دانشکده مهندسی کامپیوتر و فناوری اطلاعات

پایان نامه کارشناسی ارشد

دسته‌بندی اسناد وب با استفاده از گراف نمایه سازی اسناد

نرجس رمضانی اومالی

استاد راهنما :

دکتر مرتضی زاهدی

شهریور ۱۳۹۲

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

گروه هوش مصنوعی

دسته‌بندی اسناد وب با استفاده از گراف نمایه سازی اسناد

نرجس رمضانی اومالی

استاد راهنما :

دکتر مرتضی زاهدی

استاد مشاور:

دکتر حمید حسن‌پور

پایان نامه ارشد جهت اخذ درجه کارشناسی ارشد

شهریور ۱۳۹۲

دانشگاه صنعتی شاهرود

دانشکده : مهندسی کامپیوتر و فناوری اطلاعات

گروه : هوش مصنوعی

پایان نامه کارشناسی ارشد خانم نرجس رمضانی اومالی

تحت عنوان: دسته‌بندی اسناد وب با استفاده از گراف نمایه سازی اسناد

در تاریخ ..... توسط کمیته تخصصی زیر جهت اخذ مدرک کارشناسی ارشد مورد ارزیابی و با درجه ..... مورد پذیرش قرار گرفت.

اساتید راهنما	امضاء	اساتید مشاور	امضاء
نام و نام خانوادگی :		نام و نام خانوادگی :	
نام و نام خانوادگی :		نام و نام خانوادگی :	

اساتید داور	امضاء	نماینده تحصیلات تکمیلی	امضاء
نام و نام خانوادگی :		نام و نام خانوادگی :	
نام و نام خانوادگی :			
نام و نام خانوادگی :			
نام و نام خانوادگی :			

تشکر و قدردانی

حمد و سپاس بیکران به درگاه قادر بی همتا که زیور خرد را بر آدمیان ارزانی داشت تا راه زندگی را با نیروی تعقل و دانش اندوزی بپیمایند. کردگار عظیم را شاکرم که به من توفیق آموختن عطا فرمود تا بدانم آنچه را که نمی‌دانم و قدرت بی انتهایش را بیشتر درک کنم.

حال که به لطف و رحمت لایتناهی حضرت حق مراحل این پایان نامه رو به اتمام نهاده بر خود لازم میدانم تا از همه کسانی که در پیشبرد اهداف این پایان نامه مرا یاری نمودند سپاس و قدردانی به عمل آورم.

از خانواده عزیزم که به حق در تمام مراحل زندگی و تحصیل مرا یاری نموده‌اند و بعد از خداوند متعال تکیه‌گاه و پشتیبان من بوده‌اند بسیار سپاسگزارم.

از زحمات و پشتیبانی بی دریغ و بی شائبه استاد ارجمندم جناب آقای دکتر زاهدی که راهنمایی این تحقیق را بر عهده داشتند کمال تشکر را دارم. بی‌شک بدون حمایت و هم فکری ایشان انجام این پایان نامه مقدور نبود.

در پایان از اساتید گران‌قدر جناب آقای دکتر حسن‌پور و جناب آقای دکتر پویان که سعادت شاگردی ایشان را در دوره‌های کارشناسی و کارشناسی ارشد داشته ام قدردانی کرده و از خداوند متعال برای این‌دو بزرگوار موفقیت و بهروزی مسألت دارم.

تعهد نامه

اینجانب نرجس رمضانی اومالی دانشجوی دوره کارشناسی ارشد رشته رشته هوش مصنوعی دانشکده مهندسی فنآوری اطلاعات و کامپیوتر دانشگاه شاهرود نویسنده "پایان نامه دسته‌بندی اسناد وب با استفاده از گراف نمایه سازی اسناد" تحت راهنمائی آقای دکتر مرتضی زاهدی متعهد می شوم .

- تحقیقات در این پایان نامه توسط اینجانب انجام شده است و از صحت و اصالت برخوردار است .
- در استفاده از نتایج پژوهشهای محققان دیگر به مرجع مورد استفاده استناد شده است .
- مطالب مندرج در پایان نامه تاکنون توسط خود یا فرد دیگری برای دریافت هیچ نوع مدرک یا امتیازی در هیچ جا ارائه نشده است .
- کلیه حقوق معنوی این اثر متعلق به دانشگاه صنعتی شاهرود می باشد و مقالات مستخرج با نام « دانشگاه صنعتی شاهرود » و یا « Shahrood University of Technology » به چاپ خواهد رسید .
- حقوق معنوی تمام افرادی که در به دست آمدن نتایج اصلی پایان نامه تأثیرگذار بوده اند در مقالات مستخرج از پایان نامه رعایت می گردد.
- در کلیه مراحل انجام این پایان نامه ، در مواردی که از موجود زنده (یا بافتهای آنها) استفاده شده است ضوابط و اصول اخلاقی رعایت شده است .
- در کلیه مراحل انجام این پایان نامه، در مواردی که به حوزه اطلاعات شخصی افراد دسترسی یافته یا استفاده شده است اصل رازداری ، ضوابط و اصول اخلاق انسانی رعایت شده است .

تاریخ

امضای دانشجو

مالکیت نتایج و حق نشر

- کلیه حقوق معنوی این اثر و محصولات آن (مقالات مستخرج ، کتاب ، برنامه های رایانه ای ، نرم افزار ها و تجهیزات ساخته شده است) متعلق به دانشگاه صنعتی شاهرود می باشد . این مطلب باید به نحو مقتضی در تولیدات علمی مربوطه ذکر شود .
- استفاده از اطلاعات و نتایج موجود در پایان نامه بدون ذکر مرجع مجاز نمی باشد.

چکیده:

با رشد روزافزون اطلاعات در وب، اغلب پروژه‌های تحقیقاتی در این حوزه با هدف سازماندهی اطلاعات شکل می‌گیرند، به گونه‌ای که کاربر نهایی بتواند راحت‌تر و با سرعت بیشتر به اطلاعاتی با دقت بالا و کارایی بهینه دست‌یابد.

دسته‌بندی اسناد ابزاری مهم در بسیاری از امور مربوط به بازیابی اطلاعات است. اغلب تکنیک‌های خوشه‌بندی اسناد مانند مدل فضای برداری برپایه‌ی تحلیل کلمات منفرد، در مجموعه داده موجود در سند هستند. کلمات منفرد به تنهایی فاقد اطلاعات کافی بوده و باعث بروز خطا در دسته‌بندی می‌شود. جهت دستیابی به خوشه‌بندی دقیق‌تر استفاده از ویژگی‌هایی حاوی اطلاعات بیشتر مانند عبارات و وزن آن عبارات در اسناد می‌تواند بسیار مفید باشد. روش‌های دیگری چون درخت پسوندی اگرچه از عبارات جهت دسته‌بندی استفاده می‌کنند ولی با افزایش تعداد اسناد به دلیل افزونگی[1] بالا فاقد کارایی لازم هستند.

در این میان مدل جدید نمایه‌سازی سند براساس عبارت با عنوان Document Index Graph یک روش دسته‌بندی مبتنی بر گراف است که در سال ۲۰۰۴ مطرح شده است. این مدل به دلیل استفاده از عبارات نسبت به مدل‌های مبتنی بر کلمات منفرد بسیار کاراتر است. در این روش به صورت موثر انطباق عبارات جهت بررسی شباهت بین اسناد انجام می‌شود. این مدل به دلیل استفاده از ساختار گراف فاقد افزونگی بوده و در دسته‌بندی از هر تعداد سند پشتیبانی می‌کند. همچنین به دلیل ساختار افزایشی الگوریتم دسته‌بندی، قابلیت به‌کارگیری به صورت آنلاین در وب را نیز دارد. استفاده از این مدل، نتایج دسته‌بندی اسناد وب را در مقایسه با روش‌های سنتی تاحد چشم‌گیری بهبود می‌بخشد.

این پایان‌نامه به بررسی روش‌های مختلف دسته‌بندی اسناد و نقاط قوت و ضعف هرکدام پرداخته و با تمرکز بر روش دسته‌بندی مبتنی بر گراف به بررسی این روش و مزایای آن نسبت به روش‌های قبلی

[1] Redundancy

می‌پردازد، در ادامه با توجه به این‌که این سیستم قابلیت استفاده در موتور جستجو را جهت دسته-بندی اسناد بازیابی شده دارد، با نگاهی دیگر از زاویه موتور جستجو به بررسی عملکرد این سیستم پرداخته و سعی در بهبود کارایی این سیستم در قالب موتور جستجو داریم.

اسناد بازیابی شده توسط موتور جستجو غالباً براساس میزان بازدید کاربران در لیست نتایج مرتب شده و در اختیار کاربر قرار می‌گیرند، با به‌کارگیری سیستم معرفی شده و اضافه کردن وزن‌هایی به نودها و یال‌های گراف می‌توان وزن عبارت مورد جستجو را در اسناد مختلف محاسبه و آن‌ها را براساس وزن عبارت مورد جستجو مرتب کرد، این کار سبب می‌شود کاربر با دقت و سرعت بیشتر به اطلاعات مورد نظر خود دست‌یابد.

برای اضافه کردن وزن با اصلاح ساختار گراف به ازای هر سند وزن نودها را با شمارش و وزن یال‌ها را با استفاده از یک شبکه‌عصبی پرسپترون محاسبه کرده و عملکرد سیستم را به عنوان بخشی از یک موتور جستجو بهبود می‌دهیم.

کلمات کلیدی:

دسته‌بندی اسناد، نمایه‌سازی مبتنی بر عبارت، گراف نمایه‌سازی اسناد، گراف فازی

فهرست مطالب

عنوان **صفحه**

فهرست شکل‌ها

عنوان **صفحه**

فهرست جدول‌ها

عنوان **صفحه**

فصل اول

مقدمه

۱-۱- مقدمه‌ای بر دسته‌بندی اسناد:

با توسعه‌ی سیستم‌های اطلاعاتی، داده به یکی از منابع پراهمیت سازمان‌ها مبدل گشته است، بنابراین روش‌ها و تکنیک‌هایی برای دستیابی کارا به داده، اشتراک داده، استخراج اطلاعات از داده و استفاده از این اطلاعات، مورد نیاز می‌باشد. طی چندین دهه‌ی گذشته به سبب ایجاد و گسترش وب و ارتباطات موجود در فضای مجازی اینترنت، اطلاعات با رشد چشمگیری همراه بوده. در حال حاضر در نتیجه‌ی استفاده از اینترنت هر کسی امکان دسترسی به منابع بی‌پایان اطلاعات را از طریق وب دارد؛ در نتیجه امر سازماندهی این حجم بالا از اطلاعات هر روز بیش از پیش دچار چالش می‌شود.

افزایش چشمگیر حجم اطلاعات و توسعه وب، محققان بسیاری را به تلاش جهت ارائه‌ی روش‌ها و تکنیک‌های مختلف برای سازماندهی این حجم بالا از منابع اطلاعاتی ترغیب کرده است. معرفی موتورهای جستجو مانند Google و Yahoo با هدف بازیابی اطلاعات انجام شده‌است. این ابزار در جواب جستجوی کاربر لیستی از اسناد مرتبط با عبارت مورد جستجو را باز می‌گردانند. کاربر از ابتدای لیست شروع کرده و یکی یکی جواب‌ها را تا رسیدن به پاسخ مورد نظر بررسی می‌کند. با اینکه موتورهای جستجو برای جستجوهای دقیق مثل پیدا کردن صفحه‌ی اصلی سایت یک سازمان بسیار خوب عمل می‌کنند ولی برای جستجوهای غیردقیق و مبهم نتایج خوبی را ارائه نمی‌دهند. در این موارد، نتایج با زیرموضوع‌ها و معانی متفاوت به صورت ترکیبی در لیست موجود بوده، بنابراین کاربر باید برای پیدا کردن مطلب مورد نظر حجم زیادی از اطلاعات نامربوط را مورد بررسی قرار دهد. از طرف دیگر راهی وجود ندارد که بتوان دقیقاً تعیین کرد چه مبحثی به جستجوی کاربر مربوط است، زیرا عبارت جستجو اغلب بسیار کوتاه و تفسیر آن بدون وجود متن ذاتاً مبهم است. با این شرایط سازماندهی اطلاعات می‌تواند سبب کاهش چشمگیر زمان جستجو، توسط کاربر شود.

اساس بسیاری از سیستم‌های پردازش اسناد وب، بر تکنیک‌های داده‌کاوی[1] استوار است. داده‌کاوی شامل تکنیک‌هایی است که می‌تواند ساختار ذاتی داده‌ها را آشکار کند، یکی از این تکنیک‌ها خوشه-بندی[2] است. با اعمال این تکنیک به داده‌های متنی، متدهای خوشه‌بندی تلاش می‌کنند تا دسته-بندی‌های ذاتی مجموعه اسناد را شناسایی کنند؛ به‌گونه‌ای که در مجموعه‌ی دسته‌های ایجاد شده شباهت درون‌دسته‌ای[3] بالا و شباهت برون‌دسته‌ای[4] پایین باشد، به عبارت دیگر هر دسته مباحثی را ارائه دهد که با مباحث ارائه شده توسط دیگر دسته‌ها متفاوت است. با به‌کارگیری متن‌کاوی[5] در حوزه‌ی وب این فرایند با نام وب‌کاوی شناخته می‌شود.

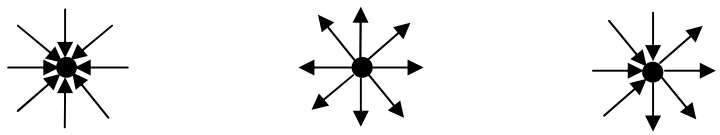
۱-۱-۱- مروری بر انواع وب‌کاوی[6]

روش‌های وب‌کاوی براساس آن‌که چه نوع داده‌ای را مورد کاوش قرار می‌دهند، به سه دسته تقسیم می‌شوند[۳۲]:

۱. **وب‌کاوی ساختاری**[7]: منظور از وب‌کاوی ساختاری کشف الگوی ارتباطی میان اسناد وب از طریق استخراج لینک‌های[8] میان اسناد است؛ درواقع لینک‌های میان اسناد وب، مولفه‌های ساختاری شبکه‌های اسناد مرتبط با هم هستند. برای به‌کارگیری الگوریتم‌های وب‌کاوی ساختاری و محاسبه معیارهای مربوطه، ابتدا لازم است، ساختار وب با استفاده از مدلی بازنمایی شود. وب را می‌توان به صورت گرافی که گره‌های آن اسناد و یال‌های آن لینک‌های

[1] Data mining
[2] Clustering
[3] Intra-cluster similarity
[4] Inter-cluster similarity
[5] Text mining
[6] Web mining
[7] Web structure mining
[8] Hyperlink

بین اسناد است، مدل‌سازی کرد. مدل‌های مبتنی بر گراف، می‌توانند از یک یا چند گره تشکیل شوند برای مثال شکل(۱-۱) نمونه‌ای از یک مدل تک گره‌ای را نشان می‌دهد.



شکل(۱-۱) : نمونه‌ای از یک مدل مبتنی بر گراف تک گره‌ای

در این شکل، مدل (a) یک نوع صفحه وب را بازنمایی می‌کند که به آن مرجع[1] گفته می‌شود. یک صفحه مرجع، صفحه‌ای است که صفحات زیاد دیگری به آن اشاره کرده‌اند. مدل (b) نوع دیگری از صفحات وب را بازنمایی می‌کند که به آن هاب[2] گفته می‌شود. یک صفحه هاب، صفحه‌ای است که به صفحات زیاد دیگری اشاره می‌کند. مدل (c) نیز ترکیبی از دو مدل قبل می‌باشد.

۲. **وب‌کاوی برمبنای استفاده[3]**: وب‌کاوی برمبنای استفاده، کاربرد تکنیک‌های داده کاوی برای کشف الگوهای استفاده از وب، به منظور درک و برآوردن بهتر نیازهای کاربران می‌باشد. این نوع از وب‌کاوی، داده‌های مربوط به استفاده‌ی کاربران از وب را مورد کاوش قرار می‌دهد.

۳. **وب‌کاوی محتوایی[4]**: وب‌کاوی محتوایی فرآیند استخراج اطلاعات مفید از محتوای مستندات وب است. محتوای یک سند وب متناظر با مفاهیمی است که آن سند درصدد انتقال آن به کاربران است. این محتوا می‌تواند شامل متن، تصویر، ویدئو، صدا و یا رکوردهای ساخت یافته مانند لیست‌ها و جداول باشد. در این میان کاوش متن بیش از سایر زمینه‌ها مورد تحقیق

[1] Authority
[2] Hub
[3] Web usage mining
[4] Web content mining

قرار گرفته است. از جمله این تحقیقات می‌توان به تشخیص موضوع[1]، استخراج الگوهای ارتباط[2]، خوشه‌بندی[3] و طبقه‌بندی[4] اسناد وب اشاره کرد. روش‌ها و تکنیک‌های موجود در این گروه، از تکنیک‌های بازیابی اطلاعات و پردازش زبان طبیعی نیز استفاده می‌کنند.

باید توجه داشت که مرز مشخصی میان سه گروه وب‌کاوی وجود ندارد. به عنوان مثال تکنیک‌های وب‌کاوی محتوایی می‌توانند علاوه بر به‌کارگیری متن مستندات، از اطلاعات کاربران هم استفاده کنند. همچنین می‌توان از ترکیب تکنیک‌های فوق برای حاصل شدن نتایج بهتر استفاده کرد. در ادامه به‌طور مختصر به برخی کاربردهای انواع وب‌کاوی می‌پردازیم.

۱-۱-۱-۱- کاربرد وب‌کاوی ساختاری

با توجه به افزایش حجم وب، پیمایش و جستجوی آن از اهمیت بالایی برخوردار است. در پیمایش این حجم وسیع از صفحات بهتر است، صفحاتی ابتدا پیمایش شوند که مرتبط با موضوع موردنظر می‌باشند. "پیمایش متمرکز[5]" روشی است که برای پیمایش صفحات مرتبط با یک موضوع به کار می‌رود. در این روش سعی بر آن است که در هنگام پیمایش، صفحات هاب به خوبی تشخیص داده شوند تا از آن‌ها به عنوان منبعی برای رسیدن به صفحات مرجع استفاده شود.

روش دیگری به نام "پیمایش هوشمند[6]" نیز برای پیمایش صفحات وب پیشنهاد شده‌است. این روش علاوه بر ساختار پیوند وب از ویژگی های دیگری نیز استفاده می‌کند. از جمله این ویژگی‌ها، می‌توان به محتوای صفحه، نشانه‌های[7] URL، مانند برخی کلمات کلیدی مشخص که اهمیت یک URL در

[1] Topic discovery
[2] Association pattern
[3] Clustering
[4] Classification
[5] Focused crawling
[6] Intelligent crawling
[7] Token

ارتباط با یک موضوع خاص را نشان می دهند و ... اشاره کرد. با استفاده از این ویژگی‌ها اولویتی برای پیمایش هر یک از صفحات تعریف می‌شود.

۱-۱-۱-۲- کاربردهای وب‌کاوی برمبنای استفاده

داده‌های استفاده از وب[1] مشخصات کاربران و رفتار پیمایش آن‌ها در سایت‌های وب را مشخص می-نمایند. این داده‌ها معمولا از سه منبع اصلی جمع‌آوری می‌شوند: سرورهای وب، سرورهای پراکسی[2] و کلاینت‌های[3] وب. این اطلاعات می‌تواند برای بهبود سایت‌های وب از دید کاربران به کار رود. سه کاربرد اصلی این نوع کاوش در این قسمت بررسی می‌شوند.

۱. شخصی‌سازی محتوای وب

تکنیک‌های وب‌کاوی برمبنای استفاده ، می‌توانند برای شخصی‌سازی استفاده‌ی کاربران از وب به کار روند. برای مثال می‌توان رفتار کاربر را از طریق مقایسه‌ی الگوی پیمایش فعلی وی با الگوهای پیمایش استخراج شده از فایل‌های ثبت، به صورت بلادرنگ پیش‌بینی کرد. سیستم‌های توصیه[4] که یک کاربرد واقعی در این زمینه هستند، پیوندهایی که کاربر را به صفحات مورد علاقه وی هدایت می‌کنند، به او پیشنهاد می‌کنند. برخی سایت‌ها نیز کاتولوگ محصولات خود را براساس علایق پیش‌بینی شده برای کاربر خاص سازماندهی و به او ارائه می‌نمایند.

[1] Usage data
[2] Proxy server
[3] Client server
[4] Recommendation system

۲. پیش بازیابی

نتایج به دست آمده از وب‌کاوی برمبنای استفاده، می‌تواند برای بهبود کارایی سرورهای وب و برنامه-های کاربردی مبتنی بر وب به کار رود. وب‌کاوی برمبنای استفاده، می‌تواند برای ایجاد استراتژی‌های پیش‌بازیابی[1] و caching استفاده شود و با پیش‌بینی مولفه‌هایی از صفحات وب که در آینده نزدیک توسط کاربران مورد استفاده قرار می‌گیرند عملکرد وب‌سایت‌ها را بهبود بخشیده و به این ترتیب زمان پاسخ سرورهای وب را کاهش دهد.

۳. بهبود طراحی سایت‌های وب

قابلیت استفاده[2] یکی از مسائل مهم در طراحی و پیاده‌سازی سایت‌های وب است. نتایج به دست آمده از وب‌کاوی برمبنای استفاده، می‌توانند به طراحی مناسب سایت‌های وب کمک کنند. سایت‌های وب تطبیقی، یک کاربرد از این نوع کاوش می‌باشند. در این سایت‌ها محتوا و ساختار سایت وب، به صورت پویا بر اساس داده‌های استخراج شده از رفتار کاربر سازماندهی مجدد می‌شوند.

۱-۱-۱-۳- کاربردهای وب‌کاوی محتوایی

دو مورد از مهم‌ترین کاربردهای وب‌کاوی محتوایی عبارتند از:

۱. طبقه بندی[3]

طبقه‌بندی مستندات به معنای مرتبط نمودن یک سند به یک طبقه‌ی از پیش تعریف شده، است. به عبارت دیگر هدف از طبقه‌بندی مستندات، یافتن طبقه‌ی موضوعی مناسبی است که با کمترین خطا موضوع بحث یک سند را نشان می‌دهد. این کار می‌تواند با مربوط کردن یک سند به یکی از طبقات از پیش تعریف شده، صورت پذیرد و یا در طبقه‌بندی پویا منجر به تعریف طبقه‌ی موضوعی جدیدی

[1] Pre-fetching
[2] Usability
[3] Classification

برای سند در دست بررسی گردد. طبقه‌بندی جزء روش‌های یادگیری با نظارت به شمار می‌آید. به آن معنی که ابتدا مجموعه اسنادی به سیستم داده می‌شود که طبقه‌ی آنها مشخص شده است. سپس انتظار می‌رود سیستم با دیدن این نمونه‌ها بتواند نمونه‌های جدید را طبقه‌بندی کند. .هدف طبقه-بندی، تحلیل نمونه‌های آموزشی و ساخت مدل دقیقی برای هر طبقه با استفاده از ویژگی‌های موجود در داده‌ها و سپس استفاده از این مدل‌ها برای طبقه‌بندی داده‌های آتی است.

۲. خوشه بندی[1]

خوشه‌بندی (دسته‌بندی) فرایند گروه‌بندی اشیاء در کلاس‌هایی از اشیاء مشابه است. خوشه‌بندی یکی از روش‌های یادگیری بدون نظارت به شمار می‌آید. به آن معنی که بر خلاف طبقه‌بندی که در ابتدا مثال‌هایی از کلاس‌های معلوم به سیستم داده می‌شود، در خوشه‌بندی هیچ‌گونه اطلاع قبلی از کلاس-ها در دسترس‌نیست و این وظیفه سیستم است که با بررسی داده‌ها، خوشه‌ها و ویژگی‌های هریک را تشخیص دهد.

به عنوان یک تکنیک وب‌کاوی، خوشه‌بندی داده‌ها، خوشه‌ها یا نواحی متراکم[2] را در مجموعه بزرگی از داده‌های چند بعدی براساس معیاری برای اندازه‌گیری فاصله، پیدا می‌کند. در یک مجموعه بزرگ از نقاط داده‌ای چندبعدی، معمولاً فضای داده‌ای بطور یکنواخت توسط نقاط پر نمی‌شود. خوشه‌بندی داده‌ها، محل‌های خلوت[3] و متراکم را تشخیص داده و در نتیجه الگوی کلی توزیع اطلاعات را تشخیص می‌دهد.

کاربردهای دسته‌بندی اسناد عبارتند از: دسته‌بندی اسناد بازیابی شده جهت ارائه‌ی نتایج سازماندهی شده و قابل فهم به کاربر، دسته‌بندی اسناد در یک مجموعه مانند کتابخانه‌ی دیجیتال، اتوماتیک-سازی ایجاد دسته‌بندی اسناد در وب و بازیابی اطلاعات به صورت کارا، با تمرکز بر روی زیرمجموعه-

[1] Clustering
[2] Densely populated
[3] Sparse

های مرتبط، نه کل مجموعه‌ی اسناد. همانطور که قبلا نیز عنوان شد دسته‌بندی اسناد سبب کاهش زمان جستجو و افزایش دقت و سرعت در جستجو می‌گردد.

تمرکز ما در این پایان‌نامه بر روی دسته‌بندی اسناد وب از طریق وب‌کاوی محتوایی معطوف می‌گردد.

۱-۲- بررسی الزامات سیستم دسته‌بندی اسناد

در فرایند دسته‌بندی علاوه بر فاکتورهای شاخصی چون معیار شباهت[1] و الگوریتم دسته‌بندی که براساس مشخصات داده، نوع کاربرد و ویژگی‌های مسئله‌ی مورد نظر، تعیین می‌شوند؛ ضروری است الزامات الگوریتم‌های خوشه‌بندی به‌طور خاص، در زمینه کار با اسناد، مورد بررسی قرار گیرد. این امر ما را قادر می‌سازد، راه‌حل‌هایی به مراتب کاراتر و قوی‌تر در راستای خوشه‌بندی اسناد ارائه دهیم. در ادامه به بررسی این نیازمندی‌ها می‌پردازیم[۱]:

۱-۲-۱- استخراج ویژگی‌های مفید و حاوی بیشترین اطلاعات

ریشه‌ی حل هر مسئله‌ی خوشه‌بندی نهفته در انتخاب مجموعه‌ای از شاخص‌ترین ویژگی‌هاست، که بهترین نماینده جهت توصیف مدل داده زیربنایی باشد. ویژگی‌های استخراج شده باید دارای اطلاعات کافی برای نمایش داده‌ی درحال تجزیه و تحلیل باشد؛ درغیر این‌صورت، هرقدر هم که الگوریتم خوشه‌بندی خوب عمل کند، در صورت به‌کارگیری ویژگی‌های فاقد اطلاعات مفید، نتایج قابل قبولی را ارائه نمی‌کند. به‌علاوه از آن‌جا که ابعاد بالای بردار ویژگی همواره تاثیر شدیدی برروی مقیاس-پذیری[2] الگوریتم دارد، مهم است که تعداد ویژگی‌ها را تا حد امکان کاهش دهیم.

[1] Similarity measure
[2] Scalability

در این میان مدل نمایش داده نیز از اهمیت بالایی برخوردار است. مدل نمایش داده به نوعی بیانگر چگونگی استخراج ویژگی از داده‌ی مورد دسته‌بندی است و ویژگی‌های استخراج شده در قالب این مدل برای دسته‌بندی مورد استفاده قرار می‌گیرند.

۱-۲-۲- همپوشانی[1] مدل خوشه‌بندی

درهنگام دسته‌بندی اسناد، ممکن است سند به بیش از یک خوشه تعلق داشته باشد، یک مدل خوشه‌بندی همپوشان از دسته‌بندی چنین اسنادی پشتیبانی می‌کند. الگوریتم‌های خوشه‌بندیی چون فازی و درخت پسوندی از این دسته‌اند. با توجه به اینکه اغلب اسناد موجود در وب، بیش از یک موضوع را پوشش می‌دهند، دقت به این نکته در انتخاب الگوریتم مناسب حائز اهمیت است.

۱-۲-۳- مقیاس‌پذیری

در حوزه‌ی وب، با یک درخواست جستجوی ساده ممکن است صدها یا شاید هزاران صفحه وب به عنوان پاسخ به کاربر ارائه شود. یک سیستم دسته‌بندی باید قادر باشد عملیات دسته‌بندی را در زمان معقولی بر روی نتایج انجام دهد. لازم به ذکر است که برخی از سیستم‌های ارائه شده در این زمینه تنها اطلاعات خلاصه‌ی[2] استخراج شده توسط موتورهای جستجو را دسته‌بندی می‌کنند و از متن اصلی صفحات استفاده نمی‌کنند. این روش اگرچه هزینه محاسباتی و زمانی بالایی به سیستم تحمیل نمی‌کند، با این وجود، خلاصه صفحات حاوی اطلاعات کافی درمورد محتویات واقعی اسناد نیستند، در نتیجه احتمال خطا در دسته‌بندی مبتنی بر این روش نسبتاً بالاست.

یک الگوریتم خوشه‌بندی آنلاین باید قادر باشد عملیات دسته‌بندی را نسبت به تعداد اسناد حدالامکان در زمانی خطی و با دقتی مطلوب انجام دهد.

---

[1] Overlapping
[2] Snippets

۱-۲-۴- قدرت تحمل خطا[1]

یک مشکل بالقوه که بسیاری از الگوریتم‌های خوشه‌بندی با آن مواجه هستند وجود نویز در مجموعه داده است. یک الگوریتم خوشه‌بندی خوب باید قادر باشد با انواع مختلف نویز مقابله کرده و از تاثیر آن بر روی کیفیت دسته‌بندی جلوگیری کند. در روش‌هایی که تنها از کلمات منفرد بدون درنظر گرفتن ارتباط میان آن‌ها، جهت دسته‌بندی استفاده می‌شود، احتمال بروز خطای متاثر از نویز بالاست؛ با این وجود در تکنیک‌هایی که از عبارات یا به‌طور کلی ارتباط میان کلمات برای دسته‌بندی استفاده می‌کنند، کلمات نویزی تاثیری به‌مراتب کمتر در کیفیت دسته‌بندی دارند، علاوه بر مدل نمایش داده انتخاب یک معیار شباهت مناسب و متناسب با داده مورد دسته‌بندی نیز تاثیر بسزایی در افزایش کیفیت عملکرد دسته‌بندی و کاهش اثرات نویز در بروز خطا دارد.

۱-۲-۵- افزایش تدریجی[2]

برای سیستم دسته‌بندی یک ویژگی بسیار مطلوب در محیط پویایی[3] چون وب این است که قادر باشد دسته‌ها را به‌صورت تدریجی به‌روز کند، دراین‌صورت اسناد جدید اضافه شده به محض ورود به سیستم، بدون نیاز به دسته‌بندی مجدد کل مجموعه در دسته‌های مربوط به خود قرار می‌گیرند. در صورت دستیابی به این ویژگی در سیستم، قابلیت استفاده از آن به صورت آنلاین در وب فراهم می‌شود.

[1] Noise tolerance
[2] Incrementality
[3] Dynamic

۱-۳- بررسی اجمالی پژوهش‌های انجام شده در زمینه دسته‌بندی اسناد

همانطور که اشاره شد، همزمان با رشد سریع منابع اطلاعاتی در محیط وب، پژوهش‌های بسیاری از سوی محققان در جهت سازماندهی این حجم بالا از اطلاعات انجام شده است. مطالعات انجام شده حاکی از آن است که پژوهشگران در زمینه‌های تحقیقاتی مختلف به بررسی موضوع دسته‌بندی اسناد متنی پرداخته‌اند که پایگاه داده (DB)[۱] ، بازیابی اطلاعات (IR)[۲] و هوش مصنوعی (AI)[۳] شامل یادگیری ماشین (ML)[۴] و پردازش زبان‌های طبیعی (NLP)[۵] از آن دسته‌اند. از جمله متدهای مورد استفاده در دسته‌بندی متن عبارتند درخت تصمیم[۶] ، شبکه عصبی[۷]، سیستم‌های مبتنی بر قاعده[۸] و تحلیل آماری که در ادامه به‌طور مختصر به بررسی آنها می‌پردازیم.

از میان متدهای نام برده، درخت تصمیم و شبکه عصبی نیاز به آموزش دارند، درنتیجه باید مجموعه‌ی جامعی از نمونه‌های برچسب‌گذاری شده با توزیعی مناسب شامل همه‌ی کلاسترها داشته‌باشیم، تا سیستم به خوبی آموزش داده شود و پس از مرحله‌ی یادگیری در دسته‌بندی نمونه‌های مختلف عملکرد مطلوبی را ارائه کند. در متدهای مبتنی بر آموزش، با ارائه‌ی یک کلاستر جدید که با توجه به پویایی محیط وب دور از انتظار نخواهدبود، لازم است فرایند یادگیری مجدداً با حضور نمونه‌هایی از کلاستر جدید انجام شود تا سیستم از عمومیت[۹] مطلوبی جهت دسته‌بندی نمونه‌های جدید برخوردار باشد. اگرچه استفاده از سیستم‌های آموزش‌محور در محیط‌های ایستا[۱۰] نتایج قال قبولی را ارائه می-دهد، ولی استفاده از آن‌ها در محیط‌های پویا به دلیل هزینه محاسباتی و زمانی بالا توصیه نمی‌شود.

[1] Database
[2] Information retrieval
[3] Artificial intelligence
[4] Machine learning
[5] Natural language processing
[6] Decision tree
[7] Neural net
[8] Rule-based systems
[9] Generalization
[10] Static

سیستم‌های مبتنی بر قاعده که با عنوان سیستم‌های خبره[1] شناخته می‌شوند، مبتنی بر استفاده از قواعد استنتاجی هستند که توسط افراد متخصص به سیستم داده می‌شود. علی‌رغم به‌کارگیری این سیستم‌ها در دسته‌بندی متن، به دلیل پیچیدگی محاسباتی بالا و نیاز به وجود نیروی انسانی جهت تولید و اصلاح قواعد نمی‌توان از آن‌ها به عنوان یک ابزار اتوماتیک جهت دسته‌بندی اسناد در محیط وب استفاده کرد. دراین میان سیستم‌های مبتنی برتحلیل آماری بیشترین کاربرد را در دسته‌بندی اسناد داشته‌اند.

همان‌طور که در بخش قبل اشاره شد، یکی از مهم‌ترین الزامات یک سیستم دسته‌بندی صرف‌نظر از این‌که چه متدی برای دسته‌بندی انتخاب شود، مدل نمایش داده است. چندین مدل مرسوم نمایش اسناد متنی عبارتند از:

- مدل فضای برداری[2]
- مدل پنجره لغزان[3]
- مدل درخت پسوندی[4]

درحال حاضر، مدل فضای برداری یکی از مرسوم‌ترین مدل‌های نمایش داده در سیستم‌های دسته-بندی اسناد است. این مدل اسناد را به‌صورت یک بردار ویژگی از کلمات منحصر به‌فرد به‌کار رفته در کل مجموعه‌ی اسناد نمایش می‌دهد، بردار مربوط به هر سند شامل وزن (اغلب فرکانس) کلمات به‌کار رفته در آن سند است. در این مدل شباهت میان اسناد با استفاده از معیارهای مبتنی بر بردار شامل کسینوس[5] و جاکارد[6] محاسبه می‌شود. متدهای خوشه‌بندی برپایه‌ی این مدل تنها از تحلیل کلمات منفرد استفاده می‌کنند و از خاصیت مجاورت کلمات یا عبارات بهره‌ای نمی‌برند.

[1] Expert systems
[2] Vector space model
[3] n-gram model
[4] Suffix tree model
[5] Cosine
[6] Jaccard

ما برای انتقال مفاهیم از کنار هم قرار دادن کلمات در یک نوشته بهره می‌بریم. مجاورت کلمات در یک نوشته سبب ایجاد مفاهیمی می‌شود که کلمات در حالت انفرادی قادر به انتقال آن مفاهیم نخواهند بود. علاوه بر این کلماتی وجود دارند که دارای چندین معنی هستند و تنها با قرارگیری آنها در یک عبارت و درمجاورت سایر کلمات معنی موردنظر آن‌ها مشخص می‌شود. با توجه این نکته در نظر گرفتن مجاورت میان کلمات به‌منظور دسته‌بندی اسناد متنی بسیار حائز اهمیت است.

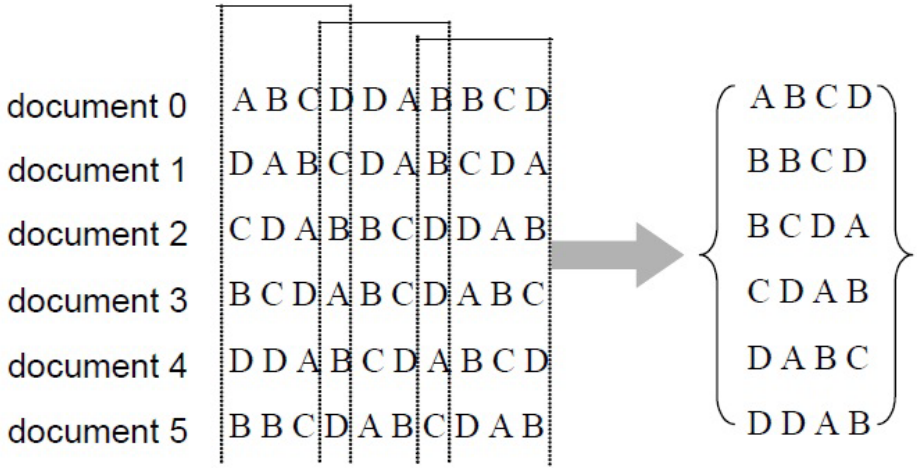
با معرفی مدل n-gram سعی شد تا حدودی از خاصیت مجاورت میان کلمات استفاده شود.

یک 1-gram یا unigram ریشه‌ی یک کلمه است. یک n-gram دنباله‌ای (t_k) از n ، unigram است که به صورت الفبایی مرتب شده باشد. دنباله t_k هنگامی در سند رخ می‌دهد که دنباله‌ای از کلمات موجود در سند پس از حذف stop word ها و ریشه‌یابی شامل جایگشتی از دنباله t_k باشد.

در این مدل فرض می‌شود سند دنباله‌ای از کلمات است، با استفاده از یک پنجره‌ی لغزان به طول n همه‌ی n-gram های سند تولید شده و تعداد n-gram های مشترک میان دو سند میزان شباهت آن‌ها را مشخص می‌کند. از معایب این مدل می‌توان به افزونگی بالا و طول ثابت پنجره لغزان اشاره کرد. نمایی از این مدل را در شکل (۱-۲) مشاهده می‌کنید.

مدل نمایش درخت پسوندی یک مدل مبتنی برعبارت است که عبارات پسوندی مشترک میان اسناد را پیدا کرده و براساس این عبارات درخت پسوندی را می‌سازد، در این درخت هر نود نماینده‌ی بخشی از یک عبارت، و شناسه‌ی اسنادی است که در این عبارت پسوندی مشترک هستند. این رویکرد به‌وضوح از اطلاعات مجاورت کلمات استفاده می‌کند که در محاسبه شباهت میان اسناد با دقت بالا بسیار حائز اهمیت است و سبب برتری این مدل نسبت به مدل‌های قبل شده است. با این وجود فاکتور انشعاب این درخت بسیار بزرگ است به‌خصوص در سطح اول درخت، که هر پسوند احتمالی تشخیص داده شده در مجموعه‌ی اسناد، از گره ریشه منشعب می‌شود، همچنین این مدل از افزونگی

بیش از حد پسوندها در گره‌های مختلف درخت رنج می‌برد. در ادامه مدلی معرفی می‌شود که علاوه بر به‌کارگیری عبارات در نمایش اسناد نواقص مدل‌های مرسوم را نیز تا حد مطلوبی مرتفع می‌کند.



شکل(۱-۲) : نمایی از مدل n-gram با پنجره‌ی لغزانی به‌طول ۴ (4-gram)

۱-۴- معرفی سیستم دسته‌بندی اسناد وب براساس گراف نمایه‌سازی اسناد

در سال ۲۰۰۴ مدل گراف نمایه‌سازی اسناد (DIG)[1] توسط "M. Hammouda" و "S. Kamel" ارائه شد[۲] که بر دو مفهوم استوار بود اول اینکه از عبارات وزن‌دار به عنوان اجزای اصلی تشکیل دهنده اسناد استفاده شده و شباهت بین اسناد براساس انطباق عبارات و وزن‌های آنهاست. دومین مفهوم، خوشه‌بندی افزایشی اسناد براساس متد مبتنی بر هیستوگرام است و طوری عمل می‌کند که شباهت درون خوشه‌ها را بیشینه کند. این سیستم از ۴ مولفه تشکیل شده:

۱. یک شمای بازسازی سند وب که بخش‌های مختلف سند را تعیین کرده و سطوح اهمیت مختلفی را به هربخش اختصاص می‌دهد.

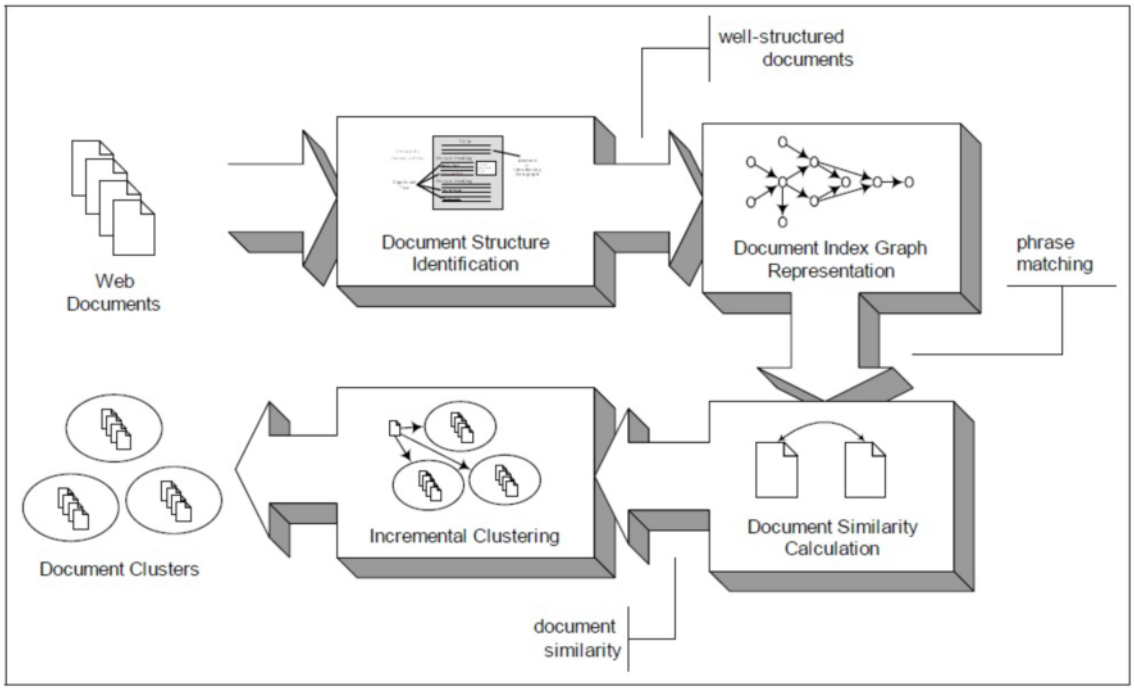
[1] Document Index Graph

۲. مدل نمایه‌سازی سند براساس عبارت، گراف نمایه‌سازی، که به جای کلمات از ساختار جمله در سند بهره می‌گیرد. مدل DIG براساس تئوری گراف طراحی شده و از خواص گراف برای انطباق عبارات (با هرطولی) یک سند با هر تعداد سندی که قبلا دیده شده در زمانی متناسب با تعداد کلمات سند بهره می‌گیرد.

۳. معیار شباهت برای ارزیابی شباهت بین دو سند براساس عبارات منطبق و وزن عبارات. این معیار ترکیبی از شباهت کلمات منفرد و شباهت عبارات است.

۴. یک متد خوشه‌بندی افزایشی که براساس شباهت هر جفت سند است.

مدل DIG ارائه شده در این سیستم به دلیل استفاده از ساختار گراف فاقد افزونگی بوده و در دسته-بندی از هر تعداد سند پشتیبانی می‌کند، همچنین به دلیل ساختار افزایشی الگوریتم دسته‌بندی، قابلیت به‌کارگیری به صورت آنلاین در وب را نیز دارد. کیفیت خوشه‌های تولید شده توسط این سیستم نسبت به خوشه‌های تولید شده با استفاده از روش‌های معمول بیشتر ارزیابی شده و بهبودی ۱۰ تا ۲۹ درصدی گزارش شده. نمای کلی این سیستم در شکل(۱-۳) آمده است.



شکل(۱-۳) : شمای کلی سیستم دسته‌بندی اسناد وب براساس گراف نمایه‌سازی اسناد[۱]

در ادامه این نوشتار در فصل دوم به‌طور مفصل به پژوهش‌های انجام شده در زمینه دسته‌بندی اسناد خواهیم پرداخت. بخش‌های مختلف سیستم دسته‌بندی اسناد برپایه مدل DIG را که تکنیک مورد مطالعه ما در این پایان نامه بوده و رویکرد پیشنهادی بر پایه‌ی این سیستم ارائه شده است در فصل سوم با جزئیات مورد بررسی قرار می‌دهیم. در فصل چهارم به ارائه ایده‌ی پیشنهادی می‌پردازیم و ارزیابی عملکرد این ایده‌ی پیشنهادی را در فصل پنجم مورد بررسی قرار خواهیم داد. در نهایت به نتیجه‌گیری و ارائه‌ی پیشنهادهایی برای کارهای آینده در فصل آخر خواهیم پرداخت.

فصل دوم

تحقیقات مرتبط

۲-۱- بررسی چندین متد مورد استفاده در دسته‌بندی اسناد

همان‌طور که در مقدمه اشاره شد تاکنون از متدهای مختلفی جهت دسته‌بندی اسناد متنی استفاده شده که هر کدام مزایا و معایب خاص خود را دارد. در این فصل قصد داریم به اختصار به این متدها، ساختار و نحوه‌ی عملکردشان در دسته‌بندی اسناد بپردازیم، همچنین در مورد هر متد به‌طور مختصر بررسی خواهیم کرد که تا چه حد الزامات یک سیستم دسته‌بندی اسناد وب را فراهم می‌کند.

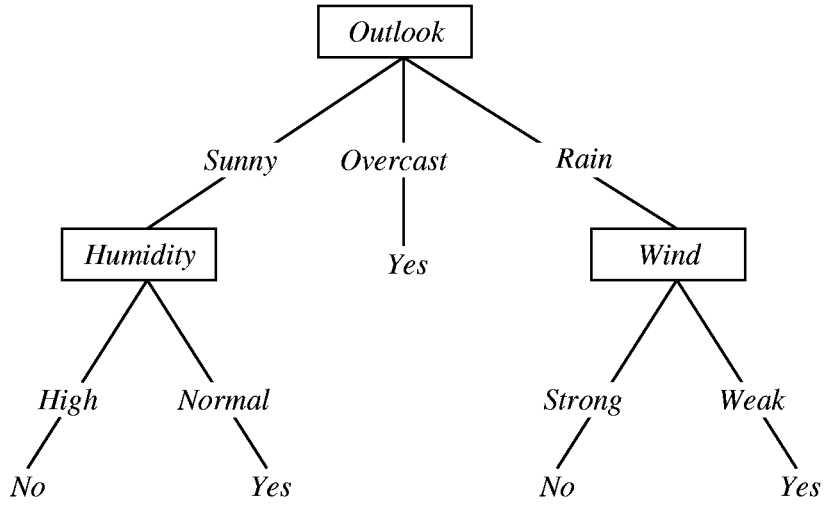
۲-۱-۱- درخت تصمیم[1]

درخت تصمیم ابزاری برای پشتیبانی از تصمیم است که از درختان برای مدل کردن استفاده می‌کند. این ابزار در آنالیز تصمیم، برای مشخص کردن استراتژی‌ای که با بیشترین احتمال به هدف برسد، بکار می‌رود. درختان تصمیم، نمونه ها را با مرتب کردن آنها در درخت از گره ریشه به سمت گره‌های برگ دسته‌بندی می‌کنند. در حالت کلی، درختان تصمیم یک ترکیب فصلی از ترکیبات قیود عطفی روی مقادیر صفات نمونه‌ها را بازنمایی می‌کنند. هر مسیر از ریشه درخت به یک برگ، متناظر با یک ترکیب عطفی صفات تست موجود در آن مسیر بوده، و خود درخت نیز متناظر با ترکیب فصلی همه این ترکیبات عطفی می‌باشد. برای مثال شکل(۲-۱) نمایی از یک درخت تصمیم را نشان می‌دهد [۳و۴].

عبارت متناظر با درخت شکل(۲-۱) به صورت زیر است:

$$(Outlook=Sunny \wedge Humidity=Normal) \vee (Outlook=Overcast) \vee (Outlook=Rain \wedge Wind=Weak)$$

[1] Decision tree



شکل(۲-۱) : مثالی از درخت تصمیم

برای ساخت یک درخت تصمیم به مجموعه‌ای از داده‌های آموزش نیاز داریم که در آن‌ها مقادیر ویژگی‌ها و کلاس هر داده مشخص شده باشد، برای مثال جدول زیر داده‌های آموزشی برای ساخت درخت شکل(۲-۱) را نمایش می‌دهد.

جدول(۲-۱) : داده‌های آموزشی برای ساخت درخت تصمیم

Day	Outlook	Temp	Humidity	Wind	Play
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

در ساخت درخت تصمیم با استفاده از الگوریتم پایه‌ای ID3 از مفهوم بی‌نظمی برای دسته‌بندی داده‌ها استفاده شده‌است و الگوریتم درصدد آن است که میزان بی‌نظمی در نودهای بالایی درخت حداقل باشد تا بتوان درختی با حداقل ارتفاع داشت. برای این منظور با استفاده از مفهوم آنتروپی و محاسبه این کمیت در هر مرحله، صفتی برای تست در هر نود انتخاب می‌شود که بی‌نظمی داده‌ها را تا حد ممکن کاهش دهد.

در حالت‌های کلی، اگر صفت هدف بتواند c مقدار مختلف بگیرد بی‌نظمی S (S = مجموعه نمونه‌های آموزش) مربوط به این دسته‌بندیc تایی به شکل زیر تعریف می‌شود:

$$Entropy(S) \equiv \sum_{i=1}^{c} -p_i log_2 p_i \qquad (۱-۲)$$

که p_i نسبتی از اعضای S می‌باشد که متعلق به دسته‌ی i هستند.

با داشتن بی‌نظمی به عنوان ناخالصی در یک مجموعه مثال آموزشی، می‌توان معیار موثر بودن یک صفت در دسته‌بندی داده‌های آموزشی را تعریف نمود. این معیار، کاهش مورد انتظار در بی‌نظمی است که با تفکیک کردن مثال‌ها برپایه‌ی این صفت حاصل می‌شود.

نفع اطلاعاتی $Gain(S,A)$ یک صفت A مربوط به مجموعه‌ی مثال‌های S به شکل زیر تعریف می شود:

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \qquad (۲-۲)$$

$Values(A)$ = مجموعه‌ی تمام مقادیر ممکن برای صفت A

S_v = زیرمجموعه‌ای از S است که در آن صفت A مقدار v را دارد. (یعنی $S_v=\{s \in S | A(s)=v\}$)

جمله‌ی اول در فرمول(۲-۲)، بی‌نظمی مجموعه‌ی اصلی S و جمله‌ی دوم، مقدار مورد انتظار بی‌نظمی بعد از تفکیک S با استفاده از صفت A می‌باشد. با این تعریف هرچقدر بی‌نظمی پس از افراز نمونه‌ها کمتر باشد نفع اطلاعاتی افزایش یافته، در نتیجه در هر مرحله صفتی انتخاب می‌شود که نفع اطلاعاتی آن بالاتر از بقیه صفات باشد.

عمل انتخاب یک صفت جدید و افراز مثال‌های آموزشی برای هر گره فرزند غیرپایانه تکرار می‌شود. در این مرحله، فقط با استفاده از مثال‌های آموزشی افراز شده به این گره، عملیات ساخت زیردرخت انجام می‌شود، بنابراین هر صفت حداکثر یکبار در هر مسیری در درخت ظاهر می‌شود.

این فرآیند برای هر گره برگ جدید تا زمانی که یکی از دو شرط زیر برقرار شوند تکرار می‌شود:

(۱) تمام صفت‌ها تاکنون در این مسیر درخت استفاده شده‌اند.

(۲) مثال‌های آموزشی افراز شده به این گره‌ی برگ، همه یک مقدار صفت هدف را دارند (بی‌نظمی صفر است).

این الگوریتم فقط قادر به دسته‌بندی داده‌ها با دامنه ویژگی‌های گسسته و محدود می‌باشد و درمورد داده‌های نویزی و مخدوش کارایی ندارد. الگوریتم C4.5 که تکمیل شده الگوریتم ID3 است، قادر به دسته‌بندی داده‌های پیوسته و نویزی نیز می‌باشد.

برای دسته‌بندی اسناد متنی توسط درخت تصمیم، کلمات کلیدی مربوط به هر کلاس را تعیین کرده و با تعیین کلماتی با بیشترین قدرت جداکنندگی اقدام به ساخت درخت تصمیم جهت دسته‌بندی اسناد می‌کنیم.

آنتروپی مجموعه اسناد با m کلاس با فرمول(۲-۳) محاسبه می‌شود. برای استفاده از این متد هر سند باید تنها به یک کلاس تعلق داشته باشد. p_i احتمال اولیه کلاس i است.

$$H(D) \stackrel{\text{def}}{=} \sum_{i=1}^{m} -p_i log_2 p_i \qquad (۳-۲)$$

نفع اطلاعاتی واژه t در مجموعه‌ی اسناد آموزشی D با فرمول(۲-۴) قابل محاسبه است.

$$I(D,t) \stackrel{\text{def}}{=} H(D) - \{\frac{|D_t|}{|D|} H(D_t) + \frac{|D_{\bar{t}}|}{|D|} H(D_{\bar{t}})\} \qquad (۴-۲)$$

D = مجموعه اسناد

D_t = زیر مجموعه‌ای از اسناد که شامل کلمه t باشند

$D_{\bar{t}}$ = زیر مجموعه‌ای از اسناد که فاقد کلمه t هستند

۲-۱-۱-۱- مزایا و معایب درخت تصمیم

در میان ابزارهای پشتیبانی تصمیم، درختان تصمیم دارای مزایایی هستند:

۱. فهم ساده: قوانین تولید شده و به کارگرفته شده قابل استخراج و قابل فهم می باشند.
۲. قابلیت ترکیب با روش‌های دیگر: به این ترتیب که تصمیم در هر نود توسط تکنیک‌های دیگر چون SVM انجام می‌شود که سبب بهبود عملکرد تصمیم می‌شود[۵].

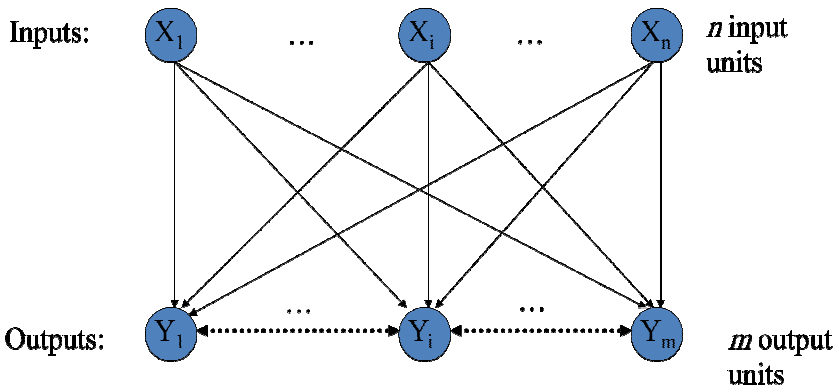
ملزومات ساخت درخت تصمیم که با دسته‌بندی آنلاین اسناد در محیط وب سازگار نیست[۶]:

۱. محاسبه احتمال اولیه کلاس‌ها که در محیط پویای وب امکان‌پذیر نیست.
۲. عدم امکان اضافه کردن کلاس جدید بدون بازسازی مجدد درخت (نداشتن خاصیت افزایشی)

۳. در درخت نمونه‌ها به کلاس‌ها افراز شده درنتیجه هر نمونه تنها به یک کلاس تعلق دارد که با ویژگی هپوشانی در محیط وب سازگار نیست.

۲-۱-۲- شبکه عصبی[1]

به‌کارگیری شبکه‌ی عصبی در دسته‌بندی داده‌ها معمولا در قالب معماری SOM[2] صورت می‌گیرد. معماری این شبکه به صورت دو لایه‌ی ورودی و خروجی است که به‌طور کامل به هم متصل هستند و بین نرون‌های لایه‌ی خروجی گاهی در مرحله‌ی آموزش اتصالاتی وجود دارد. شمای کلی شبکه SOM در شکل(۲-۲) نمایش داده شده است. تعداد نودهای لایه‌ی ورودی برابر سایز بردار ویژگی داده‌ی مورد دسته‌بندی و تعداد نودهای لایه‌ی خروجی برابر تعداد کلاس‌هاست، البته درصورت مشخص نبودن تعداد کلاس‌ها می‌توان تعداد نودهای لایه‌ی خروجی را بیشتر از تعداد تخمینی کلاس‌ها درنظر گرفت تا پس از مرحله آموزش با انجام عملیات ادغام تعداد مناسب کلاس‌ها بدست آید. جهت دسته-بندی اسناد متنی توسط SOM معمولا از مدل نمایش VSM[3] استفاده می‌شود.



شکل(۲-۲) : شمای کلی معماری شبکه SOM

[1] Neural network
[2] Self-organizing map
[3] Vector space model

آموزش شبکه SOM به صورت بدون‌نظارت[1] انجام می‌شود. برای آموزش شبکه، ابتدا وزن‌ها به‌صورت تصادفی مقداردهی اولیه می‌شوند، با آمدن اولین نمونه‌ی آموزشی فاصله‌ی آن با تمام نرون‌های خروجی محاسبه می‌شود و نرونی که کمترین فاصله را با آن دارد انتخاب شده و وزنش اصلاح می‌شود. لازم به ذکر است که در ساختار شبکه SOM پایه، برای اندازه‌گیری فاصله‌ی میان نمونه‌های ورودی و نرون‌های خروجی از معیار فاصله‌ی اقلیدسی طبق فرمول(۲-۵) استفاده می‌شود. فرایند اصلاح اوزان توسط فرمول(۲-۶) و درجهت کاهش فاصله به نمونه صورت می‌گیرد تا بتواند به خوبی نمایانگر دسته‌ی نمونه‌های مشابه باشد. درصورت تعریف شعاع همسایگی، اوزان نرون‌های موجود در همسایگی نرون برنده نیز به تناسب ضریب همسایگی اصلاح می‌شوند، البته ضریب همسایگی به مرور زمان کاهش یافته تا وقتی که تنها اوزان نرون برنده اصلاح می‌شود. مرحله آموزش تا زمانی ادامه می‌یابد که تغییرات وزن‌ها زیاد نباشد. در این مرحله شبکه آموزش دیده و آماده‌ی دسته‌بندی نمونه‌های جدید است.

$$\sum_{k=1}^{n}(i_{l,k} - w_{j,k}(t))^2 \qquad (۲-۵)$$

$$w_j(t+1) = w_j(t) + \eta(t)(i_l - w_j(t)) \qquad (۲-۶)$$

برای بهبود عملکرد شبکه SOM در دسته‌بندی اسناد متنی، روش‌هایی چون تغییر معماری شبکه و اصلاح نحوه‌ی نمایش اسناد ارائه شده است.

۲-۱-۲-۱- بهبود عملکرد شبکه SOM با تغییر معماری شبکه

شبکه‌های عصبی SOM با وجود توانایی بالا، مشکلاتی از جمله لزوم تعیین توپولوژی شبکه قبل از شروع آموزش، سرعت آموزش و اجرای پایین دارند، علت این مساله لزوم مقایسه ورودی با همه‌ی

[1] Unsupervised

نورون‌های شبکه به منظور یافتن نورون برنده می‌باشد، هزینه محاسباتی این شبکه‌ها با افزایش اندازه شبکه بصورت خطی افزایش می‌یابد. از شبکه‌های SOM سلسله‌مراتبی می‌توان برای افزایش سرعت در زمان آموزش و اجرا استفاده نمود. دراین حالت به دلیل عدم مقایسه داده ورودی با کلیه نورون‌های شبکه می‌توان به سرعت یادگیری و اجرای بیشتری در شبکه دست یافت، بنابراین اگر شبکه‌ای با N نورون داشته باشیم درحالت یک سطحی به N مقایسه نیاز می‌باشد ولی درشبکه‌ای با L سطح تعداد مقایسات به $L \times \sqrt[L]{N}$ کاهش می‌یابد این امر خصوصا در شبکه‌های بزرگی که قرار است با حجم زیادی داده آموزش داده شوند حائز اهمیت است[۷و۸].

۲-۱-۲-۲- بهبود عملکرد شبکه SOM با اصلاح نحوه‌ی نمایش اسناد

همانطور که در بالا اشاره شد، در دسته‌بندی اسناد توسط شبکه SOM معمولا از مدل نمایش VSM استفاده می‌شود، در این مدل هر سند با برداری نمایش داده می‌شود که ابعاد آن، برابر تعداد کلمات منحصربه‌فردِ به‌کار رفته در کل مجموعه‌ی اسناد است؛ با این تعریف با افزایش تعداد اسناد در یک مجموعه، تعداد کلمات منحصربه‌فرد، زیاد شده و با افزایش سایز بردار نمایش سند، تعداد نودهای ورودی شبکه نیز افزایش می‌یابد که با افت عملکرد شبکه‌عصبی همراه است.

برای بهبود عملکرد دسته‌بندی اسنادمتنی توسط شبکه SOM دو مدل ESVM[1] و HSVM[2] برای نمایش اسناد معرفی شده [۹و۱۰و۱۱].

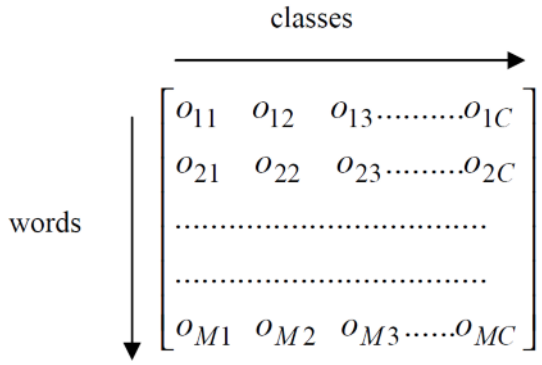
در مدل ESVM سعی شده از دانش کلاس‌بندی در نمایش سند استفاده شود. در این مدل به هر کلمه به تناسب میزان رخداد در هر کلاس، وزنی در هر کلاس تعلق می‌گیرد، و هر سند نیز با توجه به کلماتی که دارد نسبت به هر کلاس وزنی دارد که وزن بیشتر نشان‌دهنده‌ی تعلق بیشتر به آن کلاس است. در واقع در این روش هر سند با برداری به ابعاد تعداد کلاس‌ها نمایش داده می‌شود که به ازای

[1] Extended significance vector model
[2] Hypernym significance vector model

هر کلاس وزنی برای سند محاسبه می‌شود. برای ساخت این مدل از ماتریس رخداد کلمه-کلاس[1] استفاده شده که میزان رخداد کلمات مختلف در هر کلاس را در کل مجموعه اسناد آموزش نمایش می‌دهد و تعیین وزن کلمات در هر کلاس با استفاده از فرمول(۲-۷) انجام می‌شود. به این ترتیب به ازای هر کلمه، در هر کلاس وزنی تعیین می‌شود که با جمع اوزان کلمات موجود در هر سند به ازای هر کلاس، وزن آن سند در هر کلاس مشخص می‌شود.

$$W_{ij} = \frac{o_{ij}}{\sum_{c=1}^{C} o_{ic}} \times \log \frac{\sum_{m=1}^{M}\sum_{c=1}^{C} o_{mc}}{\sum_{m=1}^{M} o_{mj}} \qquad (۲-۷)$$ (محاسبه وزن کلمه i در کلاس j)

در این فرمول C تعداد کل کلاس‌ها و M تعداد کل کلمات موجود در مجموعه اسناد است.



شکل(۲-۳) : ماتریس رخداد کلمه-کلاس

در مدل HSVM که تعمیم یافته مدل ESVM است، ایده اصلی جایگزین کردن کلمات با مفهوم کلی‌تر، در جهت سازگاری و شباهت بیشتر سند با کلاستر و استفاده از مفاهیم زبانی در دسته‌بندی اسناد است. در این مدل مفاهیم جایگزین از WordNet استخراج می‌شوند.

این دو مدل نمایش داده، با این فرض به‌کار می‌روند که مجموعه‌ای از اسناد با توزیعی مناسب بر روی همه‌ی کلاس‌ها داشته‌باشیم، که با استفاده از این مجموعه‌ی مرجع، سیستم آموزش داده شود. در صورتی که در محیط پویای وب امکان ایجاد چنین مجموعه‌ای وجود ندارد، زیرا توزیع اسناد در کلاس‌های مختلف دائم در حال تغییر است. همچنین به دلیل پویایی محیط وب و امکان اضافه شدن

[1] Word-class occurrence matrix

کلاس جدید در هر زمان، درصورت استفاده از شبکه عصبی با اضافه شدن یک کلاس جدید، مرحله‌ی آموزش باید مجددا اجرا شود که در سیستم‌های آنلاین به دلیل لزوم خاصیت افزایش تدریجی[1]، قابل بهره‌برداری نیست؛ ماهیت ساختار شبکه‌عصبی به‌گونه‌ای است که قابلیت سازگاری با سیستم‌های آفلاین را دارد.

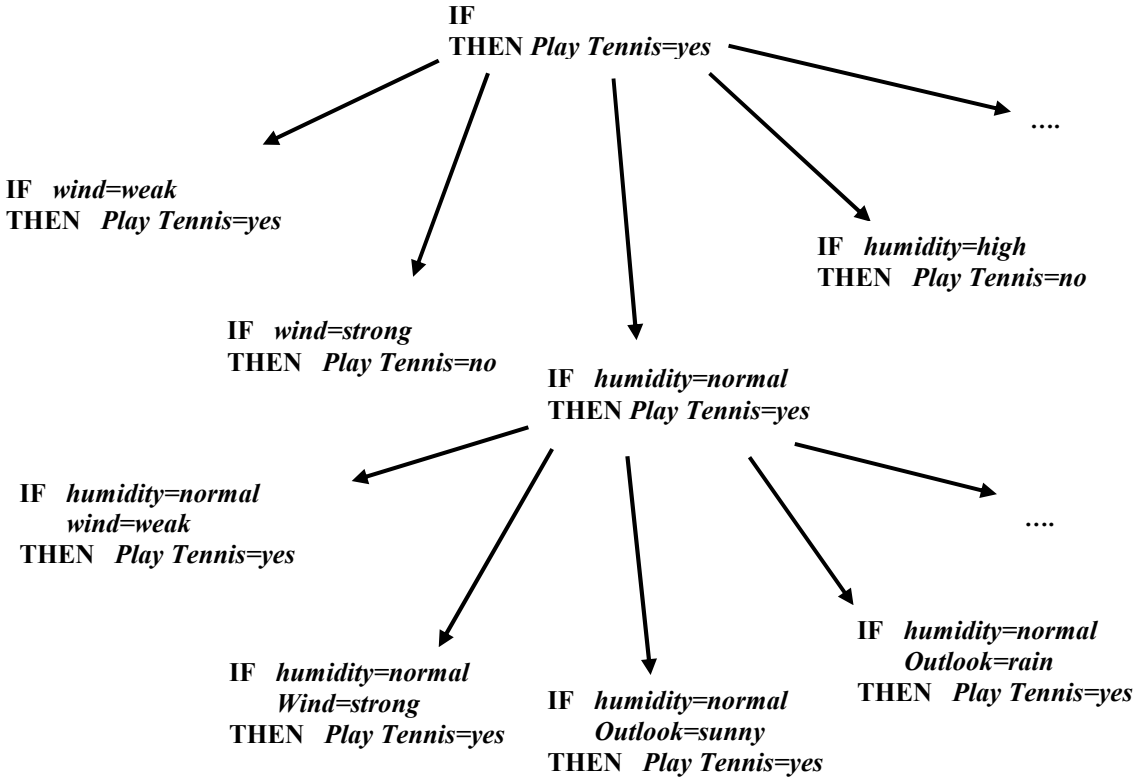
متدهایی که تا اینجا مورد بررسی قرار گرفتند تنها از تحلیل کلمات منفرد استفاده می‌کنند، همان‌طور که در مقدمه نیز اشاره شد مفاهیم تولید شده از، کنار هم قرار گرفتن کلمات در یک متن، نقش کلیدی در دسته‌بندی اسناد متنی دارد، درواقع این‌گونه مفاهیم ویژگی‌های پنهان داده‌های متنی هستند، که برای انجام یک دسته‌بندی با دقت بالا، بهره‌گیری از این ویژگی‌ها بسیار حائز اهمیت است. متدهایی که در ادامه مورد بررسی قرارخواهند گرفت به‌نوعی درصدد استخراج مفاهیم و استفاده از آن‌ها در دسته‌بندی متون هستند.

۲-۱-۳- برنامه‌نویسی منطق استنتاجی[2]

برنامه‌نویسی منطق استنتاجی، عبارتست از یادگیری قواعد منطقی[3] از مثال‌ها و دانشِ زمینه[4] ،و تولید مجموعه‌ای از قوانین اگر-آنگاه[5] که عملیات دسته‌بندی با استفاده از این مجموعه از قوانین انجام می-شود. نحوه‌ی عملکرد این متد تاحدودی شبیه فرایند یادگیری درخت تصمیم است، با این تفاوت که در برنامه‌نویسی منطق استنتاجی از الگوریتم‌های پوشش متوالی[6] استفاده می‌شود به این معنی که یادگیری قوانین بر پایه‌ی یادگیری یک قانون و حذف داده‌هایی که آن قانون پوشش می‌دهد و سپس تکرار دوباره این پروسس می‌باشد؛ در صورتی که در روش درخت تصمیم، الگوریتم‌های پوشش

[1] Incrementality
[2] Inductive logic programming
[3] Logic rules
[4] Background knowledge
[5] If then
[6] Sequential covering

همزمان[1] به‌کار می‌رود، که پس از هر مقایسه داده‌ها افراز شده، و بخش‌های افراز شده به صورت مجزا و همزمان مورد بررسی قرار میگیرند. برای نمونه، شکل(۲-۴) مثالی از عملکرد برنامه‌نویسی منطق استنتاجی را بر اساس داده‌های جدول(۲-۱) ارائه می‌دهد. همان‌طور که در شکل(۲-۴) نشان داده شده جستجو با در نظر گرفتن پیش‌شرط عمومی‌ترین قانون ممکن شروع می‌شود و سپس به صورت حریصانه ویژگی‌هایی را که بهترین افزایش کارایی قانون را روی مثال‌های آموزشی می‌دهند اضافه می‌کند. مثل ID3 این پروسس با اضافه کردن ویژگی‌های جدید به صورت حریصانه فرضیه‌ها را رشد می‌دهد تا وقتی که فرضیه‌ها به یک سطح قابل قبول از کارایی دست پیدا کنند. در تفاوت با ID3، پیاده‌سازی "یادگیری یک قانون" در هر مرحله از جستجو فقط در یک فرزند که بهترین کارایی را نتیجه می‌دهد ادامه پیدا می‌کند.



شکل(۲-۴) : مثالی از نحوه یادگیری در برنامه‌نویسی منطق استنتاجی

[1] Simultaneous covering

۲-۱-۳-۱- الگوریتم یادگیری FOIL

در این قسمت به معرفی یکی از الگوریتم‌های به‌کار رفته در سیستم‌های مبتنی بر قاعده می‌پردازیم. FOIL یک الگوریتم با خط مشی پوشش متوالی است که با دریافت نمونه‌های مثبت و منفی، مجموعه قوانینی را تولید می‌کند که همه‌ی نمونه‌های مثبت را پوشش داده و نمونه‌های منفی را رد کند.

FOIL(*Target-predicate, Predicates, Examples*)

- $Pos \leftarrow those\ Examples\ for\ which\ the\ Target-predicate\ is\ True$
- $Neg \leftarrow those\ Examples\ for\ which\ the\ Target-predicate\ is\ False$
- $Learned-roules \leftarrow \{\}$
- While *pos*, do
  Learn a NewRule
  - $NewRule \leftarrow$ $the\ rule\ that\ predicts\ Target-predicate\ \ with\ no\ precondition$
  - $NewRuleNeg \leftarrow Neg$
  - While $NewRuleNeg$, do
    Add a new literal to specialize NewRule
    - $Candidate-literals \leftarrow$ $generate\ candidate\ new\ literals\ for\ NewRule, based\ on\ predicates$
    - $Best-literal \leftarrow \text{argmax}_{L \in Candidate-litrals}\ Foil-gain(L, NewRule)$
    - $add\ Best-literal\ to\ precinditions\ of\ NewRule$
    - $NewRuleNeg \leftarrow$ $subset\ of\ NewRuleNeg\ that\ satisfies\ NewRule\ precondition$
  - $Learned-rules \leftarrow Learned-rules + NewRule$
  - $Pos \leftarrow Pos -$ {member of Pos covered by NewRule}
- Return $Learned-rules$

(شبه کد) : الگوریتم FOIL

همانطور که در الگوریتم FOIL قابل مشاهده است، در هربار تکرار حلقه‌ی while بیرونی یک قانون عمومی اضافه می‌شود که مجموعه‌ای از نمونه‌های مثبت و منفی را پوشش می‌دهد. در حلقه‌ی While داخلی این قانون با اضافه شدن پیش‌شرط‌ها خصوصی شده، تا وقتی که نمونه‌های منفی را پوشش ندهد، هنگامی که دیگر هیچ نمونه‌ی منفی مطابق با پیش‌شرط‌های قانون وجود نداشت، این قانون به مجموعه‌ی قوانین اضافه شده و نمونه‌های مثبت تحت پوشش این قانون از مجموعه‌ی نمونه‌های مثبت

حذف شده و حلقه‌ی while بیرونی تکرار می‌شود. تولید قانون تا وقتی که نمونه‌های مثبت وجود دارند ادامه می‌یابد.

۲-۱-۳-۲- استفاده از برنامه‌نویسی منطق استنتاجی در دسته‌بندی متن

در مقالات مختلف، تکنیک برنامه‌نویسی منطق استنتاجی به‌صورت مستقل در استخراج اطلاعات از متن و دسته‌بندی متون[۱۲و۱۳] و یا به صورت ترکیبی با متدهای آماری[۱۴] جهت دسته‌بندی اسناد مورد استفاده قرارگرفته است. در ادامه نمونه‌ای از قوانین استخراج شده جهت دسته‌بندی اسناد مشاهده می‌شود[۱۵].

```
invoice(Doc) :- invoice@Doc : P.
invoice(Doc) :- payment@Doc : P1 , next (P1 , P2) , within@Doc : P2.
offer(Doc) :- thank@Doc : P1, you@Doc : P2, for@Doc : P3,
              next (P1 , P2) , next (P2, 1, 3, P3).
```

- قانون اول کلاس صورت‌حساب را به هر سندی که کلمه invoice در هر جای آن رخ داده باشد نسبت می‌دهد.
- قانون دوم کلاس صورت‌حساب را به اسنادی نسبت می‌دهد که عبارت payment within در آن‌ها رخ داده باشد.
- آخرین قانون برای انتساب کلاس Offer اسناد را جهت رخ دادن عبارت thank you و کلمه for با فاصله‌ی دو کلمه مثلاً "thank you very much for" بازبینی می‌کند.

همانطور که در این مثال مشاهده می‌شود برای انتساب یک سند به کلاسی خاص باید سند را از بابت رخداد کلمات یا عبارات خاص آن کلاس مورد بررسی قرار داد، با این توصیف در مرحله یادگیری باید کلمات و عبارات کلیدی هر کلاس مشخص شود و این کلمات دقیقا در اسناد مربوط به آن کلاس

وجود داشته باشند. با این شرایط این متد جهت دسته‌بندی اسنادی با ساختار خاص مانند فرم‌های مربوط به خرید و فروش که شامل کلمات و عباراتی معین و خاص این امور هستند، مناسب‌تر است.

۲-۱-۳-۳- مزایا و معایب متد برنامه‌نویسی منطق استنتاجی

مزیت این روش نسبت به متدهای قبل این است که با اضافه شدن کلاس جدید نیاز به اجرای مجدد کل فرایند آموش نیست بلکه تنها عبارات و کلمات کلیدی کلاس جدید و قوانین کلاس‌بندی مربوط به آن استخراج شده و عملیات دسته‌بندی با مجموعه قوانین به‌روز شده ادامه می‌یابد در واقع این متد به نوعی دارای خاصیت افزایش تدریجی است.

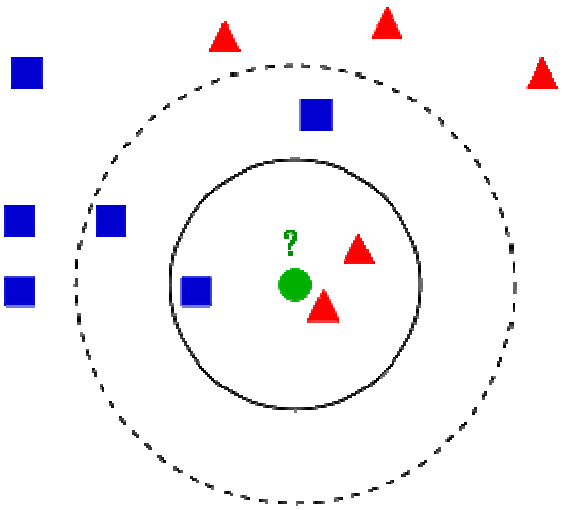
با استفاده از این متد در دسته‌بندی اسناد متنی، امکان تعریف قوانین پیچیده‌ای را داریم که توالی کلمات را در متن بررسی کند، که این مزیت سبب افزایش دقت دسته‌بندی می‌شود، با این وجود بررسی کل سند جهت یافتن کلمه یا عبارت مورد نظر، موجب می‌شود این متد از پیچیدگی محاسباتی نسبتا بالایی برخوردار باشد که امکان استفاده از این متد را به‌صورت آنلاین فراهم نمی‌کند.

۲-۱-۴- الگوریتم‌های خوشه‌بندی افزایشی[1]

خوشه‌بندی افزایشی یک استراتژی ضروری برای کاربردهای آنلاینی است که زمان فاکتوری مهم در قابلیت استفاده از آن‌هاست. الگوریتم‌های خوشه‌بندی افزایشی با پردازش یک داده در هر زمان، به‌طور افزایشی داده را به کلاستر مربوط به آن نسبت می‌دهند. این فرایند علی‌رغم سادگی ظاهری، درگیر چالش‌هایی چون ترتیب ورود داده‌ها و امکان انتساب داده به چند کلاستر است. در ادامه به اختصار به دو متد خوشه‌بندی افزایشی مورد استفاده در دسته‌بندی اسناد می‌پردازیم.

[1] Incremental clustering

- **خوشه‌بندی تک مرحله‌ای**[1] : این الگوریتم اسناد را به‌صورت ترتیبی[2] پردازش کرده و هر سند را با همه‌ی کلاسترهای موجود مقایسه می‌کند، شباهت میان سند جدید با هر کلاستر از یک مقدار آستانه بیشتر شد، سند به آن کلاستر اضافه می‌شود درغیر این‌صورت کلاستر جداگانه‌ای را تشکیل می‌دهد. معمولا شباهت میان سند و یک کلاستر با محاسبه‌ی میانگین شباهت میان سند و اسناد آن کلاستر اندازه‌گیری می‌شود.
- **خوشه‌بندی k نزدیکترین همسایه**[3] : اگرچه این روش اغلب در کلاس‌بندی کاربرد دارد ولی برای خوشه‌بندی نیز مورد استفاده قرار می‌گیرد. در این روش با ورود سند جدید شباهت آن با تمام اسناد محاسبه شده، K تا از شبیه‌ترین اسناد انتخاب شده و سند جدید به کلاستری اختصاص می‌یابد که اکثریت این K سند به آن تعلق دارند. همانطور که در شکل(۲-۵) مشاهده می‌شود کلاستر مربوط به دایره در صورتی که شعاع همسایگی برابر ۳ باشد مثلث و با شعاع همسایگی ۵ برابر مربع است.



شکل(۲-۵) : شمایی از خوشه‌بندی K نزدیکترین همسایه

[1] Single-pass clustering
[2] Sequentially
[3] K-nearest neighbor clustering

۲-۲- بررسی مدل‌های نمایش مورد استفاده در دسته‌بندی اسناد

همان‌طور که پیش‌تر عنوان شد، یکی از مسائل مهم در زمینه‌ی دسته‌بندی، نحوه‌ی نمایش داده-هاست که با عنوان مدل نمایش داده از آن یاد می‌شود. اهمیت به‌کارگیری یک مدل نمایش مناسب و حاوی ویژگی‌هایی با حداکثر اطلاعات در مقدمه مورد بررسی قرار گرفت. در این بخش پس از تعریف مدل نمایش سند متنی، مدل‌های مرسوم مورد استفاده در نمایش اسناد متنی را مورد بررسی قرار داده و به بیان مزایا و معایب هر کدام می‌پردازیم.

مدل سند متنی در واقع مفهومی است که توصیف می‌کند چگونه یک مجموعه‌ی با معنی از ویژگی‌ها طی عملیات پیش‌پردازش از کلمات سند استخراج می‌شود. با معنی بودن به این معناست که تابعی وجود داشته‌باشد که بتواند نگاشتی را از مجموعه ویژگی‌های دو سند به بازه [1 0] داشته باشد که این تابع همان معیار شباهت است. هرچقدر مقدار تابع به یک نزدیک باشد نشان‌دهنده‌ی شباهت دو سند و درصورت عدم تشابه این مقدار نزدیک به صفر خواهد بود[۱۶].

لازم به ذکر است که معیار شباهت وابسته به مدل نمایش داده بوده و با توجه به آن انتخاب می‌شود.

مدل‌های مرسوم نمایش اسناد متنی عبارتند از:

- مدل فضای برداری[1]
- مدل پنجره لغزان[2]
- مدل درخت پسوندی[3]

[1] Vector space model
[2] n-gram model
[3] Suffix tree model

۲-۲-۱- مدل فضای برداری

درحال حاضر مدل فضای برداری مرسوم‌ترین مدل نمایش اسناد در دسته‌بندی اسناد متنی است که توسط Salton در سال ۱۹۷۵ معرفی شده است[۱۷]. این مدل اسناد را به‌صورت یک بردار ویژگی از کلمات منحصر به‌فرد به‌کار رفته در کل مجموعه‌ی اسناد نمایش می‌دهد:

$$\overrightarrow{d_i} = (w_{i1}, w_{i2}, \dots, w_{i|z|})$$

که |z| تعداد کلمات منحصر به فرد در کل مجموعه‌ی اسناد و w_{ij} وزن j امین کلمه از i امین سند است. وزن کلمات به چند روش قابل محاسبه است[۱۸و۱۹].

- ساده‌ترین متد وزن‌دهی بولی است. در این روش وزن کلمه درصورت رخ دادن در سند ۱ و درغیر این‌صورت برابر صفر است (فرمول۲-۸).

$$w_{ij} = \begin{cases} 1 & if \;\; TF(t_j, d_i) > 0 \quad (TF = term\ frequency) \\ 0 & otherwise \end{cases} \qquad (۸-۲)$$

- روش دیگر استفاده از فرکانس کلمات در سند، به عنوان وزن آن‌هاست (فرمول ۲-۹).

$$w_{\mathrm{ij}} = TF(t_j, d_i) \qquad (۹-۲)$$

- در رویکردی دقیق‌تر جهت محاسبه‌ی وزن کلمات از معکوس فرکانس اسناد[1] استفاده می‌شود. فلسفه استفاده از معکوس فرکانس اسناد افزایش تاثیر کلماتی است که در اسناد کمتری

[1] Inverse document frequency

ظاهر شده‌اند و به همین خاطر قدرت جداکنندگی[1] بالاتری دارند و کاهش تاثیر کلماتی است که در اغلب اسناد ظاهر شده‌اند و درنتیجه قدرت جداکنندگی کمتری دارند[۱۶].

$$w_{ij} = TF(t_j, d_i) \times IDF(t_j) \quad j = 1, \ldots, |Z|, i = 1, \ldots, |Q| \quad (۱۰-۲)$$

$$IDF(t_j) = \log\left(\frac{|Q|}{DF(t_j)}\right) \qquad j = 1, \ldots, |Z| \quad (۱۱-۲)$$

در عبارات بالا |Z| تعداد کلمات مستخرج از مجموعه‌ی اسناد، |Q| تعداد کل اسناد موجود در مجموعه و $DF(t_j)$ تعدا اسنادی است که شامل j امین کلمه هستند.

- در روشی مشابه با روش قبل با در نظرگرفتن طول متفاوت اسناد سعی در نرمال کردن وزن‌ها شده است.

$$w_{ij} = \frac{TF(t_j,d_i)\times IDF(t_j)}{\sqrt{\sum_{k=1}^{Z}[TF(t_k,d_i)\times IDF(t_k)]^2}} \quad (۱۲-۲)$$

با توجه به این‌که برای نمایش اسناد در این مدل باید از همه‌ی کلمات منحصر به‌فرد مستخرج از کل مجموعه‌ی اسناد استفاده کرد، بزرگ شدن مجموعه‌ی اسناد با افزایش ابعاد بردار ویژگی همراه است که منجر به افت عملکرد دسته‌بندی می‌شود، به همین دلیل باید ابعاد بردار ویژگی را تا حد ممکن کاهش داد و تنها از واژه‌هایی با بیشترین بار اطلاعاتی در بردار ویژگی استفاده کرد. در ادامه دو متد انتخاب ویژگی جهت کاهش ابعاد بردار ویژگی بررسی می‌شود[۱۹و۲۰].

[1] Discriminative power

- **حد آستانه فرکانس سند**[1]: فرکانس سند برای یک کلمه تعداد اسنادی است که آن کلمه در آن‌ها رخ داده. در این روش فرکانس سند به ازای تمام کلمات مستخرج محاسبه شده و کلماتی با فرکانس سند کمتر از حد آستانه تعیین شده حذف می‌شوند. فرض بر این است که کلمات نادر فاقد اطلاعات کافی جهت دسته‌بندی‌اند و ممکن است نویز باشند.
- **نفع اطلاعاتی**[2] : در این روش به‌ازی تمام کلمات مقدار نفع اطلاعاتی با فرمول(۲-۱۳) محاسبه شده و کلماتی که نفع اطلاعاتی آن‌ها کمتر از حد آستانه باشد حذف می‌شوند.

$$\begin{aligned} IG(w) = & -\sum_{j=1}^{K} P(c_j) log P(c_j) \\ & + P(w) \sum_{j=1}^{k} P(c_j|w) log P(c_j|w) \\ & + P(\overline{w}) \sum_{j=1}^{k} P(c_j|\overline{w}) log P(c_j|\overline{w}) \end{aligned} \qquad (۱۳-۲)$$

در این فرمول $P(c_j)$ نسبت اسناد متعلق به کلاس c_j به کل اسناد، $P(w)$ نسبت اسناد حاوی کلمه‌ی w به کل اسناد، $P(c_j|w)$ نسبت اسنادی از کلاس c_j که w حداقل یک‌بار در آن‌ها رخ داده و $P(c_j|\overline{w})$ نسبت اسنادی از کلاس c_j که w در آن‌ها رخ نداده است.

با توجه به نمایش برداری اسناد در مدل فضای برداری، برای اندازه‌گیری شباهت میان اسناد باید از معیارهای شباهت مبتنی بر بردار استفاده کرد. دو معیار شباهت کسینوس[3] و جاکارد[4] معیارهای مورد استفاده در دسته‌بندی اسناد با این مدل نمایش هستند. در معیار کسینوس (فرمول ۲-۱۴) در واقع کسینوس زاویه میان دو بردار محاسبه می‌شود که درصورت منطبق بودن، زاویه صفر و مقدار کسینوس برابر ۱ خواهد بود که نشان‌دهنده‌ی حداکثر شباهت است.

[1] Document frequency thresholding
[2] Information gain
[3] Cosine
[4] Jaccard

$$sim_C(\overrightarrow{d_1}, \overrightarrow{d_2}) = \frac{\overrightarrow{d_1} . \overrightarrow{d_2}}{|\overrightarrow{d_1}| \times |\overrightarrow{d_2}|} \qquad (۱۴-۲)$$

در اسناد متنی، ضریب جاکارد (فرمول ۲-۱۵) نسبت مجموع وزن کلمات مشترک دو سند را به وزن کلمات غیرمشترک دو سند محاسبه می‌کند.

$$sim_J(\overrightarrow{d_1}, \overrightarrow{d_2}) = \frac{\overrightarrow{d_1} . \overrightarrow{d_2}}{|\overrightarrow{d_1}|^2 \times |\overrightarrow{d_2}|^2 - \overrightarrow{d_1} . \overrightarrow{d_2}} \qquad (۱۵-۲)$$

۲-۲-۱-۱- مزایا و معایب مدل فضای برداری

مزایا :

- مدلی ساده براساس جبر خطی است.
- اندازه‌گیری شباهت میان اسناد در این مدل با پیچیدگی محاسباتی پایینی همراه است.

معایب:

- ابعاد (ویژگی‌های) زیاد
- از بین رفتن ترتیب کلمات و شکسته شدن ساختارهای نحوی که سبب حذف مفهوم می‌شود.
- فرض بر این است که کلمات از لحاظ آماری مستقل هستند.
- دو سند با مفهوم برابر و مجموعه لغات مختلف در این مدل ارتباطی به هم ندارند.
- وجود داده‌های نویزی

۲-۲-۲- مدل پنجره لغزان

عناصر بردار ویژگی را می‌توان به‌جای کلمات n-gram درنظر گرفت. این مدل به دو شکلِ مبتنی بر کلمه و مبتنی بر کاراکتر مورد استفاده قرار می‌گیرد. تعریفی که در مقدمه از n-gram آمده مبتنی بر کلمه است[۲۱]. به عنوان نمونه‌ای از این مدل inform retriev یک 2-gram است و عبارات محتمل رخ داده در متن مطابق با این 2-gram عبارتند از:

(a) information retrieval
(b) retrieval of information
(c) informative retrieval
(d) retrieved information
(e) retrieving information
(f) retrieves information
(g) Retrieve information!
(h) He informs the retriever

همان‌طور که در نمونه‌ها قابل مشاهده است، عبارات اسمی (a تا d)، عبارات فعلی (e و f) و جملات کامل (g و h) همه می‌توانند رخدادی از یک n-gram مشترک باشند. در مورد پراکندگی معنایی نیز همان‌طور که در موارد a و d مشاهده می‌شود، عبارات اسمی با معانی مختلف می‌توانند n-gram مشترک داشته باشند. تعریف n-gram با این روش براساس این فرض استوار است که حالات مختلف نحوی ممکن است یک مفهوم را انتقال دهند. البته درنظر گرفتن این فرضیه مسائل زیر را به همراه دارد:

- **تعمیم بیش از حد**[1]: همان‌طور که در نمونه‌های c و h مشاهده می‌شود این دو مورد به مفهومِ مورد نظر بقیه نمونه‌ها اشاره نمی‌کند.

[1] Over-generalization

- **تعمیم ناقص**[1]: این مورد هنگامی رخ می‌دهد که عبارتی مثل retrieving interesting information که قطعا به مفهوم مورد c اشاره می‌کند در این مجموعه به رسمیت شناخته نمی‌شود.

البته باید به این نکته توجه شود که صرف وقوع دو کلمه در متن تضمین کننده‌ی مرتبط بودن آن با به یک مفهوم پیچیده نیست.

کار کردن با متون معمولا نیاز به دانشی درمورد زبان مورد استفاده در متن دارد. انجام اموری چون حذف stop wordها از متن و ریشه‌یابی کلمات بدون شناخت زبانِ مورد استفاده امکان‌پذیر نیست.[۲۲] استفاده از مدل n-gram مبتنی بر کاراکتر، این امکان را به ما می‌دهد که متون را مستقل از زبان مورد استفاده‌ی آن‌ها مورد پردازش قرار دهیم[۲۲،۲۳،۲۴،۲۵،۲۶].

یک n-gram کاراکتری، زیررشته‌ای n کاراکتری، از رشته‌ای بلندتر است. مدل‌های n-gram، مبتنی بر استخراج و شمارش n-gram ها با استفاده از یک پنجره لغزان به طول n در یک سند هستند[۲۴]. در جدول(۲-۲) نمونه‌ای از این مدل را می‌بینید.

جدول(۲-۲) : نمایش n-gram کاراکتری کلمه‌ی "corpus" با طول‌های مختلف

$n=1$	$n=2$	$n=3$	$n=4$	$n=5$
_	_c	_co	_cor	_corp
c	co	cor	corp	corpu
o	or	orp	orpu	orpus
r	rp	rpu	rpus	rpus_
p	pu	pus	pus_	pus__
u	us	us_	us__	us___
s	s_	s__	s___	s____

[1] Under-generalization

برای نمایش اسناد با استفاده از n-gram کاراکتری نیازی به پیش‌پردازش‌های زبانی نیست. این تکنیک اسناد را به بردارهای ویژگی با ابعاد بسیار بالا تبدیل می‌کند. برای کاهش ابعاد بردار ویژگی از روش‌های مختلفی استفاده می‌شود. در[۲۴] انتخاب n-gram ها براساس فرکانس سند انجام می‌شود، به این‌صورت که مقادیر حد آستانه مینیمم و ماکزیمم برای مقدار فرکانس سند درنظر گرفته می‌شود و n-gram هایی با مقدار فرکانس سند در این بازه انتخاب و بقیه حذف می‌شوند. در مقاله‌ای دیگر[۲۳] برای هر سند برداری از n-gramها با طول برابر ساخته می‌شود، سپس با استفاده از معیار Chi-square، n-gram ها رتبه‌بندی شده و n-gram هایی با رتبه‌های کمتر از حد آستانه حذف می‌شوند. با وجود روش‌های مطرح شده به‌دلیل ابعاد بالای بردار ویژگی کاهش تعداد n-gram ها موضوع مورد بحث بسیاری از تحقیقات انجام شده در این زمینه است[۲۲، ۲۵، ۲۷].

برای دسته‌بندی اسناد با این مدل، پس از تشکیل بردار ویژگیِ هرسند و انتخاب ویژگی‌ها، با استفاده از بردارهای اسناد مربوط به هر کلاس، یک بردار مرکزی برای هر کلاس تولید می‌کنیم، در مرحله‌ی تست، بردار داده‌ی تست با بردارهای مرکزی کلاس‌های مختلف مقایسه شده و با هر بردار کمترین فاصله را داشته باشد به آن کلاس تعلق می‌یابد. برای اینکه بتوان یک سند را به بیش از یک کلاس ارجاع داد مقدار آستانه‌ای تعیین می‌کنیم که اگر فاصله‌ی سند با بردار مرکزیِ یک کلاس کمتر یا مساوی آن باشد متعلق به آن کلاس خواهد بود. اندازه‌گیری فاصله سند با کلاستر با فرمول(۲-۱۶) انجام می‌شود [۲۳].

$$d(D_i, C_j) = \sum_{m \in D_i \cup C_j} \left(\frac{2.(f_i(m) - f_j(m))}{f_i(m) + f_j(m)} \right)^2 \qquad (۱۶-۲)$$

در این فرمول $f_i(m)$ فرکانس n-gram، m در سند i و $f_j(m)$ فرکانس n-gram، m در کلاس j است.

۲-۲-۲-۱- مزایا و معایب مدل پنجره لغزان

مزایا:

- n-gram مبتنی بر کاراکتر در برابر خطاهای کوچک تایپی مقاوم است.
- از خاصیت مجاورت میان کلمات در دسته‌بندی اسناد استفاده می‌کند.
- n-gram مبتنی بر کاراکتر مستقل از زبان بوده و نیازی به پیش‌پردازش‌های زبانی ندارد.
- به همراه مدل مارکوف کاربرد وسیعی در بازشناسی گفتار[1] دارد.

معایب:

- ابعاد بالای بردار ویژگی
- ثابت بودن طول n-gram

۲-۲-۳- مدل درخت پسوندی[2]

مدل درخت پسوندی اسناد را به صورت دنباله‌ای از کلمات درنظر می‌گیرد. در این مدل هر سند به‌وسیله‌ی مجموعه‌ای از زیررشته‌های پسوندی در قالب یک درخت نمایش داده می‌شود. به تعبیر دیگر مدل درخت پسوندی می‌تواند یک متد n-gram با سایز انعطاف‌پذیر را برای تشخیص و استخراج عبارات همپوشان در اسناد فراهم کند[۱۶، ۲۸].

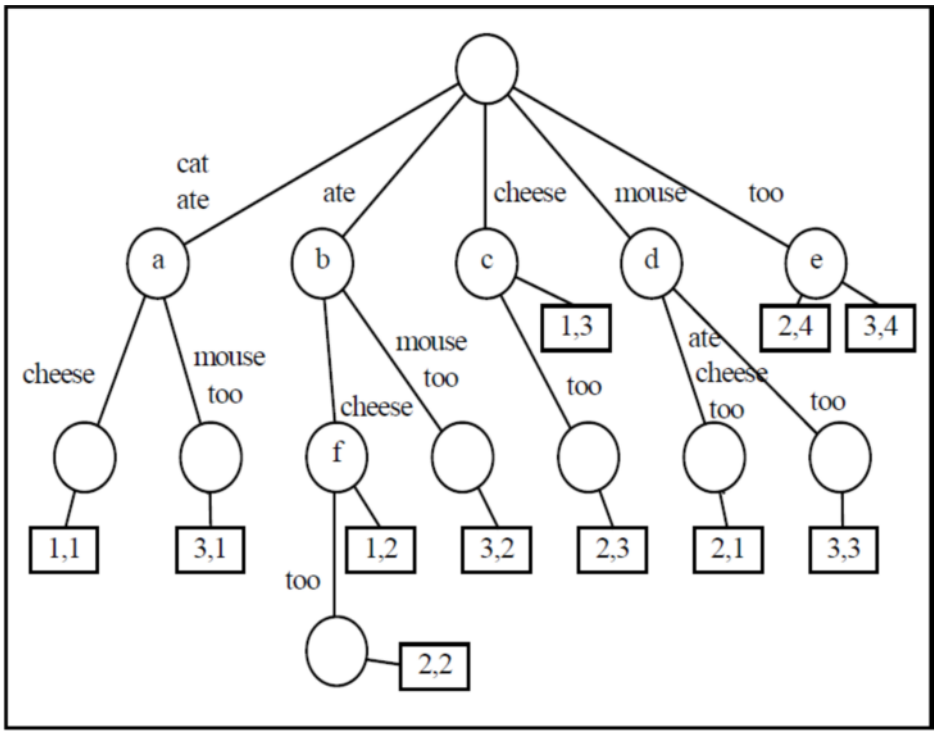
برای ساخت درخت پسوندی ابتدا باید تعریف دقیقی از پسوند داشته باشیم. اگر سند d را به صورت رشته‌ای از کلمات در نظر بگیریم ($d=w_1 w_2 \dots w_m$) i امین پسوند سند d زیر رشته‌ای از d است که با w_i شروع و به انتهای رشته ختم می‌شود. درخت پسوندی سند d، درختی برچسب‌دار شامل همه‌ی

[1] Speech recognition
[2] Suffix tree model

پسوندهای سند است که هر پسوند درطول مسیری از ریشه تا برگ است که یال‌های (نودهای) این مسیر با کلمات عبارت پسوندی برچسب‌گذاری شده‌اند.

برای ساخت درخت پسوندی به صورت زیر عمل می‌شود:

اضافه کردن iامین پسوند d با چک کردن این که از میان یال‌های منشعب شده از ریشه، یالی با w_i برچسب‌گذاری شده، شروع می‌شود. در صورت وجود داشتن چنین یالی آن را پیمایش کرده و در بین یال‌های منشعب از نود مربوط به این یال چک می‌کنیم که آیا با w_{i+1} برچسب‌گذاری شده‌اند یا خیر، اگر در عمق k نود n بدون یالی منطبق با این شرط پیدا شد، یک نود جدید ایجاد و به نود n متصل می‌کنیم که یال آن با رشته‌ی w_{i+k} برچسب گذاری می‌شود[۱۶]. شکل(۲-۶) نمونه ای از یک درخت پسوندی را که از سه رشته‌ی "cat ate cheese"، "mouse ate cheese too" و "cat ate mouse too" تشکیل شده نشان می‌دهد[۲۹]. عددهای ضمیمه شده به نودها نشان دهنده‌ی شناسه‌ی رشته و شماره‌ی پسوند در آن رشته است.



شکل(۲-۶) : نمونه‌ای از درخت پسوندی

مدل نمایش درخت پسوندی شباهت بین دو سند را برحسب همپوشانی رشته‌ها در درخت پسوندی مشترک آن‌ها تعریف می‌کند. در ساده‌ترین حالت در[۱۶] شباهت دو سند به این صورت محاسبه می-شود: اگر دو سند d^+ و d^- در یک درخت پسوندی اولیه درج شوند، هر یال درخت T برچسب + ، - یا +- را با توجه به اینکه یال e هنگام درج یک پسوند از d^+ ، d^- یا هردو پیمایش شده دریافت می‌کند، شباهت دو سند برابر است با:

$$\varphi ST = \frac{|E^+ \cap E^-|}{|E^+ \cup E^-|} \qquad (۱۷-۲)$$

(E^+ و E^- یال‌هایی در E هستند که برچسب + یا – دارند.)

این معیار فرکانس پسوندها را به صورت بولین (۰ و ۱) در نظر می‌گیرد و تعداد دفعاتی که یک یال در جریان درج پسوندها پیمایش شده ثبت نمی‌شود.

در نگاهی دقیق‌تر در[۳۰] در هنگام ساخت درخت به هر نود اطلاعاتی از جمله شناسه‌ی سندهای شامل آن عبارت و تعداد پیمایش هر کدام از اسناد از این نود ضمیمه می‌شود که با این اطلاعات می-توان در مورد هر عبارت، فرکانس و فرکانس سند را استخراج کرد، پس از ساخت درخت با نگاشت هر نود (غیر از ریشه) برچسب‌گذاری شده با یک عبارت ناتهی به یک ویژگی در برداری به ابعاد تعداد نودهای درخت (غیر از ریشه)، هر سند با یک بردار ویژگی از وزن‌های عبارات نودهای درخت نمایش داده می‌شود. درواقع به‌جای کلمات در مدل فضای برداری عبارات جایگزین می‌شوند. با این مدل نمایش درخت پسوندی، می‌توان وزن عبارات را با روش‌های مطرح شده در مدل فضای برداری محاسبه کرد. در مورد معیار شباهت نیز با توجه به نمایش برداری می‌توان از معیار کسینوس برای این منظور بهره برد.

۲-۲-۳-۱- خوشه‌بندی درخت پسوندی

الگوریتم خوشه‌بندی درخت پسوندی بر مبنای مدل درخت پسوندی توسعه یافته و ۳ مرحله دارد:

۱. ساخت درخت پسوندی: یک درخت پسوندی برای همه‌ی پسوندهای اسناد موجود در مجموعه‌ی $D=\{d_1,d_2, \dots ,d_N\}$ ساخته می‌شود. هر نود داخلی (نودی که نه ریشه باشد و نه برگ) که حداقل در دو سند مختلف وجود دارد به عنوان خوشه پایه استخراج می‌شود. در شکل(۲-۷) خوشه‌ی پایه‌ی استخراج شده از درخت پسوندی شکل(۲-۶) با ضمیمه شدن شناسه رشته‌ی حاوی آن‌ها مشخص شده‌اند.

Node	*Phrase*	*Documents*
a	cat ate	1,3
b	ate	1,2,3
c	cheese	1,2
d	mouse	2,3
e	too	2,3
f	ate cheese	1,2

شکل(۲-۷) : استخراج خوشه‌های پایه از درخت پسوندی

۲. انتخاب خوشه‌ی پایه: به هر خوشه‌ی پایه یک امتیاز s(B) اختصاص می‌یابد.

$$s(B) = |B| \,.\, f(|P|) \qquad (۱۸-۲)$$

در رابطه(۲-۱۸) |B| تعداد اسناد حاوی عبارت برچسب شده به خوشه‌ی پایه و |P| تعداد کلمات عبارت P برچسب شده به خوشه‌ی پایه است. تابع f با توجه به طول عبارات به آن‌ها امتیاز می‌دهد،

این تابع برای عبارات تک کلمه‌ای مقدار بسیار کم، برای عبارات ۲ تا ۶ کلمه‌ای به صورت خطی عمل می‌کند و برای عبارات با طول بیشتر مقداری ثابت است.

پس از امتیازدهی، تمام خوشه‌های پایه برحسب امتیاز مرتب شده و K تا از خوشه‌های پایه با بالاترین امتیاز برای عملیات ادغام مرحله‌ی بعد انتخاب می‌شوند.

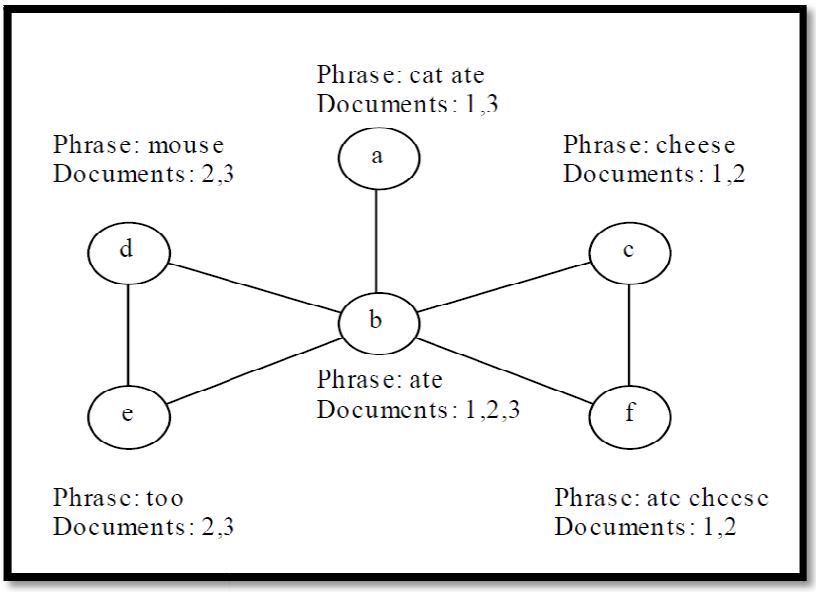
۳. ادغام خوشه‌ها: در این مرحله یک گراف شباهت شامل K خوشه‌ی پایه تولید می‌شود. در این گراف دو خوشه پایه‌ی B_i و B_j در صورتی با یال به هم متصل می‌شوند که ضریب جاکارد بین آن‌ها بیشتر از ۰/۵ باشد.

$$\frac{|B_i \cap B_j|}{|B_i \cup B_j|} > 0.5 \qquad (۱۹-۲)$$

در این رابطه $|B_i \cap B_j|$ تعداد اسنادی است که در این عبارت مشترک هستند و $|B_i \cup B_j|$ مجموع تعداد اسناد حاوی هر کدام از دو عبارت است.

پس از تشکیل گراف شباهت، مولفه‌های همبند در گراف خوشه‌های نهایی را شکل می‌دهند. برای مثال در درخت شکل (۲-۶) نودهای a، b، c، d، e و f به عنوان خوشه‌های پایه انتخاب شده‌اند و در نهایت ۶ خوشه پایه پس از ادغام خوشه نهایی را شکل می‌دهند، شکل(۲-۸).

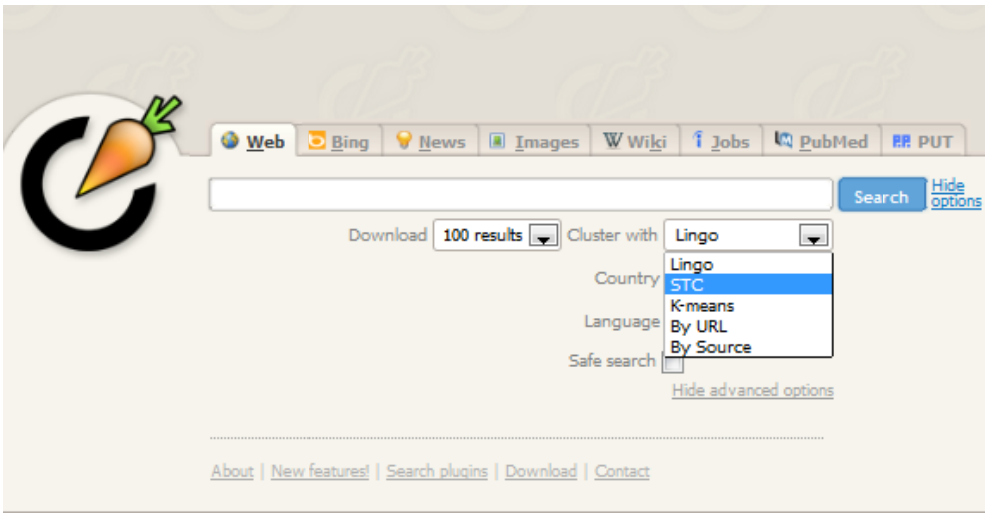
در این مثال تنها یک مولفه‌ی همبند در گراف شباهت وجود دارد و به همین دلیل تنها یک خوشه نهایی تولید می‌شود ولی در صورتی که کلمه‌ی "ate" جزو کلمات حذف شده در مرحله پیش‌پردازش باشد آن‌گاه با حذف خوشه‌ی پایه‌ی نود b در گراف شباهت سه مولفه‌ی همبند و در نتیجه ۳ خوشه نهایی ایجاد خواهد شد.



شکل(۲-۸) : گراف شباهت درخت پسوندی شکل(۲-۶)

در عمل الگوریتم خوشه‌بندی درخت پسوندی جهت خوشه‌بندی بریده‌های[1] متن اسناد بازیابی شده توسط موتورهای جستجو به خوبی کار می‌کند[۳۰].

به عنوان نمونه‌ی عملی کاربرد درخت پسوندی، در استفاده از موتور جستجوی carrot تنظیماتی وجود دارد که می‌توان تعیین کرد عملیات خوشه‌بندی با چه متدی انجام شود، یکی از متدهای موجود درخت پسوندی است، شکل(۲-۹).



شکل(۲-۹) : نمایی از موتور جستجوی carrot

[1] Snippets

۲-۲-۳-۲- مزایا و معایب مدل درخت پسوندی

مزایا:

- استفاده از خاصیت مجاورت میان کلمات با بهره‌گیری از عبارات در دسته‌بندی
- در این مدل می‌توان از امکان همپوشانی در خوشه‌بندی استفاده کرد.
- درخت پسوندی در زمانی خطی متناسب با سایز سند و به صورت تدریجی همزمان با آمدن اسناد ساخته می‌شود.

معایب:

- فاکتور انشعاب این درخت بسیار بزرگ است به‌خصوص در سطح اول درخت، که هر پسوند احتمالی تشخیص داده شده در مجموعه‌ی اسناد، از گره ریشه منشعب می‌شود.
- افزونگی بیش از حد پسوندها در گره‌های مختلف درخت

۲-۳- جمع‌بندی فصل

در این فصل تلاش کردیم ابتدا متدهای به‌کار رفته در دسته‌بندی اسناد را به‌طور مختصر مورد بررسی قرار دهیم. متدهایی چون درخت تصمیم، شبکه‌ی عصبی، برنامه‌نویسی منطق استنتاجی و تحلیل آماری که حاصل اشتراک زمینه‌های تحقیقاتی مختلفی چون پایگاه داده، بازیابی اطلاعات و هوش مصنوعی شامل یادگیری ماشین و پردازش زبان‌های طبیعی هستند.

از میان نیازمندی‌های یک سیستم دسته‌بندی آنلاین اسناد دو مورد افزایش تدریجی و همپوشانی به‌طور مستقیم تحت تاثیر مدل خوشه‌بندی قرار دارند که در اغلب متدهای مورد بررسی این دو نیاز اساسی پشتیبانی نمی‌شوند. در این میان دو الگوریتم خوشه‌بندی افزایشی ترتیبی و k نزدیکترین همسایه

تنها متدهایی هستند که این شرایط را برای حضور در یک سیستم دسته‌بندی آنلاین اسناد فراهم می‌کنند.

پس از بررسی متدهای خوشه‌بندی، مدل‌های مختلف نمایش اسناد را مورد بحث قرار دادیم. در بین مدل‌های نمایش اسناد، مدل درخت پسوندی با به‌کارگیری عبارات در نمایش اسناد به خوبی از خاصیت مجاورت میان کلمات، که مسئله‌ی کلیدی در انتقال مفاهیم توسط متن است، بهره می‌گیرد با این حال با مشکل افزونگی بسیار بالای پسوندها مواجه است.

در ادامه در فصل بعد مدلی مورد بررسی قرار خواهد گرفت که همانند مدل درخت پسوندی از عبارات در نمایش اسناد استفاده می‌کند ولی معایب این مدل را ندارد.

فصل سوم

تکنیک مورد مطالعه

۳-۱- مقدمه

همان‌طور که در فصل‌های قبل اشاره شد، تکنیک‌های خوشه‌بندی اسناد اغلب بر پایه‌ی تحلیل کلمات منفرد اسناد موجود در مجموعه‌ی داده هستند. جهت دستیابی به خوشه‌بندی دقیق‌تر، استفاده از ویژگی‌های حاوی اطلاعات بیشتر، مانند عبارات و وزن آن‌ها در اسناد می‌تواند بسیار مفید واقع شود. در این فصل مدلی مورد بررسی قرار خواهد گرفت که از عبارات برای نمایش اسناد بهره می‌گیرد، اما برخلاف مدل درخت پسوندی، جهت تسهیل در نمایش عبارات از تئوری گراف برای این منظور استفاده می‌کند.

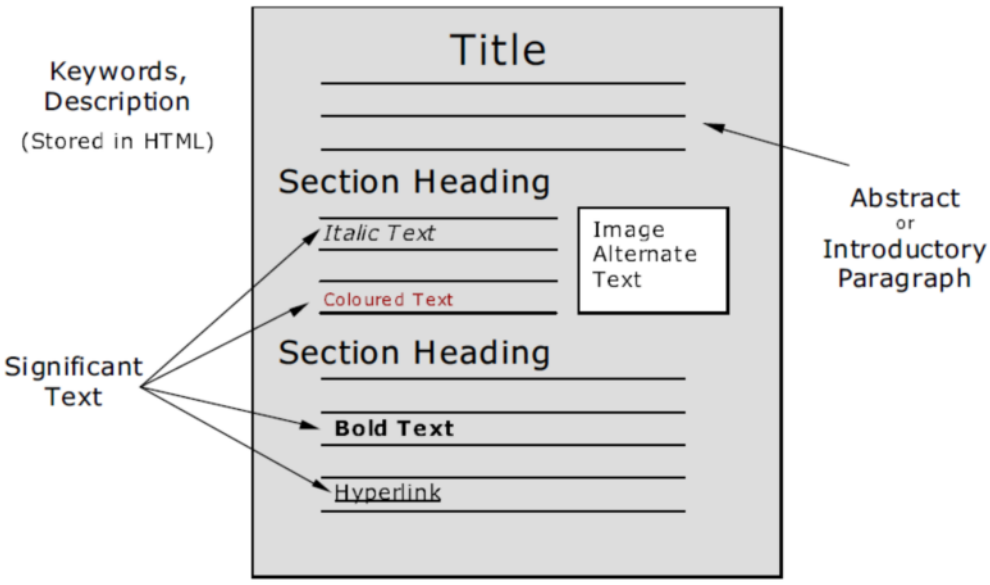
در ادامه‌ی این فصل ابتدا ساختار اسناد وب را مورد بررسی قرار داده و توضیح می‌دهیم که چگونه از این ساختار در جهت بهبود دقت خوشه‌بندی بهره می‌بریم، سپس به معرفی مدل گراف نمایه‌سازی اسناد و نحوه ساخت آن می‌پردازیم و معیار شباهت متناسب با این مدل را مورد بحث قرار خواهیم داد، و نهایتا به شریح فرایند خوشه‌بندی اسناد در قالب این مدل می‌پردازیم.

۳-۲- بررسی ساختار اسناد وب

زبان نشانه‌گذاری ابرمتنی[1] زبان توصیف ساختار صفحه‌های وب است. دستورعمل‌های این زبان، برچسب[2] نام دارند که محتوای یک صفحه‌ی وب، با آن‌ها، نشانه‌گذاری شده و بدین‌ترتیب، نحوه‌ی نمایش آن صفحه برای مرورگرهای وب، توصیف می‌شود. هر یک از برچسب‌های HTML، معنا و مفهوم خاصی دارند و تأثیر مشخصی بر محتوا می‌گذارند؛ مثلاً برچسب‌هایی برای تغییر شکل ظاهری متن، نظیر درشت و ضخیم کردن یک کلمه یا برقراری پیوند به صفحات دیگر در HTML تعریف شده‌اند. در واقع برچسب‌ها برای تعیین بخش‌های مختلف سند مورد استفاده می‌گیرند و براساس این

[1] HTML: hypertext markup language
[2] Tag

قابلیت می‌توان بخش‌های کلیدی سند را تعیین کرد. ایده‌ی مطرح شده در این مدل این است، که بخش‌هایی از سند دارای اطلاعات بیشتری نسبت به دیگر بخش‌ها هستند. بنابراین قسمت‌های مختلف سند دارای سطوح اهمیت[1] مختلف هستند که با برچسب‌های HTML قابل تشخیص است. سیستم پیشنهادی، سند HTML را تحلیل می‌کند و آن را براساس ساختار از پیش تعیین شده که سطوح اهمیت مختلفی را به بخش‌های مختلف سند نسبت می‌دهد، بازسازی می‌کند. در شکل(۳-۱) نمای کلی ساختار اسناد وب را مشاهده می‌کنید. همانطور که در این شکل نیز مشخص است بخش-های مختلف سند بنا به میزان اهمیت به فرم ایتالیک، رنگی یا با فونت درشت نمایش داده می‌شوند.



شکل(۳-۱): شمای کلی ساختار اسناد وب[۱]

در این مدل چهار سطح اهمیتِ خیلی زیاد، زیاد، متوسط و کم تعریف شده که متناسب با اهمیت هر بخش به آن نسبت داده می‌شود. در جدول(۳-۱) سطوح اهمیت مرتبط با هر بخش آمده است. این سطوح اهمیت در بخش‌های بعد در محاسبه شباهت میان اسناد مورد استفاده قرار می‌گیرد.

[1] Levels of significance

جدول(۳-۱): سطوح اهمیت بخش‌های مختلف سند[۱]

Part	Significance Level
Title	Very High
Keywords	Very High
Description	Very High
Section Headings	High
Captions	Medium
Introductory Paragraph (Abstract)	Medium
Significant Text (Bold, Emphasized, Italics, Colored)	Medium
Image Alternate Text	Medium
Hyper-linked Text	Medium
Body Text	Low

۳-۳- گراف نمایه‌سازی اسناد[1]

در این بخش به معرفی مدل جدید گراف نمایه‌سازی اسناد (DIG) می‌پردازیم. این مدل اسناد را با حفظ ساختار اصلی جملات، نمایه‌سازی کرده و امکان بهره‌گیری از انطباق عبارات با هر طولی را به‌جای کلمات مجزا، فراهم می‌کند. در ادامه، ساختار این مدل مورد بررسی قرار می‌گیرد[۱].

۳-۳-۱- بررسی ساختار گراف نمایه‌سازی اسناد

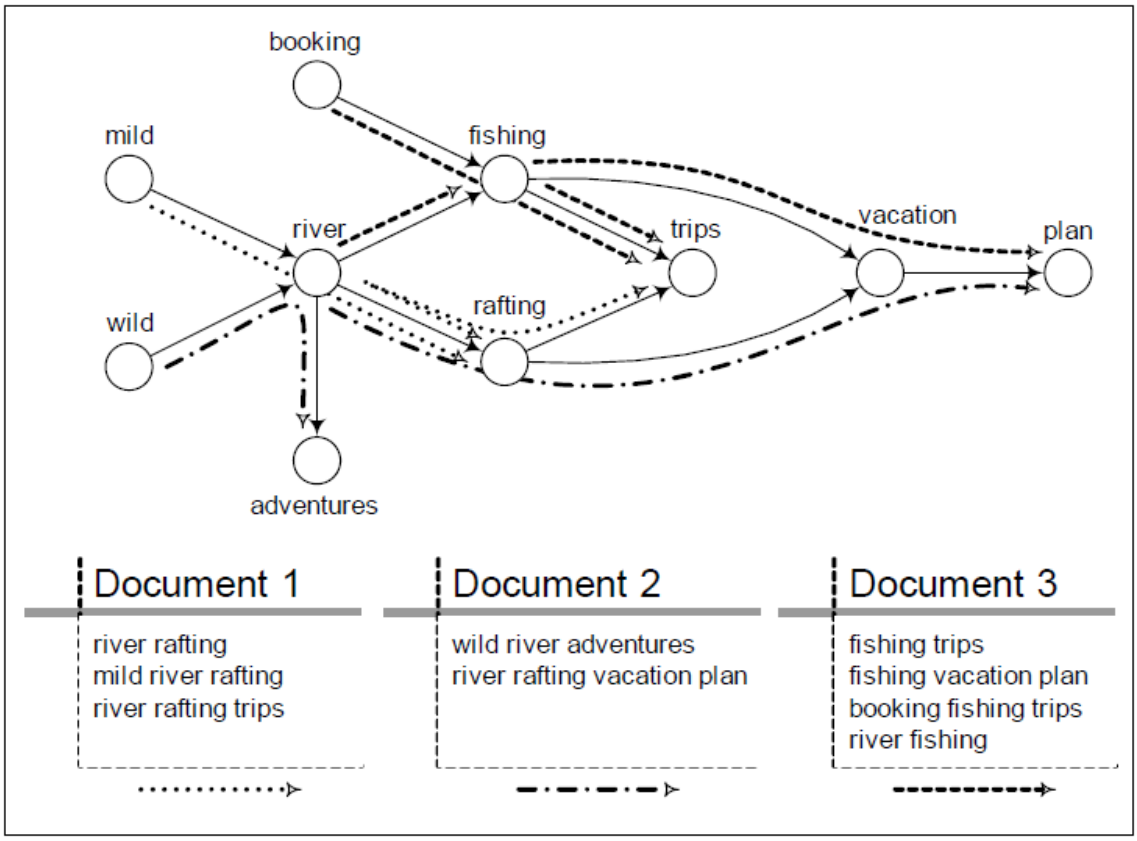
گراف نمایه‌سازی اسناد گرافی است جهت‌دار $G=(V,E)$ که:

V: مجموعه رئوس گراف است $\{v_1, v_2, \ldots, v_n\}$ که هر راس v بیانگر یک کلمه‌ی منحصر به فرد از کل مجموعه‌ی اسناد است.

E: مجموعه یال‌های گراف است $\{e_1, e_2, \ldots, e_m\}$ که هر یال e یک دوتایی مرتب از رئوس به صورت (v_i, v_j) است. یال (v_i, v_j) یالی از راس v_i به راس v_j است که در مجاورت هم قرار دارند. یالی از مبدا v_i به مقصد v_j وجود دارد اگر و تنها اگر در سندی کلمه‌ی v_j بلافاصله بعد از کلمه‌ی v_i رخ داده باشد.

[1] Document index graph

تعریف بالا مشخص می‌کند که تعداد رئوس گراف برابر تعداد کلمات منحصر به فرد کل مجموعه‌ی اسناد است. شکل(۳-۲) نمایی از یک گراف نمایه‌سازی اسناد به همراه اسناد تشکیل دهنده‌ی آن را نشان می‌دهد.



شکل(۳-۲): مثالی از گراف نمایه‌سازی اسناد[۱]

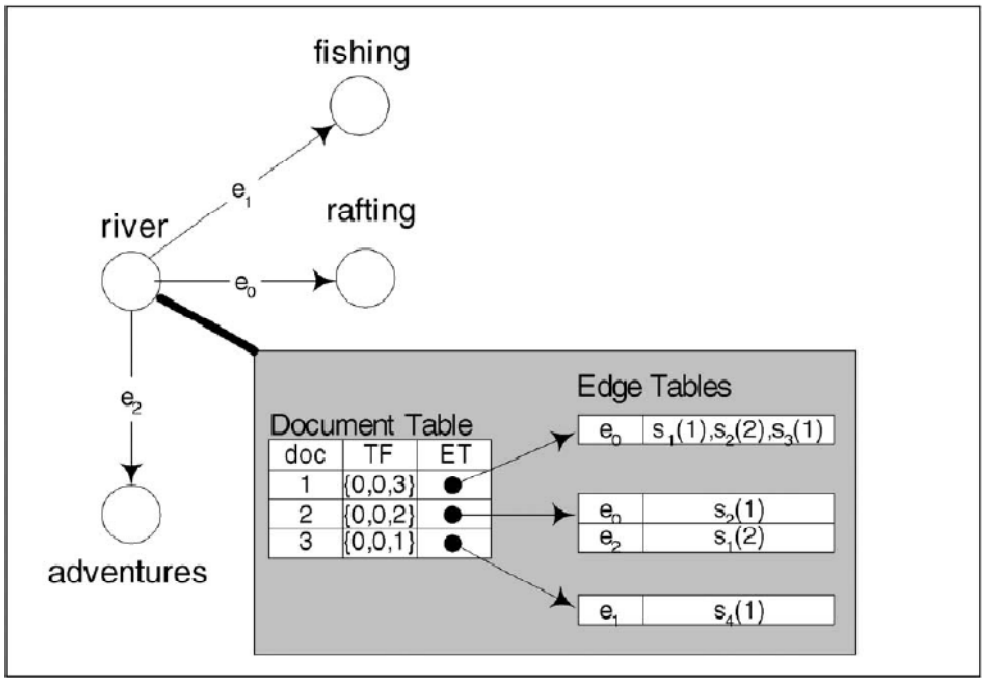
یک جمله در گراف با اتصال نودهای متوالی که نمایانگر کلمات در جمله هستند نمایش داده می‌شود و مسیری را در گراف ایجاد می‌کند که نماینده‌ی آن جمله است. اطلاعات مسیر در نودهای آن ذخیره می‌شود تا هر جمله به‌طور منحصر به‌فرد قابل شناسایی باشد. نودها علاوه بر اطلاعات مسیر شامل اطلاعاتی در مورد اسنادی که در آن‌ها رخ داده‌اند نیز هستند.

ساختاری که در هر نود برای ثبت اطلاعات مورد استفاده قرار می‌گیرد جدول اسناد است. هر مدخل در این جدول مرتبط با یک سند بوده و فرکانس کلمه مربوط به آن نود، در آن سند را ثبت می‌کند.

با توجه به این‌که کلمه ممکن است چندین بار و با سطوح اهمیت مختلف در سند رخ دهد، فرکانس کلمات ثبت شده به سطوح اهمیت مختلف شکسته شده و فرکانس کلمه در هر سطح، جداگانه ثبت می‌شود. این ساختار به محاسبه‌ی دقیق‌تر شباهت براساس سطوح اهمیت کمک می‌کند. به ازای هر مدخل در جدول سند، لیستی از یال‌های خروجی ثبت می‌شود که مشخص میکند هر جمله از طریق چه یالی امتداد می‌یابد، و به این طریق ساختار جملات در گراف حفظ می‌شود.

همان‌طور که در شکل(۳-۳) مشاهده می‌شود ساختار جدول سند که در هر نود ذخیره می‌شود شامل موارد زیر است:

- شناسه‌ی سند
- فرکانس کلمه (به ازای سطوح مختلف اهمیت)
- جدول یال



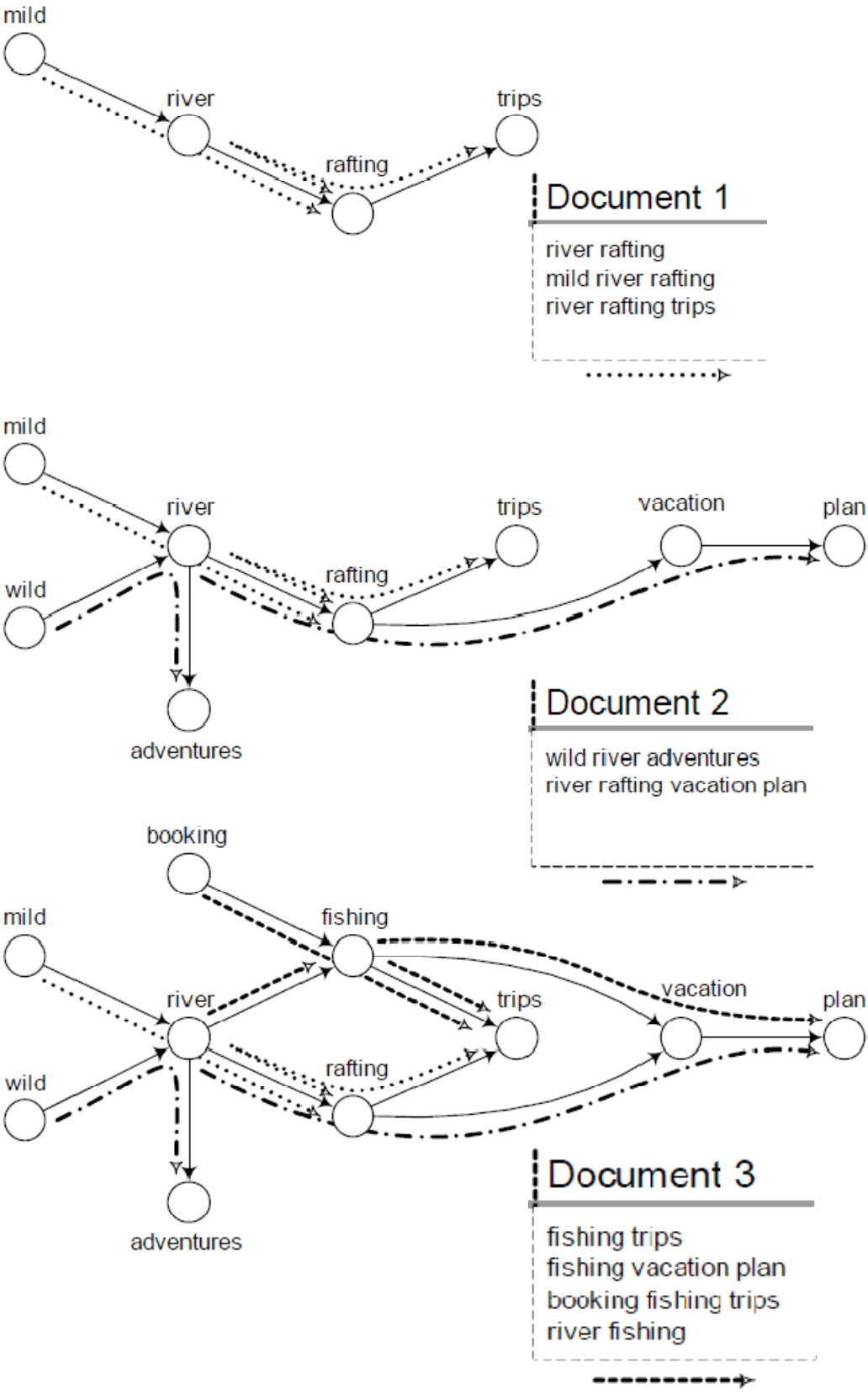
شکل(۳-۳): جزئیات ساختار گراف نمایه‌سازی اسناد[۱]

از آنجا که یک جمله یا عبارت ممکن است چندین بار در یک سند رخ دهد، همان‌طور که در شکل(۳-۳) مشاهده می‌شود جدول یال اطلاعات کلیه این عبارات را به فرم $s_i(j)$ ثبت می‌کند که بیان کننده‌ی jامین کلمه از iامین جمله‌ی سند مربوطه است.

۳-۳-۲- ساخت گراف نمایه‌سازی اسناد

گراف نمایه‌سازی اسناد به‌صورت افزایشی و با پردازش یک سند در هر زمان ساخته می‌شود. هر سند جدید، پس از ورود به روش ترتیبی پیمایش شده و گراف با اطلاعات جملات جدید بروز می‌شود، کلمات جدید به گراف اضافه شده و اتصال میان نودها جهت بازتاب ساختار جملات برقرار می‌شود. درصورتی که کلمه‌ی جدیدی به گراف افزوده نشود یا تعداد کلمات افزوده شده اندک باشد، فرایند ساخت گراف نیازمند حافظه‌ی زیادی نیست که این موجب ثبات بیشتر گراف شده و تنها عمل مورد نیاز ایجاد روابط جدید میان نودها و بروز کردن اطلاعات جداول جهت نمایش جملات جدید در گراف است.

نکته‌ی قابل توجهی که موجب کارایی این مدل می‌شود این است که با ورود هر سند تنها کلماتی به گراف اضافه می‌شوند که در سند جدید رخ داده‌اند ولی نودی معادل آن‌ها در گراف موجود نیست، این امر موجب می‌شود گراف عاری از هر گونه افزونگی باشد، مسئله‌ای که مدل درخت پسوندی از آن رنج می‌برد. برای مثال شکل(۳-۴) فرایند افزایشی ساخت گراف شکل(۳-۲) را نشان می‌دهد. همان‌طور که در این شکل مشاهده می‌شود با اضافه شدن هر سند به صورت تدریجی نودها اضافه شده و یال-های جدید بین نودها ایجاد شده و گراف به‌طور افزایشی به‌روز می‌شود.



شکل(۳-۴) : نمایی از فرایند ساخت افزایشی گراف نمایه‌سازی اسناد[۱]

همزمان با ساخت گراف نمایه‌سازی اسناد، فرایند تشخیص عبارات مشترک میان اسناد نیز به سادگی امکان‌پذیر است. الگوریتم شکل(۳-۵) فرایند ساخت افزایشی گراف نمایه‌سازی اسناد به همراه تشخیص عبارات مشترک میان اسناد را تشریح می‌کند.

Algorithm 3.1 DIG incremental construction and phrase matching

Require: G_{i-1} : cumulative graph up to document $\boldsymbol{d}_{i-1}$
or G_0 if no documents were processed previously

1: $d_i \leftarrow$ Next document
2: $M \leftarrow$ Empty list {M is a list of matching phrases from previous documents}
3: **for** each sentence s_{ij} in d_i **do**
4: $v_1 \leftarrow t_{ij1}$ {first word in s_{ij}}
5: **if** v_1 is not in G_{i-1} **then**
6: Add v_1 to G_{i-1}
7: **end if**
8: **for** each term $t_{ijk} \in s_{ij}$, $k = 2, \ldots, l_{ij}$ **do**
9: $v_k \leftarrow t_{ijk}$; $v_{k-1} \leftarrow t_{ij(k-1)}$; $e_k = (v_{k-1}, v_k)$
10: **if** v_k is not in G_{i-1} **then**
11: Add v_k to G_{i-1}
12: **end if**
13: **if** e_k is an edge in G_{i-1} **then**
14: Retrieve a list of document entries from v_{k-1} document table that have a sentence on the edge e_k
15: Extend previous matching phrases in M for phrases that continue along edge e_k
16: Add new matching phrases to M
17: **else**
18: Add edge e_k to G_{i-1}
19: **end if**
20: Update sentence path in nodes v_{k-1} and v_k
21: **end for**
22: **end for**
23: $G_i \leftarrow G_{i-1}$
24: Output matching phrases list M

شکل(۳-۵) : فرایند ساخت افزایشی گراف نمایه‌سازی اسناد به همراه تشخیص عبارات مشترک میان اسناد[۲]

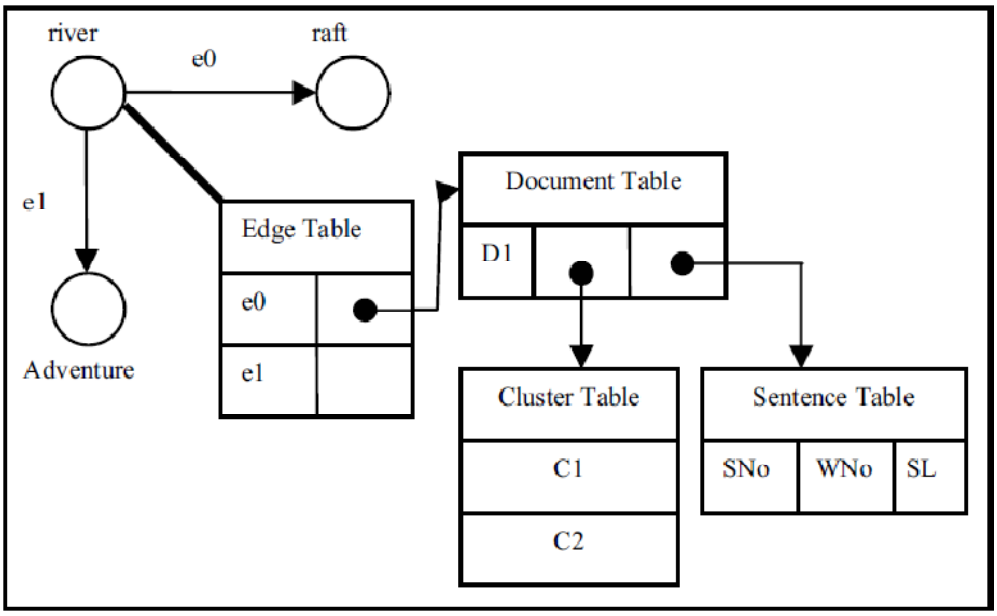
لازم به ذکر است که هر سند قبل از ورود به ساختار گراف تحت یک‌سری عملیات پیش‌پردازش قرار می‌گیرد که شامل حذف stop word ها، اعداد و نمادها، ریشه‌یابی و تشخیص مرز جملات است.

الگوریتم با ورود سند جدید جهت پردازش آغاز می‌شود، لیست M برای ثبت عبارات مشترک میان سند ورودی و اسناد قبلی ایجاد می‌شود که به ازای هر سند موجود در گراف که عبارتی مشترک با سند جاری دارد، یک مدخل در لیست درنظر گرفته می‌شود. به ازای هر جمله، کلمات را به ترتیب پردازش کرده، کلمات جدید را به عنوان نودهای جدید به گراف اضافه کرده و با ایجاد یال‌های جدید مسیر جدیدی را در گراف ایجاد می‌کنیم که نماینده‌ی جمله‌ی درحال پردازش است. همزمان با به‌روز شدن گراف، عبارات مشترک جدید نیز به لیست اضافه شده یا عبارات مشترک موجود در لیست توسعه می‌یابند. جهت تشخیص عبارت مشترک در صورت وجود یال e_k، به جدول اسناد نود v_{k-1} مراجعه می‌کنیم تا اسناد حاوی جملاتی که از طریق e_k امتداد می‌یابند را پیدا کنیم. این اسناد حداقل در دو کلمه (معادل یال e_k) با جمله‌ی درحال پردازش مشترک هستند، پس از آن لیست M به ازای هر عبارت مشترک قبلی (از دور قبلی حلقه‌ی تکرار) برای توسعه‌ی عبارت دو کلمه‌ای بر روی e_k چک می‌شود. به این ترتیب می‌توان عبارات مشترک قبلی را توسعه داد و درنتیجه عبارات مشترک با هر طولی را بین اسناد پیدا کرد.

در صورت عدم وجود یال e_k، یال جدید به گراف اضافه شده و جداول جهت نمایش مسیر جمله‌ی جدید به‌روز می‌شوند. پس از پردازش تمام سند، لیست M شامل اطلاعات کلیه‌ی عبارات مشترک سند جدید و اسناد قبلی است که در محاسبه‌ی شباهت میان اسناد مورد استفاده قرار می‌گیرد.

در این مدل جهت کاهش زمان مراجعه به جداول از ساختار hash table برای پیاده‌سازی جداول اسناد و یال‌ها استفاده شده است که تا حد قابل توجهی در کاهش زمان مراجعه موثر بوده است ولی با این وجود استفاده از این متد در پیاده‌سازی نیازمند حافظه بیشتری است، بر همین اساس B.F.Momin در سال ۲۰۰۶ [۳۱] با اصلاح ساختار اطلاعاتی نود در گراف دسترسی ساده‌تری را به

جداول جهت تشخیص عبارات مشترک فراهم کرده است. شکل(۳-۶) این ساختار اصلاح شده را نشان می‌دهد.



شکل(۳-۶) : ساختار اصلاح شده‌ی گراف نمایه‌سازی اسناد[۳۱]

با استفاده از این ساختار اصلاح شده دسترسی به جداول و اطلاعات گراف بدون نیاز به hash table و در زمانی کمتر به سادگی امکان‌پذیر است.

۳-۴- معیار شباهت مبتنی بر عبارت

همان‌طور که در فصل‌های پیش اشاره شد، عبارات دربردارنده‌ی مفاهیم محلی هستند که به‌کارگیری آن‌ها در تعیین دقیق شباهت میان اسناد بسیار موثر می‌باشد. در این مدل جهت بهره‌گیری از این ویژگی معیار شباهتی معرفی شده که به‌جای کلمات مجزا، براساس عبارات طراحی شده است. این معیار از اطلاعات لیست M که در جریان به‌روز شدن گراف، استخراج شده برای تعیین شباهت میان اسناد استفاده می‌کند.

معیار شباهت معرفی شده تابعی از ۴ مولفه زیر است:

- تعداد عبارات منطبق بین اسناد P
- طول عبارات منطبق ($l_i : i = 1,2, \dots, P$)
- فرکانس عبارات منطبق در دو سند (f_{1i} and $f_{2i} : i = 1,2, \dots, P$)
- سطح اهمیت عبارات منطبق در هر دو سند (w_{1i} and $w_{2i} : i = 1,2, \dots, P$)

معیار شباهت مبتنی بر عبارت بین دو سند d_1 و d_2 با استفاده فرمول(۳-۱) محاسبه می‌شود:

$$sim_p(d_1, d_2) = \frac{\sqrt{\sum_{i=1}^{P}[g(l_i) \cdot (f_{1i}w_{1i} + f_{2i}w_{2i})]^2}}{\sum_j |s_{1j}| \cdot w_{1j} + \sum_k |s_{2k}| \cdot w_{2k}} \qquad (۱-۳)$$

در این عبارت $g(l_i)$ تابعی است که بر اساس طول عبارت منطبق امتیازی در نظر می‌گیرد، هرچه طول عبارت به جمله‌ی اصلی نزدیکتر باشد امتیاز اختصاص داده شده بیشتر است.$|s_{1j}|$ و $|s_{2k}|$ به ترتیب طول جملات اصلی در اسناد d_1 و d_2 هستند. این تابع به صورت فرمول(۳-۲) تعریف می‌شود:

$$g(l_i) = \left({l_i}/{|s_i|} \right)^{\gamma} \qquad (۲-۳)$$

که $|s_i|$ طول جمله‌ی اصلی و ضریب γ مقداری برابر یا بزرگتر از یک دارد. اگر γ برابر ۱ باشد، درصورتی که دو نیمه‌ی جمله به‌طور مستقل با عبارات دیگر منطبق باشند، رفتاری مشابه انطباق جمله‌ی کامل بروز می‌دهد، بنابراین با افزایش این مقدار از بروز این اتفاق جلوگیری می‌کنیم و هرچه طول عبارت به جمله‌ی اصلی نزدیکتر باشد امتیاز بیشتری به آن اختصاص می‌دهیم. در آزمایشات انجام شده با درنظر گرفتن γ برابر ۱/۲ بهترین نتایج حاصل شده است.

با توجه به فرمول(۳-۱) شباهت دو سند براساس عبارات مشترک میان آن‌ها سنجیده می‌شود، براین اساس هرچه طول عبارات به جملات اصلی نزدیکتر و فرکانس و سطوح اهمیت آن‌ها در دو سند بیشتر باشد شباهت دو سند بیشتر خواهد بود. عبارت مخرج نیز جهت نرمال‌سازی کمیت آمده تا حاصل با مقادیر مشابه اسناد دیگر قابل مقایسه باشد.

۳-۴-۱- ترکیب معیار شباهت مبتنی بر عبارت و مبتنی بر کلمه

اگر شباهت اسناد تنها بر اساس عبارات منطبق سنجیده شود و کلمات نقشی در محاسبه‌ی شباهت نداشته باشند، اسناد مشابهی که عبارات منطبق کافی جهت سنجش تشابه ندارند ممکن است نامشابه ارزیابی شوند. بدون شک عبارات در استخراج مفاهیم محلی، از متون، بسیار حائز اهمیت هستند اما گاهی ارزیابی شباهت صرفا براساس عبارات کافی نیست. جهت حل این مشکل و تولید خوشه‌های دقیق‌تر، در این مدل دو معیار شباهت مبتنی بر کلمه و معیار شباهت مبتنی بر عبارت با هم ترکیب شده‌اند. در این مدل از معیار کسینوس فرمول(۲-۱۴) به عنوان معیار شباهت مبتنی بر کلمه استفاده شده است که وزن کلمات به صورت فرکانس کلمه- معکوس فرکانس سند[1] مطابق فرمول(۲-۱۰) درنظر گرفته می‌شود. ترکیب این دو معیار شباهت، یک میانگین وزنی از دو فرمول (۳-۱) و (۲-۱۴) را ارائه می‌دهد که به صورت فرمول(۳-۳) نمایش داده می‌شود. علت تفکیک کلمات از عبارات در محاسبه‌ی شباهت، به‌جای درنظر گرفتن کلمات به عنوان عبارات تک کلمه‌ای، ارزیابی فاکتور ترکیب میان این دو معیار و مشاهده‌ی تاثیر عبارات درمقابل کلمات در محاسبه شباهت میان اسناد است.

$$sim(d_1, d_2) = \alpha \,.\, \mathrm{sim_p}(\mathrm{d_1}, \mathrm{d_2}) + \ (1 - \alpha) \,.\, \mathrm{sim_t}(\mathrm{d_1}, \mathrm{d_2}) \qquad (۳-۳)$$

[1] TF-IDF (term frequency– inverse document frequency)

در این فرمول α عددی در بازه‌ی $[0,1]$ است که ضریب ترکیب نامیده می‌شود. بر اساس آزمایشات مقدار این ضریب بین ۰/۶ و ۰/۸ بالاترین میزان بهبود را در کیفیت خوشه‌بندی به همراه دارد.

با ترکیب این دو معیار شباهت، معیاری قوی و مقاوم در برابر نویز ایجاد می‌شود که تاثیر بسزایی در بهبود کیفیت خوشه‌بندی خواهد داشت. لازم به ذکر است که شباهت محاسبه شده بین سند جدید و اسناد قبلی موجود در گراف برای ساخت ماتریس شباهت افزایشی که در فرایند خوشه‌بندی به‌کار می‌رود مورد استفاده قرار می‌گیرد.

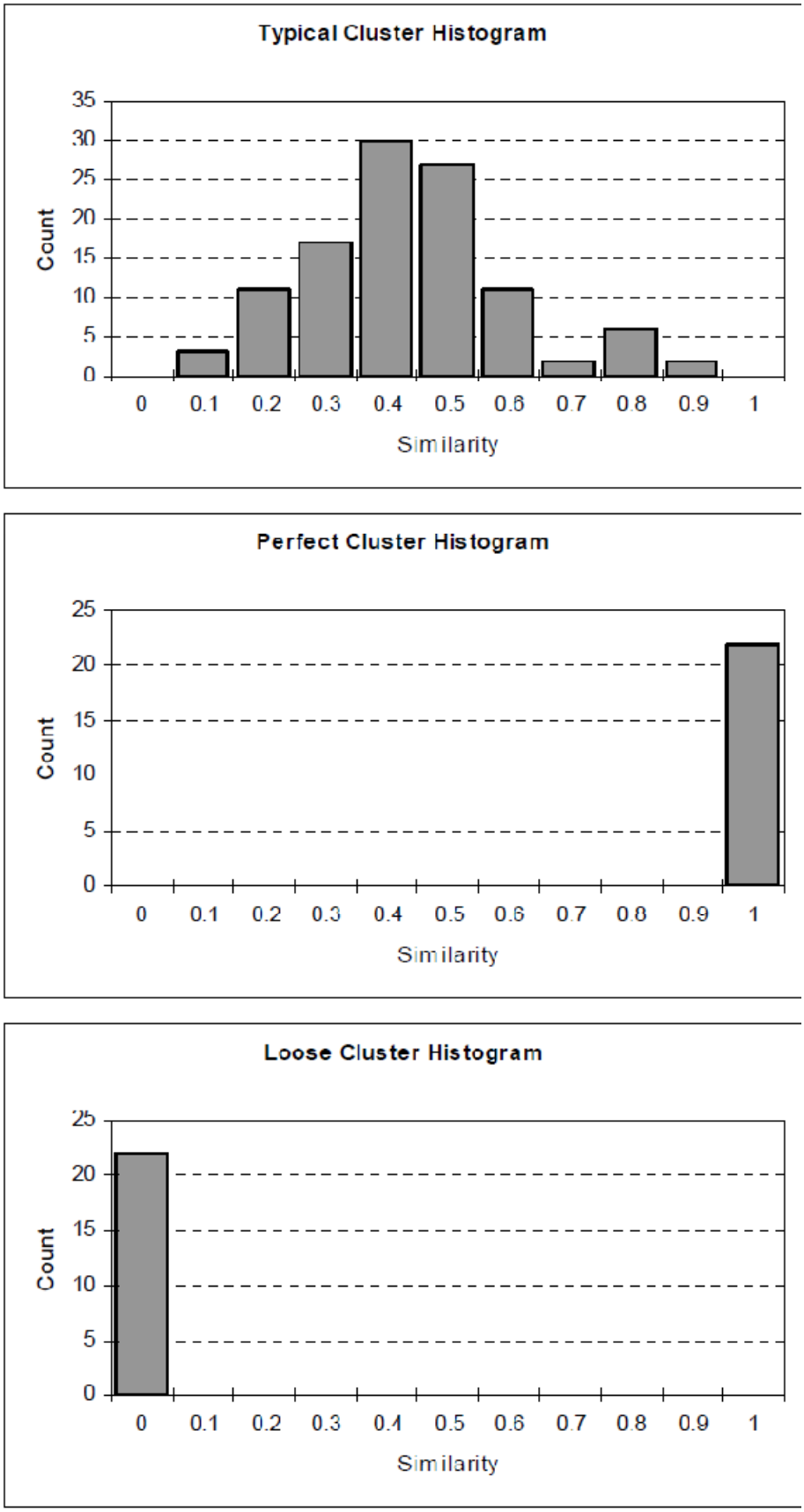
۳-۵- خوشه‌بندی افزایشی برمبنای هیستوگرام مبتنی بر شباهت

رویکرد خوشه‌بندی در این مدل یک متد افزایشی پویا و درعین حال همپوشان را برای ساخت خوشه-ها ارائه می‌دهد. مفهوم کلیدی در متد خوشه‌بندی مبتنی بر هیستوگرام شباهت، حفظ هر خوشه در درجه‌ی بالایی از انسجام است. با دستیابی به انسجام بالا در خوشه‌ها، نیاز به افزایش شباهت درون خوشه‌ای برآورده می‌شود.

۳-۵-۱- هیستوگرام شباهت[1]

در این مدل، انسجام[2] یک خوشه با مفهومی جدید به‌نام هیستوگرام شباهت معرفی می‌شود. هیستوگرام شباهت یک نمایش آماری مختصر از توزیع شباهت دوبه‌دوی اسناد در خوشه است. هیستوگرام شباهت ستون‌هایی[3] را منطبق بر بازه‌های ثابت مقادیر تشابه درنظر می‌گیرد که هر ستون شامل تعداد شباهت‌های جفت اسناد در بازه‌ی شباهت مربوطه است به‌طوری که با محاسبه‌ی یک مقدار تشابه جدید میان دو سند ارتفاع ستون بازه‌ی مربوط به مقدار آن در هیستوگرام افزایش می-یابد.

[1] Similarity histogram
[2] Coherency
[3] Bins



شکل(۳-۷) : نمونه‌های هیستوگرام شباهت[۱]

شکل(۳-۷) سه نمونه از هیستوگرام شباهت را نشان می‌دهد که شامل: یک هیستوگرام شباهت معمولی که تقریبا دارای یک توزیع نرمال است، یک هیستوگرام شباهت مربوط به یک خوشه‌ی کامل که همه‌ی شباهت‌ها در خوشه، حداکثر هستند و یک هیستوگرام شباهت خوشه ضعیف که همه‌ی شباهت‌ها در خوشه، حداقل هستند.

۳-۵-۲- ایجاد افزایشی خوشه‌های منسجم

در این متد هدف حفظ هر خوشه در منسجم‌ترین حالت ممکن است، که به ماکزیمم کردن شباهت درون خوشه‌ها تفسیر می‌شود. در حالت ایده‌آل هیستوگرام شباهت هر خوشه باید مشابه هیستوگرام شباهت خوشه‌ی کامل باشد، بنابراین ما باید توزیع شباهت‌ها در خوشه را دربازه‌های بالای شباهت و در حد بالایی نگه داریم.

برای دستیابی به این هدف می‌توان تاثیر اضافه شدن یک سند جدید به خوشه‌ای خاص را مورد سنجش قرار داد. درصورتی که ورود سند جدید موجب تنزل زیاد توزیع شباهت‌ها در خوشه شود، از اضافه شدن سند به خوشه جلوگیری می‌شود و در غیر این‌صورت سند به خوشه اضافه می‌شود. حتی می‌توان محدودیت بیشتری را جهت افزودن یک سند منظور کرد تا فقط درصورت بهبود توزیع شباهت‌ها در یک خوشه، سند به خوشه افزوده شود. اگرچه این شرایط سخت‌گیرانه می‌تواند سبب بروز مشکل نیز شود، درصورتی که بخواهیم سندی را به خوشه‌ای اضافه کنیم که در حال حاضر توزیع شباهت بسیار خوبی دارد و اضافه کردن آن سند سبب تنزل اندک توزیع شود. سناریوی افزودن سند به یک خوشه‌ی کامل را در نظر بگیرید؛ ممکن است سند با اغلب اسناد خوشه شباهت بالایی داشته باشد و با بقیه‌ی اسناد نیز شباهت کمی داشته باشد، در این شرایط اضافه شدن سند سبب تنزل اندک توزیع شباهت می‌شود (به‌خاطر کامل بون خوشه).

برای جلوگیری از رد سند صرفاً به‌خاطر تنزل اندک توزیع شباهت‌ها نحوه‌ی ارزیابی تاثیر افزودن سند جدید به یک خوشه را به‌طور دقیق‌تر مورد بررسی قرار می‌دهیم. همان‌طور که اشاره شد یک خوشه‌ی منسجم شامل تعداد بالای شباهت‌ها در فواصل مقادیر بالای شباهت و تعداد اندک شباهت‌ها در فواصل مقادیر پایین شباهت است. در این متد انسجام خوشه‌ها با محاسبه‌ی نسبت تعداد شباهت‌های بیشتر از یک مقدار شباهت آستانه S_T به تعداد کل شباهت‌ها بدست می‌آید، این نسبت درواقع درصد شباهت‌های بیشتر از یک مقدار آستانه است. هرچقدر که این نسبت بیشتر باشد خوشه منسجم‌تر است.

اگر n تعداد اسناد در یک خوشه باشد، تعداد شباهت‌های دوبه‌دوی اسناد در خوشه برابر:

$$m = \frac{n(n-1)}{2} \qquad (۴-۳)$$

اگر $S=\{s_i : i=1,2,\ldots, m\}$ مجموعه‌ی شباهت‌ها در خوشه باشد هیستوگرام شباهت‌ها در خوشه به‌صورت زیر نمایش داده می‌شود:

$$H = \{h_i : i = 1, \ldots, B\} \qquad (۵-۳)$$

$$h_i = count(s_k) \qquad s_{li} < s_k < s_{ui} \qquad (۶-۳)$$

که:

- B تعداد ستون‌های هیستوگرام
- h_i : تعداد شباهت‌ها در ستون i
- s_{li} : کران پایینی شباهت ستون i
- s_{ui} : کران بالایی شباهت ستون i

نسبت هیستوگرام یک خوشه یا بنا به تعریف معیار انسجام خوشه به‌صورت زیر محاسبه می‌شود:

$$HR(C) = \frac{\sum_{i=T}^{B} h_i}{\sum_{j=1}^{B} h_j} \qquad (۷-۳)$$

$$T = \lfloor S_T \,.\, B \rfloor \qquad (۸-۳)$$

که:

- HR : نسبت هیستوگرام
- C : خوشه‌ی تحت بررسی
- S_T : مقدار آستانه شباهت
- T : شماره‌ی ستون مطابق با آستانه‌ی شباهت

به‌طور کلی هدف این متد این است که نسبت هیستوگرام هر خوشه را در حد بالایی حفظ کند. با این وجود، با اضافه شدن اسنادی که موجب تنزل اندک نسبت هیستوگرام می‌شوند ممکن است یک تاثیر زنجیره‌ای از کاهش نسبت هیستوگرام با اضافه شدن اسنادی که به‌جای بهبود تنها موجب کاهش نسبت می‌شوند ایجاد شود که مقدار نسبت هیستوگرام را به صفر نزدیک کند یا به عبارت دیگر اغلب شباهت‌ها کمتر از حد آستانه قرار گیرند. برای جلوگیری از این اثر زنجیره‌ای، باید یک نسبت هیستوگرام حداقل HR_{min} تعیین شود که خوشه‌ها ملزم به حفظ آن باشند. همچنین از ورود اسنادی که نسبت هیستوگرام را به‌طور قابل توجهی کاهش می‌دهند (حتی اگر نسبت هنوز بیشتر از HR_{min} است) جلوگیری شود. این امر به سبب جلوگیری از افت شدید کیفیت خوشه تنها با افزودن یک سند نامناسب انجام می‌شود. یک سند درصورتی به خوشه افزوده می‌شود که نسبت هیستوگرام خوشه را بیش از ε کاهش ندهد.

شکل(۳-۸) الگوریتم خوشه‌بندی افزایشی مبتنی بر چارچوب ارائه شده را نمایش می‌دهد. این الگوریتم به صورت افزایشی با دریافت سند جدید، به ازای هر خوشه نسبت هیستوگرام را قبل و بعد از اضافه شدن سند محاسبه می‌کند (خط ۳-۵)، نسبت‌های قدیم و جدید هیستوگرام مقایسه شده و اگر نسبت جدید بیشتر یا برابر نسبت قدیم باشد سند اضافه می‌شود. اگر نسبت جدید از مقدار قدیم کمتر باشد و تفاوت کمتر از ε بوده و نسبت هنوز بیشتر از HR_{min} باشد نیز سند اضافه می‌شود (خط ۶-۸). اگر پس از بررسی تمام خوشه‌ها، سند به هیچکدام از خوشه‌ها منتصب نشد، یک خوشه‌ی جدید ایجاد شده و سند در خوشه‌ی جدید قرار می‌گیرد (خط ۱۰-۱۳).

Algorithm 3.2 Similarity Histogram-based Incremental Document Clustering

1: $D \leftarrow$ New Document documents
2: **for** each cluster C **do**
3: $HR_{old} = HR(C)$
4: Simulate adding D to C
5: $HR_{new} = HR(C)$
6: **if** $(HR_{new} \geq HR_{old})$ OR **(**$(HR_{new} > HR_{min})$ AND $(HR_{old} - HR_{new} < \varepsilon)$**)** **then**
7: Add D to C
8: **end if**
9: **end for**
10: **if** D was not added to any cluster **then**
11: Create a new cluster C
12: ADD D to C
13: **end if**

شکل(۳-۸) : الگوریتم خوشه‌بندی افزایشی مبتنی بر هیستوگرام شباهت[۱]

۳-۶- ارائه‌ی نتایج آزمایشات

برای تست کارایی سیستم خوشه‌بندی ارائه شده مجموعه آزمایشاتی صورت گرفته و مدل پیشنهادی با متدهای خوشه‌بندی HAC[1]، Single-pass و KNN مورد مقایسه قرار گرفته است. آزمایشات بر روی دو مجموعه پایگاه داده‌ی DS1 و DS2 انجام شده که اطلاعات آن‌ها در جدول(۳-۲) آمده است.

جدول(۳-۲): توصیف پایگاه داده‌های مورد آزمایش[۱]

Data Set	Description	Categories	Documents
DS1	UW web site, Canadian web sites	10	314
DS2	Reuters news articles (from Yahoo! news)	20	2340

در ارزیابی عملکرد متدها از سه معیار F-measure، آنتروپی[2] و شباهت کلی[3] استفاده شده است. معیار F-measure از ترکیب دو مفهوم صحت[4] و فراخوانی[5] تشکیل شده و درواقع میزان همپوشانی کلاستر تشکیل شده با کلاس اصلی را می‌سنجد و به صورت زیر محاسبه می‌شود:

$$P = Precision(i,j) = \frac{N_{ij}}{N_j}$$
$$R = Recall(i,j) = \frac{N_{ij}}{N_i} \quad \rightarrow \quad F(i) = \frac{2PR}{P+R} \qquad (۹-۳)$$

در این فرمول N_{ij} تعداد اعضای کلاس i موجود در کلاستر j ، N_j تعداد اعضای کلاستر j و N_i تعداد اعضای کلاس i است.

معیار آنتروپی میزان یکنواختی کلاستر را می‌سنجد و هرچه کلاستر یکنواخت‌تر باشد آنتروپی کاهش می‌یابد. در کلاستری با یک سند آنتروپی برابر صفر است. معیار شباهت کلی میانگین تمام شباهت-

[1] Hierarchical agglomerative clustering
[2] Entropy
[3] Overall similarity
[4] Precision
[5] Recall

های دوبه‌دوی اسناد در یک کلاستر است که معمولا در غیاب برچسب کلاس برای اسناد، مورد استفاده قرار می‌گیرد.

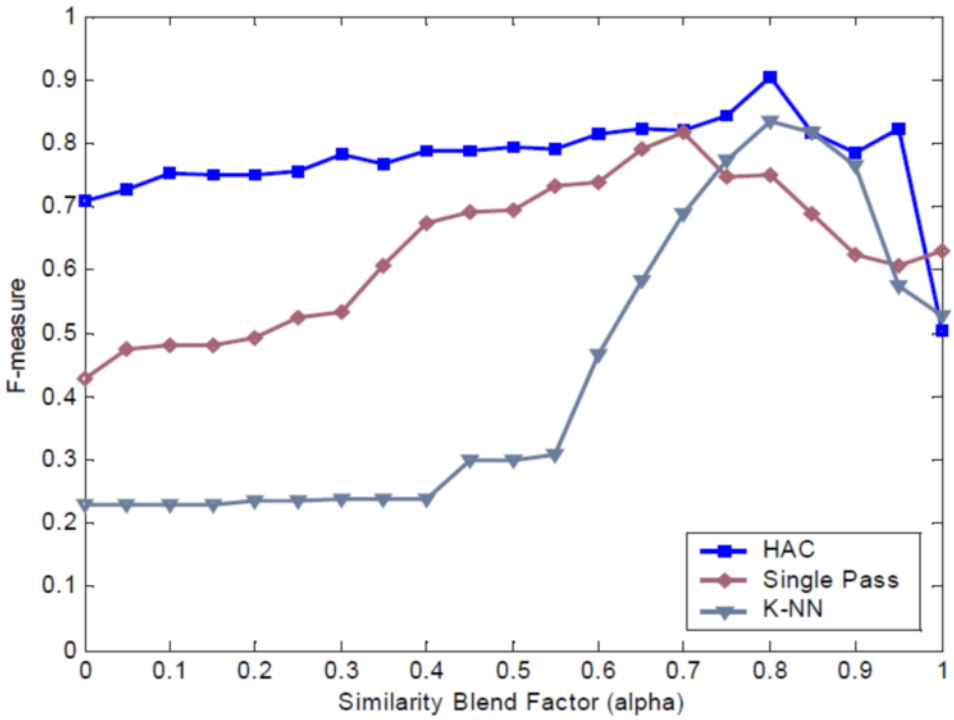
جهت بهبود عملکرد متدها، لازم است دو معیار F-measure و شباهت کلی با افزایش و آنتروپی با کاهش همراه باشد.

برای ارزیابی میزان تاثیر معیار شباهت ترکیبی ارائه شده، ماتریس شباهت اسناد را با دو معیار شباهت ترکیبی و مبتنی بر کلمه تشکیل داده و عملیات خوشه‌بندی را با متدهای مختلف انجام می-دهیم. جدول(۳-۳) میزان تاثیر استفاده از معیار شباهت ترکیبی ارائه شده در این مدل را در مقایسه با معیار شباهت مبتنی بر کلمه در کیفیت خوشه‌بندی اسناد با متدهای مورد مقایسه نشان می‌دهد. نتایج بدست آمده حاکی از بهبود قابل توجه کیفیت خوشه‌بندی با معیار معرفی شده در متدهای مورد آزمایش است.

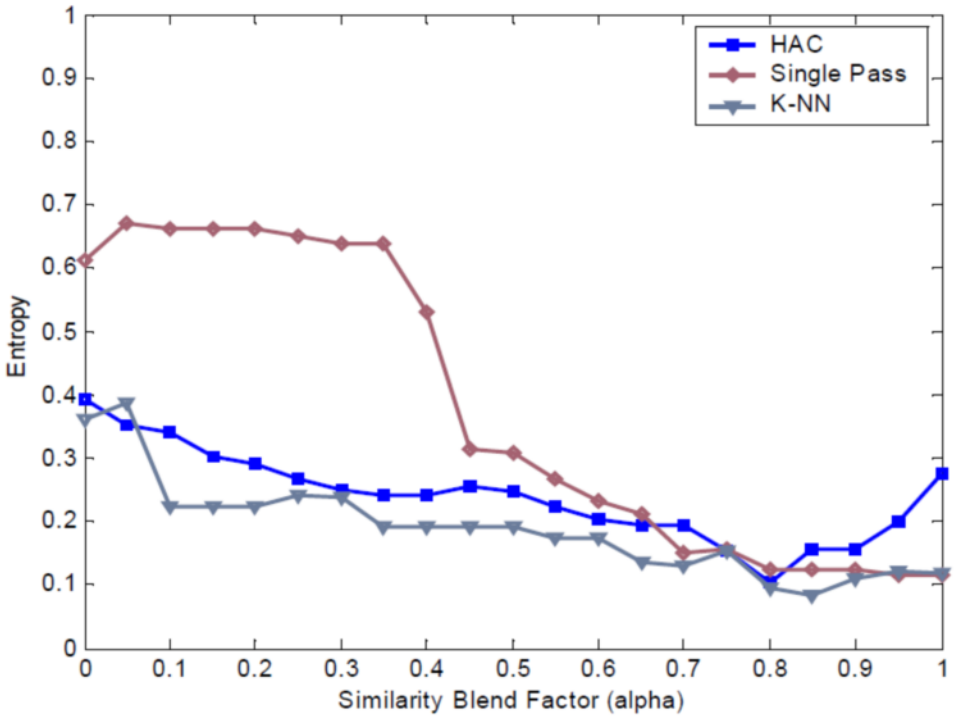
جدول(۳-۳): تاثیر استفاده از معیار شباهت ترکیبی در مقایسه با معیار شباهت مبتنی بر کلمه در کیفیت خوشه‌بندی اسناد[۱]

	Single-Term Similarity		Combined Similarity	
	F-measure	Entropy	F-measure	Entropy
HAC [a]	0.709	0.351	0.904	0.103
Single Pass [b]	0.427	0.613	0.817	0.151
K-NN [c]	0.228	0.173	0.834	0.082

شکل(۳-۹) نیز تاثیر مقادیر مختلف فاکتور ترکیب معیار شباهت ارائه شده (α) را در متدهای مورد مقایسه براساس دو معیار F-measure و آنتروپی نشان می‌دهد. همان‌طور که قبلا نیز عنوان شد در هر دو معیار، مقادیر α در بازه‌ی (۰/۸ و ۰/۶) بهینه‌ترین نتایج را ارائه می‌دهد.



(a) Effect of Phrase Similarity on F-measure



(b) Effect of Phrase Similarity on Entropy

شکل(۳-۹): تاثیرمقادیر مختلف فاکتور ترکیب در متدهای مورد مقایسه[۱]

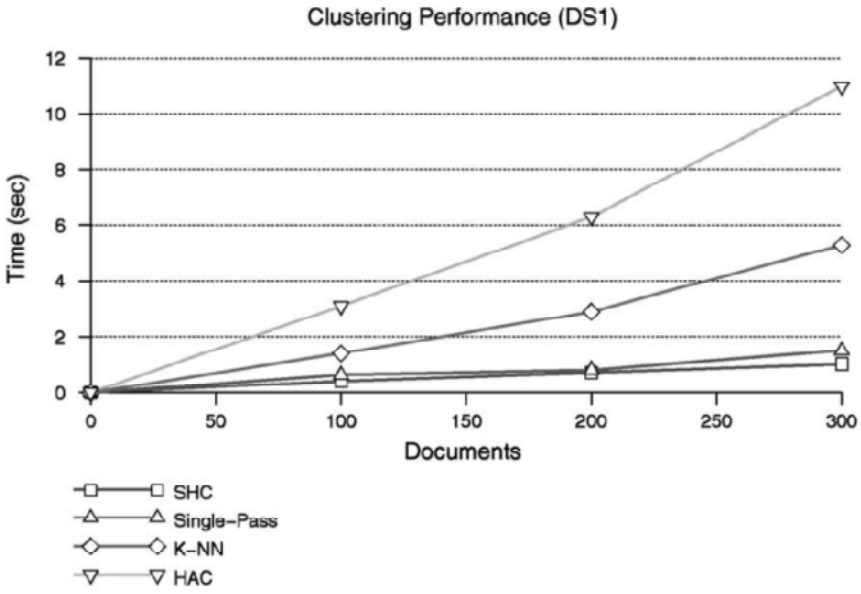
برای ارزیابی عملکرد متد خوشه‌بندی ارائه شده جدول(۳-۴) نتایج آزمایشات صورت گرفته بر روی متد ارائه شده در مقایسه با متدهای HAC، Single-pass و KNN را نمایش می‌دهد. همان‌طور که مشاهده می‌شود در پایگاه داده اول بهبود کیفیت بسیار قابل توجه است و بهبود بالای ۷۰ درصد نسبت به متد KNN، ۲۵ درصدی نسبت به متد HAC و ۵۰ درصدی را نسبت به Single-pass شاهد هستیم و در پایگاه داده دوم هم متد پیشنهادی با بهبود ۱۰ تا ۱۸ درصدی نسبت به متدهای دیگر همراه است.

جدول(۳-۴): مقایسه عملکرد متد خوشه‌بندی ارائه شده با دیگر متدهای مرسوم خوشه‌بندی[۱]

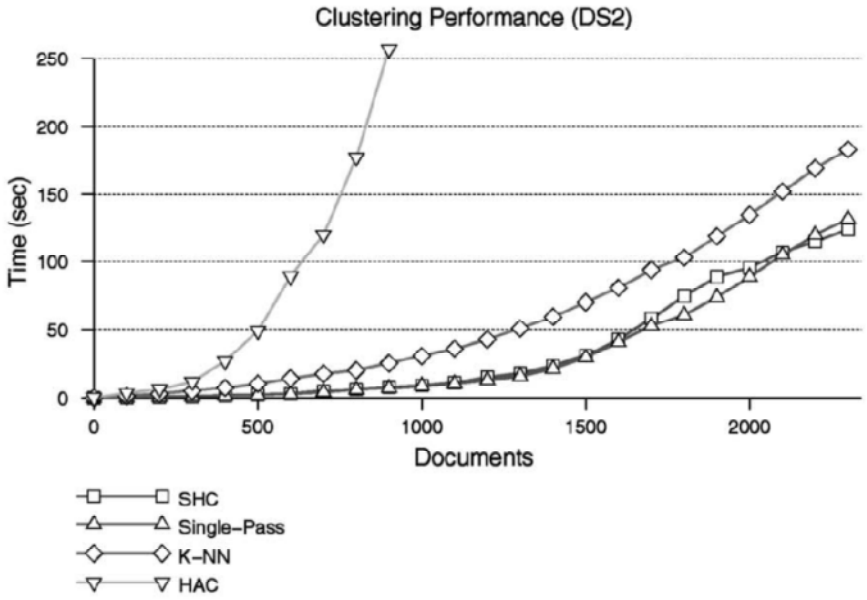
	Data Set 1			Data Set 2		
	F-measure	Entropy	O-S[a]	F-measure	Entropy	O-S
Proposed Method	0.931	0.119	0.504	0.682	0.156	0.497
HAC	0.709	0.351	0.455	0.584	0.281	0.398
Single Pass	0.427	0.613	0.385	0.502	0.250	0.311
K-NN	0.228	0.173	0.367	0.522	0.161	0.452

[a]Overall-Similarity

شکل(۳-۱۰) و (۳-۱۱) نیز عملکرد زمانی متدهای مورد مقایسه را بر روی دو پایگاه داده مورد مطالعه نمایش می‌دهد.



شکل(۳-۱۰): عملکرد خوشه‌بندی از لحاظ زمانی بر روی پایگاه داده DS1 [۲]



شکل(۳-۱۱): عملکرد خوشه‌بندی از لحاظ زمانی بر روی پایگاه داده DS2 [۲]

همان‌طور که در شکل‌ها مشاهده می‌شود از نظر پیچیدگی زمانی نیز متد پیشنهادی قابل مقایسه با KNN و Single-pass و به مراتب بهتر از HAC می‌باشد. علت زمان‌بر بودن متد HAC محاسبه‌ی شباهت‌ها در هر مرحله از ادغام خوشه‌هاست.

۳-۷- جمع‌بندی فصل

در این فصل به معرفی یک سیستم جدید دسته‌بندی اسناد پرداختیم که از ۴ مولفه تشکیل شده است و سعی در بهبود مساله‌ی خوشه‌بندی اسناد در حوزه‌ی وب دارد. این ۴ مولفه عبارتند از:

۱- مولفه‌ی تحلیل‌گر اسناد وب که سطح اهمیت بخش‌های مختلف سند را جهت پردازش‌های بعدی تعیین می‌کند. ۲- مدل جدید گراف نمایه‌سازی اسناد، که اسناد را برحسب عبارات و سطوح اهمیتشان نمایه‌سازی می‌کند ۳- معیار شباهت مبتنی بر عبارت و ۴- متد خوشه‌بندی افزایشی اسناد که براساس حفظ انسجام بالای خوشه‌ها به خوشه‌بندی اسناد می‌پردازد.

این ۴ مولفه در کنار هم سیستمی قدرتمند را جهت خوشه‌بندی اسناد وب فراهم کرده و تقریبا تمام الزامات یک سیستم دسته‌بندی اسناد را که در مقدمه مورد بررسی قرار گرفت تامین می‌کنند. نتایج گزارش شده نیز نشان از بهبود قابل توجه کیفیت خوشه‌بندی توسط این سیستم در مقایسه با دیگر متدهای مرسوم دسته‌بندی اسناد دارد.

از آنجایی که این سیستم قابلیت استفاده در موتور جستجو را جهت دسته‌بندی اسناد بازیابی شده دارد، در ادامه در فصل بعد با نگاهی دیگر از زاویه موتور جستجو به عملکرد این سیستم پرداخته و سعی در بهبود کارایی این سیستم در قالب موتور جستجو داریم.

فصل چهارم

رویکرد پیشنهادی جهت بهبود عملکرد سیستم در بستر موتور جستجو

۴-۱- مقدمه

در فصل قبل به بررسی یک سیستم دسته‌بندی اسناد جدید پرداختیم که تا حد زیادی نواقص موجود در مدل‌های نمایش و متدهای خوشه‌بندی مرسوم را بهبود بخشید. است. معرفی مدل نمایش جدید گراف نمایه‌سازی اسناد به لحاظ بهره‌گیری از عبارات به عنوان ویژگی‌هایی با بالاترین بار اطلاعاتی و ظرفیت مقیاس‌پذیری بالا به دلیل عدم وجود افزونگی در مدل، تبیین معیار شباهت ترکیبی در جهت افزایش قدرت تحمل خطا و معرفی متد خوشه‌بندی مبتنی بر هیستوگرام شباهت با قابلیت همپوشانی و افزایش تدریجی در کنار هم سیستمی را تشکیل می‌دهند که بهبود قابل توجهی را در کیفیت خوشه‌بندی اسناد به همراه دارد.

گنجاندن چنین سیستمی در ساختار موتور جستجو جهت دسته‌بندی اسناد بازیابی شده و ارائه نتایج قابل قبول در قبال عبارت مورد جستجوی کاربر، مسائلی را مطرح می‌کند که در این فصل به طرح یکی از این مسائل پرداخته و با اصلاح مدل پیشنهادی سعی در بهبود آن داریم.

۴-۲- طرح مسئله

در فرایند استفاده از موتور جستجو، با وارد کردن عبارت مورد جستجو توسط کاربر لیستی از اسناد مرتبط که حاوی آن عبارت هستند برای کاربر به نمایش در می‌آید. درصورت به‌کارگیری سیستم معرفی شده، در ساختار موتور جستجو، نتایج به‌صورت لیستی از کلاسترها خواهد بود که شامل اسناد حاوی آن عبارت هستند، با این‌که در مورد دوم اسناد، دسته‌بندی شده هستند و کاربر با مراجعه به دسته‌ی مورد نظر خود سریعتر به نتایج دلخواه دست می‌یابد، با این حال اسناد موجود در هر کلاستر نه به‌ترتیب میزان اهمیت نسبت به عبارت مورد جستجو بلکه اغلب با توجه به میزان بازدید کاربران از اسناد بازیابی شده در لیست ظاهر می‌شوند، به‌عبارت دیگر میزان رخداد عبارت مورد جستجو در

ترتیب اسناد در لیست مدنظر نخواهد بود و ممکن است سندی که بیشترین رخداد عبارت مورد نظر را دارد و احیاناً درخواست کاربر را تامین می‌کند در لیست نتایج در مراتب پایین‌تر قرار بگیرد.

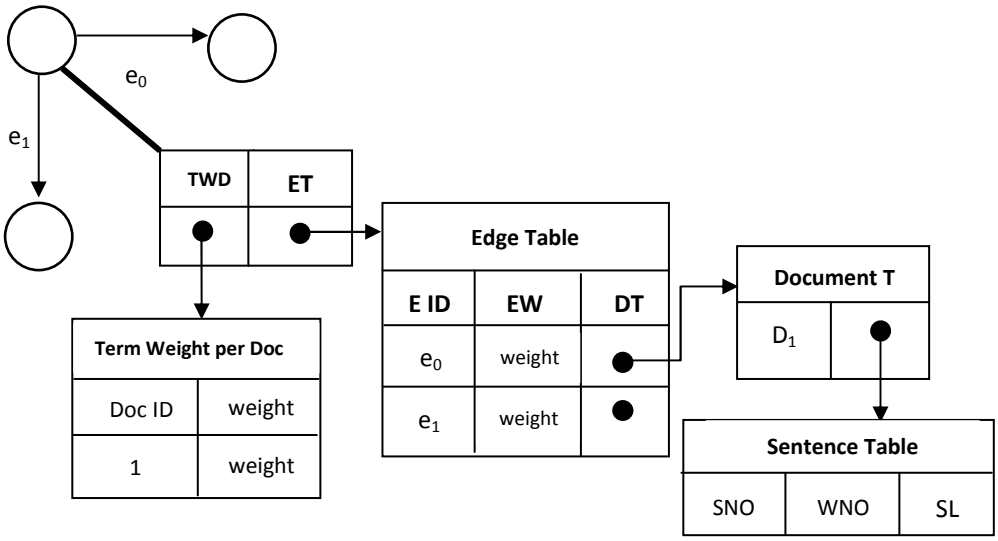
جهت برطرف کردن این مسئله باید چاره‌ای اندیشیده شود که بر مبنای عبارت مورد جستجو اسنادی که بیشترین ارتباط را با عبارت مورد نظر دارند به‌ترتیب در مراتب اول نتایج ظاهر شوند تا کاربر در زمان بهینه‌تر و با صرف وقت و انرژی کمتر به نتیجه دلخواه دست یابد.

۴-۳- ارائه‌ی مدل گراف فازی

رویکرد پیشنهادی جهت رفع مسئله‌ی مطرح شده در بخش قبل ارائه‌ی مدل گراف فازی است. در این مدل به ازای هر سند، به هر نود و یال وزنی اختصاص می‌یابد که می‌توان با استفاده از آن‌ها وزن فازی هر عبارتی را، در هر سند محاسبه کرد و در ارائه‌ی نتایج از طریق موتور جستجو لیستی از اسناد را به ترتیب درجه‌ی اهمیت و میزان ارتباط با عبارت مورد جستجو در اختیار کاربر قرار داد.

جهت فازی‌سازی گراف، ساختار گراف را به نحوی اصلاح می‌کنیم که وزن‌های فازی نیز در گراف ذخیره شوند. مدل پیشنهادی به صورت شکل(۴-۱) نمایش داده شده است.

همان‌طور که در شکل(۴-۱) مشاهده می‌شود، برای ذخیره‌ی وزن هر کلمه در هر سند یک جدول جدید (TWD) و برای نگهداری وزن یال‌ها نیز بخش جدیدی به جدول یال‌ها افزوده شده است. در ادامه شیوه‌ی محاسبه این وزن‌ها مورد بررسی قرار خواهد گرفت.



شکل(۴-۱): مدل پیشنهادی ساختار گراف اصلاح شده

۴-۳-۱- محاسبه‌ی وزن‌های فازی

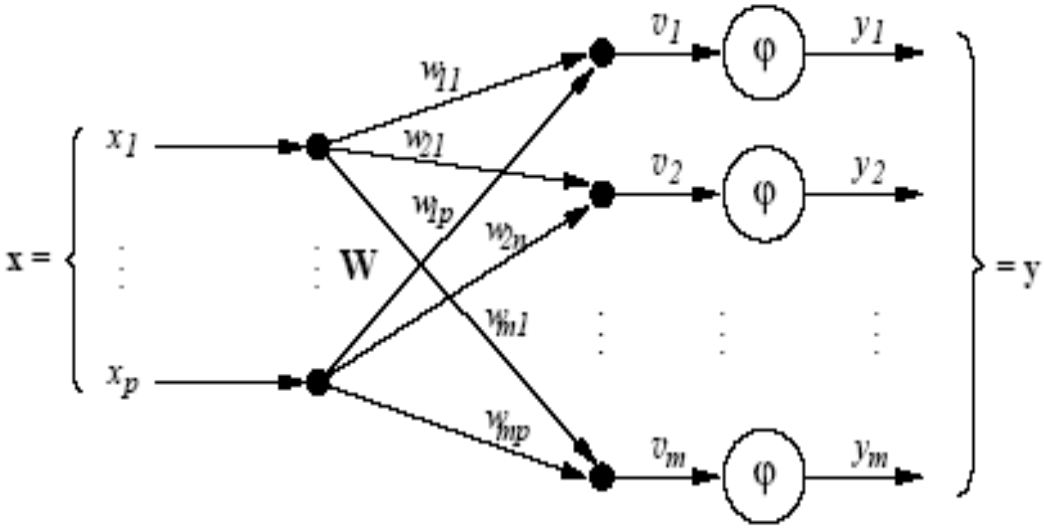
هدف از تعیین وزن‌های فازی ارزیابی میزان امکان رخداد هر کلمه در سند و رابطه‌ی میان هر دو کلمه در سند است. در واقع هر کلمه که بیشترین تکرار را در سند دارد وزن بالاتر و هر دو کلمه که بیشترین امکان رخداد را به‌صورت متوالی در سند دارند یال میان این دو کلمه در گراف وزن بیشتری را به خود اختصاص می‌دهد.

برای تعیین وزن هر نود یا به عبارت دیگر کلمه، به ازای هر سند، فرکانس رخداد آن کلمه را در سند بدست آورده و برای ایجاد قابلیت مقایسه با سایر اسناد، این مقدار را بر کل کلمات سند تقسیم کرده و وزن نرمال شده‌ای را به ازای هر کلمه در سند بدست می‌آوریم.

$$weight(v,d) = \frac{f_{vd}}{l_d} \qquad (۱-۴)$$

در این عبارت v شناسه‌ی نود، d شناسه‌ی سند، f_{vd} فرکانس کلمه v در سند d و l_d طول سند یا به عبارتی تعداد کل کلمات سند مورد بررسی است. فرایند محاسبه‌ی وزن هر نود حین پردازش هر سند در هنگام ورود به سادگی قابل محاسبه است.

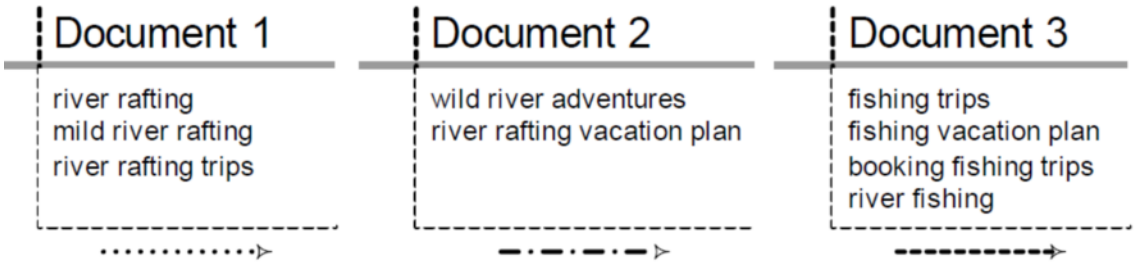
محاسبه‌ی وزن یال‌ها قدری متفاوت است. برای این امر برخلاف وزن کلمات که از طریق شمارش محاسبه می‌شود و در واقع احتمال وقوع کلمات در سند را بدست می‌آوریم، می‌خواهیم به‌جای احتمال وقوع، امکان وقوع دوبه‌دوی کلمات را محاسبه کنیم و به همین دلیل از تئوری فازی برای این منظور استفاده می‌کنیم. رویکرد پیشنهادی برای محاسبه‌ی وزن فازی یال‌ها استفاده از یک شبکه عصبی پرسپترون تک لایه است، شکل(۴-۲) معماری کلی یک شبکه عصبی پرسپترون را نشان می-دهد.



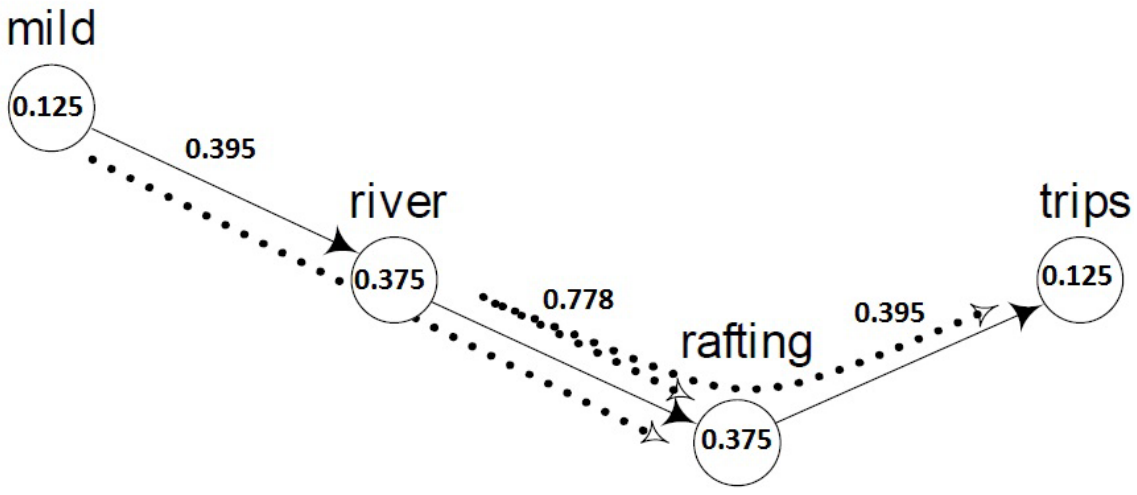
شکل(۴-۲): معماری شبکه پرسپترون

برای محاسبه‌ی وزن یال‌ها به ازای هر سند یک شبکه عصبی پرسپترون تک لایه در نظر می‌گیریم که تعداد نودهای ورودی و خروجی شبکه برابر تعداد کل کلمات منحصر به فرد سند درحال پردازش و وزن‌های شبکه متناظر با یال‌های گراف است. برای آموزش به ازای هر دو کلمه متوالی در سند، نود

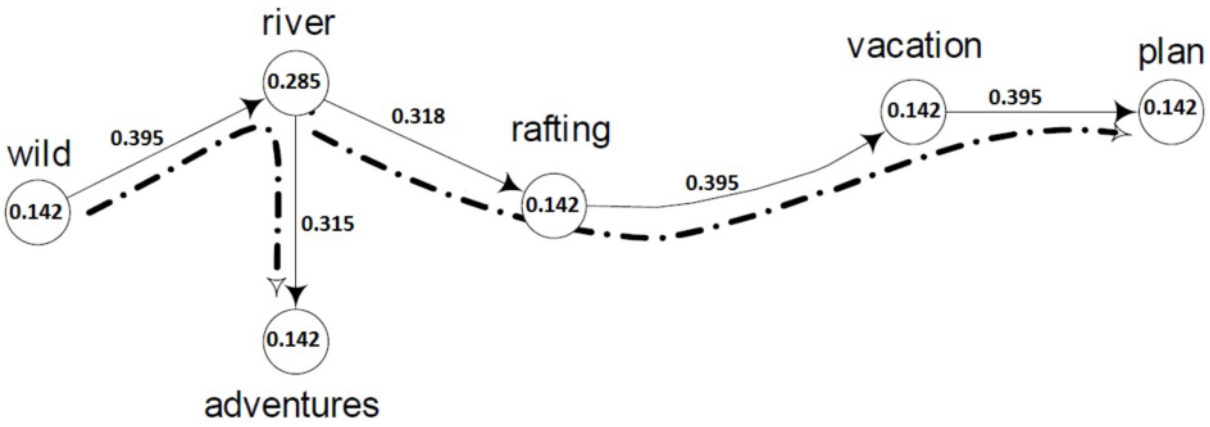
متناظر با کلمه اول در ورودی و نود متناظر با کلمه دوم در خروجی را یک و بقیه را صفر در نظر گرفته و شبکه را تا رسیدن به تغییرات اندک در خروجی آموزش می‌دهیم. این عمل را به ازای تمام جفت کلمات متوالی موجود در سند با در نظر گرفتن مرز جملات انجام داده و پس از آموزش شبکه، وزن‌های اصلاح شده‌ی شبکه را به عنوان وزن‌های فازی یال‌های متناظر با آن‌ها، در گراف ذخیره می-کنیم. شکل‌های (۴-۴) (۴-۵) و(۴-۶) گراف‌های فازی اسناد گراف شکل(۳-۲) را نشان می‌دهند.



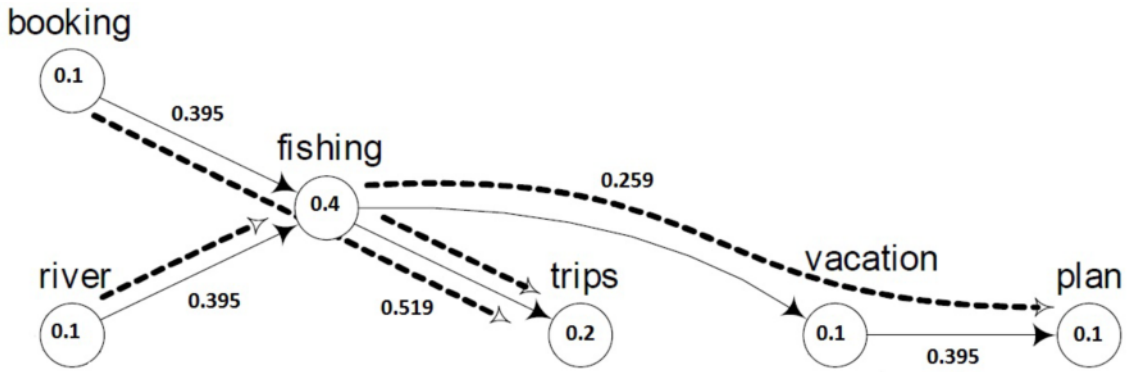
شکل(۴-۳): اسناد مثال شکل(۳-۲)



شکل(۴-۴): گراف فازی سند اول مثال شکل(۳-۲)

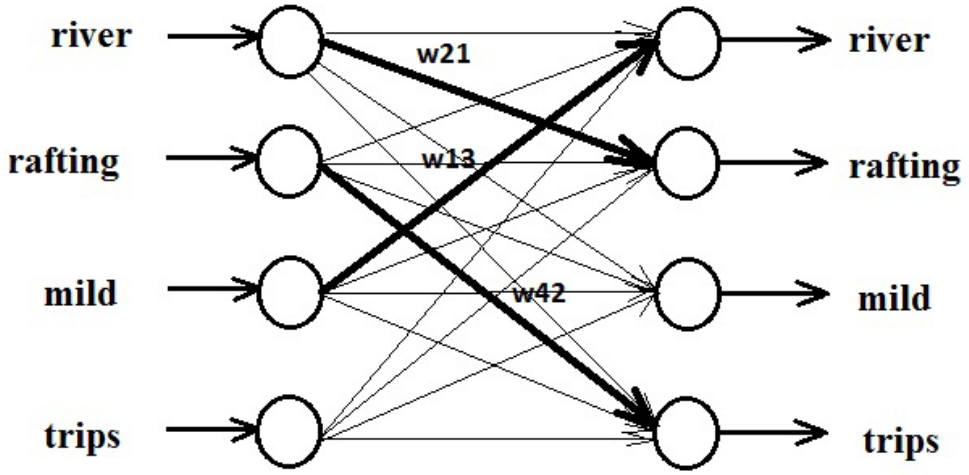


شکل(۴-۵): گراف فازی سند دوم مثال شکل(۳-۲)



شکل(۴-۶): گراف فازی سند سوم مثال شکل(۳-۲)

برای محاسبه‌ی وزن‌های فازی اسناد مثال شکل(۳-۲) سه شبکه عصبی پرسپترون تک لایه پیاده-سازی شد که شبکه عصبی سند اول دارای ۴ نود ورودی و خروجی، شبکه عصبی سند دوم دارای ۶ نود ورودی و خروجی و شبکه عصبی سند سوم نیز دارای ۶ نود ورودی و خروجی است و به شیوه‌ای که بیان شد شبکه‌ها را آموزش دادیم. وزن نودها نیز به با استفاده از فرمول(۴-۱) محاسبه شده‌اند. شکل(۴-۷) ساختار شبکه عصبی موبوط به سند ۱ را نشان می‌دهد. وزن‌های متناسب با یال‌های گراف نیز در شکل مشخص شده است.



شکل(۴-۷): ساختار شبکه عصبی سند اول

۴-۳-۲- استفاده از وزن‌های فازی در بهبود ارائه نتایج توسط موتور جستجو

پس از وارد کردن عبارت مورد جستجو توسط کاربر و بازیابی اسناد جهت ارائه‌ی نتایج، در هر سند وزن‌های متناظر با عبارت مورد جستجو (وزن نودها و یال‌های مسیر عبارت) را در هم ضرب کرده، به این ترتیب وزن عبارت را در اسناد مختلف بدست آورده و اسناد را براساس وزن‌های محاسبه شده به ترتیب در لیست ارائه‌ی نتایج در اختیار کاربر قرار می‌دهیم که با توجه به ترتیب لیست اسناد براساس عبارت مورد جستجو انتظار می‌رود سبب کاهش زمان دستیابی کاربر به اطلاعات مورد نیازش شود.

برای مثال وزن عبارت "river rafting" که عبارتی مشترک میان سند اول و دوم در گراف شکل(۳-۲) است در سند اول برابر ۰/۱۰۹ و در سند دوم برابر ۰/۰۱۲ است که به وضوح نشان دهنده‌ی فرکانس بالاتر این عبارت در سند اول است جدول(۴-۱).

جدول(۴-۱): وزن‌های مربوط به عبارت "river rafting" در اسناد اول و دوم

weight	Node: river	Edge: (river, rafting)	Node: rafting	Phrase: "river rafting"
Document 1	0.375	0.778	0.375	0.109
Document 2	0.285	0.318	0.142	0.012

۴-۴- جمع‌بندی فصل

در این فصل با اشاره به عملکرد سیستم معرفی شده در قالب بخشی از ساختار موتور جستجو، به یکی از مسائل مطرح شده دراین زمینه پرداخته و با اصلاح ساختار گراف در سیستم مورد بررسی، سعی کردیم آن را بهبود دهیم.

با تعیین وزن‌های فازی برای نودها و یال‌ها امکان رتبه‌بندی اسناد برمبنای عبارت مورد جستجو فراهم شده و اسناد بازیابی شده علاوه بر قرارگیری در خوشه‌های مربوط به خود براساس عبارت مورد نظر کاربر نیز مرتب می‌شوند و این امکان را به کاربر می‌دهد تا در زمان بهینه‌تر به اطلاعات مورد نیاز خود دسترسی پیدا کند.

در فصل بعد با مطالعه‌ی موردی[1] یک مثال به بررسی عملکرد وزن‌های فازی در بهبود ارائه‌ی نتایج جستجو به کاربر توسط موتور جستجو می‌پردازیم.

[1] Case study

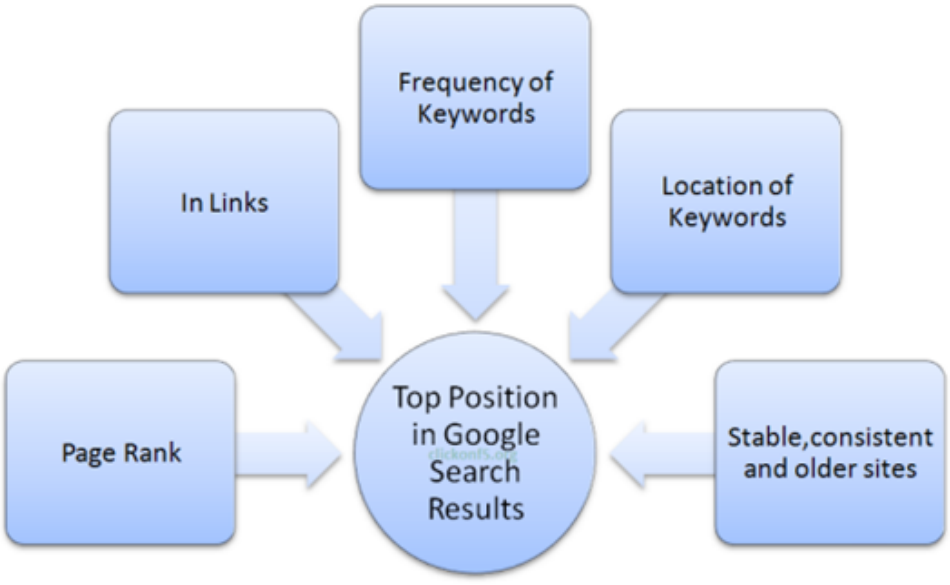
فصل پنجم

ارزیابی عملکرد سیستم بهبود یافته

۵-۱- مقدمه

همان‌طور که در فصل قبل مطرح شد، رویکرد پیشنهادی جهت بهبود سیستم معرفی شده در فصل سوم، افزودن وزن به نودها و یال‌ها در گراف، جهت محاسبه‌ی وزن عبارات مورد جستجو برای ارائه‌ی نتایج دقیق‌تر توسط موتورجستجو است. در واقع برای ارزیابی میزان تاثیر تغییر اعمال شده باید سیستم پیشنهادی دسته‌بندی اسناد در ساختار موتور جستجو گنجانده شده و نتایج یکبار بدون اعمال سیستم دسته‌بند و یکبار با استفاده از سیستم به کاربر ارائه شود تا بتوان میزان بهبود یا عدم بهبود را ارزیابی کرد.

بحث در مورد نحوه‌ی عملکرد موتورهای جستجو نیازمند تحقیقات بیشتر بوده و در این مجال نمی-گنجد، با این حال بررسی‌های اجمالی در مورد Google نشان می‌دهد که این موتور جستجو از چندین فاکتور جهت ایجاد ترتیب نتایج برای نمایش به کاربر بهره می‌گیرد. شکل زیر به‌طور خلاصه فاکتورهای موثر در ایجاد لیست نتایج در موتور جستجوی گوگل را نشان می‌دهد.



شکل(۵-۱): فاکتورهای موثر در ایجاد لیست نتایج در موتور جستجوی گوگل

[http://www.clickonf5.org/3261/google-search-engine-made-simple]

همان‌طور که مشاهده می‌شود این موتور جستجو به لحاظ متنی، تنها از کلمات کلیدی جهت مرتب-سازی نتایج بهره می‌گیرد در نتیجه پیش‌بینی می‌شود که استفاده از وزن عبارات بتواند ترتیب ارائه‌ی نتایج را بهبود دهد. در این فصل تنها در مقیاس کوچک به بررسی یک مثال پرداخته و تاثیر بهبود انجام شده را در این مثال ارزیابی می‌کنیم.

دلیل این‌که تنها در یک مطالعه موردی به بررسی عملکرد بهبود اعمال شده می‌پردازیم، مؤلفه‌های مؤثر در فرایند ارزیابی است که فراهم آوردن همه‌ی آن‌ها برای انجام یک ارزیابی معتبر مقدور نمی-باشد. از جمله‌ی این مؤلفه‌ها پایگاه داده مناسب برای انجام ارزیابی است. اغلب پایگاه داده‌های موجود جهت انجام پژوهش‌های مبتنی بر وب، پایگاه داده‌های خبری هستند. علی‌رغم مناسب بودن این پایگاه داده‌ها در پژوهش‌های مبتنی بر دسته‌بندی اسناد، با توجه به این‌که اغلب افراد برای جستجوی مطلبی خاص در اینترنت به اسنادی غیر از اسناد خبری مراجعه می‌کنند، به نظر می‌رسد به‌کارگیری آن‌ها برای بررسی عملکرد چنین سیستمی مناسب نیستند. مؤلفه‌ی مهم دیگر تعداد کاربران جهت انجام آزمایشات است. از آنجایی که ارزیابی چنین سیستمی کاملاً کاربر محور بوده و وابسته به نظر شخصی کاربر است نمی‌توان با استفاده از معیارهای معمول عملکرد سیستم را مورد بررسی قرار داد، به‌همین دلیل به تعداد قابل قبولی کاربر جهت ارزیابی سیستم نیاز داریم تا با برایند میزان رضایت کاربران از نتایج ارئه شده به‌صورت کیفی میزان بهبود عملکرد سیستم را ارزیابی کنیم. مؤلفه‌ی دیگر ایجاد یک بستر موتور جستجو است که سیستم ارائه شده در این بستر مورد ارزیابی قرار گیرد البته لازم به ذکر است که فاکتور وزن عبارت مورد جستجو باید به صورت ترکیبی با فاکتورهای دیگری که توسط موتور جستجو به‌کار گرفته می‌شود مورد استفاده قرار گیرد تا بهترین نتیجه حاصل شود.

در ادامه با بررسی یک مثال نشان می‌دهیم که وزن عبارت مورد جستجو می‌تواند در ارائه‌ی ترتیب بهتری از نتایج بازیابی شده توسط موتور جستجو موثر باشد.

۵-۲- ارزیابی سیستم ارائه شده در مقیاس کوچک

برای ارزیابی عملکرد سیستم بهبود یافته ابتدا جستجویی را در موتور جستجو (Google) انجام داده و لیست نتایج را بررسی می‌کنیم تا اسنادی را که تا حدودی درخواست ما را تامین می‌کنند تعیین کنیم. سپس اسناد بازیابی شده را وارد گراف کرده و وزن‌های نودها و یال‌ها را به ازای هر سند تعیین می‌کنیم، سپس وزن عبارت مورد جستجو را محاسبه کرده اسناد را برحسب وزن عبارت مورد جستجو مرتب می‌کنیم.

در صورتی که اسناد تعیین شده‌ای که درخواست ما را تامین می‌کنند، پس از مرتب شدن براساس وزن عبارت مورد جستجو، در مراتبی بالاتر نسبت به لیست نتایج ارائه شده توسط موتور جستجو قرار بگیرند، می‌توان گفت عملکرد بازیابی اطلاعات با بهبود همراه بوده است.

برای شروع عبارت "ponseti method" را مورد جستجو قرا می‌دهیم و ۱۰ سند ابتدای لیست را بررسی می‌کنیم. از بین اسناد بازیابی شده اسناد ۷، ۴ و ۳ به ترتیب بیشترین اطلاعات را برای ما فراهم می‌کنند. برای مرتب کردن اسناد براساس وزن عبارت مورد جستجو، گراف نمایه‌سازی این ۱۰ سند را ساخته و به ازای هر سند وزن نودها و یال‌ها را محاسبه می‌کنیم. پس از مرتب کردن اسناد براساس وزن عبارت مورد جستجو ترتیب اسناد به صورت جدول(۳-۱) تغییر می‌کند.

همان‌طور که در جدول مشاهده می‌شود از بین اسناد مورد نظر ما اسناد ۷ و ۴ با سه و یک رتبه صعود و سند ۳ با سه درجه نزول در لیست نتایج در رتبه‌های ۴، ۳ و ۶ قرار گرفته‌اند. به نظر می‌رسد استفاده از وزن عبارات موجب بهبود نسبی قرارگیری اسناد در لیست نتایج شده است. البته با توجه به مقیاس کوچک این ارزیابی اثبات بهبود عملکرد سیستم قطعی نیست ولی به نظر می‌رسد در مقیاس بزرگ نیز نتایج با بهبود همراه باشد.

جدول(۵-۱): لیست اسناد بازیابی شده بر اساس وزن عبارت مورد جستجو

شناسه‌ی سند	رتبه‌ی قرارگیری در لیست نتایج
سند ۹	۱
سند ۸	۲
سند ۴	۳
سند ۷	۴
سند ۶	۵
سند ۳	۶
سند ۱	۷
سند ۵	۸
سند ۲	۹
سند ۱۰	۱۰

استفاده از این روش با وجود بهبود نسبی در ارائه‌ی نتایج، توسط موتور جستجو با مسائلی روبروست که در ادامه به برخی از آن‌ها می‌پردازیم:

- ممکن است عبارتی مورد جستجو قرار گیرد که کلمه یا کلماتی از آن در گراف موجود نباشد، به‌واسطه‌ی ترکیبات اضافی و وصفی یا کلمات stop word که ممکن است در سند اصلی موجود بوده ولی در فرایند پیش‌پردازش حذف شده، در چنین شرایطی برای مقادیر ناموجود، به‌جای مقدار صفر، کوچک‌ترین وزن محاسبه شده در بین اسناد را قرار می‌دهیم تا در صورت وجود دیگر زیررشته‌های عبارت در گراف، بتوان نتایج قابل قبولی را به کاربر ارائه داد.
- برخی صفحات وب نیز بدون داشتن هیچ‌گونه اطلاعات مفید، تنها با قرار دادن مکرر کلمات و عباراتی که بیشترین میزان جستجو را در میان کاربران دارند، سعی در افزایش بازدید این صفحات و بالا بردن رنکینگ در موتور جستجو با اهداف تبلیغاتی دارند، که استفاده از این روش در مورد این صفحات با خطا مواجه می‌شود. در این موارد می‌توان با تعیین یک مقدار

آستانه برای فرکانس کلمات درسند پس از عملیات پیش‌پردازش، در صورت تکرار بیش از این مقدار آستانه، از ورود این سند به مجموعه پایگاه داده جلوگیری کرد.

۵-۳- جمع‌بندی فصل

همان‌طور که مشاهده شد در این فصل تنها در مقیاس کوچک به بررسی عملکرد سیستم ارائه شده پرداختیم، نکته‌ی قابل توجه در ارزیابی عملکرد چنین سیستمی کاربر محور بودن آن است، برای مثال ممکن است دو کاربر با عبارت یکسان عمل جستجو را انجام دهند و لیست نتایج هم کاملاً یکسان باشد، ولی کاربر اول با سومین سند به اطلاعات مورد نظرش دست یابد و کاربر دوم با دهمین سند، بر این اساس ارزیابی کاملاً کاربر محور بوده و بستگی به نظر شخصی کاربر دارد. در نتیجه عملکرد سیستم بدون درنظر گرفتن نظر کاربر، براساس فرمول خاصی قابل ارزیابی نیست.

جهت ارزیابی دقیق این سیستم باید این سیستم به عنوان بخشی از یک موتور جستجو و همراه با یک پایگاه داده معتبر در اختیار تعداد قابل قبولی کاربر قرار گرفته و پس از انجام جستجو توسط آن‌ها میزان رضایت کاربران از نتایج ارائه شده، بررسی شود و به صورت کیفی میزان بهبود عملکرد سیستم مورد ارزیابی قرار گیرد.

فصل ششم

نتیجه گیری و کارهای آینده

۶-۱- نتیجه‌گیری و کارهای آینده

در این پایان‌نامه به‌طور مفصل و از زوایای مختلف به بحث درمورد دسته‌بندی اسناد پرداختیم. در ابتدای تحقیق الزامات یک سیستم دسته‌بندی اسناد را برشمرده و در ادامه به بررسی ساختار و عملکرد متدهای خوشه‌بندی مختلف و مدل‌های نمایش مورد استفاده در دسته‌بندی اسناد پرداخته و نقاط قوت و ضعف آن‌ها را مورد بررسی قرار دادیم.

پس از بررسی متدهای مرسوم، به معرفی یک سیستم جدید دسته‌بندی اسناد پرداختیم که با بهره-گیری از ۴ مولفه‌ی، تحلیل‌گر اسناد وب، مدل جدید گراف نمایه‌سازی اسناد، معیار شباهت مبتنی بر عبارت و متد خوشه‌بندی مبتنی بر هیستوگرام شباهت، بهبود قابل توجهی را در دسته‌بندی اسناد وب فراهم کرده است. این سیستم با معرفی مدل نمایش جدید مبتنی بر گراف برخلاف مدل‌های مرسوم نمایش اسناد، از عبارات به عنوان ویژگی‌هایی با بیشترین بار اطلاعاتی جهت نمایه‌سازی اسناد بهره می‌گیرد که سبب ایجاد خوشه‌هایی با دقت و کیفیت بالا می‌شود. از آنجا که این سیستم قابلیت به-کارگیری در موتور جستجو را جهت دسته‌بندی اسناد بازیابی شده در پاسخ به جستجوی کاربر دارد با ارائه‌ی ساختار اصلاح شده‌ی گراف و افزودن وزن‌هایی به نودها یال‌های گراف سعی کردیم عملکرد این سیستم را به عنوان بخشی از یک موتور جستجو بهبود دهیم.

انتظار می‌رود بهبود اعمال شده علی‌رغم تغییر به‌ظاهر کوچک ساختار گراف، زمان عملیات جستجو توسط کاربر را کاهش دهد که با توجه به رشد روز افزون اطلاعات در محیط وب امری ضروری به نظر می‌رسد.

در ادامه‌ی این پژوهش می‌توان به پیشنهادات زیر اشاره کرد:

- می‌توان با درنظر گرفتن سطوح اهمیت در محاسبه‌ی وزن نودها و یال‌ها، نتایج به مراتب دقیق‌تری را در قبال جستجوی کاربر ارائه داد.

- با توجه به این‌که ممکن است عبارت مورد جستجو دقیقا در اسناد وجود نداشته باشد، تعیین وزن‌های مناسب برای کلماتی که در عبارت مورد جستجو وجود دارند ولی در اسناد وجود ندارند یا در مرحله‌ی پیش پردازش حذف شده‌اند جهت ارائه‌ی نتایج قابل قبول به کاربر امری ضروری است.
- تعیین مقدار آستانه‌ی مناسب برای حذف اسناد تبلیغاتی و فاقد اطلاعات مفید.

فصل هفتم

مراجع

مراجع:

[1] Hammouda K, (2002), Master. Thesis, "Web Mining: Identifying Document Structure forWeb Document Clustering ", Systems Design. Depart. Waterloo University.

[2] Khaled M. Hammouda, Mohamed S. Kamel, (2004), "Efficient Phrase-Based Document Indexing For Web Document Clustering", *IEEE*, NO. 10, VOL. 16, pp 1279-1296.

[3] QUINLAN J.R, (1968), "Induction of Decision Trees ", Kluwer Academic Publishers, Machine Learning, pp 81-106.

[4] Tom Mitchell, (1997), "Machine Learning", McGraw Hill, pp 52-80.

[5] Ramaswamy S, (2006), "Multiclass Text Classification A Decision Tree based SVM Approach", Berkeley, pp 30-39.

[6] Yang y, (2000), "Decision Trees in Text Categorization", AI, pp 45-66.

[7] Pise N, (2007),"Document Classification using Neural Networks", Maharshra Institute of Technology

[۸]بهمنی ز، صفابخش ر، (۱۳۸۹)، "معرفی یک شبکه عصبی خودسازمانده رشدیابنده سلسله مراتبی سریع"، شانزدهیم کنفرانس ملی سالانه انجمن کامپیوتر ایران، ص ۶۷۳، تهران

[9] Chihli Hung, Stefan Wermter, (2005), "Neural Network-based Document Clustering Using WordNet Ontologies".

[10] Chihli Hung, Stefan Wermter, Peter Smith, (2003), "Hybrid Neural Document Clustering Using Guided Self-Organization and WordNet", *IEEE*, pp 68-77.

[11] Samuel Kaski, Timo Honkela, Krista Lagus, Teuvo Kohonen, (1998), " WEBSOM Self-organizing maps of document collections", *Elsevier*, Neurocomputing 21, pp 101-117.

[12] James Stuart Aitken, "Learning Information Extraction Rules: An Inductive Logic Programming approach", Artificial Intelligence Applications Institute, Division of Informatics, University of Edinburgh.

[13] Ganesh Ramakrishnan, Sachindra Joshi, Sreeram Balakrishnan, Ashwin Srinivasan, (2007), "Using ILP to Construct Features for Informatio Extraction from Semi-Structured Text".

[14] Nuanwan Soonthornphisaj, Boonserm Kijsirikul, (2007), "Combining ILP with Semi-supervised Learning for Web Page Categorization", *World Academy of Science*, pp 682-685.

[15] Markus Junker, Michael Sintek, Matthias Rinck, (2000), "Learning for text categorization and information extraction with ILP", *German research center for artificial intelligence*, pp 84-93.

[16] Eissen S., Stein B., Potthast M., "The Suffix Tree Document Model Revisited", 5th International Conference on Knowledge Management, Journal of Universal Computer Science, pp. 596-603, Tochtermann

[17] Scott S., Matwin S., (1999), "Feature engineering for text classification", In *Proceedings of the 16th International Conference on Machine Learning* , pp 379–388.

[18] E. Fersini, E. Messina, F. Archetti, (2010), "A probabilistic relational approach for web document clustering", *Elsevier,* Information Processing and Management 46, pp 117–130.

[19] K. Aas and L. Eikvil, (1999), "Text Categorisation: A Survey," Technical Report 941, Norwegian Computing Center.

[20] Yang Y., Pedersen J.P., (1997), "A Comparative Study on Feature Selection in Text Categorization," Proc. 14th Int'l Conf. Machine Learning (ICML '97), pp. 412-420.

[21] M.F. Caropreso, S. Matwin, F. Sebastiani, (2000), "Statistical Phrases in Automated Text Categorization", Technical Report IEI-B4-07-2000, pp. 1-15.

[22] Kurt Hornik, Patrick Mair, Johannes Rauch, Wilhelm Geiger, Christian Buchta, Ingo Feinerer, (2013),"The textcat Package for n-Gram Based Text Categorization in R", *Journal of Statistical Software*, Volume 52, Issue 6, pp 1-17.

[23] Mason J., Shepherd M., Duffy J., Keselj V., Watters C., (2010), "An *n*-gram Based Approach to Multi-labeled Web Page Genre Classification", International Conference on System Sciences *IEEE*, Hawaii.

[24] Yingbo Miao, Vlado Keselj, Evangelos Milios, (2005), "Document Clustering using Character Ngrams: A Comparative Evaluation with Termbased and Wordbased Clustering", *ACM*.

[25] Mahdi Shafiei, Singer Wang, Roger Zhang, Evangelos Milios, Bin Tang, Jane Tougas, Ray Spiteri, (2007), "Document Representation and Dimension Reduction for Text Clustering", IEEE, pp 770-779.

[26] C. Y. Suen, (1979), "N-gram statistics for natural language understanding and text Processing", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, VOL PAMI-1, NO 2,pp 164–172.

[27] Kim M., Whang K., Lee J., Lee M, (2005), "n-Gram/2L: A Space and Time Efficient Two-Level n-Gram Inverted Index Structure", Proceedings of the 31st VLDB Conference, pp. 325-328, Trondheim, Norway.

[28] O. Zamir and O. Etzioni, (1999), "Grouper: A Dynamic Clustering Interface to Web Search Results", Computer Networks, vol. 31, no. 11-16, pp. 1361-1374.

[29] Zamir O., Etzioni O., (1998), "Web Document Clustering: A Feasibility Demonstration", Proc. 21st Ann. Int'l ACM SIGIR Conf., pp. 46-54, Melbourne.

[30] Hung Chim, Xiaotie Deng, (2008), "Efficient Phrase-Based Document Similarity for Clustering", *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, NO. 9, VOL. 20, pp. 1217-1229.

[31] Momin B. F., Kulkarni P. J., Chaudhari A, (2006), " Web Document Clustering Using Document Index Graph", *conf. IEEE*, pp. 32-37.

[32] Wang Y., (2000), " Web Mining and Knowledge Discovery of Usage Patterns", Project report.

Abstract:

With the tremendous growth of the World Wide Web, many research projects were targeted on how to organize such information in a way that will make it easier for the end users to find the information they want efficiently and accurately.

Document Clustering is an important tool for many Information Retrieval (IR) tasks.

Document clustering techniques mostly rely on single term analysis of the document data set, such as the Vector Space Model. To achieve more accurate document clustering, more informative features including phrases and their weights are particularly important in such scenarios. Although other techniques such as suffix tree find out any length common phrases between documents but it suffers from high redundancies stored in the form of suffixes.

The new model Phrase-Based Document Index graph is proposed in 2004. This model incrementally constructs a graph by phrase-based indexing the document set; this allows us to make use of more informative phrase matching rather than individual words matching which provides efficient similarity calculation between documents. This model has no redundancy and supports any number of documents in clustering. This efficient performance of construction/phrase-matching lends itself to online incremental processing, such as processing the results of a Web search engine retrieved list of documents. The quality of the clusters produced using this system was higher than those produced using traditional clustering methods.

This thesis is studying on different methods of document clustering and discusses their strengths and weaknesses, then focuses on new proposed document clustering system and its advantages over previous methods. Since that this new system is able to be used in a search engine to cluster retrieved documents, we consider operation of this system from search engine point of view and try to improve its efficiency as a part of search engine structure.

Search engine often considers the rate of visiting documents to create the order of the search results displayed for user. By using proposed system in search engine and adding weights to nodes and edges of the graph we can compute the weight of the search phrase in different documents and sort them based on phrase weight, this causes user achieve her/his desired information more accurately and quickly.

To adding weights by modifying graph structure, for each document compute node weights by counting and edge weights by using a single layer-perceptron and improve the System Performance as a part of search engine structure.

Key words:
Document clustering, phrase-based indexing, document index graph, fuzzy graph.

Shahrood University

Faculty of Computer Engineering & Information Technology

M. Sc. Thesis

Web Document Clustering Using Document Index Graph

Narjes Ramazani Oomali

Supervisor:

Dr. Morteza Zahedi

Advisor:

Dr. Hamid Hasanpour

September 2013