

بهبود روش های هوشمند عامل ترکیب اطلاعات

کاوه کاوسی

فارغ التحصیل کارشناسی ارشد گروه مهندسی برق و کامپیوتر - دانشکده فنی - دانشگاه تهران

بهزاد مشیری

دانشیار گروه مهندسی برق و کامپیوتر - قطب کنترل و پردازش هوشمند - دانشکده فنی - دانشگاه تهران

(تاریخ دریافت ۸۰/۷/۱۴، تاریخ دریافت روایت اصلاح شده ۸۱/۱۱/۹، تاریخ تصویب ۸۳/۳/۳)

چکیده

مسأله ی ترکیب اطلاعات که در سطحی فراتر از ترکیب داده ها قرار دارد، این امکان را فراهم می سازد تا از روی اطلاعاتی که هریک از زوایای مختلف نسبت به یک موضوع گردآوری می شوند و آمیخته با عدم قطعیت می باشند، نتیجه ای کاملتر و خالصتر حاصل گردد. امروزه نیاز به سیستم های هوشمندی که قادر باشند تا به صورت فراجویشگر شخصی کاربر اطلاعات مورد نیاز او را از میان حجم عظیم اطلاعات در دسترس تهیه کنند و در اختیار وی قرار دهند، بسیار محسوس است و اقدامات انجام شده در این زمینه هنوز نواقص فراوانی دارند. در همین راستا در این مقاله تلاش شده است تا معماری یک عامل هوشمند سفارشی، که قادر است اطلاعات مورد نیاز کاربر را با توجه به علائق وی از سرویس دهنده های اطلاعاتی مختلف بازیابی نماید، طراحی و پیاده سازی گردد. این عامل برای نیل به هدف خود از تکنیک های ترکیب اطلاعات بهره می گیرد. در این مقاله از عملگر کارآمد OWA که زیر مجموعه ای از خانواده ی بزرگ انتگرال های فازی است برای شکل دهی مدل عملیاتی عامل هوشمند استفاده شده است. تنظیم

در این عامل هر سه سطح ترکیب اطلاعات، یعنی ترکیب داده ها، ترکیب ویژگی ها، و ترکیب تصمیم ها وجود دارند. این عامل قادر است تا با خوشه بندی موضوعات، مدل رفتاری هر سرویس دهنده ی اطلاعاتی را نسبت به هر خوشه شناسایی نماید و در تصمیم گیری نهایی مورد استفاده قرار دهد

واژه های کلیدی : ترکیب اطلاعات، عامل هوشمند، قواعد فازی، ترکیب مجموعه ها، فراجویشگر

مقدمه

Excite, Open Text و... اشاره نمود. ولی هیچکدام از این ابزارها، آنقدر که مورد انتظار ماست، مفید و مؤثر نیستند و عملاً در بسیاری از موارد، نتیجه رضایت بخش نیست. مثلاً الگوریتم Altavista به لغات غیر معمول و خاص امتیاز بالاتر میدهد. برای مثال اگر شخصی albatross boat fishing را در جعبه جستجوی Altavista تایپ نماید، مستنداتی^۱ که در آنها تعداد زیاد لغت albatross، وجود دارد و یا این لغت در ابتدای متن و یا در تیتیر متن وجود دارد، امتیاز بالاتری دریافت می کنند. در این صورت این مستندات باعث کاهش امتیاز متن هایی می شوند که هر سه لغت را با هم دارند. بوضوح مشخص است که این بهترین روش جستجو نیست. بنابر این باید امکانات دیگری به این الگوریتم افزوده شود تا نتایج بهتری حاصل شود. افزایش انفجار آمیز اطلاعات فقط در وب مطرح نیست. بلکه در بسیاری از پایگاههای داده برخط^۲ و یا برون خط^۳ هم این مسأله به چشم

بی شک مسأله بازیابی مناسب اطلاعات از اینترنت و سایر پایگاه های داده با مقیاس بزرگ^۱ و بسیار بزرگ از مهمترین مشکلات موجود بر سر راه استفاده بهینه از منابع اطلاعاتی می باشد. امروزه وب بزرگترین پایگاه داده از مستندات و سایر فرمهای اطلاعات است و زمینه مناسبی برای ارزیابی تکنیک های مختلف بازیابی اطلاعات^۲ است. اغلب اطلاعات در وب بصورت ابرمتن^۳ سازماندهی شده اند و کاربران می توانند با انتخاب پیوندهای^۴ موجود در ابرمتن ها به اکتشاف اطلاعات جدید بپردازند و به ابرمتن های دیگر دسترسی پیدا کنند.

هرچه وب گسترده تر می شود نیاز به ابزارهای قوی جستجو در اینترنت بیشتر می شود. در حال حاضر سرویس های زیادی برای جستجوی وب^۵ موجود است. به عنوان نمونه میتوان به Google, Infoseek, Altavista،

می کند و سپس از کاربر می خواهد تا برای اهمیت حضور و یا عدم حضور آن کلمه یا عبارت در متن، وزنی در نظر بگیرد. این وزن به صورت مفهومی زبانی^{۱۳} می باشد. به این صورت که کاربر می تواند تعیین کند که حضور و یا عدم حضور یک لغت یا عبارت در متن بدون اهمیت، با اهمیت کم، با اهمیت متوسط، با اهمیت زیاد، و یا با اهمیت بسیار زیاد می باشد.

سپس عامل توسط واحدی به نام Query Generator به تعداد سرویس دهنده های اطلاعاتی^{۱۴} (مثلا جوینده های اینترنتی و یا پایگاه های داده) که قالب Query آنها را می شناسد، Query آماده می کند.

پس از ارسال Query ها هر سرویس دهنده لیستی از مستندات را باز می گرداند که این لیست با توجه به الگوریتم کاری همان سرویس دهنده بر اساس نزدیکی موضوع به Query رتبه بندی شده است. سپس عامل این لیست ها را بررسی می کند و موارد تکراری را حذف می کند و آنها را بر اساس الگوریتم های مناسب ترکیب مجموعه ها به صورتی با یکدیگر ترکیب می کند که یک لیست مرتب شده از مستندات ایجاد شود. از این بابت به هر سند بر اساس رتبه اش در این لیست یک امتیاز تعلق می گیرد [۱۳،۱۱].

سپس تک تک مستندات موجود در این لیست مورد پردازش قرار می گیرند و وضعیت آنها از لحاظ حضور و یا عدم حضور لغات و عباراتی که کاربر مشخص کرده است تعیین می گردد و بر اساس تعداد و پراکندگی حضور آنها دو امتیاز به هر سند تعلق می گیرد. در این مکانیزم، عملگر قدرتمند OWA^{۱۵} نقش اساسی بر عهده دارد.

همینطور هر بار که کاربر قصد استفاده از عامل را دارد مشخص می کند که موضوع مورد علاقه اش در کدام خوشه^{۱۶} ی موضوعی قرار می گیرد. منظور از یک خوشه موضوعی، یک دسته بندی منطقی از موضوعات مورد علاقه است. هر بار که کاربر جستجوی جدیدی را آغاز می کند می تواند یکی از خوشه های موجود را برگزیند و یا یک خوشه ی جدید ایجاد نماید. دسته بندی هایی که در حال حاضر در عامل پیش بینی شده اند در جدول (۱) آمده اند.

می خورد. انقلابی که تولید کامپیوترهای شخصی بوجود آورده است، دسترسی به کتابخانه های دیجیتال را بسیار آسان نموده است. ولی با این وجود کاربران استفاده کننده از موتورهای بازاریابی اطلاعات از پایگاه های داده، همانند کاربران اینترنت با موانع فراوانی مواجهند. از جمله آنها می توان به دشواری یافتن اطلاعات مفید، بدست آمدن مطالب بسیار حجیم در ارتباط با یک موضوع، بدست آمدن اطلاعات کم و ناقص، و دسته بندی و گزارش گیری نامناسب اشاره کرد.

اشاره به کار گذشتگان

در این قسمت به اختصار به کارهایی که تا کنون در زمینه استفاده عملی از تکنیک های ترکیب اطلاعات در محیط های نرم افزاری برای اتخاذ تصمیم های بهینه انجام شده است می پردازیم. در قسمتی از پروژه تهاب^۹ که مربوط به جمع آوری اطلاعات مورد نیاز مدیران بانکی بر اساس مدل سازی شخصیت آنها می باشد از تکنیک های ترکیب اطلاعات برای تصمیم گیری بهینه در این مورد استفاده شده است. قسمتی از کار انجام شده در [۱۷] ارائه شده است. در مورد استفاده از نظریه ترکیب اطلاعات در فراجویشگر ها تا کنون دو مورد گزارش شده است [۱۸،۱۹]. ولی در هیچیک از آنها این استفاده مبتنی بر پایه های تئوریک این نظریه نبوده است. در ضمن مساله ترکیب لیست ها و نیز مدل سازی رفتار جوینده های اینترنتی تاکنون از زاویه ای که در این مقاله به آنها پرداخته شده است مورد تعمق قرار نگرفته اند. ضمناً عملگر OWA تا کنون در هیچ جوینده و یا فراجویشگری به منظور امتیاز دهی به مستندات مورد اتفاده قرار نگرفته است.

خصوصیات عامل هوشمند سفارشی^{۱۰} ترکیب اطلاعات^{۱۱}

در این مقاله سعی ما بر این است تا با استفاده از تکنیک های ترکیب اطلاعات عاملی هوشمند ایجاد نماییم تا نقش یک فراجویشگر^{۱۲} سفارشی را برای کاربر ایفاء نماید. این عامل کلمات و عبارات مورد علاقه ی کاربر را که مایل است مطالب مرتبط با آن کلمات را مطالعه نماید دریافت

جدول ۱: لیست خوشه های اطلاعاتی که عامل هوشمند در حال حاضر سوابق آنها را برای شناسایی مدل رفتاری سرویس دهنده های اطلاعاتی نگهداری می کند .

شماره خوشه	نام خوشه
1	Data / Information Fusion
2	Context Sensitive Web Searching
3	Dempster Shafer Theory
4	Computer Science , Hardware
5	Computer Science , Software
6	Case Based Reasoning
7	Fuzzy Controllers
8	Mobile Robot Navigation
9	Intumescent Coatings
10	Paint & Resin Technology
11	Robotics
12	TBM model
13	Neural Networks
14	Neuro Fuzzy Systems
15	NTFS Filing System
16	Intelligent Agents & Ecommerce
17	WAP Technology and Applications

عامل با توجه به عملکرد سابق هر سرویس دهنده ی اطلاعاتی در هر دسته بندی، به هر سرویس دهنده یک امتیاز اختصاص داده است. زمانی که کاربر خوشه ی خاصی را انتخاب نمود، به هر سند براساس سرویس دهنده ای که آنرا بازیابی نموده است امتیاز همان سرویس دهنده را اختصاص می دهد. به این ترتیب برای هر سرویس دهنده اطلاعاتی ۴ امتیاز مختلف بدست می آید که در نهایت عامل باید امتیاز مستندات مختلف را بر اساس آنها محاسبه نماید و در اختیار کاربر قرار دهد. این کار از طریق ترکیب این امتیازات با روش های ترکیب اطلاعات انجام می گیرد و در آن برای هر امتیاز وزنی در نظر گرفته می شود که این وزن نشاندهنده اهمیت آن معیار در تصمیم گیری نهایی می باشد. همچنین در نهایت امتیاز هر سرویس دهنده در آن دسته بندی خاص تصحیح می گردد.

به خاطر گستردگی بحث، در این مقاله روش ها و الگوریتم های مورد استفاده در زمینه ی چگونگی امتیازات تخصیصی در ارتباط با حضور و یا عدم حضور لغات و عباراتی که کاربر مشخص کرده است و نیز بر اساس تعداد و پراکندگی حضور آنها، و نیز نحوه ی مدلسازی عامل هوشمند از سرویس دهنده های اطلاعاتی مورد بررسی

کامل قرار می گیرند.

نحوه ی استفاده از مکانیزم ترکیب مجموعه ها ، در مقاله دیگری، به تفصیل تشریح می گردد.

مکانیزم ترکیب اطلاعات با استفاده از عملگر میانگیر وزنی مرتب (OWA)

در ادامه، در جایی که امتیازات مربوط به تعداد و پراکندگی لغات را برای هر سند محاسبه می کنیم، برای هر سند به تعداد لغات و عبارات امتیاز بدست می آید که باید از روی آنها بوسیله ی عملگر OWA یک امتیاز برای هر سند استخراج نماییم که در امتیاز نهایی دخالت داده می شوند. همچنین در محاسبه ی امتیاز نهایی برای هر سند، ۴ معیار مختلفی را که برای آن سند به دست آمده اند، جهت اخذ تصمیم مناسب توسط این عملگر با یکدیگر ترکیب می نماییم. بنابر این در این قسمت به تعریف و ذکر خصوصیات عملگر OWA می پردازیم.

یکی از مسائل مهم ، جمع آوری توابع معیار و تشکیل تابع تصمیم کلی میباشد. از یک سو گاهی می خواهیم که همه معیارها رعایت شود و از سوی دیگر در بعضی موارد میخواهیم که حداقل یکی رعایت شود. این دو جهت به استفاده از عملگرهای and و or برای ترکیب توابع معیار بر می گردند. عملگر OWA نوعی جمع آوری انجام میدهد که به صورت کاملاً منعطف^{۱۷} در میان دو حالت ذکر شده قرار می گیرد و به همین جهت میتوان به آن عملگر orand گفت.

کاربرد عملگر OWA را در موضوعات مختلف شامل تصمیم سازی، سیستمهای خبره ، کنترل فازی، سیستمهای پایگاه داده ، فشرده سازی تصویر، و بسیاری زمینه های دیگر می توان یافت [۱،۲،۵،۸،۹،۱۱،۱۵] گستره کاربرد آن در موضوعات متعدد و متنوع، کارآیی و اهمیت عملگر OWA را اثبات می کند.

Ronald. R. Yager برای اولین بار این عملگر را معرفی نموده است [۱۳] . اخیراً در منابع مختلفی اثبات شده است که این عملگر، زیر مجموعه ای از طیف وسیع انتگرال های فازی می باشد [۳،۷] .

نگاشت F از $[0,1]^n$ به $[0,1]$ یک عملگر OWA با بعد n نامیده میشود اگر یک بردار وزنی W وابسته به F

بصورت

تاثیری ندارد. برای اثبات میتوانید به [۱۳] مراجعه کنید. توجه کنید که عملگر میانگین معمولی نیز نوعی OWA با وزنهای مساوی است:

$$w_i = \frac{1}{n} \rightarrow F_{\text{Avg}}(A) = \frac{1}{n} \sum_{i=1}^n a_i$$

همچنین با توجه به اینکه در بین وزنهای w_1, \dots, w_n ، w_1 همواره ضریب بزرگترین عنصر خواهد بود، پس اگر وزنهای را بصورت

$$\mathbf{W}^* = \begin{cases} w_1 = 1 \\ w_i = 0, i \neq 1 \end{cases}$$

بگیریم عملگر OWA همان عملگر max یا or خواهد بود:

$$F^*(A) = (\mathbf{W}^*)^T \mathbf{B} = \sum_{j=1}^n w_j b_j = b_1 = \max[a_1, \dots, a_n]$$

نیز اگر \mathcal{F} را انتخاب کنیم عملگر OWA تبدیل به عملگر min خواهد شد که همان and است:

$$F_*(A) = (\mathbf{W}_*)^T \mathbf{B} = \sum_{j=1}^n w_j b_j = b_n = \min[a_1, \dots, a_n]$$

در همان مرجع ثابت شده است که هر عملگر OWA دیگر بین این دو قرار خواهد گرفت:

$$F_*(A) \leq F(A) \leq F^*(A)$$

و بنابراین هر عملگر OWA، مابین max (کمترین or) و min (بیشترین and) قرار میگیرد. پس طبیعتاً اگر بعنوان and از نورم t دلخواه T و بعنوان or از نورم s دلخواه S استفاده کنیم، خواهیم داشت:

$$T(A) \leq F(A) \leq S(A)$$

ویژگی اساسی عملگر OWA، فرایند مرتب سازی آن است که آرگومانها را به وزنهای متناسب میکند. این یک فرایند غیر خطی است. OWA را میتوان بصورت ضرب

$$F_w(a_1, \dots, a_n) = \mathbf{W}^T \mathbf{B}$$

داخلی نشان داد که \mathbf{W} بردار وزنهای و \mathbf{B} بردار a_i های مرتب شده به ترتیب بزرگی است. بنابراین ملاحظه میشود که OWA از دو قسمت خطی و غیر خطی تشکیل شده است، ضرب داخلی بردارها بخش خطی آن و فرایند مرتب سازی بخش غیر خطی آن را تشکیل می دهد. در این مقاله، در سه مرحله از عملگر OWA برای هر سند استفاده می شود. بدین ترتیب اگر تعداد کل مستندات N

$$\mathbf{W} = [w_1, w_2, \dots, w_n]^T$$

وجود داشته باشد طوریکه

$$1) w_i \in [0, 1]$$

$$2) \sum_{i=1}^n w_i = 1$$

$$F(a_1, \dots, a_n) = \sum_{j=1}^n w_j b_j \quad \text{و}$$

که در آن، b_j عبارتست از j امین عنصر بزرگتر در مجموعه مقادیر a_1, a_2, \dots, a_n [۱۳]. عبارت دیگر اگر a_i ها را از لحاظ بزرگی مرتب کنیم طوریکه تبدیل به مجموعه مقادیر b_1, \dots, b_n شوند، b_1 بزرگترین آنها باشد و بعد b_2 از همه b_j ها غیر از b_1 بزرگتر باشد و ...

$$F(a_1, a_2, \dots, a_n) = \mathbf{W}^T \mathbf{B}$$

می توان نوشت \mathbf{B} بردار شامل b_j ها است که همان a_i های مرتب شده هستند طوریکه:

$$i > j \Rightarrow b_i \leq b_j$$

توجه به این نکته مهم ضروری است که بر خلاف عملگرهای وزنی معمولی که در آنها هر وزن به عنصر خاصی تعلق میگیرد، در اینجا وزن ها به مرتبه عناصر از لحاظ بزرگی نسبت داده میشوند و نه به خود عناصر. عبارت دیگر این طور نیست که هر مقداری که بجای a_1 در لیست آرگومانهای تابع F قرار بگیرد وزن مشخصی داشته باشد، بلکه بر حسب اینکه چه رتبه ای بین سایر آرگومانها دارد و وزنش متغیر خواهد بود. بنابراین ترتیب آمدن خود عناصر بعنوان آرگومان تابع F مهم نیست چون در هر صورت عملگر OWA خودش مجدداً ترتیب را عوض خواهد کرد. همچنین همواره داریم:

$$0 \leq F(a_1, a_2, \dots, a_n) \leq 1$$

عملگر OWA دارای خواص متعدد است، که از جمله آنها یکنوایی بودن و متقارن بودن است. خاصیت یکنوایی به این معنی است که اگر $A = [a_1, \dots, a_n]$ و $B = [b_1, \dots, b_n]$ را بعنوان آرگومانهای تابع F اعمال کنیم و برای هر i ، $a_i \geq b_i$ باشد، آنگاه $F(A) \geq F(B)$ خواهد بود. متقارن بودن نیز به این معنی است که جابجایی آرگومانها در نتیجه جمع آوری

توجه نمود آن است که منظور ما از کلمه و عبارت یک n-gram است.

مفهوم n-gram به دنباله ای از لغات اطلاق می شود که در آن n نشاندهنده تعداد لغات حاضر در این دنباله می باشد [۶].

در پیاده سازی این عامل، تک کلمه ها (1-gram)، زوج کلمه ها (2-gram)، و عبارات سه تایی (3-gram) از کاربرد دریافت می گردد و مکان آنها در متن شناسایی می گردند. کاربرد می تواند برای حضور یا عدم حضور هر یک از این n-gram ها در متن امتیازی از مجموعه { کم، متوسط، زیاد، بسیار زیاد } در نظر بگیرد. شکل (۱) بخشی از معماری این عامل هوشمند را نشان می دهد.

واسط کاربر در این بخش وظیفه دریافت علائق کاربر و نیز ارائه نتایج به او را برعهده دارد. همانطور که مشخص است علائق کاربر برای ایجاد پرس و جو^{۱۹} به واحد تولید کننده پرس و جو در شکل (۲) ارسال می شوند و یک لیست رتبه بندی شده از شکل (۲) به عنوان نتیجه ترکیب لیست ها به شکل (۱) وارد می شود. مکانیزم و الگوریتم ترکیب لیست ها در این مقاله بررسی نشده است. مستندات حاصل از شکل (۲) به واحد تجزیه کننده^{۲۰} در شکل (۱) ارسال می شوند و در آنجا تعداد و موقعیت n-gram های مورد علاقه کاربر از این مستندات استخراج می شوند.

الگوریتمی که عامل هوشمند برای امتیاز دهی به مستندات از بابت حضور و یا عدم حضور n-gram ها مورد استفاده قرار می دهد در این بخش و امتیاز ناشی از پراکندگی n-gram ها در متن در بخش بعد توضیح داده می شوند. در نهایت این امتیاز ها برای هر مستند به واحد ترکیب امتیازات نهایی در شکل (۳) فرستاده می شود و در آنجا با دو امتیاز دیگر (امتیاز ناشی از الگوریتم ترکیب لیست ها و امتیاز ناشی از مدل رفتاری جویشگری که مستند را بازیابی کرده است) ترکیب می شود.

پس از پردازش مستندات، تعداد هریک از n-gram ها در متن مشخص می شود و با توجه به امتیازی که کاربر برای آنها در نظر گرفته است به هر متن به تعداد n-gram ها امتیاز تعلق می گیرد.

باشد، تعداد استفاده از عملگر OWA برابر $3N$ خواهد بود.

در ادامه و در بخش های مربوطه، نحوه ی استفاده از عملگر OWA را به شیوه ای نوین، بیان می کنیم و نشان می دهیم که چگونه می توان با استفاده از قوانین فازی، یک تابع یکنوای افزایشی (RIM)^{۱۸} مناسب برای محاسبه وزنه های OWA پیدا نمود.

معماری عامل هوشمند ترکیب اطلاعات امتیاز تعداد کلمات و عبارات مورد علاقه کاربر در مستندات

یکی از مهمترین پارامترها در امتیاز دهی به مستندات، نحوه ی حضور و یا عدم حضور کلمات و عبارات در مستندات، و چگونگی پراکندگی آنها در متن می باشد. مثلاً فرکانس تکرار یک کلمه در متن، تا حدود زیادی می تواند ارتباط آن متن را با آن کلمه نشان دهد. البته بحث در این مورد، در حیطه ی همبافت شناسی متون به صورت وسیعی تعقیب شده است. ولی نتایج عملی آن در جوینده ها و فراجویشگر های امروزی کمتر مورد استفاده قرار گرفته است.

نکته ی دیگری که باید به آن توجه داشت، این است که در بسیاری از پیاده سازی ها، فقط تعداد کلماتی که کاربر به حضور آنها در متن علاقه دارد در امتیاز دهی مورد استفاده قرار می گیرند که مطلوب نیست. زیرا کار بر ممکن است به پالایش بیشتری در مستندات احتیاج داشته باشد. یعنی عدم حضور لغات و عباراتی که کاربر حضور آنها در متن را نمی پسندد، نیز به اندازه حضور عبارات مورد علاقه اهمیت دارد. بنابر این حضور این لغات در متن باید امتیاز منفی در پی داشته باشد تا پالایش بیشتری بر روی لیست نهایی صورت گیرد.

یکی از مهمترین پارامترهایی که می تواند میزان ارتباط یک سند را با علائق کاربر مشخص نماید، تعداد کلمات و عبارات مورد نظر کاربر در آن متن و نیز عدم حضور کلماتی است که کاربر تأکید دارد که در متن حضور نداشته باشند. این فاکتور به قدری مهم است که بدون در نظر گرفتن آن امکان امتیاز دهی مناسب به نحو چشمگیری کاهش می یابد. نکته ی مهمی که باید به آن

که در آن داریم:

$$\alpha_j \in \{-1, -0.75, -0.5, -0.25, 0, 0.25, 0.5, 0.75, 1\}$$

• امتیاز مطلق سند k ام نسبت به n -gram

$$\varepsilon_{kj} = \text{زام}$$

• امتیاز نسبی سند k ام نسبت به n -gram

$$\theta_{kj} = \text{زام}$$

$$0 \leq \theta_{kj} \leq 1$$

در تمام تعاریف فوق داریم:

$$j = 1, \dots, T; k = 1, \dots, N$$

در این صورت امتیاز مطلق سند k ام نسبت به n -gram

زام را به صورت زیر تعریف می کنیم:

$$\varepsilon_{kj} = p_{kj}$$

حال برای اینکه مرجع مقایسه ی مناسبی در دست داشته

باشیم، امتیاز نسبی سند k ام نسبت به n -gram j ام را

به صورت زیر تعریف می کنیم:

$$\theta_{kj} = \frac{\varepsilon_{kj}}{\max_{l=1}^N (\varepsilon_{lj})}$$

یعنی امتیاز نسبی هر سند نسبت به تعداد یک n -gram

با تقسیم امتیاز مطلق به بیشترین امتیازی که یک سند

نسبت به تعداد همان n -gram دارد محاسبه می شود.

به این ترتیب برای هر سند T امتیاز مختلف بدست

می آید. حال هدف ما این است که بتوانیم از ترکیب این

T امتیاز برای هر سند، یک امتیاز بدست آوریم، که این

امتیاز در تصمیم گیری نهایی مؤثر خواهد بود.

برای این منظور در این مرحله از عملگر OWA استفاده

می کنیم. برای ترکیب این معیارها از OWA با بردار

وزنی

$$W = [w_1, w_2, \dots, w_T]^T$$

استفاده می کنیم. تابع تصمیم نهایی با استفاده از OWA

بصورت زیر داده می شود:

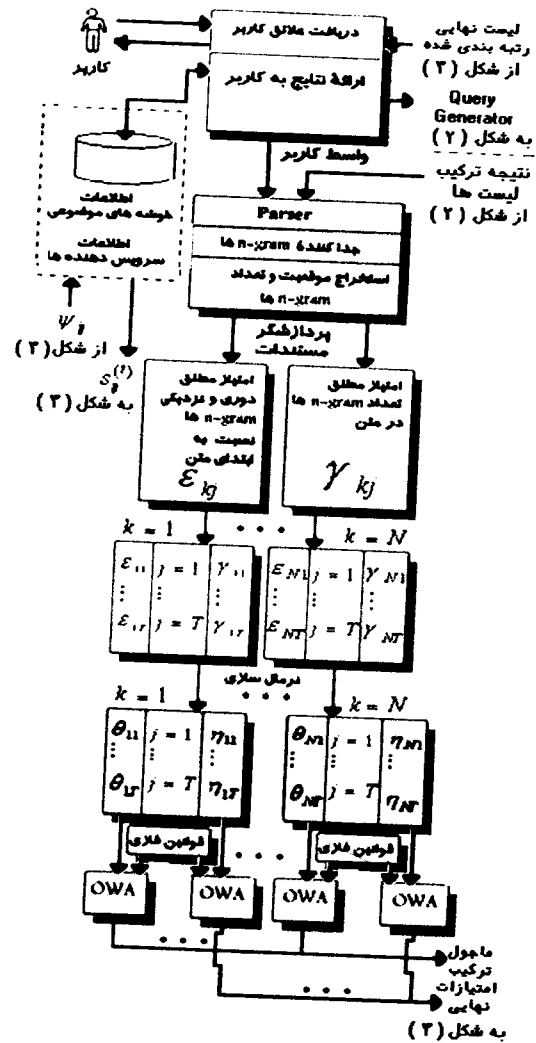
$$U_k = \sum_{j=1}^T w_j \varepsilon_{kj}$$

پارامتر U_k از میان پارامترهای فوق در تصمیم گیری

نهایی مورد استفاده قرار خواهد گرفت که در آن ε_{kj} ها،

$\alpha_j \theta_{kj}$ ها هستند که به ترتیب بزرگی قدر مطلقشان

مرتب شده اند.



شکل ۱: دریافت علائق کاربر، استخراج ویژگی های متن و ترکیب امتیازات ناشی از آنها.

در نهایت این امتیازها با استفاده از یک عملگر OWA

هم ترکیب می شوند و بدین ترتیب به هر متن یک امتیاز

تعلق می گیرد که در رتبه بندی نهایی مستندات مورد

استفاده قرار می گیرد.

در ادامه به تفصیل به توضیح الگوریتم می پردازیم. برای

این منظور تعاریف زیر را ارائه می نماییم:

• تعداد کل مستندات $N =$

• تعداد n -gram های مورد علاقه $T =$

• تعداد زامین n -gram مورد علاقه در

سند k ام $p_{kj} =$

• اهمیت حضور یا عدم حضور n -gram j ام

در سند $\alpha_j =$

$$w_1 = \left(\frac{1}{5}\right)^a; w_2 = \left(\frac{2}{5}\right)^a - \left(\frac{1}{5}\right)^a; w_3 = \left(\frac{3}{5}\right)^a - \left(\frac{2}{5}\right)^a;$$

$$w_4 = \left(\frac{4}{5}\right)^a - \left(\frac{3}{5}\right)^a; w_5 = 1^a - \left(\frac{4}{5}\right)^a;$$

توجه داریم که شرط $\sum_{i=1}^T w_i = 1$ برقرار است.
پس از انجام محاسبات خواهیم داشت:

$$a < \frac{\ln\left(\frac{0.65}{0.8}\right)}{\ln\left(\frac{4}{5}\right)} \Rightarrow a < 0.93052$$

برای مقادیر $\begin{cases} m' = 60 \\ y' = 0.4 \\ x' = 0.75 \end{cases}$ و قاعده ی دوم ذکر شده در بالا خواهیم داشت:

$$(w_1 + w_2) \times 1 + (w_3 + w_4 + w_5) \times 0.4 \leq 0.75$$

پس از انجام محاسبات خواهیم داشت:

$$a > \frac{\ln\left(\frac{0.35}{0.6}\right)}{\ln\left(\frac{2}{5}\right)} \Rightarrow a > 0.58824$$

بنا بر این اگر $a \in [0.58824, 0.93052]$ باشد دو شرط فوق ارضاء خواهند شد. با انتخاب $a = 0.75$ (که عملگر OWA را به یک عملگر Or-like تبدیل می کند) نتیجه نهایی چنین خواهد بود:

$$w_1 = 0.29907; w_2 = 0.20390; w_3 = 0.17876;$$

$$w_4 = 0.16416; w_5 = 0.15410;$$

و بنا بر این:

$$U_7 = \sum_{j=1}^5 w_j \zeta_{7j} = 0.57307$$

که این مقدار به عنوان امتیاز تعداد کلمات کلیدی در متن سند هفتم، در امتیاز دهی نهایی این سند مورد استفاده قرار می گیرد.

در این مرحله از چندینگر^{۲۱} $Q_a(r) = r^a$ برای محاسبه $a \geq 0$

ی وزن ها استفاده شده است [۱۴]. بنابر این با توجه به شرایطی که وزن های OWA باید داشته باشند، می توان این وزن ها را از رابطه ی زیر بدست آورد:

$$w_j = Q\left(\frac{j}{T}\right) - Q\left(\frac{j-1}{T}\right); \quad j = 1, 2, \dots, T$$

به سادگی دیده می شود که $\sum_{j=1}^T w_j = 1$ می باشد.

اینک با دو قاعده ی ساده ی فازی^{۲۲} می توان محدوده مناسب و دقیقتری برای a بدست آورد [۱۶].

- قاعده ی اول: اگر بیشتر از m درصد امتیازهای تخصیصی به یک سند بزرگتر از y باشند، امتیاز نهایی تعداد لغات در آن سند باید بزرگتر از x باشد.
- قاعده ی دوم: اگر بیشتر از m' درصد امتیازهای تخصیصی به یک سند کوچکتر از y' باشند، امتیاز نهایی تعداد لغات در آن سند باید کوچکتر از x' باشد.

$$\text{مثلا مقادیر } \begin{cases} m = 80 \\ y = 0.8 \\ x = 0.65 \end{cases} \text{ برای قاعده ی اول}$$

$$\text{و برای قاعده ی دوم مقادیر نسبتا مناسبی } \begin{cases} m' = 60 \\ y' = 0.4 \\ x' = 0.75 \end{cases} \text{ می باشند.}$$

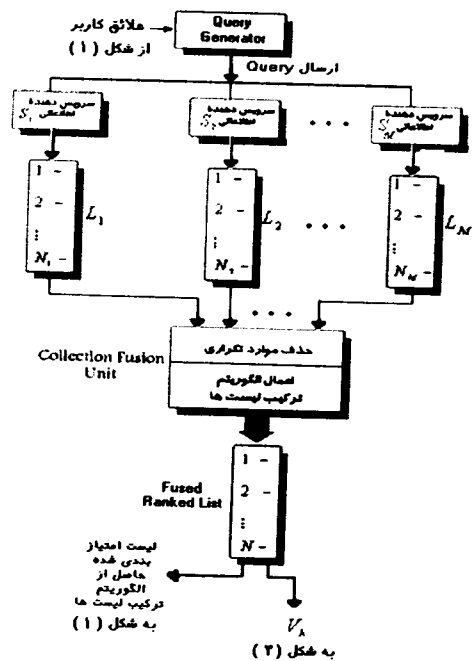
مثال عملی زیر نحوه ی انجام این عملیات را بهتر مشخص می کند:

مثال: فرض کنید که $n\text{-gram}$ ($T = 5$) به عنوان کلمات کلیدی در مستندات مورد جستجو می باشند و اکنون سند هفتم در لیست مرتب شده مورد پردازش می باشد. اطلاعات حاصل از پردازش مطابق جدول (۲) می باشد. از طرفی با محاسبه ی وزن های عملگر OWA داریم:

جدول ۲: اطلاعات مربوط به تعداد n-gram ها در یک سند مرتبط با تکنولوژی CORBA.

ζ_{γ_j}	$\alpha_j \theta_{\gamma_j}$	θ_{γ_j}	$Max_{l=1}^N(\epsilon_{lj})$	$\epsilon_{\gamma_j} = p_{\gamma_j}$	p_{γ_j}	α_j	کلمه ی کلیدی j ام	j
1	1	1	84	84	84	بسیار زیاد	CORBA	1
0.510000	0.333333	0.333333	6	2	2	بسیار زیاد	JAVA RMI	2
0.419642	0.419642	0.839285	56	47	47	متوسط	BANKSERVER	3
0.333333	0.510000	0.680000	25	17	17	زیاد	IDL	4
0.261364	0.261364	0.348485	66	23	23	زیاد	ORB	5

گردد. در این صورت با این معیار امتیاز کمتری به متن اول نسبت به متن دوم تعلق خواهد گرفت که نا مطلوب است. ولی با این وجود به خاطر آنکه در بسیاری از موارد پراکندگی یک n-gram در متن ارتباط مستقیمی با میزان ارتباط آن متن و n-gram با یکدیگر دارد، نمی توان از اهمیت آن صرفنظر نمود.



شکل ۲: ارسال Query به سرویس دهنده های اطلاعاتی و دریافت نتایج و اعمال الگوریتم ترکیب لیست ها به نتایج که در اثر آن یک لیست رتبه بندی شده ایجاد می گردد.

برای مدل نمودن این معیار، برای یک n-gram خاص، تعاریف زیر را ارائه می نمایم:

- تعداد کل مستندات $N =$
- تعداد n-gram های مورد علاقه $T =$

امتیاز پراکندگی n-gram های مورد علاقه کاربر (میزان دوری و نزدیکی یک کلمه یا عبارت به ابتدای متن)

مسأله ی ظریف دیگری که در ابزار های جستجوی فعلی کمتر به آن پرداخته شده است، این است که تعداد n-gram ها به تنهایی فاکتور کاملی برای تصمیم گیری در مورد ارتباط متن با آن لغات نمی باشد. بلکه نحوه توزیع کلمات و عبارات در متن، و نیز حجم متن از اهمیت زیادی برخوردار هستند. البته این بحث بسیار پیچیده است. مثلاً فرض کنید که تعداد n-gram های مورد علاقه در متنی که کلاً ۱۰۰۰ لغت دارد، برابر تعداد کلمات مورد علاقه در متنی است که ۲۰۰۰ لغت دارد. در این صورت نمی توان به صورت حتمی ادعا کرد که ارتباط متن اول با کلمات مورد علاقه بیش از ارتباط متن دوم می باشد. ولی از طرفی نیز نمی توان تأثیر آنها انکار نمود. یکی از پارامترهایی که جوینده هایی مانند Altavista، در امتیاز دهی به مستندات اهمیت زیادی به آن می دهند میزان دوری و نزدیکی یک n-gram نسبت به ابتدای متن می باشد [۴]. بنا بر این، این پارامتر باید به گونه ای در تصمیم گیری نهایی و امتیاز نهایی هر متن دخیل باشد.

البته باید توجه داشت که این پارامتر نیز مانند بسیاری دیگر از فاکتورهای دخیل در تصمیم گیری، فقط تا حدودی می تواند بیانگر میزان ارتباط یک سند با n-gram مورد نظر باشد.

مثلاً ممکن است از یک سو قسمتی از بخش های آخر یک سند طولانی ارتباط عمیقی با کلمه ی مورد نظر کاربر داشته باشد و از سوی دیگر همان کلمه چندین بار در ابتدای یک متن که ارتباط ضعیفی با آن کلمه دارد ظاهر

$$\gamma_{kj} = \sum_{d'_{ikj} \in D'_{kj}} (n' - d'_{ikj})$$

اما از آنجا که در هر سری از مستندات که مورد پردازش قرار می گیرند، باید یکی از آنها به عنوان بهترین سند ملاک ارزیابی سایرین قرار گیرد، بنابراین امتیاز نسبی سندی که بالاترین امتیاز را دارد به ۱ تبدیل می کنیم و سپس از روی آن باید امتیاز سایر مستندات را نسبت به پراکندگی این n-gram محاسبه کنیم.

برای این منظور افزودن مقدار تفاوت ۱ با بالاترین امتیاز، به همه ی مستندات کار صحیحی نیست. زیرا در این صورت ممکن است سندی که امتیاز ۰ دارد به امتیازات نسبتا بالایی برسد. عملگر ضرب عملگر مناسبی برای این منظور است.

$$\eta_{kj} = \frac{\gamma_{kj}}{\text{Max}_{l=1}^N(\gamma_{lj})} = \frac{\sum_{d'_{ikj} \in D'_{kj}} (n' - d'_{ikj})}{\text{Max}_{l=1}^N(\gamma_{lj})}; j = 1, \dots, T$$

همانطور که ملاحظه می شود امتیاز مطلق هر سند نسبت به یک n-gram به صورت γ_{kj} محاسبه می شود و سپس امتیاز نسبی هر سند نسبت به دوری و نزدیکی یک n-gram به ابتدای متن با تقسیم امتیاز مطلق به بیشترین امتیازی که یک سند نسبت به همان n-gram دارد محاسبه می شود.

به این ترتیب برای هر سند T امتیاز مختلف بدست می آید. حال هدف ما این است که بتوانیم از ترکیب این T امتیاز برای هر سند، یک امتیاز بدست آوریم که این امتیاز در تصمیم گیری نهایی مؤثر خواهد بود.

برای این منظور در این مرحله از عملگر OWA استفاده می کنیم. برای ترکیب این معیار ها از OWA با بردار وزنی

$$W = [w_1, w_2, \dots, w_T]^T$$

استفاده می کنیم. تابع تصمیم نهایی با استفاده از OWA

$$P_k = \sum_{j=1}^T w_j \omega_{kj} \quad \text{بصورت زیر داده می شود:}$$

پارامتر P_k از میان پارامتر های فوق در تصمیم گیری نهایی مورد استفاده قرار خواهد گرفت که در آن ω_{kj} ها، $\alpha_j \eta_{kj}$ ها هستند که به ترتیب بزرگی قدر مطلقشان مرتب شده اند.

• تعداد j امین n-gram مورد علاقه در سند k ام p_{kj}

• مکان i امین باری که n-gram j ام در

سند k ظاهر می شود (نسبت به اول متن) d_{ikj}

• تعداد کل کلمات موجود در سند k ام $n_k =$

• اهمیت حضور یا عدم حضور n-gram j ام در سند $\alpha_j =$

که در آن داریم:

$$\alpha_j \in \{-1, -0.75, -0.5, -0.25, 0, 0.25, 0.5, 0.75, 1\}$$

• مجموعه مکانهای n-gram j ام در سند $D_{kj} =$ ام k

$$D_{kj} = \{d_{1kj}, d_{2kj}, \dots, d_{p_{kj}kj}\}$$

یعنی $D_{kj} = \{d_{1kj}, d_{2kj}, \dots, d_{p_{kj}kj}\}$

$$D'_{kj} = \{d'_{1kj}, d'_{2kj}, \dots, d'_{p_{kj}kj}\}; d'_{ikj} = d_{ikj} \cdot \frac{\text{Max}_{r=1}^N(n_r)}{n_k}$$

• امتیاز مطلق سند k ام نسبت به n-

$$\gamma_{kj} = \text{gram } j \text{ ام}$$

• امتیاز نسبی سند k ام نسبت به n-gram

$$\eta_{kj} = \text{ } j \text{ ام}$$

$$0 \leq \eta_{kj} \leq 1$$

در تمام تعاریف فوق داریم: $k = 1, \dots, N$;

$$j = 1, \dots, T; i = 1, \dots, p_{kj}$$

در ابتدا باید مستندات را اصطلاحا هم حجم نماییم. یعنی برای آنکه متریک مناسبی برای محاسبات داشته باشیم فرض می کنیم که همه ی مستندات تعداد کلمات مساوی دارند. سپس مکان جدید کلمات و عبارات در هر سند را بدست می آوریم:

$$\text{با تعریف مقدار } n' = \text{Max}_{r=1}^N(n_r) \text{ به منظور هم حجم}$$

سازی مستندات، مکان جدید n-gram ها در مستندات به صورت زیر می باشند:

$$D'_{kj} = \{d'_{1kj}, d'_{2kj}, \dots, d'_{p_{kj}kj}\}; d'_{ikj} = d_{ikj} \cdot \frac{\text{Max}_{r=1}^N(n_r)}{n_k}$$

بنا بر این امتیاز مطلق سند k ام نسبت به j امین

n-gram به شکل زیر محاسبه می گردد:

جدول ۳: اطلاعات مربوط به پراکندگی تعداد n-gram ها در یک سند مرتبط با تکنولوژی CORBA.

ω_{γ_j}	$\alpha_j \eta_{\gamma_j}$	η_{γ_j}	$\text{Max}_{i=1}^N(\gamma_{ij})$	$\gamma_{\gamma_j} = \sum_{d_{\gamma_j} \in D_{\gamma_j}} (n' - d_{\gamma_j})$	p_{γ_j}	α_j	ام j کلمه ی کلیدی	j
1	1	1	641512	641512	84	بسیار زیاد	CORBA	1
0.470238	0.157680	0.157680	123218	19429	2	بسیار زیاد	JAVA RMI	2
0.409214	0.470238	0.940467	218300	205304	47	متوسط	BANKSERVER	3
0.317350	0.409214	0.545619	244817	133577	17	زیاد	IDL	4
0.157680	0.317350	0.423133	269851	114183	23	زیاد	ORB	5

• امتیاز تعداد n -gram های موجود در سند k ام
 $U_k =$

• امتیاز پراکندگی n -gram ها در سند k ام $P_k =$

• امتیاز سرویس دهنده ی بازیابی کننده ی سند k ام
 در خوشه ی j ام $Q_k =$

در اینجا نیز به نظر می رسد که استفاده از عملگر OWA برای ترکیب این اطلاعات مناسب می باشد. مشابه

قسمت های قبل در اینجا نیز برای محاسبه بهینه وزن های OWA وضع چند قاعده ی فازی راه گشا می باشد.

روشی که ما در این سیستم از آن استفاده کرده ایم این است که از ترکیب هر چهار معیار استفاده کنیم.

برای این کار معیار های Q_k, P_k, U_k, V_k را با استفاده از OWA با بردار وزنی

$$W = [w_1, w_2, w_3, w_4]^T$$

مطابق بلوک دیاگرام شکل (۳) با هم ترکیب می کنیم.

واضح است که اهمیت همه ی معیار ها در تصمیم گیری یکسان نیست. β_i ها ضرایب اهمیت هستند که به هر

یک از معیار ها تعلق می گیرند و در طول زمان تغییر می کنند. بطور مثال، امتیاز ناشی از سرویس دهنده های

اطلاعاتی در ابتدا که هنوز مدل رفتاری سرویس دهنده ها ناقص است، زیاد معتبر نیست و بنابراین اهمیت کمتری

خواهد داشت و بمرور این اهمیت بیشتر خواهد شد.

برای آموزش β_i ها، می توان استراتژی های مختلفی در پیش گرفت. سعی ما بر این بوده است که حتی الامکان،

کمتر کاربر را درگیر پاسخگویی به سؤالات عامل هوشمند نماییم.

در این مرحله نیز از چند دیگر $Q_a(r) = r^a; a \geq 0$ استفاده شده است. در اینجا نیز مانند آنچه که در قست قبل ذکر شد با دو قاعده ی فازی مشابه قوانینی که در بخش مذکور آمده اند می توان مقدار مناسب a برای محاسبه ی وزنه ای OWA را بدست آورد.

در ادامه ی مثال قبلی، اینک به مثال زیر توجه می کنیم.

مثال: در مثال قبل پس از پردازش مشخص می شود که تعداد کل کلمات موجود در این سند $n_7 = 8781$ و

$n' = \text{Max}_{r=1}^N(n_r) = 10021$ ، و سایر اطلاعات حاصل از پردازش به شرح جدول (۳) می باشند.

$$\begin{cases} m = 80 \\ y = 0.8 \\ x = 0.65 \end{cases}$$

و انجام محاسبات مشابه مثال قبل و انتخاب $a = 0.75$ خواهیم داشت:

$$w_1 = 0.29907; w_2 = 0.20390; w_3 = 0.17876;$$

$$w_4 = 0.16416; w_5 = 0.15410;$$

$$P_7 = \sum_{j=1}^5 w_j \omega_{\gamma_j} = 0.5445$$

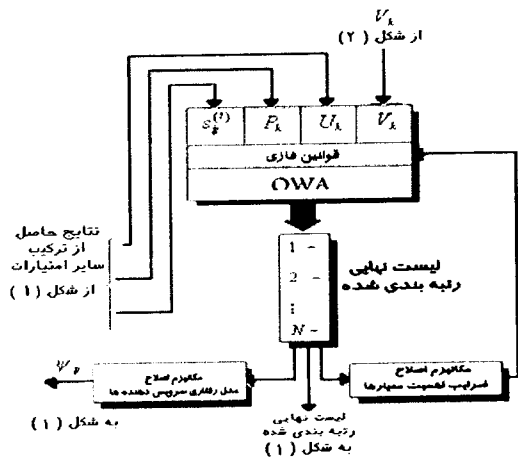
و بنا بر این:

که این مقدار به عنوان امتیاز توزیع کلمات کلیدی در متن سند هفتم، در امتیاز دهی نهایی این سند مورد استفاده قرار می گیرد.

محاسبه ی امتیاز نهایی

همانطور که دیدیم، برای هر سند ۴ امتیاز به شرح زیر بدست می آید:

• امتیاز ناشی از ترکیب لیست های رتبه بندی شده برای سند k ام $V_k =$



بنابر این روشی که عامل در این مورد به کار می بندد، اصلاح وزنها از روی نتایج نهایی می باشد. البته کاربرد فقط نظر خود را اعلام می کند که آیا از نتایج بدست آمده راضی بوده است یا خیر. در بخش بعدی این موضوع را بیشتر توضیح می دهیم.

تابع تصمیم نهایی با استفاده از OWA بصورت زیر داده میشود:

$$D_k = \sum_{i=1}^4 w_i B_i$$

که در آن B_i ها همان مقادیر $\beta_1 V_k, \beta_2 U_k, \beta_3 P_k, \beta_4 Q_k$ هستند که بترتیب بزرگی قدر مطلق هایشان مرتب شده اند ($0 \leq \beta_i \leq 1 ; i = 1, 2, 3, 4$).

با استفاده از این تابع تصمیم ترکیبی، امتیاز نهایی همه مستندات مشخص شده و سپس این مستندات بترتیب امتیاز در اختیار کاربر قرار می گیرند. ضرایب اهمیت اولیه که برای هر یک از این معیارها در نظر گرفته شده است به صورت زیر است:

$$\beta_1 = 0.24, \beta_2 = 0.32, \beta_3 = 0.2, \beta_4 = 0.24$$

در اینجا نیز دو قاعده ی ساده ی فازی ما را در تعیین مقادیر وزن های عملگر OWA یاری می کنند. وزن های OWA با توجه به مطالب بیان شده در بخش های

$$\text{پیشین برای چندینگر} \begin{cases} Q_a(r) = r^a \\ a \geq 0 \end{cases} \text{ به صورت زیر}$$

می باشند:

$$w_1 = \left(\frac{1}{4}\right)^a; w_2 = \left(\frac{2}{4}\right)^a - \left(\frac{1}{4}\right)^a;$$

$$w_3 = \left(\frac{3}{4}\right)^a - \left(\frac{2}{4}\right)^a; w_4 = 1^a - \left(\frac{3}{4}\right)^a$$

قوانین فازی مورد استفاده برای محاسبه ی وزن های OWA به صورت زیر هستند:

• قاعده ی اول: اگر حداقل دو معیار از معیار های فوق برای یک سند امتیازی بالاتر از 0.8 داشته باشند، امتیاز نهایی آن سند باید بزرگتر از 0.5 باشد.
یعنی:

$$(w_1 + w_2) \times 0.8 + (w_3 + w_4) \times 0 \geq 0.65$$

شکل ۳: ترکیب امتیازات ناشی از بلوک های شکل های ۱ و ۲، تولید لیست رتبه بندی شده نهایی، و اصلاح ضرایب اهمیت معیارها و مدل رفتاری سرویس دهنده ها.

پس از انجام محاسبات خواهیم داشت:

$$a < \frac{\ln\left(\frac{0.5}{0.8}\right)}{\ln\left(\frac{2}{4}\right)} \Rightarrow a < 0.67807$$

• قاعده ی دوم: اگر بیشتر از سه معیار از معیار های فوق برای یک سند امتیازی کوچکتر از 0.4 باشند، امتیاز نهایی آن سند باید کوچکتر از 0.75 باشد.

$$\text{یعنی: } (w_1) \times 1 + (w_2 + w_3 + w_4) \times 0.4 \leq 0.7$$

پس از انجام محاسبات خواهیم داشت:

$$a > \frac{\ln\left(\frac{0.7 - 0.4}{0.6}\right)}{\ln\left(\frac{1}{4}\right)} \Rightarrow a > 0.5$$

بنا بر این اگر $a \in [0.5, 0.67807]$ باشد دو شرط فوق ارضاء خواهند شد. با انتخاب $a = 0.58$ نتیجه نهایی چنین خواهد بود:

$$w_1 = 0.4475; w_2 = 0.22145;$$

$$w_3 = 0.17736; w_4 = 0.15368$$

آموزش ضرایب اهمیت معیارها

همانطور که در بخش پیشین ذکر شد، برای مقادیر ضرایب اهمیت معیارها

$$\left\{ \begin{array}{l} \beta_i^{(t+1)} = \frac{1 - e_i^{(t+1)}}{\sum_{j=1}^4 (1 - e_j^{(t+1)})} \\ i = 1, 2, 3, 4 \end{array} \right.$$

همانطور که مشخص است شرط $\sum_{i=1}^4 \beta_i^{(t+1)} = 1$ نیز برقرار می باشد. بدین ترتیب پس از گذشت مدت زمانی از آغاز فعالیت عامل در هر خوشه ی موضوعی، مدلی از رفتار هر سرویس دهنده در آن موضوع بدست می آید.

نتیجه گیری

در این مقاله قسمتی از معماری یک عامل هوشمند که نقش یک فراجویشگر سفارشی را برای کاربر ایفاء می نماید، مورد بررسی قرار گرفته است. شماتیک کلی معماری این عامل در شکل (۱) مشخص شده است. در مراحل ابتدایی طراحی و پیاده سازی عامل، ترکیب اطلاعات در دو سطح انجام می شد. یعنی ابتدا ترکیب لیست ها انجام می شد و سپس ترکیب همه $(2T+2)$ امتیاز متعلق به یک سند به صورت یکجا و توسط یک اپراتور OWA انجام می شد.

پس از اصلاح معماری و افزایش سطوح ترکیب اطلاعات به سه سطح، نتایج حاصله به نحو محسوسی بهتر گردید که در مقاله ی جدا گانه ای این امر و دلایل آن تحلیل شده است. به طور خلاصه دلیل این بهبود را می توان به تفکیک سطح ترکیب ویژگی ها از سطح ترکیب تصمیمات نسبت داد. زیرا در سطحی که اطلاعات مربوط به تعداد و پراکندگی n-gram ها با یکدیگر ترکیب می شوند، عملوند^{۳۳} های OWA در واقع ویژگی هایی از متن هستند که از پردازش متن به دست آمده اند. ولی در سطح سوم، چهار معیاری که باید با هم ترکیب شوند، هر یک به نوعی تصمیم راجع به میزان منطبق بودن متن با خواست کاربر می باشند. بنا بر این در این سطح تصمیم ها با هم ترکیب می شوند و تصمیم نهایی اتخاذ می گردد. بنا بر این این اطلاعات از یک جنس نیستند و ترکیب آنها به صورت توأم منطقی نیست.

($0 \leq \beta_i \leq 1$; $i = 1, 2, 3, 4$) دو شیوه ی کلی وجود دارد که یکی با گرفتن پسخور از کاربر و دیگری به صورت خودکار و از روی قرائن انجام می شود. در حالت اول استفاده از روش هایی مانند Q-Learning مناسب به نظر میرسد و تا کنون این روش در آموزش رفتار عامل های هوشمند زیادی به کار گرفته شده است. چنانچه بخواهیم این کار را به صورت خودکار انجام دهیم استفاده از شبکه های عصبی یکی از گزینه های است که می توان به نتیجه گیری مناسب از آن امیدوار بود. اما در این مقاله روش جدیدی برای این منظور ابداع شده است. مکانیزم این کار به این صورت است که پس از اینکه در یک دوره ی کامل، لیست نهایی مستندات بر اساس امتیاز ها آماده شد، امتیاز نهایی با هر یک از مقادیر P_k ، U_k ، V_k و Q_k مقایسه می شود و به هر کدام از این امتیازات که نزدیکتر باشد، ضریب اهمیت آن امتیاز باید افزایش بیشتری داشته باشد. برای این منظور میانگین خطای امتیاز نهایی از امتیاز هر یک از معیار های چهارگانه، ملاک عمل قرار می گیرد. هر بار این مقدار خطا با توجه به داده های جدید مجددا محاسبه می شود [۱۶].

مطلب فوق را می توان به صورت زیر بیان نمود:

$$e_1^{(t+1)} = e_1^{(t)} \cdot \frac{t}{t+1} + \frac{\frac{1}{N} \sum_{k=1}^N |D_k - V_k|}{t+1}$$

$$e_2^{(t+1)} = e_2^{(t)} \cdot \frac{t}{t+1} + \frac{\frac{1}{N} \sum_{k=1}^N |D_k - U_k|}{t+1}$$

$$e_3^{(t+1)} = e_3^{(t)} \cdot \frac{t}{t+1} + \frac{\frac{1}{N} \sum_{k=1}^N |D_k - P_k|}{t+1}$$

$$e_4^{(t+1)} = e_4^{(t)} \cdot \frac{t}{t+1} + \frac{\frac{1}{N} \sum_{k=1}^N |D_k - Q_k|}{t+1}$$

که در آن شاخص t نشاندهنده ی تعداد دفعات استفاده از عامل از ابتدا (زمان ری ست کردن امتیازات) تا کنون می باشد. بنا بر این هرچه مقدار خطای $e_i^{(t+1)}$ برای معیار i ام کوچکتر باشد، آن معیار، معیار مناسبتری می باشد. بنا بر این مقادیر $0 \leq \beta_i \leq 1$; $i = 1, 2, 3, 4$ به صورت مقابل اصلاح می شوند:

جمع بندی و پیشنهادات

ارزیابی قرار گیرند و نقاط قوت و ضعف هر یک از این روش ها در مقایسه با روش فعلی مشخص شوند. در زمینه ی نحوه تطبیق پذیری عامل هوشمند با کاربران متفاوت و نگهداری پروفایل کاربران و بروز کردن آن ، نیز جای کار فراوانی در این پابن نامه وجود دارد. تکنیک های مختلف مدلسازی کاربران و روش های مختلف یادگیری ماشینی در ساختار این عامل هوشمند قابل جایگزینی با الگوریتم های فعلی این عامل می باشند.

همچنین ساختار ذهنی این عامل در حال حاضر ضعیف است و می توان برای بالا بردن هوشمندی آن، معماری های مختلفی که برای ساختار ذهنی عاملها وجود دارند، مانند مدل BDI^{26} و سایر مدل ها را به طور کامل پیاده سازی نمود.

موضوع ترکیب اطلاعات در محیط های نرم افزاری موضوع جدید و پیشتازی است و هم اکنون جنبه های زیادی از این مقوله نیاز به تحقیق و بررسی بیشتر دارد. در حال حاضر تنوری مجموعه های تصادفی و نسخه خاص آن، $FISST^{24}$ که برای مقاصد ترکیب داده/ اطلاعات طراحی شده است، از موضوعات جدیدی است که محققان حوزه ی ترکیب اطلاعات مشغول تحقیق و کسب تجربه در این باب هستند. در این مقاله معماری عامل هوشمند ترکیب اطلاعات بر اساس عملگر OWA که عضوی از خانواده ی انتگرال های فازی است پی ریزی شده است. در ادامه ی این طرح لازم است کارآیی سایر روش ها و علی الخصوص روش های مبتنی بر $FISST$ و TBM^{25} با حفظ ساختار اصلی این معماری مورد

مراجع

- ۱ - کاوسی، ک. "بهبود عملکرد روش های هوشمند عامل نرم افزاری ترکیب اطلاعات در اینترنت." پایان نامه کارشناسی ارشد، گروه مهندسی برق و کامپیوتر، دانشگاه تهران، (۱۳۸۰).
- 2 - Cutello, V. and Montero, J. (1994). "Hierarchies of aggregation operators." *International Journal of Intelligent Systems*, Vol. 9, PP. 1025-1045.
- 3 - Engemann, K. J., Miller, H. E. and Yager, R. R. (1992). "Decision making with belief structures: an application in risk management." *Technical Report #MII-1234, Machine Intelligence Institute, Iona College, New Rochelle, NY.*
- 4 - Glockner, I. (2000). *A Broad Class of Standard DFSes*, Bielefeld University Report TR-2000.
- 5 - Jansen, J. (1997). "Using An Intelligent Agent to Enhance Search Engine Performance." *Peer-Reviewed Journal on the Internet*, <http://www.eecs.usma.edu/usma/academic/eecs/instruct/jansen/>, 1997.
- 6 - J. Kacprzyk, (1990). "Inductive learning from considerably erroneous examples with a specificity based stopping rule." *Proceedings of the International Conference on Fuzzy Logic and Neural Networks*, Iizuka, Japan, 819.
- 7 - Mladenic, D. (1999). "Text-Learning and Related Intelligent Agents: A Survey." *IEEE Intelligent Systems Journal*, PP. 44-54, July/August.
- 8 - Ovchinnikov, S. (2000). "Piecewise Linear Aggregation Functions." *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, Vol. 1, No. 1. PP.11-22.
- 9 Yager, R. R. (1992). "Decision making under Dempster-Shafer uncertainties." *International Journal of General Systems*, Vol. 20, PP. 233-245.
- 10 - Yager, R. R. and Filev, D. P. (1992). "Fuzzy logic controllers with flexible structures." *Proceedings of Second International Conference on Fuzzy Sets and Neural Networks*, Iizuka, Japan, PP. 317-320.

- 11 - Yager, R. R. and Rybalov, A. (1997). "On the Fusion of Documents from Multiple Collection Information Retrieval Systems." *Journal of the American Society for Information Science*.
- 12 - Yager, R. R. (1991). "Fuzzy quotient operators for fuzzy relational data bases." *Proc. Int. Fuzzy Engineering Symposium*, PP. 281-296, Yokohama, Japan.
- 13 - Yager, R. R. and Kreinovich, V. (1997). "On How to Merge Sorted Lists Coming from Different Web Search Tools." *American Association for Artificial Intelligence (AAAI) Symposium on Frontiers in Soft Computing and Decision Systems*, MIT, Boston, MA, November 8-10.
- 14 - Yager, R. R. (1988). "On Ordered Weighted Averaging Aggregation Operators in Multicriteria Decisionmaking." *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 18, No. 1, January/February.
- 15 - Yager, R. R. (1994). "Quantifier Guided Aggregation Using OWA Operators." *Technical Report, #MII-1504, Machine Intelligence Institute, Iona College, New York*.
- 16 - Yager, R. R. (1987). "A note on weighted queries in information retrieval systems." *Journal of American Society of Information Sciences*, Vol. 28, PP. 23-24.
- 17 - Moshiri, B., Moghadassi A. and Eydgahi, A. (1999). "Application of Information/Data Fusion in a Financial Network." *International Conference on Data Fusion, Stratford-upon-Avon, UK, Proceedings of EuroFusion 99, PP. 243-249, 5th-7th October*.
- 18 - Gauch, S., Wang, G. and Gomez, M. (1996). "ProFusion: Intelligent Fusion from Multiple, Distributed Search Engines." *Journal of Universal Computer Science*, Vol. 2, No. 9, PP. 637-649.
- 19 - Smeaton, A. F. and Crimmins, F. "Using a Data Fusion Agent for Searching the WWW." *WWW6 conference, Stanford, USA, April 1997, available at <http://lorca.compapp.dcu.ie/fusion/papers/fusion-www6.html>*.

واژه های انگلیسی به ترتیب استفاده در متن

- | | |
|---|--------------------------------|
| 1 - Large Scale Data Bases | 19 - Query |
| 2 - Information Retrieval | 20 - Parser |
| 3 - Hyper Text | 21 - Quantifier |
| 4 - Links | 22 - Fuzzy Rules |
| 5 - Web Searching | 23 - Operand |
| 6 - Documents | 24 - Finite-Set Statistics |
| 7 - On-Line | 25 - Transferable Belief Model |
| 8 - Off-Line | 26 - Belief-Desired-Intention |
| 9 - TAHAB (تحلیل هوشمند آمار بانکی) | |
| 10 - Customized Intelligent Agent | |
| 11 - Information Fusion | |
| 12 - Meta Search Engine | |
| 13 - Linguistic Concept | |
| 14 - Information Servers | |
| 15 - Ordered Weighted Adveraging Operator | |
| 16 - Cluster | |
| 17 - Soft | |
| 18 - Regular Increasing Monotone | |