ارائه یک چارچوب به منظور توسعه خودکار توابع رتبه‌بندی در وب با استفاده از برنامه‌نویسی ژنتیک

ناصر مزینی
دانشگاه علم‌وصنعت ایران
mozayani@iust.ac.ir

سید ناصر رضوی
دانشگاه علم‌وصنعت ایران
razavi@Comp.iust.ac.ir

چکیده

توابع رتبه‌بندی نقش بسیار مهمی در کارآیی سیستم‌های بازیابی اطلاعات (IR) و موتورهای جستجو دارند. گرچه در ادبیات IR توابع رتبه‌بندی بسیاری وجود دارند، اما مطالعات تجربی مختلفی که به منظور ارزیابی کارآیی آنها انجام گرفته‌اند، نشان داده‌اند که این توابع در زمینه‌های مختلف (پرس‌وجوها، کلکسیون‌ها، کاربران) عملکرد خوبی ندارند. علاوه بر این، طراحی توابع رتبه‌بندی بهینه توسط انسان و به صورت دستی که بتوانند برای پرس‌وجوها، کلکسیون‌ها و کاربران مختلف به خوبی عمل کنند، کاری بسیار دشوار و پرهزینه می‌باشد. ما در این مقاله یک چارچوب جدید بر اساس برنامه‌نویسی ژنتیک به منظور طراحی و توسعه خودکار توابع رتبه‌بندی، ارائه و پیاده‌سازی نموده‌ایم و بوسیله آزمایش‌های متعدد نشان داده‌ایم که چگونه چنین چارچوبی می‌تواند به فرآیند طراحی و توسعه خودکار توابع رتبه‌بندی کمک کند.

1- مقدمه

در سال‌های اخیر زمینه بازیابی اطلاعات (IR) به دلیل پیشرفت‌های انجام گرفته در تکنولوژی اطلاعات (IT) و تکنیک‌های محاسباتی، به شدت در معرض توسعه و تغییر قرار گرفته است. میزان بسیار زیاد اطلاعات دیجیتالی در جوامع بشری که به طور روزافزون در حال افزایش می‌باشد، باعث شده است که تحقیقات در زمینه بازیابی اطلاعات به یکی از مهیج‌ترین و مهم‌ترین زمینه‌های تحقیقاتی در حال حاضر تبدیل گردد. برطبق *SearchEngineWatch.com*[1] بیش از 75 درصد کابران از موتورهای جستجو برای یافتن اطلاعات مورد نظر خود در وب استفاده می‌کنند، که این مطلب به خوبی بیانگر اهمیت و نقش بازیابی اطلاعات در زندگی روزمره ما می‌باشد. متأسفانه، برخلاف اهمیت فراوانی که سیستم‌های بازیابی اطلاعات در زندگی ما انسان‌ها دارند، مطالعات مختلفی که به منظور ارزیابی این سیستم‌ها انجام شده‌اند نشان می‌دهند که در حال حاضر عملکرد این سیستم‌ها در حد انتظار کابران آنها نمی‌باشد[1]. دلایل متعددی در نارضایتی کاربران از عملکرد سیستم‌های بازیابی اطلاعات نقش دارند که یکی از مهم‌ترین آنها نحوه رتبه‌بندی اسناد بازیابی شده توسط سیستم برای ارائه به کاربر می‌باشد. سیستم‌های بازیابی اطلاعات از توابع رتبه‌بندی برای مرتب نمودن نتایج حاصل از جستجو به صورت نزولی و بر اساس میزان مرتبط بودن آنها با درخواست کاربر استفاده می‌کنند.

اغلب سیستم‌های بازیابی اطلاعات موجود برای تمامی کاربران و پرس‌وجوهای متفاوتی که از طرف آنها مطرح می‌شوند، از یک استراتژی ثابت به منظور رتبه‌بندی نتایج بازیابی شده استفاده می‌کنند (جستجوی همگانی). کاربران در بسیاری از موارد ترجیح می‌دهند که نتایج

[1] http://www.searchenginewatch.com/reports/seindex.html

جستجو براساس نیازمندی‌ها و سلیقه‌های شخصی آنها تنظیم شوند (جستجوی شخصی). اغلب موتورهای جستجوی فعلی، از چنین ویژگی پیشرفته‌ای پشتیبانی نمی‌کنند[2].

از طرفی، اغلب توابع رتبه‌بندی موجود بر اساس مشاهدات و به صورت تجربی طراحی شده‌اند و لذا تئوری خاصی هم در ایجاد آنها وجود ندارد و همچنین این توابع در اغلب موارد از دقت لازم برخوردار نمی‌باشند. حتی آن دسته از توابع رتبه‌بندی نیز که بر اساس نظریه احتمالات طراحی شده‌اند، عملکردشان کاملاً وابسته به متن (کاربران, پرس‌وجوها و کلکسیون‌ها) می‌باشد؛ به این معنا که ممکن است این توابع برای برخی از پرس‌وجوهای خاص بسیار خوب عمل نمایند، اما برای برخی دیگر عملکرد ضعیفی از خود نشان دهند. از طرفی طراحی یک تابع رتبه‌بندی بهینه برای هر شخص و هر پرس‌وجو به صورت دستی توسط متخصصین مربوطه نیز به تلاش و زمان بسیار زیادی نیاز دارد و در عمل غیرممکن می‌باشد.

با توجه به نقش و اهمیت سیستم‌های بازیابی اطلاعات در زندگی ما انسان‌ها و همچنین مشکلاتی که در بالا مطرح شد، توسعه یک سیستم خودکار به منظور توسعه توابع رتبه‌بندی هم در انجام جستجوی شخصی (بر اساس اولویت‌ها و سلیقه‌های شخصی کاربران) و هم در انجام جستجوی همگانی کاملاً ضروری می‌باشد. ما در این مقاله یک چارچوب بر اساس برنامه‌نویسی ژنتیک به همین منظور ارائه و پیاده‌سازی نموده‌ایم که قادر است برای یک پرس‌وجو و یا مجموعه‌ای از پرس‌وجوها توابع رتبه‌بندی بهتری نسبت به توابع موجود به صورت خودکار توسعه دهد. در بخش دوم این مقاله مدل فضای برداری به عنوان مدل استفاده شده در پیاده‌سازی چارچوب ارائه شده و نیز مشکلات موجود در زمینه رتبه‌بندی اسناد و همچنین نقاط ضعف اصلی سیستم‌های فعلی بازیابی اطلاعات و ضرورت بهبود آنها مطرح شده اند. در بخش سوم ابتدا مسأله توسعه توابع رتبه‌بندی به شکل رسمی‌تری تعریف شده است و در نهایت یک چارچوب به منظور توسعه خودکار توابع رتبه‌بندی با استفاده از برنامه‌نویسی ژنتیک ارائه شده است. در این بخش، برنامه‌نویسی ژنتیک و عناصر ضروری آن معرفی شده‌اند و دلایل استفاده از این تکنیک در این پروژه بررسی شده‌اند. در بخش چهارم با انجام مطالعات موردی بر روی یک کلکسیون استاندارد موجود در زمینه بازیابی اطلاعات به نام Med و با انجام دو آزمایش مختلف قدرت چارچوب ارائه شده در توسعه و طراحی توابع رتبه‌بندی شخص و همگانی مورد بررسی قرار گرفته است. همچنین در این بخش به منظور ایجاد درک بهتر از سودمندی چهارچوب ارائه شده در توسعه خودکار توابع رتبه‌بندی، عملکرد توابع توسعه یافته توسط سیستم GP با دو نمونه از توابع رتبه‌بندی مطرح موجود به نام‌های Okapi و PTFIDF مقایسه شده است. بخش پنجم نیز شامل نتیجه‌گیری و ارائه زمینه‌هایی برای انجام تحقیقات بیشتر می‌باشد.

2- مروری بر کارهای قبلی

سیستم‌های بازیابی اطلاعات از توابع رتبه‌بندی به منظور مرتب‌سازی اسناد بازیابی شده بر طبق تخمین میزان مرتبط بودن آنها با پرس‌وجوی کاربر استفاده می‌کنند. جهت ساده‌تر شدن این فرآیند نیاز است که هم اسناد و هم پرس‌وجوی کاربر به گونه‌ای بازنمایی شوند که بتوانند به شکلی کارآ و مؤثر توسط کامپیوترها مورد پردازش قرار بگیرند. مدل فضای برداری (VSM) در این زمینه یکی از موفق‌ترین مدل‌ها می‌باشد[3, 4]. ما نیز در این مطالعه به دلیل سادگی این مدل و نیز به دلیل موفقیت چشگمیر این مدل در مطالعات مختلفی به منظور ارزیابی کارآیی سیستم‌های بازیابی اطلاعات صورت گرفته است از همین مدل استفاده نموده‌ایم[4, 5, 6, 7, 8].

در این مدل هم اسناد و هم پرس‌وجوهای کاربران، به عنوان بردارهایی از کلمات شاخص نمایش داده می‌شوند. اگر در تمامی یک کلکسیون کلاً t کلمه شاخص وجود داشته باشد، یک سند داده شده مانند D و یک پرس‌وجو مانند Q را می‌توان به شکل زیر نمایش داد:

$D = (w_{d1}, w_{d2}, w_{d3}, \ldots, w_{dt})$
$Q = (w_{q1}, w_{q2}, w_{q3}, \ldots, w_{qt})$

که در این نمایش‌ها w_{di} و w_{qi} (بازای i از 1 تا t) به ترتیب وزن‌های اختصاص یافته به کلمات مختلف موجود در سند D و پرس‌وجوی Q می‌باشند. در این نحوه نمایش، شباهت میان یک سند با یک پرس‌وجو را می‌توان از طریق معیار پر استفاده کسینوسی به شکل زیر محاسبه نمود[8]:

$$Similarity(Q,D)=\frac{\sum_{i=1}^{t} w_{qi}\times w_{di}}{\sqrt{\sum_{i=1}^{t}(w_{qi})^2\times\sum_{i=1}^{t}(w_{di})^2}} \qquad (1)$$

سپس می‌توان اسناد را براساس مقدار نزولی محاسبه شده توسط این معیار مرتب نمود.

هنر طراحی توابع رتبه‌بندی خوب، بستگی به استراتژی وزن‌دهی به کلمات دارد. یک استراتژی وزن‌دهی، وزن‌هایی را به کلمات بکار رفته در یک سند، یعنی w_{di}، اختصاص می‌دهد (w_{qi} معمولاً برابر 1 در نظر گرفته می‌شود و بنابراین به راحتی می‌توان از آن صرف‌نظر کرد). استراتژی‌های وزن‌دهی متفاوت بر روی میزان شباهت محاسبه شده توسط رابطه (1) تاثیر می‌گذارند. در ادامه به عنوان مثال، دو نمونه از توابع رتبه‌بندی شناخته‌شده آورده شده‌اند[9].

- OKAPI BM25

$$\sum_{T\in Q}\frac{3\times tf}{0.5+1.5\times\frac{length}{length_{avg}}+tf}\times\log\frac{N-df+0.5}{df+0.5}\times QTW \qquad (2)$$

- Pivoted TFIDF

$$\sum\frac{1+\log(tf)}{1+\log(tf_{avg})}\times\log\left(\frac{N+1}{df}\right)\times\frac{1}{0.8+0.2\times\frac{length}{length_{avg}}}\times QTW \qquad (3)$$

واضح است که تفاوت این روابط، در استراتژی‌های وزن‌دهی آنها یعنی در نحوه ترکیب کردن ویژگی‌های وزن‌دهی می‌باشد. تغییر یک استراتژی وزن‌دهی، به طور حتم باعث ایجاد تغییر در رفتار یک تابع رتبه‌بندی خواهد شد. ما از اینجا به بعد از عبارت‌های «استراتژی وزن‌دهی» و «تابع رتبه‌بندی» به یک منظور استفاده خواهیم نمود.

البته در بازیابی اطلاعات، توابع بسیار زیاد دیگری نیز به منظور رتبه‌بندی وجود دارند[8, 9, 10, 11]. مطالعات مختلفی که به منظور ارزیابی کارآیی توابع رتبه‌بندی انجام گرفته‌اند، نشان می‌دهند که عملکرد این توابع در رتبه‌بندی اسناد به میزان بسیار زیادی وابسته به متن می‌باشد[8, 9, 11]. یعنی یک تابع رتبه‌بندی مفروض ممکن است برای انواع خاصی از پرس‌وجوها به خوبی عمل نماید، درحالی‌که برای پرس‌وجوهای دیگر عملکرد مناسبی را از خود نشان ندهد. تفاوت در نحوه مشخص کردن و بیان نمودن پرس‌وجوها و کلمات تشکیل دهنده آنها، توزیع‌های آماری مختلف برای کلمات موجود در یک کلکسیون و تنوع کاربران، برخی از دلایل مربوط به ناسازگاری این توابع رتبه‌بندی می‌باشند.

این مطالعات، مشاهدات زیر را در مورد وضعیت کنونی توابع رتبه‌بندی آشکار می‌کنند:

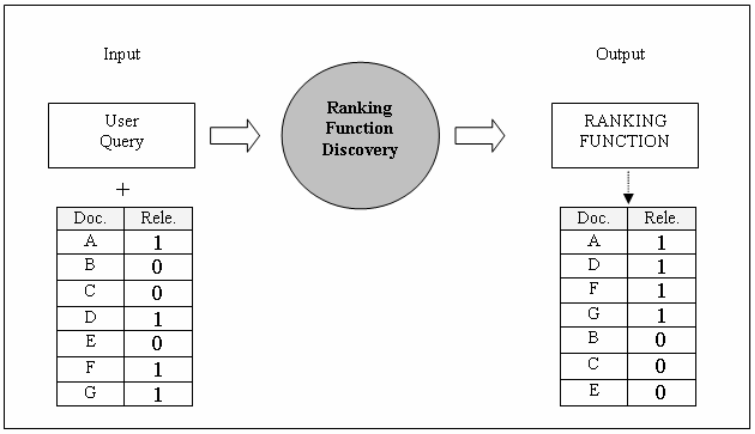
الف) هیچ تضمینی مبنی بر بهینه بودن توابع رتبه‌بندی موجود وجود ندارد و هنوز ممکن است که بتوان توابع قوی‌تری را به منظور رتبه‌بندی طراحی نمود[12].

ب) هیچ‌گونه توافق عمومی برای اینکه از کدام تابع رتبه‌بندی در کدام زمینه و متن استفاده شود وجود ندارد[4, 11].

واضح است که یک چارچوب به منظور توسعه و طراحی توابع رتبه‌بندی خوب, بیش از هر چیز دیگری در این زمینه مورد نیاز می‌باشد.

3- مسأله طراحی توابع رتبه بندی

مسأله یافتن یک تابع رتبه‌بندی خوب در شکل (1) نشان داده شده است. در این شکل در ستون "Rele" در هر دو جدول مربوط به اسناد، نمادهای "1" و "0" به ترتیب بیانگر «مرتبط بودن» و «مرتبط نبودن» سند با پرس‌وجوی مورد نظر می‌باشند.



شکل 1 مسأله طراحی تابع رتبه‌بندی

می توان مسأله یافتن یک تابع رتبه‌بندی را به طور رسمی به شکل زیر بیان نمود:

«با داشتن یک پرس‌وجوی کاربر (مجموعه ای از پرس‌وجوها) و یک مجموعه از اسناد آموزشی به عنوان ورودی که درباره میزان مرتبط بودن آنها با این پرس‌وجو (مجموعه پرس‌وجوها) قضاوت شده باشد، به دنبال توسعه یک تابع رتبه‌بندی بوسیله چارچوب مطرح شده می‌باشیم که بالقوه قادر باشد تمام اسناد مرتبط را در بالا و اسناد غیر مرتبط را در پایین لیست رتبه‌بندی قرار دهد.»

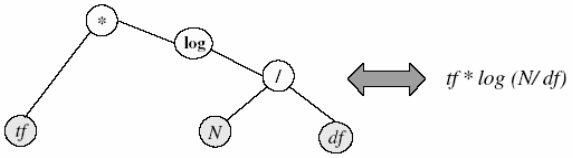
به دلیل خصوصیات و مشخصات این مسأله که در بالا به آنها اشاره شد، به نظر می‌رسد که استفاده از یک تکنیک یادگیری استقرایی به نام برنامه‌نویسی ژنتیک[13]، به عنوان تکنیک یادگیری در این مسأله به منظور یادگیری تابع رتبه‌بندی، بسیار مناسب باشد. دلایل ما برای استفاده از برنامه‌نویسی ژنتیک در این مطالعه به شرح زیر می‌باشند:

- عدم سخت‌گیری در مورد تابع هدف

این خاصیت به ما امکان می‌دهد که بتوانیم از معیارهای کارآیی گسسته و متداول در بازیابی اطلاعات، مانند دقت میانگین، به عنوان یک تابع هدف در فرآیند یادگیری استفاده نماییم.

- سادگی نمایش توابع رتبه‌بندی به عنوان راه‌حل‌های مسأله

در برنامه‌نویسی ژنتیک شکل‌های بسیاری به منظور نمایش راه‌حل‌ها - در این مورد، توابع رتبه بندی- ممکن می‌باشد[14]. یک روش بازنمایی متداول برای نمایش راه‌حل‌ها در برنامه‌نویسی ژنتیک استفاده از ساختار درختی می‌باشد. یک مثال برای نمایش یک تابع رتبه‌بندی با استفاده از ساختار درختی در شکل (2) نشان داده شده است. نمایش مبتنی بر درخت در برنامه‌نویسی ژنتیک باعث ایجاد سهولت در تجزیه و پیاده‌سازی توابع رتبه‌بندی می‌شود.



شکل 2 یک تابع رتبه‌بندی نمونه به شکل‌های درختی و رابطه‌ای

- مفید بودن برنامه‌نویسی ژنتیک در اکتشاف توابع و ساختارهای غیرخطی

برنامه‌نویسی ژنتیک به دلیل ویژگی ذاتی و منحصر بفردش در جستجو کردن فضای جستجو به صورت موازی، در بسیاری از مسائل طراحی و بهینه‌سازی مهندسی، مسایل زمان‌بندی و مسائل مربوط به اکتشاف ساختاری، در حالی که بسیاری از روش‌های سنتی بهینه‌سازی قابل اعمال نبوده و یا به خوبی عمل نمی‌کنند، با موفقیت بکار گرفته شده است[13, 15].

- یافتن راه‌حل های نزدیک به راه‌حل بهینه

به تجربه ثابت شده است که راه‌حل‌های یافته شده توسط برنامه‌نویسی ژنتیک معمولاً بهتر از راه‌حل‌های طراحی شده توسط انسان می باشند. در بسیاری از موارد، راه‌حل‌های یافته شده توسط برنامه‌نویسی ژنتیک بسیار نزدیک به بهترین راه‌حل ممکن می‌باشند[13].

ارائه یک چارچوب به منظور توسعه توابع رتبه بندی

همانطور که قبلاً بیان شد، موتورهای جستجو از یک تابع رتبه‌بندی برای مرتب‌سازی اسناد برحسب میزان مرتبط بودن آنها با یک پرس‌وجوی خاص استفاده می‌کنند. به این دلیل که بهترین تابع رتبه‌بندی برای یک پرس‌وجو (یا مجموعه‌ای از پرس‌وجوها) شناخته‌شده نمی‌باشد، مسأله توسعه توابع رتبه‌بندی به صورت یک مسأله جستجو در برنامه‌نویسی ژنتیک مدل‌سازی شده است.

در این بخش یک چارچوب که قادر است به طور خودکار به طراحی و توسعه توابع رتبه‌بندی مناسب بپردازد، ارائه شده است. تعدادی از مؤلفه‌های کلیدی بکار رفته در سیستم برنامه‌نویسی ژنتیک در جدول (1) تعریف شده‌اند. همچنین جزئیات این چارچوب که برمبنای برنامه‌نویسی ژنتیک می‌باشد در شکل (3) خلاصه شده است.

مؤلفه‌ها	معنی
پایانه‌ها	گره‌های برگی در ساختار درختی مانند tf, N و df در شکل (2)
توابع	گره‌های غیربرگی استفاده شده به منظور ترکیب گره‌های برگی. معمولاً عملگرهای عددی مانند: / ,- ,* ,+
تابع برازندگی	تابع هدفی که باید توسط سیستم بهینه‌سازی شود.
تکثیر	یک عملگر ژنتیکی که افراد با بهترین مقادیر برازندگی را مستقیماً به جمعیت نسل بعدی و بدون انجام شدن عملگرهای ژنتیکی دیگر مانند تبادل کپی می‌کند.
تبادل	یک عملگر ژنتیکی که زیردرخت‌های دو کروموزوم والد را برای ایجاد دو فرزند جدید تعویض می‌کند. هدف این عملگر, ایجاد تنوع در جمعیت و نیز بهبود شایستگی جمعیت می‌باشد.
جهش	یک عملگر ژنتیکی که به طور تصادفی یک زیردرخت را انتخاب و آن را با یک زیردرخت دیگر که به طور تصادفی ایجاد می‌شود, جایگزین می‌کند. هدف این عملگر, ایجاد تنوع در جمعیت و نیز اجتناب از گیرافتادن در راه‌حل‌های بهینه محلی می باشد.

جدول 1 مؤلفه‌های ضروری در برنامه‌نویسی ژنتیک

ورودی: یک مجموعه از پرس‌وجوها، بعلاوه یک مجموعه نمونه از اسناد آموزشی به همراه اطلاعاتی درباره میزان مرتبط بودن آنها با هر یک از پرس‌وجوهای مذکور .

خروجی: یک تابع رتبه‌بندی در زمینه مورد نظر

رویه:

1) اسناد آموزشی را به دو مجموعه تقسیم کن: مجموعه آموزشی و مجموعه اعتبارسنجی

2) یک جمعیت اولیه از توابع رتبه‌بندی تصادفی تولید کن.

3) مراحل زیر بر روی مجموعه آموزشی برای 30 نسل تکرار کن

الف) برای هر یک از افراد موجود در جمعیت، از تابع رتبه‌بندی به منظور امتیازدهی و رتبه‌بندی اسناد موجود در مجموعه آموزشی برای یک پرس‌وجو استفاده کن. این کار را برای تمام پرس‌وجوهای دیگر نیز انجام بده.

ب- برازندگی هر یک از توابع رتبه‌بندی را به‌ازای تمام پرس‌وجوها محاسبه کن

ج- با انجام اعمال ژنتیکی زیر بر روی جمعیت فعلی, یک جمعیت جدید از توابع رتبه‌بندی ایجاد کن

4) توابع رتبه‌بندی کاندیدای ثبت شده را بر روی اسناد مجموعه اعتبارسنجی اعمال کن و از میان آنها تابع رتبه‌بندی با بهترین کارآیی را به عنوان خروجی تولید کن.

شکل 3 چارچوب توسعه توابع رتبه‌بندی

در ادامه جزئیات پیاده‌سازی چارچوب فوق تشریح شده است.

- پایانه‌ها (ترمینال‌ها)

پایانه‌ها در مسأله توسعه توابع رتبه‌بندی، در حقیقت ویژگی‌های وزن‌دهی مربوط به آمارهای لغوی می‌باشند که در تابع رتبه‌بندی از آنها استفاده می‌شود. در این مقاله، پایانه‌ها پس از در نظر گرفتن برخی از توابع رتبه‌بندی مختلف موجود در این زمینه انتخاب شده‌اند. این پایانه‌ها در جدول (2) فهرست شده‌اند.

پایانه ها	معنای آماری
tf	تعداد دفعاتی که یک کلمه در یک سند ظاهر شده است
tf_max	حداکثر *tf* در یک سند
tf_avg	میانگین *tf* در یک سند
tf_doc_max	حداکثر *tf* در کل کلکسیون اسناد
df	تعداد اسناد متفاوتی که یک کلمه در آنها ظاهر شده است
df_max	حداکثر *df* برای تمامی کلمات یک پرس‌وجو
N	تعداد کل اسناد در تمامی کلکسیون اسناد متنی
Length	طول یک سند (برحسب کلمه)
Length_avg	طول متوسط یک سند در تمامی کلکسیون
n	تعداد کلمات متفاوت برای یک سند
R	یک عدد حقیقی تصادفی که توسط سیستم GP تولید می شود

جدول 2 پایانه های بکار رفته در سیستم GP

- توابع

توابع، عملیاتی هستند که برای ترکیب پایانه‌ها و یا زیردرخت‌ها به منظور تولید درخت‌های جدید اعمال می شوند. در این پیاده‌سازی، از توابع جمع، ضرب، تقسیم و لگاریتم استفاده شده است.

● تولید جمعیت اولیه

یک جمعیت، مجموعه‌ای از افراد می‌باشد که هر کدام بیانگر رابطه‌ای جهت وزن‌دهی به کلمات موجود در اسناد می‌باشند. درخت‌های واقع در جمعیت اولیه، همگی دارای این محدودیت هستند که حداکثر ارتفاع مجاز آنها برابر 4 می‌باشد و بوسیله روش *ramped half-and-half* تولید شده‌اند[13].

● توابع برازندگی

تابع برازندگی کارآیی یک تابع رتبه‌بندی را که بوسیله یک درخت بازنمایی شده است، در رتبه‌بندی اسناد اندازه‌گیری می‌کند. در این پیاده‌سازی دقت میانگین (*P_avg* که در جدول 3 تعریف شده است) در DCV[2] (که یک مقدار اختیاری می باشد و ما در این پیاده‌سازی از مقدار 100 برای DCV استفاده نموده‌ایم) سند بالایی به عنوان تابع برازندگی استفاده شده است. (برای چندین پرس‌وجو، تابع برازندگی برابر مقدار میانگین مقادیر *P_avg* برای تمامی پرس‌وجوها خواهد بود).

معیار	تعریف
P_avg	میانگین امتیازهای مربوط به دقت، که هر زمان که یک سند مرتبط جدید یافت شود محاسبه می‌شود و می‌تواند براساس اسناد مربوطه در کلکسیون نرمال‌سازی شود.
R_P	دقت زمانی که *TRel*(تعداد کل اسناد مرمرتبط در کلکسیون) سند بازیابی می‌شود.
T_Recall	نرخ فراخوانی وقتی که مثلاً 1000 سند بازیابی می‌شوند.

جدول 3 تعریف برخی از معیارهای کارآیی متداول در ارزیابی سیستم‌های بازیابی اطلاعات

● عملگرهای GP

تکثیر. این عملگر به تعداد *population_size * rate_reproduction* از بهترین درخت‌های موجود در جمعیت فعلی را بدون انجام هیچ‌گونه تغییری مستقیماً در جمعیت نسل بعدی کپی می‌کند. نرخ تکثیر یعنی پارامتر *rate_reproduction* برابر 0/1 و یا کمتر می باشد و *population_size* برابر با اندازه جمعیت می باشد.

تبادل. این عملگر با ایجاد درخت‌هایی که با والدین‌شان متفاوت می‌باشند، باعث ایجاد تنوع در جمعیت می‌شود. ما برای این عملگر از روشی به نام انتخاب دوره‌ای استفاده کرده‌ایم. انتخاب دوره‌ای این‌گونه عمل می‌کند که در ابتدا *k*(ما از مقدار 6 استفاده نموده ایم) درخت را با جایگذاری به طور تصادفی از جمعیت انتخاب می‌کند. سپس دو درخت بهتر در میان این *k* درخت، مبادرت به تعویض زیر درخت‌هایشان به طور تصادفی می‌نمایند.

● معیار توقف

ما در این پیاده‌سازی، فرآیند برنامه‌نویسی ژنتیک را به دو دلیل پس از تولید 30 نسل متوقف نموده‌ایم. دلیل اول این است که این فرآیند از لحاظ محاسباتی بسیار پر هزینه می‌باشد. دلیل دیگر اینکه آزمایشات اولیه نشان دادند که 30 نسل، یک دوره زمانی کافی برای توسعه توابعی با کارآیی بالا برای رتبه‌بندی می‌باشد.

[2] Document Cutoff Value

4- آزمایش ها

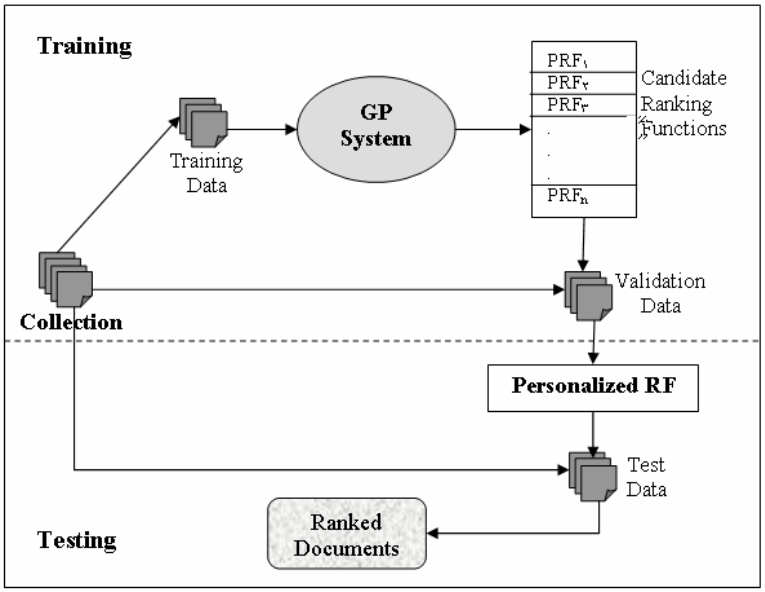
ما در این بخش به منظور ارزیابی توابع رتبه‌بندی توسعه یافته توسط سیستم GP از یک کلکسیون استاندارد به نام Med استفاده نموده‌ایم که پیش از این به طور گسترده‌ای توسط محققین در زمینه بازیابی اطلاعات بکار رفته است. این کلکسیون حاوی 1023 سند در زمینه پزشکی و 30 پرس‌وجوی انجام شده بر روی این مجموعه از اسناد می باشد. علاوه بر این، کلکسیون مذکور حاوی اطلاعاتی در مورد مرتبط بودن و یا نبودن هر یک از پرس‌وجوها با هر یک از اسناد موجود در کلکسیون می‌باشد. از طرفی، تمامی اسناد و پرس‌وجوها به صورت شاخص‌بندی شده در دسترس می‌باشند. ما در انجام آزمایش‌هایمان از 600 سند (اسناد با شماره‌های 1 تا 600) و 17 پرس‌وجوی انجام شده بر روی این اسناد استفاده نموده‌ایم.

ما در آزمایش‌های انجام شده، از یک‌چهارم اسناد انتخاب شده (150 سند) به عنوان «داده‌های آموزشی»، از یک‌چهارم دیگر به عنوان «داده‌های اعتبارسنجی» و از نیمه دیگر به عنوان «داده‌های آزمایشی» استفاده نموده‌ایم. در این آزمایش‌ها دلیل استفاده از داده‌های سه بخشی غلبه بر مشکل «جفت و جوری بیش از حد»[3] می‌باشد[12]. بنابراین انتظار می‌رود که این مجموعه داده‌های سه بخشی بتوانند اثر «آموزش بیش از حد» را کاهش دهند. از طرفی این استراتژی که ما در آزمایش‌هایمان بکار برده‌ایم، در آزمایش‌های مربوط به یادگیری ماشین یک استراتژی متداول می‌باشد[16].

بنابراین می‌توان طرز کار کلی سیستم یادگیری توابع رتبه‌بندی را با توجه به مطالب بالا به صورت زیر خلاصه نمود:

1) تقسیم اسناد موجود در کلکسیون به سه بخش شامل داده‌های آموزشی، داده‌های اعتبارسنجی و داده‌های آزمایشی؛
2) اعمال سیستم GP بر روی داده‌های آموزشی و به دست آوردن تعدادی تابع رتبه‌بندی نامزد؛
3) اعمال هر یک از توابع رتبه‌بندی بدست آمده در مرحله قبل بر روی مجموعه داده‌های اعتبارسنجی به منظور ارزیابی آن و ذخیره نمودن بهترین تابع به عنوان تابع رتبه‌بندی نهایی
4) اعمال تابع رتبه‌بندی نهایی بر روی مجموعه اسناد آزمایشی به منظور ارزیابی آن

در شکل (4) این مراحل نشان داده شده‌اند.



شکل 4 مراحل توسعه و ارزیابی توابع رتبه‌بندی شخصی

[3] Overfitting problem

ما در انجام همه آزمایش‌ها از معیارهای تعریف شده در جدول (4) استفاده نموده‌ایم. در بین سه معیار تعریف شده در جدول مذکور، معیار اولیه ما برای مقایسه بین سیستم‌های مختلف معیار دقت میانگین (P_avg) می‌باشد.

در ادامه به منظور بررسی و ارزیابی چارچوب ارائه شده، دو آزمایش با اهداف متفاوت ارائه شده است. در آزمایش اول، چارچوب ارائه شده در سطح پرس‌وجوها به طور جداگانه و با هدف توسعه توابع رتبه‌بندی شخصی‌سازی شده (جستجوی شخصی) و در آزمایش دوم، این چارچوب را برای گروهی از پرس‌وجوها با هم و به منظور توسعه توابع رتبه‌بندی عمومی (جستجوی همگانی) اعمال نموده‌ایم.

آزمایش اول: توسعه توابع رتبه‌بندی در جستجوی شخصی

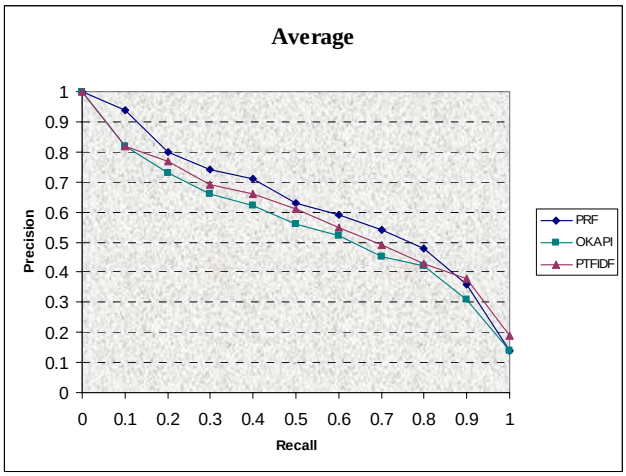
همانطور که اشاره شد، هدف از انجام این آزمایش توسعه بهترین تابع رتبه‌بندی ممکن برای هر یک از پرس‌وجوهای موجود در کلکسیون به طور جداگانه می‌باشد. با توجه به استراتژی بکار رفته (استفاده از داده‌های سه‌بخشی)، در این آزمایش، اسناد و پرس‌وجوها به گونه‌ای انتخاب شده‌اند که برای هر پرس‌وجو به تعداد کافی سند مرتبط در داده‌های سه‌بخشی وجود داشته باشد. به این معنا که ما پرس‌وجوهایی را که تعداد اسناد مرتبط کمی در داده‌های آموزشی، داده‌های اعتبارسنجی و یا داده‌های آزمایشی داشته‌اند حذف نموده‌ایم؛ زیرا این پرس‌وجوها نمی‌توانند نقش مفیدی در آزمایش و ارزیابی چارچوب ارائه شده داشته باشند. در جدول (4) شماره پرس‌وجوهای انتخاب شده از کلکسیون و همچنین تعداد اسناد مرتبط با آنها در هر یک از مجموعه داده‌ها آورده شده است.

Query No. in Collection	Number of relevant documents		
	Training data	Validation data	Test data
1	7	18	12
4	4	11	8
5	11	3	12
6	5	4	4
9	12	10	6
10	3	8	13
11	5	4	9
13	6	6	9
15	10	6	13

جدول 4 شماره پرس‌وجوهای بکار رفته در آزمایش اول به همراه تعداد اسناد مرتبط با آنها در هر یک از داده‌های سه‌بخشی

در این آزمایش ابتدا سیستم GP بر روی هر یک از پرس‌وجوها سه مرتبه اعمال شده و سپس بهترین تابع رتبه‌بندی بدست آمده برای هر پرس‌وجو به طور جداگانه ذخیره شده است. پس از آن هر یک از پرس‌وجوها با استفاده از تابع رتبه‌بندی به دست آمده برای آن بر روی کلکسیون اعمال شده و نتیجه ثبت شده است. علاوه بر این به منظور انجام مقایسه، هر یک از پرس‌وجوها توسط توابع رتبه‌بندی OKAPI و PTFIDF نیز بر روی کلکسیون اعمال شده‌اند[17]. در این آزمایش اندازه جمعیت برابر 200، احتمال تبادل برابر 0.9، نرخ تکثیر برابر 0.1 و تعداد نسل‌ها برابر 30 در نظر گرفته شده است.

به منظور جمع‌بندی نتایج به دست آمده در این آزمایش، در شکل (5) یک نمودار میانگین برای تمام پرس‌وجوها رسم شده است. برای رسم این نمودار، میانگین مقادیر دقت بازیابی برای تمام پرس‌وجوها در هر یک از سطوح فراخوانی به طور جداگانه محاسبه شده است. این کار برای هر سه سیستم به طور جداگانه انجام شده است و برای هر کدام یک نمودار به دست آمده است[17]. در اینجا به منظور انجام یک مقایسه کلی بین سیستم‌ها هر سه نمودار در شکل زیر رسم شده‌اند.



شکل 5 نمودار میانگین دقت توابع رتبه بندی PRF, Okapi و PTFIDF در سطوح مختلف فراخوانی

با دقت در شکل (5) می‌توان به سادگی دریافت که برای این مجموعه از پرس‌وجوها در کلکسیون مورد آزمایش، توابع یافته شده توسط سیستم GP به طور میانگین عملکرد بهتری (از نظر دو معیار دقت و فراخوانی) نسبت به توابع رتبه‌بندی Okapi و PTFDF از خود نشان داده‌اند. تفاوت عملکرد توابع توسعه‌یافته توسط سیستم GP نسبت به عملکرد Okapi کاملاً قابل ملاحظه می‌باشد (در حدود 10٪+) و این اختلاف به خوبی می‌تواند مقرون به صرفه بودن این سیستم را توجیه کند. اما اختلاف عملکرد در مقایسه با PTFIDF ناچیز می‌باشد (در حدود 4٪+). اما همین اختلاف ناچیز بسیار امید بخش می‌باشد زیرا این ایده را که می‌توان توابع رتبه‌بندی موجود بهتری نسبت به توابع موجود طراحی نمود، تقویت می‌کند. از طرفی سیستم برنامه‌نویسی ژنتیک دارای طبیعتی غیر قطعی می‌باشد و همچنین نتایج حاصل از آن تا حدی به مقادیر پارامترهای آن (مانند جمعیت اولیه, اندازه جمعیت, تعداد نسل‌ها, احتمال عملگر تبادل و ...) بستگی دارد و همه اینها بدان معنا می‌باشند که ممکن است بتوان با صرف وقت بیشتر نتایج بهتری بدست آورد.

در جدول (5) میانگین معیارهای تعریف شده در جدول (3) برای هر سه روش محاسبه شده است. بر اساس این جدول مشاهده می‌شود که توابع رتبه‌بندی توسعه یافته بوسیله GP در کل بهتر از دو تابع دیگر عمل می‌کنند.

Approach	P_avg	R_P	T_Recall
PTFIDF	57.03 %	56.97 %	88.73 %
OKAPI	52.63 %	50.74 %	81.42 %
Personalized RFs by GP	60.47 %	60.30 %	87.30 %

جدول 5 مقایسه هر سه روش براساس معیارهای تعریف شده در جدول (3) در جستجوی شخصی

آزمایش دوم: توسعه توابع رتبه‌بندی در جستجوی همگانی

همان‌طور که قبلاً ذکر شد، هدف از انجام این آزمایش توسعه یک تابع رتبه‌بندی همگانی برای گروهی از پرس‌وجوها می‌باشد. در این آزمایش از هر 17 پرس‌وجوی تعریف شده بر روی 600 سند اول موجود در کلکسیون استفاده شده است. در این آزمایش برای محاسبه میزان برازندگی یک تابع رتبه‌بندی، ابتدا دقت میانگین آن را به ازای هر یک از پرس‌وجوها محاسبه نموده‌ایم و سپس میانگین این مقادیر را به عنوان مقدار برازندگی آن در سیستم GP در نظر گرفته‌ایم.

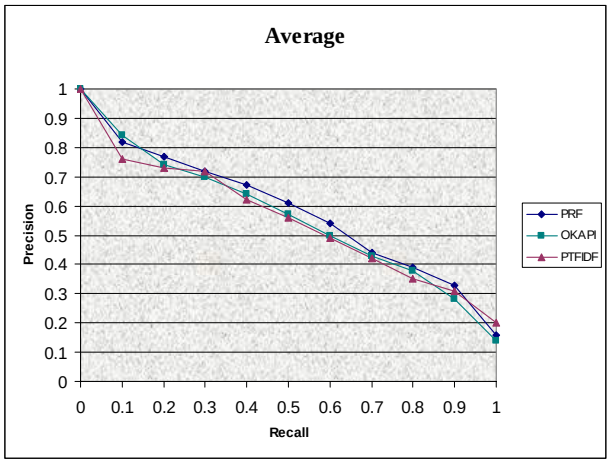
ما در این آزمایش سیستم GP را برای کل پرس‌وجوها با هم پنج مرتبه اعمال نموده‌ایم و سپس بهترین آنها را که در شکل (6) نشان داده‌ایم ثبت نموده‌ایم. لازم به ذکر است که تنظیمات سیستم GP در این آزمایش مانند آزمایش اول می‌باشد.

/(+(*(log(*tf*), log(*tf_doc_max*)), +(+(*df_max*, *df_max*),/(*tf_avg*, *N*))), /(+(log(*N*), +(*N*, *df*)), +(/(*tf_doc_max*, *N*), /(*tf_avg*, *N*))))

شکل 6 بهترین تابع رتبه‌بندی توسعه یافته توسط سیستم GP در آزمایش دوم

اگرچه ممکن تابع رتبه‌بندی نشان داده شده در شکل (6) کمی پیچیده به نظر برسد، در عمل آنچه که برای کاربران سیستم‌های بازیابی اطلاعات مهم است نتایج حاصل از جستجو در پاسخ به پرس‌وجوی مطرح شده از جانب وی می‌باشد و محاسبات انجام شده برای رتبه‌بندی اسناد اهمیت کمتری دارد.

ما در این آزمایش نیز مانند آزمایش اول دقت بازیابی هر سه روش (شامل بهترین تابع رتبه‌بندی توسعه یافته توسط سیستم GP, توابع رتبه‌بندی Okapi و PTFIDF) را در سطوح فراوانی مختلف برای هر یک از پرس‌وجوها محاسبه نموده‌ایم. در نهایت میانگین تمامی این نمودارها را برای هر سه روش محاسبه و رسم نموده‌ایم[17]. نمودار میانگین برای هر سه روش در شکل (7) نشان داده شده است.



شکل 7 نمودار میانگین دقت توابع رتبه بندی PRF, Okapi و PTFIDF در سطوح مختلف فراخوانی در جستجوی همگانی

در جدول (6) میانگین مقادیر سه معیار تعریف شده در جدول (3) نشان داده شده‌اند. در این جدول مشاهده می‌شود که تابع رتبه‌بندی توسعه‌یافته توسط سیستم GP، عملکرد بهتری را با توجه به این سه معیار از خود نشان می‌دهد. تنها در مورد معیار T_Recall تابع رتبه‌بندی PTFIDF کمی بهتر از تابع رتبه‌بندی توسعه یافته توسط سیستم GP عمل نموده است. نتایج نشان داده شده در این جدول به خوبی نشان می‌دهند که می‌توان توابع رتبه‌بندی بهتری را نسبت به توابع موجود توسعه داد و چارچوب ارائه شده می‌تواند در این زمینه بسیار مفید باشد.

Approach	P_avg	R_P	T_Recall
PTFIDF	54.02 %	49.94 %	86.49 %
OKAPI	50.41 %	51.87 %	79.55 %
Consensus RF by GP	57.07 %	54.66 %	84.68 %

جدول 6 مقایسه هر سه روش براساس معیارهای تعریف شده در جدول (3) برای مجموعه‌ای از پرس‌و جوها در جستجوی همگانی

5- نتیجه گیری و ارائه پیشنهادات

ما در این پروژه با بکارگیری ویژگی‌های وزن‌دهی متفاوتی که در اکثر سیستم‌های بازیابی اطلاعات استفاده می‌شوند (مانند *tf*, *idf*)، نشان دادیم که استفاده از یک ابزار یادگیری ماشین همانند برنامه‌نویسی ژنتیک می‌تواند به ما در خودکار نمودن فرآیند توسعه توابع رتبه‌بندی و طراحی توابع رتبه‌بندی بهتر در زمینه‌های گوناگون کمک کند. همانطور که قبلاً اشاره نمودیم، این فرآیند بدون استفاده از یک ابزار خودکار برای هر انسانی بسیار خسته کننده و مشکل می‌باشد. بویژه نشان دادیم که چارچوب ارائه شده می‌تواند به شکلی مؤثر هم در زمینه توسعه توابع رتبه‌بندی شخصی برای هر یک از پرس‌وجوها به طور جداگانه و هم در زمینه توسعه توابع رتبه‌بندی عمومی برای گروهی از پرس‌وجوها بکار گرفته شود. ما با مقایسه این رهیافت با دو نمونه از توابع رتبه‌بندی مطرح نشان دادیم که این رهیافت می‌تواند بر روش‌های موجود برتری داشته باشد.

نتایج بدست آمده در بکارگیری چارچوب ارائه شده با توجه به آزمایش‌های انجام گرفته نشان دادند که این روش می‌تواند بسیار امیدوار کننده باشد، اما هنوز برای بدست آوردن نتایج بهتر نیاز به تحقیقات بیشتری می‌باشد. در این بخش چند راهنمایی به منظور بهبود این چارچوب ارائه شده است[17].

- استفاده از ویژگی‌های وزن‌دهی ترکیبی مفید مانند $tf * \log(N / df)$ با استفاده از توابع تعریف شده خودکار[4] در GP
- استفاده از توابع برازندگی متفاوت در سیستم GP و بررسی و مقایسه تأثیر آنها در توابع رتبه‌بندی توسعه یافته
- استفاده از اطلاعات ساختاری موجود در اسناد وب (علاوه بر محتویات اسناد) و بکارگیری چارچوب ارائه شده به منظور جستجو در وب
- بکارگیری چارچوب ارائه شده در وظایف دیگر کاوش متن مانند طبقه‌بندی اسناد و خلاصه‌سازی

منابع

[1] Gordon, M., & Pathak, P. (1999). Finding information on the World Wide Web: the retrieval effectiveness of search engines. Information Processing and Management, 35(2), 141–180.

[2] Pitkow, J., Schutze, H., Cass, T., Cooley, R., Turnbull, D., Edmonds, A., Adar, E., & Breuel, T. (2002). Personalized search. Communications of the ACM, 45(9), 50–55.

[3] Salton, G. (1971). The SMART retrieval system: experiments in automatic document processing. New Jersey: Prentice Hall.

[4] Salton, G. (1989). Automatic text processing. Reading, MA: Addison-Wesley Publishing Co.

[5] Jones, W. P., & Furnas, G. W. (1987). Pictures of relevance: a geometric analysis of similarity measures. Journal of the American Society for Information Science, 38(6), 420–442.

[6] Harman, D. K. (1993). Overview of the first text retrieval conference (TREC-1). In D. K. Harman (Ed.), Proceedings of the first text retrieval conference. NIST Special Publication 500-207 (pp. 1–20).

[7] Harman, D. K. (1996). Overview of the fourth text retrieval conference (TREC-4). In D. Harman D. K. (Ed.), Proceedings of the fourth text retrieval conference. NIST Special Publication 500-236 (pp. 1–24).

[4] Automatically Defined Functions (ADT)

[8] Salton, G., & Buckley, C. (1988). Term weighting approaches in automatic text retrieval. Information Processing and Management, 24(5), 513–523.
[9] Singhal, A., Salton, G., Mitra, M., & Buckley, C. (1996). Document length normalization. Information Processing and Management, 32(5), 619–633.
[10] Salton, G., & McGill, M. J. (1983). Introduction to modern information retrieval. McGraw-Hill.
[11] Zobel, J., & Moffat, A. (1998). Exploring the similarity space. SIGIR Forum, 32(1), 18–34.
[12] Fan, W., Gordon, M.D., Pathak, P. (2003). Discovery of context-specific ranking functions for effective information retrieval using genetic programming, IEEE Transactions on knowledge and Data Engineering, in press.
[13] Koza, J. R. (1992). Genetic programming: on the programming of computers by means of natural selection. Cambridge, MA, USA: MIT Press.
[14] Langdon, W. B. (1998). Data structures and genetic programming: genetic programming + data structures¼automatic programming. Kluwer Publishing.
[15] Banzhaf, W., Nordin, P., Keller, R. E., & Francone, F. D. (1998). Genetic programming: an introduction—on the automatic evolution of computer programs and its applications. San Francisco, CA: Morgan Kaufmann Publishers.
[16] Mitchell, T. M. (1997). Machine learning. New York, NY: McGraw Hill.

[17] سید ناصر رضوی . مطالعه و پیاده‌سازی یک چارچوب به منظور توسعه خودکار توابع رتبه‌بندی در وب با استفاده از برنامه‌نویسی ژنتیک. پایان‌نامه برای دریافت درجه کارشناسی ارشد، دانشکده مهندسی کامپیوتر دانشگاه علم و صنعت ایران، 2005.