیل‌گر Ampel را برای تحلیل کلمات زبان فارسی منطبق ساخته است. کار [6] تمام موارد لحاظ شده در پروژه شیراز به علاوه چند نکته جدید از جمله در نظرگرفتن «ه صامت» و هم‌چنین افعال متصل را دربردارد.

همانطور که گفته شد بیش‌تر فعالیت‌های انجام شده در زمینه ساخت تحلیل‌گرهای ساخت‌واژی بر تصریف متکی بوده اند. علت این امر آن است که قوانین تصریف محدود و تعریف‌شده هستند ولی قوانین اشتقاق بسیار زیاد، غیر مدون و مملو از استثنائات و حالات خاص می‌باشند. به عبارت دیگر گلوگاه اصلی ایجاد یک سیستم تحلیل‌گر ساخت‌واژی جامع اکتساب وتهیه مجموعه مدون قوانین ساخت‌واژی زبان است و نه طراحی و پیاده‌سازی الگوریتم‌های تحلیل ساخت‌واژی. به دلیل کثرت، تنوع و استثناپذیری این قوانین، اکتساب دستی آن‌ها کاری زمان‌بر پرهزینه است. به همین جهت دسته دیگری از فعالیت‌های تحقیقاتی بر اکتساب خودکار آن‌ها متمرکز شده است. از جمله این تحقیقات می‌توان به [7] اشاره نمود که در آن واژک‌های زبان فارسی به صورت بدون نظارت از روی پیکره‌هایی از لغات فارسی استخراج می‌شوند. در این کار سیستم با دیدن لغات مختلف نحوه‌ی شکستن لغات به اجزاء سازنده را به تدریج می‌آموزد و می‌تواند بدون داشتن قوانین صریح عمل تجزیه ساخت‌واژی تصریفی و اشتقاقی را انجام دهد. برای اطلاعات بیش‌تر می‌توانید به فصل ریشه‌یابی مراجعه کنید.

1.2.3. تهیه منابع زبانی

یکی از گلوگاه‌های پردازش زبان فارسی در دسترس نبودن منابع زبانی کافی و معتبر برای فارسی است. منابع مورد نیاز شامل واژگان محاسباتی، دستور زبان محاسباتی، پیکره‌های خام و برچسب خورده، هستان‌شناسی‌های عمومی و تخصصی، قواعد صرفی و الگوهای

معنـایی مـی‌باشـد. در رابطـه بـا تهیـه ایـن منـابع نیـز فعالیـت‌هـایی صـورت گرفتـه اسـت. اکثـر تلاش‌های صورت گرفتـه بـرای سـاخت واژگـان محاسـباتی ماننـد [8] بـه طراحـی سـاختار واژگـان و تهیـه مجموعـه محـدودی از اطلاعـات واژی در یـک سـاختار تعریـف شـده انجامیـده‌انـد. هـدف این فعالیت‌ها بـیش‌تـر تحقیـق بـر مبـانی نظـری تهیـه واژگـان‌هـای محاسـباتی و طراحـی سـاختار مناسب برای آن‌ها بـوده اسـت و در نهایـت محصـول خاصـی کـه قابـل اسـتفاده بـرای عمـوم باشـد به دست نیامده است [10] در کار خـود عـلاوه بـر طراحـی سـاختار، بـه ورود دانـش مـورد نیـاز در ایـن سـاختار نیـز پرداختـه اسـت. ایـن واژگـان قـرار اسـت بـر روی وب و بـه صـورت آزاد بـرای استفاده در فعالیت‌های دیگر ارائه شود.

پیکره‌های متنی منابع مهم بعـدی هسـتند کـه بـه صـورت خـام و برچسـب خـورده مـورد اسـتفاده قرار می‌گیرنـد. در [11] یکـی از پیکـره‌هـا معرفـی شـده اسـت. هـم‌چنـین پیکـره همشـهری [12] پیکـره دیگـری حـاوی 345 مگابایـت ازاخبـار و مقـالات روزنامـه همشـهری مـی‌باشـد. در [13] پیکره‌ی « محک» معرفـی شـده اسـت. ایـن مجموعـه کـه از خبرگـزاری‌هـا جمـع‌آوری شـده اسـت شـامل اخبـار و مقـالاتی در انـدازه‌ی نـیم صـفحه تـا چنـدین صـفحه مـی‌باشـد. « محـک» شـامل 3007 سـند، 216 پـرس و جـو در مـورد آن‌هـا و لیسـت اسـناد مـرتبط بـا ایـن پـرس و جوهـا مـی‌باشـد. از منـابع تهیـه شـده دیگـر مـی‌تـوان بـه لیسـت کلمـات غیرمفیـد فارسـی [14] اشـاره نمود.

1.3. فعالیت‌های کاربردی

در بخش‌های قبل به فعالیت‌های پایه‌ای اشاره شد که در کاربردهای بزرگ و واقعی پردازش زبان‌های طبیعی مورد استفاده قرار می‌گیرند. این بخش به بررسی فعالیت‌های کاربردی انجام شده مرتبط با پردازش متون فارسی می‌پردازد.

از فعالیت‌های مرتبط با پردازش متون فارسی می‌توان به خلاصه‌سازی متون فارسی [15]، سیستم‌های پرسش و پاسخ به زبان فارسی [16] ، پیش بینی رایانه‌ای کلمه [17]، ویرایش ادبی جملات فارسی [18]، اعراب گذاری متون فارسی [19]، استخراج اطلاعات از مستندات متنی [20] و کارهای دیگری به تفصیل اشاره شده است.

2. بازیابی اطلاعات و استخراج کلمات کلیدی

2.1. مقدمه

در دنیای کنونی این کمبود اطلاعات نیست که مسأله است بلکه کمبود دانشی است که از این اطلاعات می‌توان حاصل کرد . میلیون‌ها صفحه‌ی وب، میلیون‌ها کلمه در کتابخانه‌های دیجیتال و هزاران صفحه اطلاعات در هر شرکت تنها چند دست از این منابع اطلاعاتی هستند. اما نمی‌توان به طور مشخص منبعی از دانش را در این بین معرفی کرد. دانش خلاصه‌ی اطلاعات است و نیز نتیجه گیری و حاصل فکر و تحلیل بر روی اطلاعات.

2.2. بازیابی اطلاعات

در واقع بیش‌تر دانش ما اگر به صورت غیردیجیتال نباشند، کاملاً غیر ساخت‌یافته‌اند. کتابخانه‌های دیجیتال، اخبار، کتابهای الکترونیکی، بسیاری از مدارک مالی، مقالات علمی و تقریباً هر چیزی که شما می‌توانید در داخل وب بیابید، ساخت‌یافته نیستند. در نتیجه ما نمی‌توانیم آموزه‌های داده‌کاوی را در مورد آن‌ها به طور مستقیم استفاده کنیم.

با این حال، سه روش اساسی در مواجهه با این حجم وسیع از اطلاعات غیر ساخت‌یافته گسترده شده در جهان وجود دارد. بازیابی اطلاعات[1]، استخراج اطلاعات[2] و کشف دانش در متن، این سه روش برخورد با این مسأله هستند.

بازیابی اطلاعات اصولاً مرتبط است با بازیابی مستندات و مدارک. کار معمول در بازیابی اطلاعات این است که بسته به نیاز مطرح شده از سوی کاربر، مرتبط‌ترین متون و مستندات را از میان دیگر مستندات یک مجموعه بیرون بکشد. این یافتن دانش نیست بلکه تنها آن بقچه‌ای از کلمات را که به نظرش مرتبط‌تر به نیاز اطلاعاتی جست و جوگر است را به او تحویل می‌دهد. در بازیابی سند، اطلاعات یک زیرمجموعه از اسناد هستند که در ظاهر مرتبط با پرس و جو می‌باشد. تمام روش‌های جست و جو مبتنی بر مقایسه بین پرس و جو و سند ذخیره شده می‌باشند. بعضی مواقع این مقایسه به صورت غیر مستقیم و با مقایسه پرس و جو با کلمات کلیدی انجام می‌گیرد.

یک پیکره متنی (c) از اسناد (d) و یک پرس وجو (q) را در نظر بگیرید، هدف بازیابی اطلاعات، استخراج و برگرداندن اسنادی است که بهترین ارضاء کننده پرس وجو باشند [21]. فرض کنید که (r) سند از (p) سند بازیابی شده توسط سیستم بازیابی اطلاعات با پرس وجو مرتبط هستند و کل اسناد مرتبط به پرس وجو q در پیکره (c) به تعداد R می‌باشد به طور نمونه، سیستم بازیابی اطلاعات یک لیستی از اسناد مرتبط را که با توجه به

[1] Information Retrieval

[2] Information Extraction

معیارهای ارزیابی خاصی بر اساس میزان ارتباط یا احتمال مرتب شده اند را بر می‌گرداند. امکان دارد سیستم کل اسنادی را که معیار ارزیابی شده برای آن‌ها از یک آستآن‌های بالاتر رود را بازیابی کند. P را آستانه در نظر می‌گیریم. بازخوانی با r تعریف می‌شود ولی دقت با r/p تعریف می‌شود. هدف پیدا کردن الگوریتمی است که دارای دقت بالا یا باز خوانی بالا یا هر موازنه‌ای بین این دو باشد.

بازیابی اطلاعات بر پایه رشته‌های مختلف همچون، علوم کامپیوتر، ریاضیات، علوم کتابداری، علوم اطلاعات، علوم شناختی ، زبان‌شناسی ، آمار و غیره است.

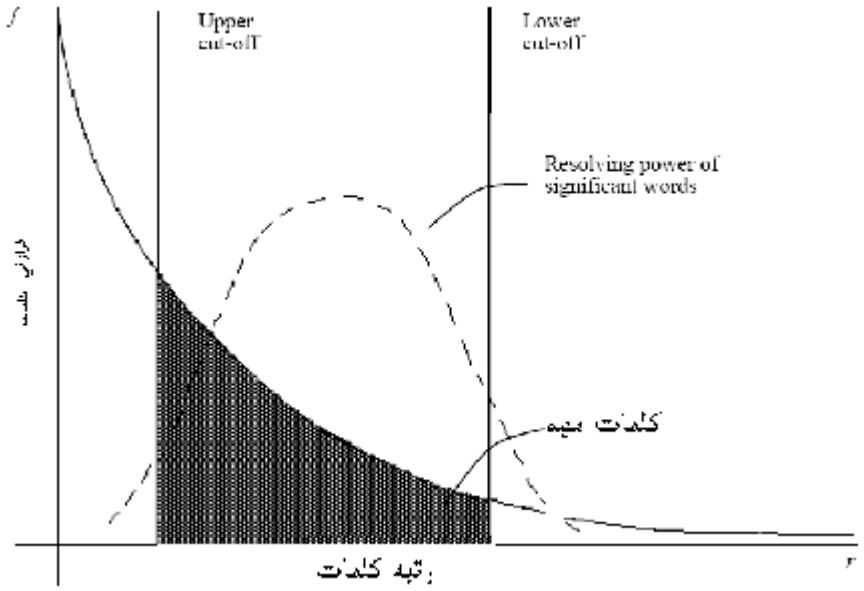
بازیابی اطلاعات اتوماتیک به منظور کاهش سربار اطلاعاتی استفاده می‌شود بسیاری از دانشگاه‌ها وکتابخانه‌های عمومی از سیستم بازیابی اطلاعات برای فراهم کردن دسترسی به کتابها، مجلات وسایر اسناد استفاده میکنند. درموتورهای جست و جو همانند yahoo , Google یا live search کاربردهای بازیابی اطلاعات بیش‌تر به چشم می‌خورد.

2.3. تئوری لان

فراوانی رخداد کلمات در یک مقاله معیار مهمی برای میزان اهمیت کلمه، فراهم می‌کند. او هم‌چنین پیشنهاد کرده است، که وضعیت نسبی دریک جمله از کلمات با درجه اهمیت معین، معیار خوبی برای تعیین اهمیت جمله‌ها فراهم می‌کند. معیار اهمیت یک جمله بر اساس ترکیب این دو معیار خواهد بود.

عقیده لان این بوده است که کلمات تکرارشونده می‌توانند برای استخراج کلمات و جملات تشریح کننده سند به کار روند.

f را فراوانی رخداد کلمات مختلف یک متن و r را رتبه آن‌ها در نظر بگیرید، سپس نمودار f,r را رسم کنید یک نمودار شبیه منحنی هزلولی همچون شکل 2-1 خواهد بود:



شکل 2-1. سهمی ارتباط بین فراوانی و رتبه کلمات در متون

در واقع این نمودار قانون zipf [4] را اثبات می‌کند که در آن حاصلضرب فراوانی کلمات در رتبه آن‌ها طبق رابطه (2-1)، تقریبا ثابت است.

$$F * r \xrightarrow{\approx} cons\tan t \qquad (1\text{-}2)$$

لان با اعمال یک آستانه‌گیر بالا و پایین بر روی این منحنی، توانست کلمات کم‌اهمیت را جدا کند.کلماتی که از آستانه‌ی بالا تجاوز کرده باشند، به عنوان کلمات عمومی درنظر گرفته می‌شوند، و کلماتی که ازآستانه پایین کم‌تر باشند، کلمات نادر و کمیاب نامیده می‌شوند و در محتویات مقاله یا سند شرکت ندارند. او هم‌چنین از روش‌های شمارشی برای تعیین اهمیت کلمات استفاده کرده همساز با این فرض و مبنی بر قدرت تمایز کلمات با اهمیت، همان توانایی در تفکیک محتوی می‌باشد.

قدرت تمایز کلمات که در بین دو آستانه به حداکثر می‌رسد و با دور شدن از نقاط انقطاع این مقدار به صفر نزدیک می‌شود. برای مشخص کردن آستانه‌ها از آزمایش و خطا استفاده شده است و هیچ پیش گویی نمی‌توان در آن مورد انجام داد. جالب توجه است که این ایده‌ها، اساس اکثر کارهای بعدی بازیابی اصطلاحات شده است . لان خودش نیز از آن برای ایجاد یک چکیده نویس خودکار استفاده کرده است.

فرض کنید که میخواهیم یک سیستم بازیابی اطلاعات ایجاد کنیم برای هر سیستم سه بخش وجود دارد:

1- حذف کلمات پرتکرار

2- حذف پس‌وندها از کلمات

3- تشخیص ریشه‌های معادل

حذف کلمات پرتکرار، یا همان کلمات عمومی[1] یک روش برای اعمال آستانه بالا برای ایده‌ی لان می‌باشد. این عمل با یک مقایسه‌ی ساده کلمات با یک لیست از قبل آماده انجام می‌گیرد. لیست‌های گوناگونی با توجه به کاربردها ارائه شده است با حذف کلمات عمومی علاوه بر این که این کلمات درحین پردازش در نظر گرفته نمی‌شوند بلکه اندازه کل فایلها بین 30 تا 50 درصد کاهش پیدا می‌کند.

گام دوم کمی پیچیده‌تر است، یک روش استاندارد برای حذف پس‌وندها استفاده از یک لیست از قبل مهیا شده می‌باشد که طولانی‌ترین پس‌وند ممکن از کلمه حذف می‌شود.

فرض در بازیابی اطلاعات این است که دو کلمه با یک ریشه‌ی مشابه مفهوم مشترکی دارند باید مثل هم شاخص شوند. بدیهی است که این فرض مشکلاتی نیز داشته باشد. برای مثال دو کلمه NEUTRON, NEUTRALISE دارای ریشه مشابهی هستند ولی ما نیاز داریم که این دو کلمه متمایز از هم باشند. ولی روشی راحت و ارزان برای این کار وجود ندارد ما به ناچار این خطاها را تحمل می‌کنیم.

[1] Stopword

2.4. قانون ZIPF

اصل کم‌ترین تلاش[1] دلیل می‌آورد که مردم به نحوی عمل می‌کنند که نرخ متوسط کار محتمل‌شان کمینه شود[23]. اگر ما، تعداد رخداد هر کلمه از زبان را دریک متنی بزرگ به دست بیاوریم و آن‌ها را براساس فراوانی رخداد آن‌ها مرتب کرده و در یک لیست قرار دهیم می‌توانیم یک رابطه‌ای بین فراوانی کلمات (F) و موقعیت آن‌ها در لیست موسوم به رتبه (r) به دست بیاوریم. قانون zipf نشان میدهد که :

$$fa\ \frac{1}{r} \qquad (2\text{-}2)$$

یا به عبارت دیگر یک ثابت k وجود دارد که

$$F.r = k \qquad (3\text{-}2)$$

برای مثال،کلمه در رتبه 50 باید سه برابر کلمه‌ای که در رتبه 150 قرار دارد تکرار شده باشد. این تئوری با نام zipf شناخته می‌شود.

[1] Least Effort

جدول 2-1. نتیجه عملی از قانون zipf را روی پیکره Tom sawyer

word	Freq	Rank	f*r
The	3332	1	3332
And	2972	2	5944
A	1775	3	5235
He	877	40	8770
But	410	20	8400
Be	224	30	8820
There	222	40	8880
One	172	50	8600
about	158	60	9480

قانون zipf نظر به این که یک تشریح از توزیع فراوانی کلمات در زبان‌های انسانی است مفید می‌باشد، تعداد کمی کلمات عمومی، تعداد متوسطی کلمات با فراوانی متوسط و تعداد زیادی کلمه با فراوانی کم وجود دارد. zipf در این اهمیت زیادی مشاهده کرده است.

با توجه به تئوری zipf هم گوینده و هم شنونده سعی می‌کنند که تلاش خود را کمینه کنند. گوینده تلاش بر کوچک نگه‌داشتن کلمه‌نامه کلمات عمومی دارد و تلاش شنونده این است که کلمه‌نامه کلمات نادر و کمیاب را بزرگ کند تا پیام کم‌ترین ابهام را داشته باشد بیش‌ترین مصالحه بین این رقابت، نیاز به نوعی رابطه‌ی دوطرفه بین فراوانی و رتبه دارد که آن هم در قانون zipf آورده شده است برای ما نتیجه اصلی از قانون zipf مسأله عملی است که برای اکثر کلمات داده‌های ما، استفاده به صورت پراکنده خواهد بود فقط برای تعداد کمی از کلمات ما مثال‌های زیادی داریم و فراوان استفاده می‌شوند.

2.5. کلمات کلیدی

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.

کلمات کلیدی در دو گروه طبقه‌بندی می‌شوند [24].

کلمات کلیدی تابعی[1]

[1] Functional

کلمات کلیدی آموزنده[1]

کلمات کلیدی تابعی یا غیرمفید برای کلمات دستوری یا مرتبط با زبان استفاده می‌شوند و ارتباط کمی با محتویات سند دارد. این نوع از کلمات کلیدی باید حذف شوند و در نظر گرفته نشوند. کلمات کلیدی آموزنده ارتباط خیلی قوی با محتویات و معنی متن سند دارد . مرز بین کلمات کلیدی تابعی و آموزنده خیلی واضح و سخت نیست و ما می‌توانیم یک مرز فازی را برای آن‌ها در نظر بگیریم.

هر چند کلمات کلیدی نقش بسیار مهمی در سیستم بازیابی اطلاعات[2] (IR) و برنامه‌های کاربردی دارند، استخراج کلمات کلیدی موثر یک کار زمان‌بر و مبتنی بر پردازش انسان می‌باشد. اخیراً استخراج اتوماتیک کلمات کلیدی زمینه جالبی در تحقیقات text mining و IR به وجود آورده است. بعضی از موضوعات مرتبط با استخراج کلمات کلیدی به شرح زیر است.

1. دسته‌بندی متون
2. شاخص‌گذاری خودکار
3. چکیده‌نویسی خودکار
4. خلاصه‌سازی متون

---

[1] Informative

[2] Information Retrieval

5. تولید عنوان برای متون
6. استخراج اصطلاحات فنی

حوزه‌ی مرتبط دیگر استخراج کلمات فنی می‌باشد که تمامی کلمات تشریح‌کننده یک دامنه یا موضوع، استخراج می‌شوند. کلمات کلیدی می‌توانند به عنوان زیرمجموعه‌ای از کلمات فنی در نظر گرفته شوند. بر اساس این تعریف، کلمات کلیدی، ترکیبی از کلمات تشریح کننده یک سند مشخص، مستقل از دامنه‌ای که در آن قرار دارد می‌باشد. کلمات کلیدی تقریبا موضوع سند را مشخص می‌کند.

2.5.1. تقسیم بندی روش‌ها

2.5.1.1. تقسیم بندی ابزاری

روش‌های استخراج کلمات کلیدی از جنبه‌های مختلف مد نظر قرار می‌گیرد.

i : اگر مجموعه‌ای از اسناد (مانند training set) با کلمه کلیدی مشخص برای هر کدام وجود داشته باشد، فرایند استخراج کلمه کلیدی یک یادگیری باناظر[1] خواهد بود. در غیر این صورت بدون ناظر[2] خواهد بود.

ii : استخراج کلمه کلیدی می‌تواند بر پایه مجموعه‌ای از نوشته‌ها[3] (اسناد) یا یک سند می‌باشد. یک کاربرد استخراج کلمه کلیدی در یک سند، مشخص سازی online کلمه کلیدی در اسناد مبتنی بر وب می‌باشد. در این حالت نمی‌توانیم از معیار و میزان مانند tfidf استفاده کنیم چون اساس آن بر فراوانی کلمه در مجموعه اسناد می‌باشد.

iii : استخراج کلمه کلیدی ممکن است از دیکشنری و یا قاموس[4] یا هیچ کدام استفاده کند. این باید مورد توجه قرار گیرد که هدف استفاده از دیکشنری ریشه‌یابی و استخراج مترادف کلمه می‌باشد که با استفاده از آن می‌توانیم کلمات را با توجه به ریشه و مترادف‌های آن طبقه‌بندی کنیم.

iv : استفاده از تکنیک پردازش زبان طبیعی (NLP) و تحلیل بخش‌های گفتار، موضوع نهایی در طبقه‌بندی روش‌های استخراج کلمات کلیدی می‌باشد.

---

[1] Supervised

[2] Unsupervised

[3] Corpus

[4] Thesaurus

2.5.1.2. تقسیم بندی تکنیکی

تمامی روش‌های پیشنهاد شده برای استخراج کلمات کلیدی ، به چهار راهکار کلی طبقه‌بندی می‌شوند:

i : روش‌های آماری[1] مبتنی بر تحلیل فراوانی کلمات.

ii : روش‌های نحوی[2] مبتنی بر تجزیه زبانی[3] و انطباق الگو.

iii : روش‌های ساختاری[4] بررسی عنوان و رئوس کلی مطالب سند.

iv : روش‌های ادراکی[5] مبتنی بر استفاده از پایگاه دانش برای تفسیر معنی و مفهوم.

در اکثر روش‌های معروف تعداد کلمات استخراج شده به عنوان کلمه کلیدی 10 الی 15 کلمه می‌باشد. اکثر روش‌های استخراج کلمات کلیدی مبتنی بر پردازش زبان طبیعی (NLP) از دیکشنری برای مشخص کردن ریشه کلمات و بخش‌های گفتار استفاده می‌کنند.

---

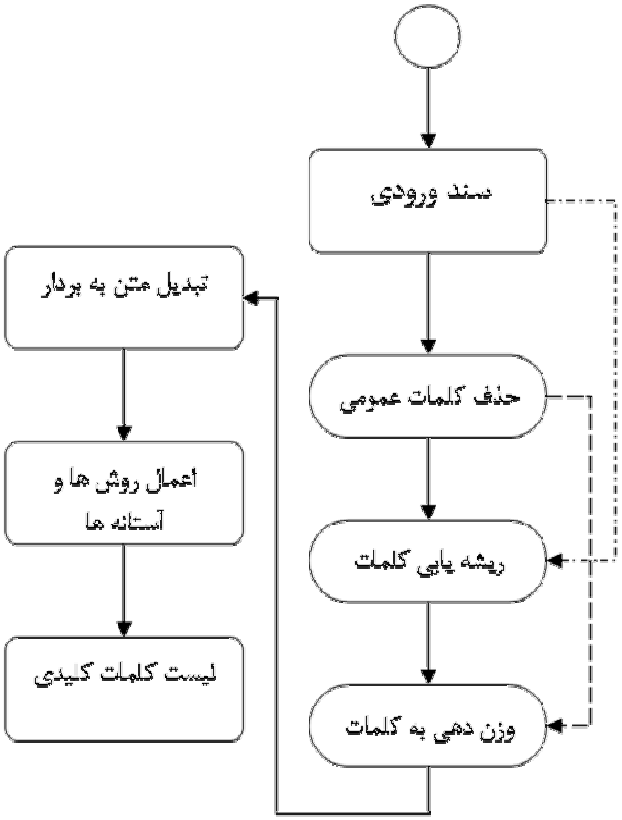
[1] Statistical

[2] Syntactic

[3] Linguistic

[4] Structural

[5] Conceptual



شکل 2-2. روند نمای استخراج کلمات کلیدی

2.5.2. مراحل استخراج کلمات کلیدی

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. برای اطلاعات بیش‌تر می‌توانید به فصل ریشه‌یابی مراجعه کنید. از کارهای دیگری که انجام می‌گیرد، می‌توانیم حذف نقطه گذاری، حذف کلمات کوچکتر از یک آستانه را نام ببریم.

روندنمای استخراج کلمات کلیدی در شکل (2-2) آمده است. اولین مرحله خواندن سند متنی و تعیین کلمات است.

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.

2.5.2.1. حذف کلمات عمومی فارسی

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 [25]. کلمات عمومی در دامنه‌های مختلف ممکن است متفاوت باشد ولی در حالت عمومی یک لیست از کلمات عمومی مطابق جدول 2-2 و 2-3 پیشنهاد شده در [14] برای استفاده در کارهای پردازش متن پیشنهاد می‌شود.

جدول 2-2. لیست کلمات عمومی (حروف پرتکرار)

در	نیز	برای	یا	را
به	تا	ها	دو	های
از	ما	آن	آن‌ها	و
که	باید	وی	اما	نمی
این	اند	یک	دیگر	هر
با	هم	خود	اگر	ای
می	هم‌چنین	بر		

جدول 2-3. لیست کلمات عمومی (افعال پرتکرار)

است	باید	بود	توانستند	داشتیم	گویم
آمد	بتوان	بودم	توانستیم	شد	گویند
آمدم	بتوانم	بودی	توانم	شده	گویی
آمدن	بتوانی	بودن	توانند	شود	گویید
آمدند	بتواند	بوده	توانی	کرد	گوییم
آمده	بتوانیم	بودیم	توانید	کردم	گیرد
آمدی	بتوانید	بودید	توانیم	کردن	گیرم

آمدید	بتوانند	بودند	خواست	کردند	گیرند
آمدیم	بخواه	بیا	خواستم	کرده	گیری
آورد	بخواهم	بیاب	خواستن	کردی	گیرید
آوردم	بخواهد	بیابد	خواستند	کردید	گیریم
آوردن	بخواهند	بیابم	خواسته	کردیم	می‌شود
آوردند	بخواهی	بیابند	خواستی	کن	هست
آورده	بخواهید	بیابی	خواستید	کند	هستم
آوردی	بخواهیم	بیابید	خواستیم	کنم	هستند
آوردید	بکن	بیابیم	خواهد	کنند	هستی
آوردیم	بکند	بیاور	خواهم	کنی	هستید
آورم	بکنم	بیاورد	خواهند	کنید	هستیم
آورند	بکنند	بیاورم	خواهی	کنیم	یابد
آوری	بکنی	بیاورند	خواهید	گرفت	یابم
آورید	بکنید	بیاوری	خواهیم	گرفتم	یابند
آوریم	بکنیم	بیاورید	داد	گرفتن	یابی

آید	بگو	بیاوریم	دار	گرفتند	یابید
آیم	بگوید	بیاید	دارد	گرفته	یابیم
آیند	بگویم	بیایم	دارم	گرفتی	یافت
آیی	بگویند	بیایند	دارند	گرفتید	یافتم
آیید	بگویی	بیایی	داری	گرفتیم	یافتن
آییم	بگویید	بیایید	دارید	گفت	یلفتند
باش	بگوییم	بیاییم	داریم	گفتم	یافته
باشد	بگیر	تواند	داشت	گفتن	یافتی
باشند	بگیرد	توانست	داشتم	گفتند	یافتید
باشی	بگیرم	توانستم	داشتن	گفته	یافتیم
باشم	بگیرند	توانستن	داشتند	گفتی	
باشید	بگیری	توانستند	داشته	گفتید	

2.5.2.2.ریشه‌یابی

یکی از مهم‌ترین کارها در استخراج کلمات کلیدی از متون فارسی، ریشه‌یابی کلمه‌ها می‌باشد. هدف از ریشه‌یابی حذف اضافات از کلمه و رسیدن به ریشه‌ی اصلی کلمه هست [26]. روش‌های مختلفی برای ریشه‌یابی کلمات فارسی پیشنهاد شده است اکثر ریشه‌یاب‌ها از روش مبتنی بر حذف پس‌وندها و پیش‌وندها استفاده می‌کنند. مباحث مربوط به ریشه‌یابی به طور مفصل در فصل ریشه‌یابی آورده شده است.

2.5.2.3. وزن‌دهی به کلمات

وزن‌دهی به کلمات بر اساس اهمیت آن‌ها در متن انجام می‌گیرد. اهمیت کلمات را می‌توان بر پایه شرایط زیر مشخص کرد [27]:

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

کلمات متضاد

کاربرد خاص کلمه:

مثلاً، اسامی در سیستمی که به دنبال اسامی خاص می‌گردد دارای اهمیت بیش‌تر است

وزن آماری کلمه:

بر پایه‌ی تکرار کلمات در متن

بر پایه‌ی توزیع کلمات در متن

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[1]

فراوانی نسبی[2]

---

[1] Absolute Frequency

[2] Relative Frequency

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. روش‌های وزن‌دهی به کلمات زیاد می‌باشد که ما در اینجا بخشی از آن‌ها را آورده ایم و بعضی از روش‌ها را نیز در بخش کارهای مرتبط خواهیم آورد.

2.5.2.3.1. پارامتر TF*IDF

پارامتر TF*IDF یکی از پرکاربردترین روابط در حوزه بازیابی اطلاعات متنی می‌باشد [27]. 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. رابطه (2-4)و(2-5) محاسبه TF*IDF را نشان می‌دهد.

$$IDF = \log_2 \frac{\mathrm{n}}{\mathrm{DOCFREQ_K}} + 1 \qquad (4\text{-}2)$$

$$WEIGHT_{ik} = IDF * FREQ_{ik} \qquad (5\text{-}2)$$

$WEIGHT_{ik}$: وزن کلمه kام در سند iام

$FREQ_{ik}$: فراوانی کلمه kام در سند iام

$DOCFREQ_K$: تعداد اسناد شامل کلمه kام

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[26].

2.5.2.3.2. پارامتر سیگنال و نویز

در این روش از تئوری اطلاعات استفاده شده است [27]. 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. رابطه (2-6) میزان اطلاعات یک کلمه با احتمال رخداد P را نشان می‌دهد.

$$INFORMATION = -LOG_2 P \qquad (6\text{-}2)$$

متوسط میزان اطلاعات یک متن با t کلمه، با استفاده از رابطه (2-7) محاسبه می‌شود.

$$\text{average information} = -\sum_{k=1}^{t} p_k . \log_2(p_k) \qquad (7\text{-}2)$$

میزان نویز بر اساس رابطه (2-8) تعریف می‌شود:

$$NOISE_K = \sum_{i=1}^{n} \left(\frac{FREQ_{ik}}{TOTFREQ_K} . \log_2 \frac{TOTFREQ_K}{FREQ_{ik}} \right) \qquad (8\text{-}2)$$

ارتباط معکوس بین نویز و تعیین‌کنندگی با پارامتر سیگنال، طبق رابطه (2-9) تعیین می‌شود:

$$signal_k = \log_2 (TOTFREQ_K) - NOISE_K \qquad (9\text{-}2)$$

در این روش وزن‌دهی به کلمات، اهمیت زیاد به کلماتی داده می‌شود که تنها در تعدادی از اسناد ظاهر شده باشند. وزن کلمه k در سند i با رابطه (2-10) محاسبه می‌شود.

$$WEIGHT_{ik} = FREQ_{ik} . signal_k \qquad (10\text{-}2)$$

2.5.2.3.3. پارامتر مقدار تمایز

در این روش، برای وزن‌دهی کلمات از قدرت تمییزدهندگی کلمات بین اسناد مختلف استفاده می‌شود [26, 27]. مقدار تمایز[1] را با استفاده از معیارهای مشابهت محاسبه

[1] Discrimination Value

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.

معیارهای مشابهت

اگر دو سند متنی X,Y به صورت زیر وجود داشته باشد:

$$X = (x_1, x_2, ..., x_t)$$
$$Y = (y_1, y_2, ..., y_t)$$

اگر X_i و y_j فراوانی کلمات در اسناد مربوط باشند معیارهای قید شده در جدول (2-4) را می‌توان برای تعیین مشابهت به کار برد.

تابع متوسط مشابهت

تابع متوسط مشابهت بیان کننده میزان مشابهت کل اسناد مجموعه به همدیگر است. و اگر کل اسناد مجموعه، مشابه باشند، این مقدار ماکزیمم می‌شود. رابطه (2-11) متوسط مشابهت را نشان می‌دهد.

$$AVGSIM = \frac{1}{n(n-1)} . \sum_{i=1}^{n} \sum_{j=1}^{n} similar(D_i, D_j) \qquad i \neq j \qquad (11\text{-}2)$$

تعریف مقدار تمایز برای کلمه kام:

$(AVGsim)_k$ متوسط مشابهت، در حالی که کلمه‌ی k از همه اسناد حذف شده باشد. فرض کنیم k کلمه‌ای باشد که به صورت گسترده در مجموعه متون به کار رفته باشد (با فراوانی زیاد و توزیع نسبتا یکنواخت). حذف k باعث کاهش مشابهت بین زوج اسناد می‌شود. در واقع $(AVGsim)_k$ کاهش

می‌یابد. در صورتی که توزیع یکنواخت نباشد حذف آن باعث افزایش $(AVGsim)_k$ می‌شود. مقدار تمایز و وزن کلمه k در سند i با روابط (2-12) و (2-13) محاسبه می‌شود.

$$Discvalue_k = (AVGSIM)_K - AVGSIM \qquad (12\text{-}2)$$

$$WEIGHT_{ik} = FREQ_{ik} . Discvalue_k \qquad (13\text{-}2)$$

پس از وزن‌دهی کلمات با استفاده از اعمال آستانه‌های مختلف روی وزن کلمات، می‌توان کلمات کلیدی را استخراج کرد.

جدول 2-4. معیارهای مشابهت

معیار مشابهت	$similarity(X,Y) =$
ضرب داخلی	$\sum_{i=1}^{t} x_i y_i$
ضریب Dice	$2 * \frac{\sum_{i=1}^{t} x_i y_i}{\sum_{i=1}^{t} x_i^2 + \sum_{i=1}^{t} y_i^2}$
ضریب کسینوسی	$\frac{\sum_{i=1}^{t} x_i y_i}{\sqrt{\sum_{i=1}^{t} x_i^2} \cdot \sqrt{\sum_{i=1}^{t} y_i^2}}$
ضریب jaccard	$\frac{\sum_{i=1}^{t} x_i y_i}{\sum_{i=1}^{t} x_i^2 + \sum_{i=1}^{t} y_i^2 - \sum_{i=1}^{t} x_i y_i}$
ضریب همپوشانی	$\frac{\sum_{i=1}^{t} x_i y_i}{\min(\sum_{i=1}^{t} x_i^2, \sum_{i=1}^{t} y_i^2)}$

2.5.3. نحوه ارزیابی کلمات کلیدی

2.5.3.1. پارامترها

2.5.3.1.1. پارامتر دربرگیری

بیانگر میزانی است که همه کلمات متن در استخراج کلمات کلیدی ظاهر شده‌اند [27]. در واقع هر چه کلمات بیش‌تری از متن در استخراج کلمات کلیدی به کار روند، میزان دربرگیری[1] کلمات کلیدی و نیز نسبت آیتم‌هایی که با آن می‌توانند بازیابی شوند زیاد خواهد بود.

2.5.3.1.2. پارامتر تعیین‌کنندگی

یعنی هر کلمه‌ی کلیدی تا چه حد دقیق، متن‌های مربوط را مشخص می‌کند [27]. کلمه کلیدی که دارای سطح بالایی از تعیین‌کنندگی[2] است، موارد نامربوط را به کلمات به کار رفته در آن نگاشت نمی‌کند. برای ارزیابی کلمات کلیدی استخراج شده به دو طریق می‌توان عمل کرد:

---

[1] Exhaustivity

[2] Specifity

2.5.3.2. داوری مبتنی بر کارشناس انسانی

در این روش، کلمات کلیدی استخراج شده توسط سیستم، به یک کارشناس موضوع سند، داده می‌شود، و کارشناس با توجه به محتوای سند، خوب یا بد بودن نتایج را مشخص می‌کند. در اکثر مواقع این کارشناس، همان نویسنده‌ی متن می‌باشد.

2.5.3.3. داوری مبتنی بر سیستم‌های بازیابی اطلاعات

در این روش، انسان هیچ نقشی در داوری کلمات کلیدی ندارد و ارزیابی کلمات کلیدی با استفاده از سیستم‌های بازیابی اطلاعات انجام می‌گیرد. برای ارزیابی کلمات کلیدی، باید یک پیکره‌ی متنی به همراه مجموعه‌ای از پرس و جوهای مرتبط با اسناد و هم‌چنین تمام اسناد مرتبط با پرس و جوها وجود داشته باشد. پرس و جوها به پیکره اعمال می‌شوند، و اسنادی با توجه به مشابهت پرس و جو و اسناد، بازیابی می‌شوند. و پارامترهای ارزیابی روی اسناد بازیابی شده اعمال می‌شود. جدول 2-5 جدول اسناد مرتبط و بازیابی شده را نشان می‌دهد. در این جدول A بیانگر تعداد اسناد مرتبط با پرس و جو و B بیانگر اسناد بازیابی شده برای پرس و جو می‌باشد. در ادامه مهم‌ترین پارامترهای ارزیابی با توجه به جدول 2-5 آورده شده است.

جدول 2-5. جدول اشتراک اسناد مرتبط و بازیابی شده

	RELEVANT	NON-RELEVANT	
RETRIEVED	$A \cap B$	$\overline{A} \cap B$	B
NOT RETRIEVED	$A \cap \overline{B}$	$\overline{A} \cap \overline{B}$	$\overline{B}$
	A	$\overline{A}$	N

2.5.3.3.1. مقدار بازخوانی

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[1] نسبت تعداد اسناد مرتبط بازیابی شده بر کل اسناد مرتبط با پرس و جو محاسبه می‌شود. رابطه‌ی (2-14) برای محاسبه پارامتر بازخوانی به کار می‌رود.

$$R = \frac{|A \mathbf{I} B|}{|A|} \qquad (14\text{-}2)$$

2.5.3.3.2. مقدار دقت

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

[1] Recall

اساس رابطه (2-15)، نسبت تعداد اسناد مرتبط بازیابی شده بر کل اسناد بازیابی شده برای پرس و جو، محاسبه می‌شود.

$$P = \frac{|\mathrm{A}\ \mathbf{I}\ \mathrm{B}|}{|\mathrm{B}|} \qquad (15\text{-}2)$$

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.

2.5.3.3.3. پارامتر $F_{measure}$

در حقیقت این پارامتر میانگین هارمونیک پارامترهای بازخوانی و دقت می‌باشد. هدف در سیستم‌های بازیابی اطلاعات بیشینه کردن این معیار می‌باشد. پارامتر $F_{measure}$ بز اساس رابطه (2-16) محاسبه می‌شود.

$$F_{measure} = \frac{2 * P * R}{P + R} \qquad (16\text{-}2)$$

2.5.3.3.4. پارامتر Fallout

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 (2-17) محاسبه پارامتر Fallout را نشان می‌دهد.

$$F = \frac{\left|\overline{A} \cap B\right|}{\left|\overline{A}\right|} \qquad (2\text{-}17)$$

2.6. روش‌های آماری

استخراج کلمات کلیدی به روش آماری، تا حد زیادی مستقل از زبان است و تنها بخش ساخت‌واژی وابسته به زبان است. به این علت چند نمونه از کارهای انجام شده در سایر زبان‌ها را در این بخش آورده‌ایم. هم‌چنین یکی از کارهای انجام شده در زبان فارسی نیز، در ادامه به همراه نتایج حاصل شده تشریح خواهد شد. برای زبان فارسی فقط یک کار در حوزه‌ی نمایه‌سازی پیدا شد.

یک راهکار برای استخراج کلمات کلیدی یادگیری باناظر هست. در این روش نیاز به اسنادی با کلمات کلیدی مشخص است. برای یادگیری، برخی ویژگی‌ها همچون tf و idf و

تعداد دفعات کلمه کلیدی بودن یک کلمه، برای تمامی کلمات محاسبه می‌شود. و با استفاده از مکانیزم‌های یادگیری، روی مجموعه آموزشی اعمال می‌شود.

در مرجع [28]، کلمات کلیدی با مشخص کردن cue-words استخراج می‌شوند که عبارت است از محموعه‌ای از کلمات یا عبارات که اشاره به اهمیت جمله‌ای که در آن هستند دارد. برای مثال "purpose of this article" , "in this paper" می‌توانند به عنوان cue-word باشند.

خوشه‌بندی (clustering) کلمه و خوشه‌بندی هم‌زمان کلمات و اسناد یک راهکار دیگر برای استخراج کلمات کلیدی هست. برای مثال در مرجع [29] کلمات کلیدی با استخراج ماتریس روابط (وابستگی) به دست می‌آیند. ماتریسی که ارتباط بین کلمات را نشان می‌دهد با استفاده از ACE (محیط خوشه‌بندی پیشرفته) که روش ACE رابطه‌ای نامیده می‌شود خوشه‌بندی می‌شود. رابطه بین کلمات با استفاده از میزان و معیاری به نام فاصله Levenshtein اندازه‌گیری می‌شود.

در برخی کاربردها نیاز به استخراج کلمات کلیدی ، مستقل از دامنه و بدون نیاز به پیکره‌های بزرگ است. یک روش خیلی قدیمی شمارش کلمات هست که این روش، به اندازه کافی مفید نیست.

مرجع [21] یک الگوریتم جدید استخراج کلمات کلیدی که بدون استفاده از پیکره، روی یک سند اعمال می‌شود، را نشان داده شده است. روش استخراج، مبتنی بر پیدا کردن جملات در متن با استفاده یک پارسر می‌باشد. با توجه به این روش، یک جمله دنباله‌ای از

کلمات است. جملات با استفاده از علامات توقف مجزا می‌شوند. علامات جداساز , "." , "!" "?" می‌باشد.

روش پیشنهاد شده، به این صورت است که ابتدا کلمات تکرار شونده استخراج می‌شود سپس مجموعه‌ای از رخدادهای هم‌زمان هر کلمه با کلمات تکرارشونده را ایجاد می‌کنیم. مثلاً رخداد در جمله‌های مشابه تولید می‌شود. توزیع رخدادهای هم‌زمان اهمیت کلمه در سند را نشان می‌دهد. اگر توزیع احتمال رخدادهای هم‌زمان بین کلمه a و کلمات تکرار شونده به یک زیر مجموعه خاص از کلمات تکرار شونده بایاس شود کلمه‌ی کلیدی بودن a محتمل است. درجه بایاس توزیع با استفاده از $\chi 2$ اندازه‌گیری می‌شود.این الگوریتم کارآیی قابل مقایسه با tfidf ، را بدون استفاده از مجموعه نوشته‌ها را ارائه می‌کند.

در اینجا ابتدا کلمات کلیدی اولیه برای هر سند (IRK) با استفاده از استنتاج فازی، استخراج می‌شود. که شامل یک مجموعه از کلمات و وزن‌های آن کلمات می‌باشد. سپس کلمات کلیدی نماینده، با استفاده از رخدادهای هم‌زمان کلمات در اسناد وزن‌دهی دوباره می‌شوند و کلمات کلیدی نهایی به دست می‌آید.

کل کلمات را به سه دسته‌ی کلمات عمومی، کلمات خاص و کلمات کلیدی تقسیم می‌کند و مرز این کلمات را به صورت فازی تعیین می‌کند. این روش نیز برای زبان انگلیسی ارائه شده است که مبتنی بر چند کلاس و چند سند می‌باشد. ایده‌ی این است که کلماتی که در کلیه اسناد کلیه کلاس‌ها پخش شده باشند، دارای ارزش و مفهوم کم‌تری هستند و به عنوان کلمات عمومی شناخته می‌شوند. کلماتی که فقط در کلیه اسناد یک کلاس پخش

شده باشند و فراوانی زیادی داشته باشند و در سایر کلاس‌ها فراوانی کمی داشته باشند به عنوان کلمات خاص یا اصطلاحات فنی آن کلاس در نظر گرفته می‌شوند. و کلماتی را که در یک سند خاص دارای فراوانی زیاد بوده و در بقیه دارای فراوانی کمی باشند، به عنوان کلمات کلیدی آن سند در نظرگرفته می‌شوند.

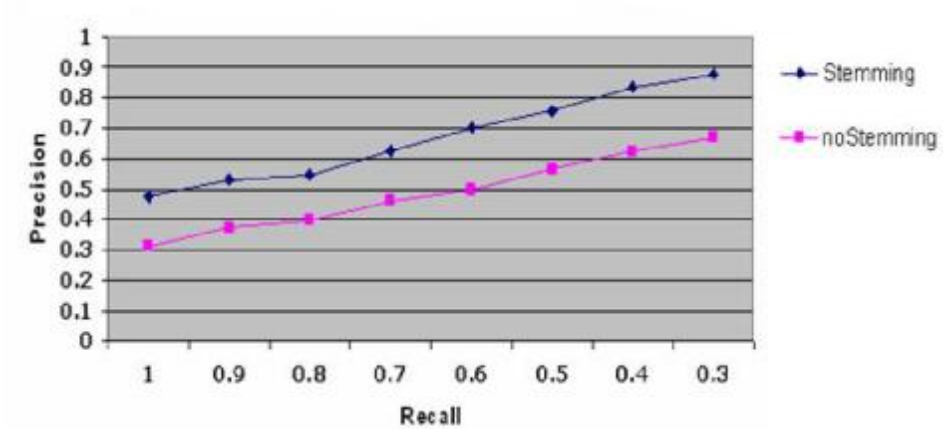
یک روش مستقل از زبان برای زبان‌های ژاپنی و انگلیسی مطرح شده است که با استفاده از تحلیل ساخت‌واژی (morphology) ، استخراج عبارات اسمی، امتیازدهی و دسته‌بندی آن‌ها، کلمات کلیدی مناسب را فقط با استفاده از یک سند استخراج می‌کند.کلمات کلیدی با انتخاب کوچکترین عضو دسته‌ها، استخراج می‌شوند که بخش ساخت‌واژی وابسته به زبان هست. و هر زبان ساخت‌واژی مخصوص به خود را دارد.

با استفاده از ضریب مشابهت Jaccard کلمات کلیدی را از وب برای ایجاد فراداده برای اشخاص استخراج کرده است.

مرجع [30] از نگاشت خودسازمانده (Som) برای استخراج کلمات کلیدی استفاده کرده است.

در [25] نمایه‌سازی متون فارسی (نمایه‌ساز سینا) کار 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. با توجه به نتیجه گیری‌های

آن‌ها دو روش tfidf و lnu بهتر جواب داده است. برای پیکره از 450 متن شامل چکیده مقالات مرتبط با کامپیوتر استفاده شده است.برای ارزیابی از معیارهای بازخوانی و دقت استفاده کرده‌اند. میانگین پارامتر دقت با ریشه‌یابی 66% و بدون ریشه‌یابی 54% بوده است. شکل 2-3 نتایج ارزیابی نمایه‌ساز سینا را نشان می‌دهد.



شکل 2-3. نتایج ارزیابی نمایه‌ساز سینا

یک حوزه نزدیک به استخراج کلمات کلیدی indexing می‌باشد. indexing یا همان ایجاد فهرست راهنما برای نمایش اسناد و عملیات جست و جو یکی از مهم‌ترین فرایندها در IR می‌باشد. با توجه به مرجع [31] «یک شاخص اساسا، مجموعه‌ای از کلمات با اشاره گرهایی به محل‌هایی از اسناد که اطلاعاتی راجع به کلمات در آنجا یافت می‌شود، می‌باشد. یک مثال از شاخص گذاری شاخص گذاری صفحات وب هست که بازیابی اطلاعات صفحه وب را راحت‌تر می‌کند» [31].

3. دشواری‌های ریشه‌یابی فارسی و معرفی روش‌هایی برای ریشه‌یابی فارسی

3.1. مقدمه

پیرایش و ویرایش بر روی دیگر زبان‌های دنیا خیلی پیشتر از این آغاز شده است. ساده‌کردن قاعده‌ها، کم‌کردن قاعده‌های پیچیده و استثناها در زبان روزمره (نه زبان ادبی)، یکسان کردن گفتار و نوشتار روزمره، به کارگیری تعداد کمی کلمه و اصطلاح، گسترش استانداردهای آماده شده برای زبان از کارهایی است که برروی بسیاری از زبان‌ها انجام شده است [32]. استادان زبان انگلیسی و زبان‌شناسان، بسیاری از قاعده‌های این زبان را پیراسته اند و یادگیری و به کارگیری این زبان را ساده نموده‌اند. برای نمونه در نوشتار امروزی انگلیسی کم‌تر حرف‌ها به هم چسبیده نوشته می‌شوند و کلمات و اصطلاح‌های کمی، به ویژه در نوشته‌های علمی، به کار گرفته می‌شود. ویرایش‌های انجام شده در زبان انگلیسی بسیار با کارهای رایانه‌ای، که بر پایه زبان انگلیسی هستند، اثر داشته است و به پیشرفت نرم‌افزارهای رایانه‌ای کمک نموده است. پیرایش‌هایی که در زبان انگلیسی انجام شده است، بسیاری از پیچیدگی‌های ساخت نرم‌افزار، برای این زبان را کاسته است. به نوبه‌ی خود ساخت نرم‌افزار رایانه‌ای گسترش استاندارد آن زبان را در پی داشته است.

3.2. قالب‌های گوناگون پرونده‌های رایآن‌های

امروزه پرونده‌های نوشتاری فارسی با نرم‌افزارهای گوناگون نوشته می‌شوند. برای ریشه‌یابی کلمات درون این نوشتارها به کمک رایانه باید قالب این پرونده‌ها خوانده شود. قالب این پرونده‌ها اغلب هم‌خوانی کمی با هم دارند. نرم‌افزارهای گوناگونی همچون pe2 ، زرنگار، کلک، نشر الف، Microsoft word (که نسخه‌های گوناگون آن فارسی را به یک شکل پشتیبانی نمی‌کنند)، pdf , latex برای نوشتن در رایانه به کار گرفته می‌شود که قالب پرونده نوشته شده در هر کدام ویژه خود آن نرم‌افزار است [33]. کسانی نیز این نوشته‌ها را به تصویر تبدیل می‌کنند تا خواننده بتواند به سادگی آن‌ها را برروی هر رایانه‌ای بخواند. آماده‌کردن یک برنامه رایانه‌ای که همه این قالبها را بخواند اگر ناممکن نباشد بسیار سخت خواهد بود. پرونده‌های با قالب xhtml (صفحه‌های شبکه جهانی) 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[1] 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 صفحه‌های سبک[2] آشنا بود؛ تا بتوان از همه توانایی‌های آن سود برد. پنجم،

---

[1] W3C

[2] Cascade Style Sheet

قابلیت حمل[1] بالایی دارد و به خوبی بر روی رایانه‌های گوناگون و سیستم عامل‌های گوناگون از آن بهره برد. به‌روزترین روش در این زمینه به کارگیری xml[2] و تبدیل آن به xhtml به کمک xsl[3] است که البته برای قالب دهی می‌توان از css یا از xsl-fo کمک گرفت.

3.3.استاندارد خط در رایانه

روند فارسی‌سازی و استاندارد نمودن خط فارسی برای رایانه فراز و نشیب‌های زیادی داشته است. کوچک‌ترین واحد نوشته نویسه[4] 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[5] نامیده می‌شود. برای نمونه نویسه‌ی «ی» دارای شکل‌های نمایشی«یی»، «ی» ، «ی»، «پیر » است. مجموعه کد به دو گونه تعریف شده است [32]:

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

---

[1] Portability

[2] Extensible Markup Language

[3] Extensible Stylesheet Language

[4] Character

[5] Glyph

2) 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 [= بدون فاصله با «خانه» باشد و «ه» در پایان خانه به شکل «خانهها» تبدیل نگردد. بنابراین نویسه‌ی فاصله مجازی[1] 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[2] 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

[1] Zero- Width Non - Joiner

[2] Zero – Width Joiner

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 223 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(مجموعه کد) را یونی‌کد[1] نامیدند. البته دراین کد نیز ترتیب چهار حرف فارسی رعایت نشده است. هم‌چنین مشکل حرف‌هایی با چندین کد (و رعایت نکردن فارسی یا

[1] Unicode

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 نرم‌افزارهای موجود 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 حداکثر6 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.

در اینجا تعدادی از قالب‌های استاندارد کدگذاری فارسی آورده شد. 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.

3.4. 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.

3.4.1. «ی» پس از «ه»

یکی از تغییرهایی که در این چند سال در نگارش فارسی به وجود آمده است، تغییر شکل کسره اضافه پس از «ه» است. در گذشته با گذاشتن «ء» بر سر «ه» این کار انجام می‌شد؛ ولی امروزه برای نشان دادن کسره اضافه پس از «ه»، «ی» به کار گرفته می‌شود.

«زبان فارسی به اندیشه ما شکل داده است»

«همه تصمیم‌ها یا گزینش‌هایی که در قسمت خلاقه مغز فرستنده به عمل آمده»

در [33] صفحه‌ی 20، همزه پیشنهاد شده است که در کتاب‌های کنونی درسی به صورت «ی» نوشته می‌شود و همین باعث سردرگمی نویسندگان خواهد شد.

ناهماهنگی دیگری در نوشتن «ۀ» دیده می‌شود که به دلیل به کار بردن کدهای گوناگون در رایانه برای آن به وجود می‌آید. نویسه «ۀ» و دیگری با دو نویسه «ه» و«ء» است که بهتر است که یکی از این دو روش برای نوشتن پیشنهاد گردد. گرچه به نظر می‌رسد که به کار بردن «ی» در «لانه گنجشک» خواناتر و زیباتر باشد و این مشکلات را در بر ندارد.

3.4.2. «ها» ی نشانه جمع

«ها (نشانه جمع) همواره به کلمه پیش از خود می‌چسبد، مانند کتابها، باغها، چاهها، کوهها، گرهها، مگر هنگامی که:

1) بخواهیم صورت مفرد کلمه را مشخص کنیم : کتاب‌ها، درس‌ها، باغ‌ها

2) کلمه به‌های غیر ملفوظ (بیان حرکت) و یاهای ملفوظی که حرف قبل از آن حرف متصل باشد، ختم شود: میوه‌ها، خانه‌ها، سفیه‌ها، فقیه‌ها، به‌ها» [34].

این سفارش فرهنگستان که مبهم است. در کتاب‌های دوره‌های دبستان، راهنمایی و دبیرستان «ها» بیش‌تر جدا نوشته می‌شود.

«تم وقتی در طویله کار داشت، این لباسها را به تن می‌کرد. ارباب آهن‌ها را از دست غلام باز کرد».

«با زبان‌های بیگانه‌ای که در کار تحقیقی با آن سروکار دارند».

«روزنامه‌نگاران که غالب کتابهای آن دوره را نوشته‌اند».

3.4.3. فاصله‌گذاری

«فاصله‌گذاری میان کلمات، خواه بسیط و خواه مرکب، امری ضروری است که اگر رعایت نشود طبعا سبب بدخوانی و ابهام معنایی می‌شود» [33]

خوش‌بختانه امروزه در نوشتار رایانه‌ای این امر تا اندازه‌ای رعایت می‌شود؛ زیرا مفهوم فاصله به خوبی روشن است. بدون رعایت کردن فاصله ریشه‌یابی سخت‌تر می‌شود. در کلماتی مانند «می نویسم» می‌خواهیم که «می» به «نویسم» چسبیده نباشد (مینویسم) و در عین حال بهتر است با آن، فاصله نیز نداشته باشد (می‌نویسم). در کلمات دیگری مانند «علاقه‌مند»، «خانه‌ها» و.... همین نیاز را داریم. یا به طور روشن‌تر در بسیاری از کلمه‌ها می‌خواهیم میان بخش‌های کلمه فاصله گذاشته نشود و در عین حال حروف پیوندناپذیر یک بخش از کلمه به بخش پس از آن نچسبد.

«همانطور که متصل کردن دو حرف نشانه اتصال است (مانند حرف ب در کلمه با) فاصله کوتاه میان دو حرف هم نشانه اتصال است. چنان که در این کلمات می‌بینیم : روزنامه، یادداشت، هم منزل».

در این حالت‌ها میان این دو کلمه فاصله‌ی مجازی گذاشته می‌شود. همین مشکل در«مند»....نیزوجود دارد. پیشنهاد می‌شود که در دستور خط فارسی بخش‌هایی برای رایانه گذاشته شود یا این که جزوه جداگانه‌ای در این زمینه آماده شود. این جزوه (یا بخش‌های درون دستور خط) دربردارنده قانون‌هایی از این دست باشد. به این ترتیب در نوشتار رایانه‌ای نیز یکسان‌سازی انجام خواهدشد.

اگر میان کلمات درون یک کلمه‌ی مرکب فاصله معمولی نباشد، پردازش رایانه‌ای نوشتار ساده‌تر خواهد شد (به ویژه در پردازش ساختاری جمله). اگر حرف پایانی یکی از کلمات میانی پیوندپذیر باشد؛ می‌توان میان این دو کلمه فاصله مجازی گذاشت. برای نمونه در «گیله‌مرد» میان دو بخش یک فاصله مجازی گذاشته شده است. هم‌چنین در حالت‌هایی که کنار هم گذاشتن کلمات درون کلمه مرکب باعث ابهام شود؛ فاصله مجازی میان بخشهای کلمه مرکب گذاشته شود.به عبارت دیگر، اگر بتوان یک کلمه مرکب را به چند گونه تجزیه کرد، باید میان کلمات آن (که مورد نظر نویسنده است) فاصله‌ی مجازی گذاشت. در این باره بهتر است، قانون مشخصی نوشته شود. هم‌چنین اگر برای نوشتن عددها نیز این قاعده رعایت شود، بهتر است. برای نمونه به جای « سی و سه»، «سی وسه» نوشته شود.

3.4.4. کلمات مرکب

درباره چگونگی نگارش کلمات مرکب فارسی، دیدگاه‌های گوناگونی وجود دارد. حتی در این زمینه که کلماتی را با هم در نظر بگیریم و آن‌ها را مرکب بنامیم، نظرات گوناگونی وجود دارد (به ویژه در فعل مرکب). با همه کوششی که برای یکسان‌سازی در نوشتن این کلمه‌ها انجام شده است؛ هنوز ابهام‌ها و ناهماهنگی‌های فراوانی دیده می‌شود.

آماده‌کردن یک فهرست کامل از این کلمات مرکب در فرهنگستان و شکل پیشنهادی آن بسیار شایسته است. در این صورت روش نوشتن آن‌ها سلیقه‌ای نخواهد بود.

3.4.5. حرکت‌گذاری در نوشتار فارسی

«در خط فارسی افزون بر حرف‌های الفبا نه نشانه خطی دیگر نیز به کار می‌رود. این نشانه‌ها ـَ ـِ ـُ ـً ـٍ ـٌ 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 (آ ـً) بیش‌تر کاربرد دارند. تنوین رفع و جر (ـٌ ـٍ) 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» [33].

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.

3.5. دگرگونی در کلمه‌ها هنگام پیوند

در هنگام پیوند کلمه‌ها یا پیش‌وند به کلمه یا پس‌وند به کلمه، نیز تغییرهای گوناگونی رخ می‌دهد. در زیر نمونه‌هایی از این تغییرها نشان داده شده است.

«زنده» + «ان» ⟸ «زندگان» ؛ «زنده» + «م» ⟸ «زنده ام»؛ «گو» + «م» ⟸ «گویم»

«ن» + «افتاد» ⇐ «نیفتد» ؛ «ن» + «آزما» ⇐ «نیازما» ؛ «زنده» + «ها» ⇐ «زنده‌ها»

متخصص رایانه در هنگام ریشه‌یابی باید این‌ها را در نظر بگیرد. همین مشکل در کلمات پربسامد پیش می‌آید.

3.6. کلمات زبان‌های دیگر در فارسی

وجود دامنه‌ی گسترده‌ای از کلمات زبان‌های دیگر در زبان فارسی (مانند کلمات زبان‌های عربی، انگلیسی، ترکی، مغولی و فرانسوی یا کلمات دیگر زبان‌های بیگانه آورده شده به زبان فارسی) ریشه‌یابی رایانه‌ای را بسیار سخت می‌کند. برخی از این کلمه‌ها به ساختارهای کلمات زبان فارسی (برای نمونه یک کلمه مرکب فارسی) نزدیک هستند، بنابراین این احتمال وجود دارد که ریشه‌یاب به نادرستی آن‌ها را ریشه یا مشتق یک کلمه فارسی در نظر بگیرد. برای نمونه «ایدئالیست» یا «تایپیست» را می‌توان ترکیب «ایدئالی» (او آدم ایدئالی است) و «ست» (است) پنداشت و آن را با این روش به دو بخش شکست. برای بسیاری دیگر از کلمه‌ها نیز مشابه این مشکل پیش می‌آید. می‌توان به کسانی که با رایانه کار می‌کنند؛ پیشنهاد داد که به گونه‌ای کلمات زبان‌های دیگر را مشخص نمایند. برای نمونه درهنگام ساختن صفحه وب که با قالب html است می‌توان از

تایپیست

لا تُ عَ دّ و لا تُ ح^ صی

یا روش دیگری برای مشخص نمودن کلمات زبان‌های دیگر در زبان فارسی به کار گرفته شود.

3.7.شناسایی ریشه فعل‌ها

به خاطر در دسترس نبودن لغت نامه تصمیم گرفته شد که بر پایه‌ی برخی قاعده‌ها رده همه‌ی کلمه‌ها شناسایی شود. در زبان فارسی شناسایی رده یک کلمه (فعل ، اسم،....) به سادگی امکان‌پذیر نیست. برای نمونه به کار گرفتن گزاره‌ی ساده «فعل کلمه‌ای است که شناسه می‌پذیرد» به هیچ روی نمی‌تواند فعل‌ها را شناسایی کند. زیرا بسیاری از اسم‌ها و صفت‌ها نیز شناسه می‌گیرند. برای نمونه اغلب «خوبیم» به جای «خوب هستیم» و «عبارتند» به جای «عبارت هستند» به کار گرفته می‌شود. فعل‌ها در زبان فارسی رده‌ی بسیار بزرگی هستند که به کمک آن‌ها بسیاری از اسم‌ها و صفت‌ها،..... ساخته می‌شوند. بنابراین با شناسایی این رده، می‌توان بسیاری از کلمات فارسی را ریشه‌یابی نمود.

«در فعل‌های ساده پس از حذف « َ ن » از مصدر، بن ماضی باقی می‌ماند و از جهت تغییری که از بن ماضی به بن مضارع انجام می‌گیرد، آن‌ها را می‌توان در هشت گروه جای داد. جدول 3-1 تغییرهای فعل‌ها را در گروه‌های هشت گانه نشان می‌دهد» [34].

تعداد کمی از فعل‌های فارسی از قاعده‌های این هشت گروه پیروی نمی‌کنند. بن گذشته و غیرگذشته این فعل‌ها جداگانه نوشته شدند.

جدول 3-1. هشت گروه فعلهای فارسی

گروه	حروف پایانی بن ماضی + پس‌وند	حروف پایانی بن ماضی	بن مضارع	مثال	
1	یدن	- ید	پس از حذف «-ید» باقی مانده بن مضارع است.	نال + ید + َ ن	نالیدن نال/نالید
2	دن	د	پس از حذف «د» باقی مانده بن مضارع است.	خور + دَ + ن	خوردن خور/ خورد
3	ودن	- ود	پس از حذف «و»، «و» به «ا» تبدیل می‌شود.	آزمو+دَ + ن	آزمودن آزما/ آزمود
4	ادن	- اد	پس از حذف «اد» باقی مانده بن مضارع	افت + دَ	افتادن

			است	+ن	افت/ افتاد
5	ختن	- خت	«ت» حذف و «خ» تبدیل به «ر» می‌شود	ساخ+ت+َن	ساختن ساز/ ساخت
6	ستن	- ست	پس از حذف «ست» باقی مانده بن مضارع است	آرا+ ست +َن	آراستن آرا/ آراست
7	شتن	- شت	پس از حذف «ت»، «ش» تبدیل به «ر» می‌شود	کاش + ت+َن	کاشتن کار/ کاشت
8	فتن	- فت	پس از حذف «ت»، «ف» تبدیل به «ب» می‌شود	تاف+ ت+َن	تافتن تاب/ تافت

این روش بر روی گردایه‌ی بزرگ کلمه‌ها به کار گرفته شد. بن گذشته و غیرگذشته بیش‌تر فعل‌های ساده‌ی فارسی به خوبی شناسایی شدند.

3.8.روش‌های ریشه‌یابی

در این جا به روش‌های موجود ریشه‌یابی و دشواری‌های آن‌ها پرداخته می‌شود. هم‌چنین روشی پیشنهاد شده و پیاده‌سازی شده است. این روش به صورت وارون کار می‌کند به این معنا که مشتق فعل را می‌سازد و در صورت درست بودن آن مشتق، از آن پس ریشه آن مشتق بن فعل خواهد بود.

3.8.1. ریشه‌یاب‌های جدولی

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.

3.8.2.ریشه‌یابی به کمک روش‌های آماری

در این روش (یا 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.

3.8.3.ریشه‌یابی به کمک روش Porter یا شبیه به آن

روش 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.

به طور کلی می‌توان روش‌های ریشه‌یابی به کمک قاعده‌های زبان را در ادامه کار همین روش دانست. برای نمونه می‌دانیم که «گفت» ریشه گذشته یک فعل است و ریشه غیر گذشته آن «گو» می‌باشد؛ بنابراین با نگاهی به دستور زبان در می‌یابیم که ریشه همه کلمات زیر «گفت» می‌باشد : «گفتم»، «نگفتم»، «گفته‌ام» «می‌گوید»، «گوینده» ، «گفتار» و..... در ضمن به خاطر رعایت نشدن دستور خط باید «می‌گفتم»، «می‌گفت» و... را نیز همین گونه ریشه‌یابی کرد. پس باید قاعده‌های بسیاری نوشته شود و این قاعده‌های دستور زبان به کمک برنامه‌نویسی پیاده‌سازی شود. برای ریشه‌یابی فعل‌های فارسی باید یک زبان برنامه‌نویسی شایسته برگزیده شود. هنگام برنامه‌نویسی، باید حالت‌های گوناگونی که هنگام ترکیب کلمات فارسی پیش می‌آید، پوشش داده شود. هم‌چنین ویرایش‌های پیچیده و بسیار زیادی بر روی قاعده‌هایی از دستور زبان فارسی که به کار گرفته شده بود، باید انجام شود. به عبارت دیگر در هنگام برنامه‌نویسی، پیاپی بسیاری از بخش‌ها دگرگون می‌شود. به هیچ روی این امکان وجود ندارد که یک روند خطی برای طراحی و پیاده‌سازی نرم‌افزار در نظر گرفته شود. با توجه به این دشواری‌ها یک زبان برنامه‌نویسی بسیار ساده و

انعطاف‌پذیر و توانمند نیاز است. این زبان باید استاندارد یونی‌کد را به خوبی و سادگی پشتیبانی کند. در این زبان باید کار با رشته‌های یونی‌کدی نیز بسیار ساده باشد. دقت کنید که اگر ماشین پذیرنده متناهی[1] نیز طراحی شود و سپس بر پایه آن برنامه‌نویسی انجام گیرد؛ همواره این امکان وجود دارد که در هنگام طراحی ماشین پذیرنده‌ی متناهی قاعده‌هایی در نظر گرفته نشوند و پس از پایان کار برنامه‌نویسی چنین قاعده‌هایی باید به ماشین پذیرنده افزوده شوند که به این ترتیب باید دوباره کد برنامه تغییر یابد. به این ترتیب روند نگهداری از برنامه ریشه‌یاب و گسترش آن بسیار هزینه پر می‌شود.

3.9. ریشه‌یاب‌های کارشده در زبان فارسی

از جمله کارهای انجام شده در زمینه ریشه‌یابی کلمات فارسی می‌توان به پروژه بن [35]، ریشه‌یاب آماری [36] و [37] اشاره نمود.

در [35] 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

[1] Deterministic Finite Automata

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 [39] 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 $AXC \rightarrow AYC$ است و در آن A و C زمینه تبدیل را مشخص می‌کنند و x رشته‌ی ورودی و y رشته تغییر یافته است.

4. روش‌های جست و جو

4.1. مقدمه

منظور از جست‌وجو، پیدا کردن و تطبیق اسناد بر اساس کلمات کلیدی با پرس‌وجوهای مطرح شده می‌باشد. ما در ادامه روش‌های کلی جست‌وجو را آورده‌ایم.

4.2. جست وجوی دوارزشی

استراتژی جست و جوی دوارزشی، اسنادی را بازیابی می‌کند که برای پرس وجو مقدار True را داشته باشند[27]. 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.

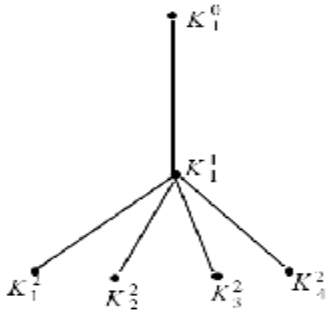
می توان برای متن D و پرس و جوی Q از یک دستورالعمل دو مرحله‌ای برای محاسبه مشابهت بین متن و پرس و جو استفاده کرد: هر کلمه‌ی q_i در Q با تابع $F(D,q_i)$ جایگزین می‌شود. مقدار این تابع در سیستم دوارزشی، اگر q_i در متن D باشد یک خواهد بود؛ و گرنه صفر خواهد بود. عملیات به وجود آمده از مرحله قبل با استفاده از یک جدول ارزیابی مثل جدول 4-1 پردازش می‌شود.

جدول 4-1. عبارات ارزیابی پرس‌وجوهای دوارزشی

عبارت	شیوه محاسبه
$F(D,t \text{ and } s)$	$\min(F(D,t),F(D,s))$
$F(D,t \text{ or } s)$	$\max(F(D,t),F(D,s))$
$F(D,\text{not } t)$	$1-F(D,s)$

بعضی از سیستم‌ها که با استفاده از جست و جوی دوارزشی پیاده‌سازی شده اند به کاربر اجازه می‌دهند تا جست و جو را محدودتر یا بازتر کند. این عمل را با دسترسی کاربر به یک لغت نامه‌ی ساخت‌یافته انجام می‌دهد که برای هر کلمه‌ی کلیدی دریافتی، کلمات کلیدی مرتبط که ممکن است خیلی عمومی‌تر یا خیلی ریزتر باشد را ذخیره می‌کند. برای مثال در ساختار درختی شکل5-1 کلمه کلیدی K_1^1 یک جزء ریزتر و دقیق‌تر برای کلمه‌ی کلیدی

K_1^0 بوده و خود K_1^1 به کلمات کلیدی K_1^2 و K_2^2 و K_3^2 و K_4^2 تقسیم می‌شود. بنابراین، اگر یکی یک سیستم فعل و انفعالی داشته باشد، جست و جو به راحتی می‌تواند دوباره فرموله‌سازی شود و از بعضی از این کلمات مرتبط نیز استفاده شود.



شکل 5-1. یک مجموعه از کلمات کلیدی مرتبط سلسله مراتبی

یک راه بدیهی برای پیاده‌سازی جست و جوی دوارزشی، به واسطه‌ی فایل وارونه می‌باشد. ما یک لیست برای هر کلمه‌ی کلیدی، در کلمه‌نامه ذخیره می‌کنیم و در هر لیست آدرس اسنادی را که کلمه‌ی مورد نظر را شامل می‌شود را قرار می‌دهیم. برای مثال، اگر لیست کلمات کلیدی به صورت زیر ذخیره شده باشند.

$K_1\ list\ : D_1, D_2, D_3, D_4$

$K_2\ list\ : D_1, D_2$

$K_3\ list\ : D_1, D_2, D_3$

$K_4\ list\ : D_1$

$$Q = (K_1\ AND\ K_2)\ OR\ (K_3\ AND\ (NOT\ \ K_4))$$

برای پاسخ دادن به بخش $(K_1\ AND\ K_2)$ از لیست K_1 و K_2 اشتراک می‌گیریم و برای پاسخ دادن به بخش $(K_3\ AND\ Not\ (K_4))$ لیست K_4 را از لیست K_3 کم می‌کنیم. OR با استفاده از گرفتن اجتماع از دو مجموعه‌ی به دست آمده برای بخشهای قبلی، پاسخ داده می‌شود. پاسخ، مجموعه اسناد $\{D_1, D_2, D_3\}$ می‌باشد که پرس و جو را ارضا می کند و هر سند در آن لیست برای پرس و جو مقدار true می‌باشد.

یک اصلاح جزئی برای روش جست و جوی دوارزشی کامل، تنها استفاده از عملگر AND می باشد، در این روش تعداد کلمات مشترک بین پرس و جو و اسناد مورد بررسی قرار می‌گیرد. این عدد به دست آمده، سطح هماهنگی نامیده می‌شود. روش جست و جو اغلب تطبیق ساده نامیده می‌شود. چون در هر سطح، ما می‌توانیم بیش از یک سند داشته باشیم، اسناد با استفاده از سطح هماهنگی مرتب می‌شوند.

برای مثال مشابه $Q = K_1\ AND\ K_2\ AND\ K_3$ را در نظر بگیرید. رتبه بندی که به دست می‌آید به صورت زیر است:

سطح هماهنگی

3	D_1, D_2
2	D_3
1	D_4

در واقع تطبیق ساده استفاده از توابع تطبیق اولیه می‌باشد. برای هر سند D، عبارت $|D \mathbf{I} Q|$ که اندازه‌ی رویهم افتادگی بین D و Q است، محاسبه می‌شود. D,Q با استفاده از کلمات کلیدی نشان داده می‌شوند.

4.3. تعمیم جست وجوی دوارزشی به فازی

وقتی که از وزن کلمات و معیارهای مشابهت برای بازیابی اسناد استفاده می‌شود، دو راه حل جهت بازیابی اسناد به نظر می‌رسد [27].

راه حل اول، استفاده از یک مقدار آستانه برای شباهت بین متن و پرس و جو می‌باشد. در این راهکار متن‌هایی که میزان مشابهت آن‌ها با پرس و جو از حد آستانه بالاتر باشد، ارزیابی می‌شوند.

راه حل دوم، استفاده از یک قطع کننده (تعداد) برای شباهت بین متن و پرس و جو می‌باشد. در این روش، اسناد بر اساس میزان مشابهت با پرس و جوها مرتب می‌شوند و به همان ترتیب بازیابی می‌شوند، با این شرط که تعداد اسناد بازیابی شده از مقدار قطع کننده بیش‌تر نشود.

$F(D,q_i)$: اگر q_i در متن D وزنی برابر k داشته باشد تابع، مقدار k را که عددی بین 0 و 1 است به خود می‌گیرد. در این روش کلمات پرس و جو وزنی به خود نمی‌گیرند. بعضی از روش‌های پیاده‌سازی مدل فازی در ادامه آورده شده است.

4.3.1. مدل mixed min & max

شباهت از ترکیب خطی کمینه‌گیری و بیشینه‌گیری روی کلمات پرس و جو در متن به صورت روابط (4-1) و (4-2) است [37]:

$$F(D,\text{t or s or}\ldots) = C_1 * \max(F(D,t), F(D,s), \ldots.) + C_2 * \min(F(D,t), \ldots.) \quad (4\text{-}1)$$

$$F(D,\text{t and s and}\ldots) = C_3 * \min(F(D,t), F(D,s), \ldots.) + C_4 * \max(F(D,t), \ldots.) \quad (4\text{-}2)$$

برای اهمیت دادن به min در and و یا max در or از قاعده زیر می‌توان استفاده کرد:

$$C_1 > C_2 \quad \text{و} \quad C_3 > C_4$$

4.3.2. مدل paice

در مدل paice وزن تمام کلمات پرس و جو در متن به حساب می‌آید [41]. اگر n تعداد کلمات پرس و جو باشد و $O_1 \ldots\ldots O_n$ وزن کلمات پرس و جو در متن D باشد که به صورت نزولی مرتب شده‌اند و $a_1 \ldots\ldots a_n$ وزن کلمات پرس و جو در متن D باشد که به صورت صعودی مرتب شده اند آن‌گاه:

$$F(D,\text{t or s or}\ldots)=\frac{\sum_{i=1}^{n} r^{i-1}o_i}{\sum_{i=1}^{n} r^{i-1}} \qquad (3\text{-}4)$$

$$F(D,\text{t and s and}\ldots)=\frac{\sum_{i=1}^{n} r^{i-1}a_i}{\sum_{i=1}^{n} r^{i-1}} \qquad (4\text{-}4)$$

$$F(D,\text{not t})=1-\text{F(D,t)} \qquad (5\text{-}4)$$

پارامتر r برای تعیین میزان تاثیرگذاری وزن‌های بالا یا پائین در محاسبات است. اگر r=1 باشد، مقدار به دست آمده برای or یا and یکسان خواهد بود.

این روش نسبت به روش mixed min & max پرهزینه‌تر است، چون در این روش نیاز به مرتب‌سازی است و هزینه‌ی زمانی آن از مرتبه O(nlogn) است در حالیکه روش mixed min & max فقط نیاز به پیدا کردن max و min دارد و مرتبه زمانی آن O(n) می‌باشد.

4.4. مدل آستانه‌ای

اگر هم کلمات متن و هم کلمات پرس و جو وزن دار باشند، می‌توان از استراتژی‌های جداگانه برای وزن کلمه‌ها در پرس و جو و متن استفاده کرد [40]. مثلاً برای متن‌ها از همان وزن کلمه استفاده شده و برای پرس و جوها از مقادیر آستانه‌ای استفاده شود.

با استفاده از مقادیر آستانه ای، اگر وزن کلمه‌ها در متن، کم‌تر از آستانه مشخص شده باشد اثر آن‌ها به صفر کاهش پیدا می‌کند. به عنوان نمونه اگر در فرمول زیر m (مقدار آستانه) وزن کلمه q_{sj} در پرس و جوی Q باشد:

$$\begin{cases} d_{ri}.q_{sj} = d_{ri} & if \ \ \mathrm{d_{ri}} \geq m \\ d_{ri}.q_{sj} = 0 & if \ \ \mathrm{d_{ri}} < m \end{cases} \qquad (6\text{-}4)$$

مثال:

در پاسخ به پرس و جوی $(A_a \text{ or } B_b)$ که a وزن کلمه A و b وزن کلمه B در پرس و جو می‌باشد، متن‌هایی بازیابی می‌شوند که کلمه A را با حداقل وزن a و یا کلمه B را با حداقل وزن b داشته باشند.

4.5. توابع تطبیق

بسیاری از روش‌های جست و جوی خوب و ماهر از تابع تطبیق استفاده می‌کنند. این تابع شبیه به یک معیار وابستگی است ولی تفاوت آن‌ها در این است که تابع تطبیق ارتباط بین پرس و جو و اسناد یا خوشه‌ها را اندازه‌گیری می‌کند، ولی معیار وابستگی روی اشیاء با یک نوع مشابه اعمال می‌شود. از لحاظ ریاضی هر دو تابع، ویژگی‌های یکسانی دارند، تفاوت

آن‌ها تنها در تفسیر آن‌هاست. مثال‌های زیادی از توابع تطبیق در کاربردها وجود دارد، که ساده‌ترین آن‌ها در ارتباط با استراتژی جست و جوی تطبیقی ساده می‌باشد.

اگر M تابع تطبیق، D مجموعه کلمات کلیدی بیانگر سند و Q مجموعه بیانگر پرس و جو باشد، آن‌گاه:

$$M = \frac{2|D \mathbf{I} Q|}{|D| + |Q|} \qquad (7\text{-}4)$$

M یک مثال از تابع تطبیق می‌باشد. M شبیه ضرائب Dice می‌باشد.

4.6. روش دوارزشی تعمیم‌یافته

روش دوارزشی ساده یکی از دو حالت درست یا غلط را دارد. مثل شرط همه یا هیچ، روش دوارزشی ساده تعداد زیادی سند را بازیابی می‌کند و یا هیچ سندی را بازیابی نمی‌کند. روش دوارزشی قدیمی هم‌چنین سعی بر ایجاد نتایج مخالف درک مستقیم دارد، به علت ویژگی همه یا هیچ، به عنوان مثال در پاسخ به یک پرسش OR چند کلمه ای، یک سند شامل همه یا بیش‌تر کلمات پرس وجو، هیچ برتری نسبت به سندی که فقط شامل یک کلمه از عبارت پرس وجو هست، ندارد. و به همین ترتیب در یک پرس وجوی AND چند کلمه ای، سندی که همه کلمات پرس وجو، به جز یکی دارد به همان اندازه‌ای بد تلقی می‌شود که انگار هیچ کدام از کلمات پرس وجو را ندارد. تعدادی مدل دوارزشی تعمیم‌یافته

برای مهیا کردن یک ترتیب در بین اسناد بازیابی شده پیشنهاد شده است نحوه‌ی عملکرد این روش‌ها به صورتی است که بعضی از اسناد خیلی بیش‌تر از سایر اسناد، شرط پرس وجو را ارضا می‌کنند و این اساس رتبه‌بندی است. به روش‌های دوارزشی تعمیم‌یافته روش‌های دوارزشی نرم نیز می‌گویند.

عملگرهای دوارزشی تعمیم‌یافته از اختصاص وزن به کلمات اسناد استفاده می‌کنند [40]. در یک عملگر دوارزشی کلاسیک این وزن‌ها برابر با 0 یا 1 بودند. اگر کلمه مورد نظر در سند بود مقدار 1(درست و در بازیابی اطلاعات، تطبیق باشند) و اگر کلمه مورد نظر در سند موجود نباشد مقدار 0 (غلط و در بازیابی اطلاعات، عدم تطبیق باشند) برگردانده می‌شود. یک عملگر دوارزشی تعمیم‌یافته آرگومان‌های خود را با اعدادی در بازه‌ی [0,1] با توجه به درجه تطبیق عبارت منطقی داده شده با سند، ارزیابی می‌کند.

Lee چند مدل از روش دوارزشی تعمیم‌یافته را ارائه داد و به وسیله‌ی معیارهای اهمیت مشخص اثبات کرد، یک روش که P-NORM نامیده می‌شود ویژگی‌های جالبی داشت. و مطلوب‌ترین بود.

"مطلوب‌ترین" به این معنی است که مدل‌های P-norm گرایش به ارزیابی میزان تطابق سند با پرس وجو با توجه به داوری انسانی، نسبت به سایر روش‌ها دارند. برای هر کدام از سایر روش‌های آزمایش شده، حالاتی وجود دارد که ارزیابی مدل‌ها از میزان تطابق سند و پرس وجو با شهود کاربر انسانی تفاوت دارد. در هرکدام از آن حالات ارزیابی تطبیق مدلهای P-norm موافق شهود کاربر انسانی است.

یک پرس وجو را با n کلمه به صورت $t_1, t_2, ..., t_n$ با وزن‌های متناظر $w_{q1}, w_{q2}, ..., w_{qn}$ در نظر بگیرید. ویک سند D با وزنهای متناظر $w_{d1}, w_{d2} .... w_{dn}$ به کلمات مشابه در پرس وجو درنظر بگیرید. مدل P-norm توابع مشابهت AND دو ارزشی توسعه داده شده و OR دو ارزشی توسعه داده شده را برای n کلمه تعریف می‌کند. تابع AND دو ارزشی تعمیم‌یافته، میزان مشابهت سند داده شده را با پرس وجو مورد نظر که کلمات آن با هم AND شده اند را محاسبه می‌کند و به همین ترتیب تابع OR دو ارزشی تعمیم‌یافته میزان مشابهت سند داده شده با پرس وجو مورد نظر که کلمات آن با هم OR شده اند را محاسبه می‌کند. هر مشابهت محاسبه شده عددی در بازه [1، 0] می‌باشد . اکثر پرس و جوهای دو ارزشی با استفاده از AND و OR ساخته می‌شود. توابع دو ارزشی تعمیم‌یافته برای مدل P-norm به صورت روابط (4-7) و (4-8) میباشد:

$$SIM_{AND}\left(d,(t_1,w_{q1})\right) AND .... AND \left(t_n,w_{qn}\right) = 1 - \left[\frac{\sum_{i=1}^{n}\left((1-w_{di})^p w_{qi}^p\right)}{\sum_{i=1}^{n} w_{qi}^p}\right]^{\frac{1}{p}} \qquad (7\text{-}4)$$

$$SIM_{OR}\left(d,(t_1,w_{q1}) OR .... OR \left(t_n,w_{qn}\right)\right) = \left[\frac{\sum_{i=1}^{n} w_{di}^p . w_{qi}^p}{\sum_{i=1}^{n} w_{qi}^p}\right]^{\frac{1}{p}} \quad 1 \le p \le \infty \qquad (8\text{-}4)$$

مدل P-norm یک پارامتر p دارد که با استفاده از آن می‌توانیم مدل را وفق (tune) دهیم. پارامتر p از یک تا بی نهایت متغیر است و تفسیر واضحی دارد. اگر p برابر بی نهایت شود،

مدل P-norm تبدیل به مدل کلاسیک دو ارزشی می‌شود. برای AND اگر تمام کلمات پرس وجو در داخل سند موجود باشد مقداری مشابهت یک خواهد بود و برای OR اگرفقط یکی از کلمات پرس و جو داخل سند باشد مقدار مشابهت یک می‌شود و هیچ تفاوتی با این که دو تا یا بیش‌تر کلمات موجود باشد ندارد. یا به عبارت دیگر AND سخت، تکلیف مطابقت می‌کند و OR با فرهنگ جامع طبقه‌ای سخت، تکلیف مطابقت می‌کند . در بازه‌های پایین p مثل 5 تا 2= p AND با عبارت سست، تکلیف مطابقت می‌کند، به طوری که "وجود کلیه کلمات پرس وجو در سند، خیلی بهتر ازوجود تنها یک کلمه با وزن کم، اساساً، امتیاز مشابهت کلی را پایین می‌آورد، هر چند که سایر کلمات وزن‌های بالایی داشته باشند به عبارت دیگر P-norm AND از AND دو ارزشی متفاوت است به طوری که یک کلمه، با وزن صفر، مثلاً یک کلمه‌ای که اصلا در سند وجود ندارد، امتیاز مشابهت دار صفر کاهش نمی‌دهد.

همین‌طور، با pهای کم، OR با فرهنگ جامع طبقه‌ای سست، تکلیف مطابقت می‌کند. بدین معنی که "وجود چند کلمه از یک کلاس یا سند خیلی بهتر از وجود فقط یک کلمه است". به عبارت دیگر، P-norm OR این محدودیت OR دو ارزشی را تعمیم میدهد که یک کلمه با وزن بالا می‌تواند یک امتیاز بالایی برای مشابهت را ایجاد کند، حتی اگر سایر کلمات دارای وزن صفر باشند. سپس P-norm OR از OR دوارزشی متفاوت است زیرا یک کلمه با وزن بالا برای بالا بردن میزان مشابهت کافی نیست، کلمات اضافی یا وزن‌های غیرصفر میزان مشابهت را بالا می‌برند.

در p=1 مدل P-norm به حالت خطی مدل " فضای برداری" ساده می‌شود که در بخش بعد تشریح می‌شود بدین معنی که کلمات از یکدیگر مستقل هستند، برتری بین عبارات و فرهنگ جامع طبقه‌ای کاملا معلوم نیست. در حقیقت در p=1 OR و AND یکسان میشوند [CAcm 83 SALTon] . هر دوی آن‌ها به معیار مشابهت cosine تبدیل می‌شوند.

OR و AND کلاسیک دو دویی بودند و در آن‌ها خاصیت انجمنی حاکم بود. مثلاً $t_1 AND (t_2 AND\ t_2)$ معادل $(t_1 AND\ t_2) AND\ t_3$ می‌باشد. ولی این خاصیت در AND و OR تعمیم‌یافته P-norm صحیح نیست.

4.7. مدل فضای برداری

مدل فضای برداری پایه‌ای‌ترین مدل در سیستم‌های بازیابی اطلاعات است که توسط Salton ابداع شد [27, 42]. 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 برداری قرار دارد. در این فضا که بسته به تعداد کلمات مجموعه یک فضای n بعدی است، بردار هر سند ترسیم می‌شود. پرسش کاربر نیز بعد از اعمال فعالیت‌های نمایه‌سازی به برداری تبدیل می‌شود که در فضای جدید 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 هر یک از بردارهای سند می‌سازد. این میزان نزدیکی، معمولا با رابطه (9-4) که به نام مشابهت کسینوسی شناخته می‌شود، محاسبه می‌گردد:

$$sim(q_i, d_j) = \frac{\vec{q}_i . \vec{d}_j}{|\vec{q}_i| \times |\vec{d}_j|} = \frac{\sum_{k=1}^{t} w_{ki} \times w_{kj}}{\sqrt{\sum_{k=1}^{t} w_{ki}^2} . \sqrt{\sum_{k=1}^{t} w_{kj}^2}} \qquad (9\text{-}4)$$

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.

5. جمع‌بندی

زبان فارسی یکی از غنی‌ترین زبان‌های دنیا می‌باشد و به طبع آن بازیابی اطلاعات فارسی امری مهم خواهد بود. کلمات کلیدی عناصر بسیار مهمی در بازیابی اطلاعات هستند. ما در این مطالعه، به بررسی جایگاه زبان فارسی و مشکلات پردازش رایانه‌ای آن، پرداختیم. یکی از مشکلات زبان فارسی، شناسایی مرز لغات و جملات در یک متن است. تعیین مرز کلمات در زبان فارسی به دلیل گوناگونی رسم‌الخط و عدم وجود استانداردهای نگارشی و هم‌چنین به دلیل وجود شکل‌های مختلف حروف (اول – وسط – آخر و چسبان و غیرچسبان) بیش از زبان انگلیسی مشکل ساز است. این مشکل در زبان انگلیسی تنها برای کلمات مرکب ممکن است رخ دهد که آن‌ها را می‌توان در مراحل بعدی پردازش مثل پردازش نحوی تشخیص داد. اما در زبان فارسی علاوه بر کلمات مرکب که مشکلی مشابه با انگلیسی ایجاد می‌کنند، مرز کلمات غیرمرکب نیز ممکن است بدرستی تشخیص داده نشود.

ما در این مطالعه روش‌های استخراج کلمات کلیدی برای متون غیرساخت‌یافته فارسی را بررسی کردیم. متون غیرساخت‌یافته متونی هستند که اطلاعات ساختاری به ما نمی‌دهند. روش‌های مختلف استخراج کلمات کلیدی به ترتیب زیر می‌باشد:

1. روش‌های آماری مبتنی بر تحلیل فراوانی کلمات.

2. روش‌های نحوی مبتنی بر تجزیه زبانی[1] و انطباق الگو.
3. روش‌های ساختاری بررسی عنوان و رئوس کلی مطالب سند.
4. روش‌های ادراکی مبتنی بر استفاده از پایگاه دانش برای تفسیر معنی و مفهوم.

چون روش‌های آماری، نیازی به اطلاعات ساختاری ندارند، یکی از روش‌های مناسب برای متون غیرساخت‌یافته هستند. انواع روش‌های آماری مختلف وزن‌دهی به کلمات در متون غیرساخت‌یافته در این گزارش معرفی شده‌اند.

بازخوانی و دقت دو معیار مهم برای ارزیابی کلمات کلیدی استخراج شده، هستند. این دو معیار با هم مصالحه دارند، و بهبود یکی باعث افت دیگری می‌شود. بیش‌ترین دقت گزارش شده برای سیستم بازیابی اطلاعات [25] فارسی 66% می‌باشد.

[1] Linguistic

مراجع

[1] شمس‌فرد م. (1385)، پردازش متون فارسی: دستاوردهای گذشته، چالش‌های پیش رو، دومین کارگاه پژوهشی زبان فارسی و رایانه، ص. 172 تا 189، تهران.

[2] داداش میری پ.، (1380)، «تشخیص انتهای کلمات و ایجاد فاصله میان کلمات»، پایان‌نامه کارشناسی، دانشگاه علم و صنعت ایران.

[3] بیجن‌خان م.، (1384)، «تشخیص کسره‌ی اضافه»، طرح تحقیقاتی، پژوهشگاه فرهنگ هنر و ارتباطات، تهران.

[4] M. Arabsorkhi, M. Shamsfard, "Unsupervised Discovery of Persian Morphemes," in 11th Conference of the European chapter of the Association for Computational Linguistics (EACL), Italy, 2006.

[5] K. Megerdoomian and Z. Rémi, “Processing Persian Text: Tokenization in the Shiraz Project," in *Memoranda in Computer and Cognitive Science* (MCCS-00-322), 2000.

[6] پورحسن م.، (1385)، «تحلیل‌گر ساخت‌واژی زبان فارسی»، پایان‌نامه‌ی کارشناسی، دانشگاه شهید بهشتی، تهران.

[7] قاسمی‌زاده ب.، رحیمی س.، سالاریان م. ترکمنی ع. سمیاری ح.، کوچاری ع.، نم نبات م.، براری ل.، (1384)، «روشی نوین برای صرف واژه‌های فارسی»، یازدهمین کنفرانس بین‌المللی انجمن کامپیوتر ایران، تهران.

[8] نوربالا، م.، (1380)، "طراحی یک واژگان جامع برای پردازشگر زبان فارسی"، پایان‌نامه کارشناسی ارشد، دانشگاه علم و صنعت.

[9] Y. Matsuo, and M. Ishizuka, "Keyword Extraction from a Single Document using Word Co-occurrence Statistical Information," in *Proceedings of the Sixteenth International Florida Artificial Intelligence Research Society Conference*, pp. 392-396, 2003.

[10] رستم‌پور، س.، (1385)، "واژگان محاسباتی زبان فارسی"، پایان‌نامه کارشناسی، دانشگاه شهید بهشتی.

[11] بیجن‌خان م.، (1383)، "پیکره متنی زبان فارسی"، مجموعه سخنرانی‌های اولین کارگاه پژوهشی زبان فارسی و رایانه، تهران.

[12] F. Oroumchian, E. Darrudi, and M.R. Hejazi, "Assessment of a modern farsi corpus, " in *Proceedings of The 2nd Workshop on Information Technology & its Disciplines (WITID)*, Iran, 2004.

[13] K. Sheykh Esmaili, H. Abolhassani, M. Neshati, E. Behrangi, A. Rostami and M. Mohammadi Nasiri, "Mahak: A Test Collection for Evaluation of Farsi Information Retrieval Systems," in Proceedings of *5th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA-07)*, Amman, Jordan, May 2007.

[14] K. Sheykh Esmaili, A. Rostami, "List of Persian Stopwords," Technical Report No. 2006-03, Semantic Web Research Laboratory, Sharif University of Technology, Tehran, Iran, June 2006.

[15] N. Mazdak, "FarsiSum-a persian text summarizer", Master thesis, Department of linguistics, Stockholm University, 2004.

[16] اشراق، ف.، سارابی، ز. (1385) "سیستم پرسش و پاسخ به زبان فارسی" پایان‌نامه کارشناسی، دانشگاه شهید بهشتی.

[17] قیومی، م.، (1383)، "پیش‌بینی رایآن‌های کلمه"، مجموعه سخنرانی‌های اولین کارگاه پژوهشی زبان فارسی و رایانه، تهران.

[18] محروقی، ح.، (1375)، " طراحی و پیاده‌سازی سیستمی برای ویرایش ادبی جملات ساده زبان فارسی"، پایان‌نامه کارشناسی ارشد، دانشکده مهندسی کامپیوتر، دانشگاه صنعتی شریف.

[19] زاهدی، (1377)، "طراحی و پیاده‌سازی یک برنامه هوشمند برای اعراب گذاری در متون فارسی"، پایان‌نامه کارشناسی ارشد، دانشگاه تهران.

[20] باقری، م.، (1372)، "استنباط موضوعات مشترک از جملات مرتبط به هم"، پایان‌نامه کارشناسی ارشد، دانشکده مهندسی کامپیوتر، دانشگاه صنعتی شریف.

[21] M. W. Berry, *Survey of Text Mining : Clustering, Classification, and Retrieval*, Springer, 2003.

[22] رضانیا ک.، (1376)، «پی‌ریزی طرح کلی واژگان و طراحی و پیاده‌سازی پردازشگر ساخت‌واژی برای زبان فارسی»، پایان‌نامه‌ی کارشناسی ارشد، دانشکره‌ی مهندسی کامپیوتر، دانشگاه صنعتی شریف.

[23] G.K. Zipf, *Human Behaviour and the Principle of Least-Effort*, Addison-Wesley, 1949.

[24] اسلامی م.، شریفی م.، علیزاده ص.، زندی ط.، (1383)، «واژگان زایای فارسی»، مجموعه سخنرانی‌های اولین کارگاه پژوهشی زبان فارسی و رایانه، تهران.

[25] بشیری، ح، کربلائی، ف.، موسوی، ش،(1384)، "طراحی و ارزیابی نمایه‌ساز خودکار متون فارسی"، مجموعه مقالات یازدهمین کنفرانس بین‌المللی کامپیوتر.

[26] تشکری، م.، (1380)، " ساخت یک نمایه‌ساز خودکار برای متون فارسی"، پایان‌نامه کارشناسی ارشد، دانشکده مهندسی کامپیوتر، دانشگاه صنعتی امیر کبیر.

[27] G. Salton, M.J Mc Gill, *Introduction to Modern Information Retrieval*, Mc Graw Hill, New York, 1983.

[28] A. Hulth, "Improved Automatic Keyword Extraction Given More Linguistic Knowledge," in *Proceedings of the Conference on Empirical Methods in Natural Language Procezsing (EMNLP'OB),* Sapporo, Japan, 2003.

[29] T.A. Runkler, and J.C. Bezdek, "Automatic keyword extraction with relational clustering and Levenshtein distances," *FUZZ-IEEE* , Vol. 2, pp. 636-640, 2000.

[30] Y. Otani, H. Kawanaka, T. Yoshikawa, K. Yamamoto, T. Shinogi and S. Tsuruoka, "Keyword Extraction from Incident Reports and Keyword Map Generation Method Using Self Organizing Map", in *Proc. of IEEE International Conference on Systems, Man and Cybernetics (IEEE SMC 2005),* pp.1024-1029, 2005.

[31] S.K. Pal, V. Talwar, and P. Mitra, "Web Mining in Soft Computinb32 Framework Relevance, State of the Art and Future Directions," *IEEE Transactions on Neural Networks*, Vol. 13, No. 5, pp.1163 -1177, 2002.

[32] یوسفان، ا.، صالحی زارعی، س.، مینایی بیدگلی، ب.، (1385)، "دشواری‌های ریشه‌یابی فارسی و روشی برای ریشه‌یابی فعل‌های ساده فارسی " دومین کارگاه پژوهشی زبان فارسی و رایانه، ص. 172 تا 189، تهران.

[33] دستور خط، فرهنگستان زبان و ادب فارسی، 1380.

[34] انوری، ح.، احمدی گیوی، ح.، (1380) "دستور زبان فارسی 2 . تهران: انتشارات فاطمی، چاپ بیست‌ویکم.

[35] 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.

[36] محمدی نصیری، م.، شیخ اسماعیلی، ک.، ابولحسنی، ح.، ،(1384)، "یک ریشه‌یاب آماری برای زبان فارسی"، مجموعه مقالات یازدهمین کنفرانس بین‌المللی کامپیوتر.

[37] K. Taghva, R. Beckley and M. Sadeh. ”A Stemming Algorithm for the Farsi Language,” in *proceedings of International Conference on Information Technology: Coding and Computing (ITXX05)* - Volume I, pp. 158-162, 2005.

[38] E.A. Fox, and S. Sharat, “A comparison of two methods for soft Boolean interpretation in information retrieval,” technical report TR-86-1, Department of Computer Science, Virginia Tech, Blacksburg, VA, 1986.

[39] C.P. Paice, “Soft evaluation of Boolean search queries in information retrieval systems,” Information Technology: Research, Development, Applications, Vol. 3 No. 1,pp. 33-41, 1984.

[40] G. Salton, *Automatic Text Processing*, Reading, MA: Addison Wesley, 1989.