رتبه بندی اسناد با استفاده از اتوماتای یادگیر توزیع شده

سعید ساعتی محمدرضا میبدی

آزمایشگاه سیستمهای نرم افزاری
دانشکده مهندسی کامپیوتر و فناوری االطلاعات
دانشگاه صنعتی امیرکبیر
تهران ایران
(saati, meybodi)@ce.aut.ac.ir

چکیده: در قسمت اول این مقاله یک ساختار خود سازمانده مبتنی بر اتوماتای یادگیر توزیع برای مجموعه های بزرگ اسناد پیشنهاد میگردد. در این ساختار پیشنهادی به هر سند یک اتوماتای یادگیر اختصاص داده می شود که وظیفه آن یادگیری ارتباطات آن سند با اسناد دیگر می باشد. در قسمت دوم مقاله روشی بر اساس ساختار پیشنهادی برای رتبه بندی اسناد ارایه میشود. از جمله مزایای این روش امکان استفاده از آن در کتابخانه های بزرگ و همچنین قابلیت گسترش آن می باشد. کارایی روش پیشنهادی برای رتبه بندی اسناد از طریق مقایسه رتبه بندی ایجاد شده توسط این روش با رتبه بندی ایده آل نشان داده میشود.

کلمات کلیدی: کتابخانه های دیجیتال، اتوماتای یادگیر توزیع شده، خودسازماندهی، رتبه بندی اسناد

۱- مقدمه

ایده رتبه بندی صفحات[1] که توسط Brin و Page ارائه گردیده است[13,14] در موتور جستجوی گوگل به منظور مرتب کردن نتایج جستجو استفاده میشود. الگوریتم رتبه بندی صفحات که بر اساس ساختار ارتباطی بین صفحات در وب عمل میکند رتبه یک صفحه p را متناسب با رتبه صفحاتی که به صفحه p ابرپیوند[2] دارند، مشخص می-کند. رتبه صفحه p طبق رابطه بازگشتی (۱) محاسبه می‌شود.

$$PageRank(p) = (1-d) + d * \sum_{all\ q\ linking\ to\ p} \left(\frac{PageRank(q)}{c(q)} \right) \qquad (1)$$

که در آن d ضریب تعدیل[3] که مقداری بین ۰ و ۱ است که معمولا ۰,۸۵ در نظر گرفته می‌شود، n تعداد کل صفحات و c(q) تعداد ابرپیوندهای موجود در صفحه q می باشد.

بیشتر تحقیقات انجام شده در زمینه رتبه بندی صفحات در باره بهبود الگوریتم حل معادله ۱ می باشد. در این روشها بدلیل اینکه نمی توان برای ابرپیوندهای موجود ارزشهای متفاوت در نظر گرفت، برای تمامی پیوندهای موجود در یک صفحه ارزشهای یکسان فرض می شود. روشهای دیگری هم که به پیوند های موجود در صفحه ارزشهای متفاوتی نسبت می دهند از اطلاعات مربوط به پیوند مانند محل ابرپیوند یا تگ html ای که در برگیرنده ابرپیوند است و توسط ایجاد کننده آن صفحه ایجاد شده است استفاده می کند[15].

در قسمت اول این مقاله یک ساختار خود سازمانده برای مجموعه های بزرگ اسناد مبتنی بر اتوماتای یادگیر توزیع شده پیشنهاد میگردد. . در این ساختار پیشنهادی به هر سند یک اتوماتای یادگیر اختصاص داده می شود که وظیفه آن یادگیری ارتباطات آن سند با اسناد دیگر می باشد. با استفاده از روش ارتباطات ایجاد شده ما بین اسناد دارای ارزشهای متفاوت میباشند. در قسمت دوم مقاله روشی بر اساس ساختار پیشنهادی برای رتبه بندی اسناد ارایه میشود. از جمله مزایای این روش امکان استفاده از آن در کتابخانه های بزرگ و قابلیت گسترش آن می باشد. کارایی روش پیشنهادی برای رتبه بندی اسناد از طریق مقایسه رتبه بندی ایجاد شده توسط این روش با رتبه بندی ایده آل نشان داده میشود.

ادامه مقاله بدین صورت سازماندهی شده است: در بخش ۲ اتوماتاهای یادگیر و اتوماتای یادگیر توزیع شده به طور مختصر شرح داده می شوند. در بخش ۳ خود سازماندهی در ساختار اطلاعاتی اسناد با استفاده از اتوماتای

[1] PageRanking
[2] Hyperlink
[3] Damping Factor

یادگیرتوزیع شده توضیح داده می شود. در بخش ۴ با استفاده ساختار اطلاعاتی پیشنهادی در بخش ۳ روش پیشنهادی برای رتبه بندی اسناد ارائه می شود. نتایج شبیه سازی در بخش ۵ آمده است. بخش ۶ نتیجه گیری میباشد.

۲- اتوماتای یادگیر[1]

اتوماتای یادگیر یک مدل انتزاعی است که تعداد محدودی عمل را می تواند انجام دهد. هر عمل انتخاب شده توسط محیطی احتمالی ارزیابی شده و پاسخی به اتوماتای یادگیر داده می شود. اتوماتای یادگیر از این پاسخ استفاده نموده وعمل خود را برای مرحله بعد انتخاب می کند. شکل ۱ ارتباط بین اتوماتای یادگیر و محیط را نشان می دهد.

شکل ۱: ارتباط بین اتوماتای یادگیر و محیط

محیط[2]: محیط را می توان توسط سه تایی $E \equiv \{\alpha, \beta, c\}$ نشان داد که در آن $\alpha \equiv \{\alpha_1, \alpha_2, ..., \alpha_r\}$ مجموعه ورودیها، $\beta \equiv \{\beta_1, \beta_2, ..., \beta_m\}$ مجموعه خروجیها و $c \equiv \{c_1, c_2, ..., c_r\}$ مجموعه احتمالهای جریمه می باشد. هر گاه β مجموعه دو عضوی باشد ، محیط از نوع P می باشد. در چنین محیطی $\beta_1 = 1$ به عنوان جریمه و $\beta_2 = 0$ به عنوان پاداش در نظر گرفته می شود. در محیط از نوع Q ، $\beta(n)$ می تواند به طور گسسته یک مقدار از مقادیر محدود در فاصله [0,1] و در محیط از نوع S، $\beta(n)$ متغیر تصادفی در فاصله [0,1] است. c_i احتمال اینکه عمل α_i نتیجه نامطلوب داشته باشد می باشد. در محیط ایستا[3] مقادیر c_i بدون تغییر می مانند، حال آنکه در محیط غیر ایستا[4] این مقادیر در طی زمان تغییر می کنند. اتوماتاهای یادگیربه دو گروه با ساختار ثابت و با ساختار متغیر تقسیم بندی میگردند. در ادامه به شرح مختصری درباره اتوماتای یادگیر با ساختار متغیر که در این مقاله از آنها استفاده شده است می پردازیم.

اتوماتای یادگیر با ساختار متغیر[5]: اتوماتای یادگیر با ساختار متغیر توسط ۴ تائی $\{\alpha, \beta, p, T\}$ نشان داده می شود که در آن $\alpha \equiv \{\alpha_1, \alpha_2, \cdots, \alpha_r\}$ مجموعه عملهای اتوماتا، $\beta \equiv \{\beta_1, \beta_2, \cdots, \beta_m\}$ مجموعه ورودیهای اتوماتا، $p \equiv \{p_1, p_2, \cdots, p_r\}$ بردار احتمال انتخاب هر یک از عملها، و $p(n+1) = T[\alpha(n), \beta(n), p(n)]$ الگوریتم یادگیری می باشد. در این نوع از اتوماتاها، اگر عمل α_i در مرحله n ام انتخاب شود و پاسخ مطلوب از محیط دریافت نماید، احتمال $p_i(n)$ افزایش یافته و سایر احتمالها کاهش می یابند. و برای پاسخ نامطلوب احتمال $p_i(n)$ کاهش یافته و سایر احتمالها افزایش می یابند. در هر حال، تغییرات به گونه ای صورت می گیرد تا حاصل جمع $p_i(n)$ ها همواره مساوی یک باقی بماند. .الگوریتم زیر یک نمونه از الگوریتمهای یادگیری خطی در اتوماتای با ساختار متغییر است.

الف- پاسخ مطلوب

$$p_i(n+1) = p_i(n) + a[1 - p_i(n)]$$
$$p_j(n+1) = (1-a)p_j(n) \qquad j \neq i \quad \forall j \qquad (2)$$

ب- پاسخ نامطلوب

$$p_i(n+1) = (1-b)p_i(n) \quad \forall j \quad j \neq i$$
$$p_j(n+1) = \frac{b}{r-1} + (1-b)p_j(n) \qquad (3)$$

در روابط فوق، پارامتر پاداش و a پارامتر پاداش و b پارامتر جریمه می باشد. با توجه به مقادیر a و b سه حالت را می توان در نظر گرفت. زمانیکه a و b با هم برابر باشند، الگوریتم را L_{RP}[6] می نامیم. زمانیکه b از a خیلی کوچکتر باشد، الگوریتم را $L_{R\varepsilon P}$[7] می نامیم. زمانیکه b مساوی صفر باشد، الگوریتم را L_{RI}[8] می نامیم. برای مطالعه بیشتر در باره اتوماتاهای یادگیر می توان به مراجع [10- 9] مراجعه کرد.

اتوماتای یادگیر توزیع شده[9]: یک اتوماتای یادگیر توزیع شده شبکه ای از اتوماتاهای یادگیر است که برای حل یک مساله خاص با یکدیگر همکاری دارند. در این شبکه از اتوماتاهای یادگیر همکار در هر زمان تنها یک اتوماتا فعال است تعداد اعمال قابل انجام توسط یک اتوماتا در DLA برابر با تعداد اتوماتاهایی است که به این اتوماتا متصل شده اند. انتخاب یک عمل توسط اتوماتای یادگیر در این شبکه باعث فعال شدن اتوماتای یادگیر متصل شده به این اتوماتای یادگیر متناظر با این عمل می گردد. به عبارت دیگر انتخاب یک عمل توسط یک اتوماتای یادگیر در این شبکه متناظر با فعال شدن یک اتوماتای یادگیر دیگر در این شبکه است.

مدلی که برای شبکه DLA در نظر می گیریم یک گراف است که هر یک از رئوس آن یک اتوماتای یادگیر است. وجود یال (LA_i, LA_j) در این گراف

[1] Learning Automata
[2] Environment
[3] Stationary
[4] Non-Stationary
[5] Variable Learning Automata
[6] Linear Reward Penalty
[7] Linear Reward Epsilon Penalty
[8] Linear Reward Inaction
[9] Distributed Learning Automata

بدین معناست که انتخاب عمل α_j^i توسط LA_i باعث فعال شدن LA_j می گردد. تعداد اعمال قابل انتخاب توسط LA_k بصورت $p^k = \{p_1^k, p_2^k, ..., p_{r_k}^k\}$ نمایش داده می شود. در این مجموعه عدد p_m^k نشان دهنده احتمال مربوط به عمل a_m^k است. انتخاب عمل a_m^k توسط LA_k باعث فعال شدن LA_m می شود. r_k تعداد اعمال قابل انجام توسط اتوماتای LA_k را نشان می دهد.

۳- خود سازماندهی در ساختار اطلاعاتی اسناد با استفاده از اتوماتای یادگیرتوزیع شده

در روش مبتنی بر اتوماتای یادگیر توزیع شده برای ایجاد یک ساختار اطلاعاتی پویا در مجموعه های بزرگ از اسناد مانند کتابخانه های دیجیتال, مجموعه اسناد و کاربران استفاده کننده از آن نقش یک محیط تصادفی را برای اتوماتاهای یادگیر موجود در DLA ایفا می کنند. خروجی DLA یک دنباله از اسناد مرور شده توسط یک کاربر هستند که مسیر حرکت کاربر را به سمت یک سند مورد نظر نشان میدهد. محیط با استفاده از این دنباله پاسخی برای DLA تولید می کند. با استفاده از این پاسخ ساختار داخلی اتوماتاهای یادگیر در اتوماتای یادگیر توزیع شده طبق الگوریتم یادگیر بروز میشود. در این قسمت از مقاله دو روش خودسازماندهی برای ساختار اطلاعاتی اسناد پیشنهاد می شود. در روش اول (DLA-FA) فرض براین است که با افزایش تعداد اسناد، تعداد اسنادی که یک سند به آن مرتبط می باشد ثابت باقی میماند و به بهمین دلیل به هر سند یک اتوماتای یادگیر با ساختار متغییر با تعداد اعمال ثابت تخصیص داده میشود. در روش دوم (DLA-VA) تعداد اسنادی که یک سند به آن مرتبط می باشد متغییر فرض شده است و به همین دلیل به هر سند یک اتوماتای یادگیر با ساختار متغییر با تعداد اعمال متغییر تخصیص داده میشود. این دو روش در ادامه شرح داده میشود و سپس از طریق شبیه سازی با یکدیگر مقایسه میگردند.

در روش DLA-FA اندازه بردار احتمال برای هر اتوماتای یادگیر در DLA با افزایش تعداد اسناد در مجموعه اسناد تغییر پیدا نمیکند. هر کدام از اعمال یک اتوماتای یادگیر، متناظر با یکی از اسناد در مجموعه اسناد و احتمال انتخاب این عمل در بردار احتمالات، ارتباط این سند با سند متناظر با آن عمل میباشد. بعبارت دیگر بردار اعمال یک اتوماتای یادگیر میتواند بعنوان شناسه سند متناظر با آن اتوماتای یادگیر و بردار احتمالات میزان ارتباط این سند با دیگر سندها در مجموعه اسناد در نظر گفته شود. بنابراین برای هر سند Doc_i یک اتوماتای یادگیر LA_i در نظر می گیریم که تعداد عملهای آن تعداد ثابتی میباشد. انتخاب عمل j توسط اتوماتای یادگیر LA_i به معنی فعال کردن اتوماتای یادگیر LA_j متناظر با سند Doc_j می باشد. در صورتیکه عمل انتخاب شده k امین عمل اتوماتای LA_i باشد (یعنی $a_k^i = j$) احتمال متناظر این عمل یعنی p_k^i بعنوان میزان ارتباط سندهای i و j در نظر گرفته می شود.

با ورود یک کاربر به سیستم و مشاهده سند Doc_i، اتوماتای یادگیر متناظر با آن سند یعنی LA_i فعال می شود. با حرکت کاربر از سند Doc_i به سند Doc_j، عمل مرتبط با این انتخاب در اتوماتای LA_i انتخاب میشود و به محیط اعمال می شود. با توجه به ثابت بودن تعداد اعمال اتوماتاهای متناظر اسناد، ممکن است عمل مرتبط با انتخاب سند Doc_j در بردار اعمال اتوماتای یادگیر Doc_i وجود نداشته باشد. در این شرایط در اتوماتای یادگیر متناظر با سند Doc_i عملی که دارای کمترین احتمال است حذف و بجای آن عمل جدید a_j^i قرار می گیرد و احتمال متناظر با این عمل برابر صفر قرار داده می شود. سپس احتمال عمل حذف شده بین احتمالهای اعمال توزیع می شود تا مجموع احتمالها همچنان ۱ باقی بماند. این مراحل تا پایان حرکت کاربر بین اسناد برای هر دو سند متوالی مشاهده شده توسط وی انجام می شود. همچنین ممکن است کاربر دوباره به Doc_i برگردد که این حرکت یک دور در مسیر حرکت او می باشد و نشاندهنده عدم رضایت از حرکت قبلی به سمت سند Doc_j می باشد. پس از اینکه کاربر سیستم را ترک کرد، با توجه به مسیر حرکت کاربر، اعمال انتخاب شده توسط اتوماتاهای یادگیر در طول مسیر طی شده در صورتیکه جزیی از یک دور نباشند، پاداش داده میشوند. هر چه مسیر طی شده توسط کاربر کوتاهتر باشد میزان پاداش داده شده توسط الگویتم یادگیری به اعمال انتخاب شده در طول این بیشتر می باشد. اعمالی که قسمتی از یک دور باشند نشاندهنده حرکت اشتباه کاربر هستند و مجازات می شوند. با این مراحل هر کاربر یک رشته از اتوماتاها را فعال نموده و احتمال اعمال آنها توسط سیستم اصلاح شده است که در نتیجه ارتباطات اسناد متناظر آن اتوماتاها اصلاح می شود.

روش دوم (DLA-VA) برای شرایطی که تعداد اسنادی که یک سند به آن مرتبط می باشد ثابت نمیباشد پیشنهاد شده است. در این روش، مانند روش (DLA-FA) بردار احتمالات اتوماتاهای یادگیر، برای نشان دادن میزان ارتباط بین اسناد استفاده می شود. نحوه استفاده از این بردار دقیقا مانند روش DLA-FA می باشد. تفاوت این روش در نحوه حذف اعمال می باشد که در ادامه توضیح داده میشود. با ورود یک کاربر به سیستم و مشاهده سند Doc_i، اتوماتای یادگیر متناظر با آن سند یعنی LA_i فعال می شود. با حرکت کاربر از سند Doc_i به سند Doc_j، عمل مرتبط با این انتخاب در

اتوماتای LA_i انتخاب میشود و به محیط اعمال می شود. در صورتیکه عمل مرتبط با انتخاب سند Doc_j در بردار اعمال اتوماتای یادگیر Doc_i وجود نداشته باشد، در اتوماتای یادگیر متناظر با سند Doc_i عمل جدید a_j^i اضافه می شود و احتمال متناظر با این عمل برابر پارامتر الگوریتم یادگیری اتوماتاهای یادگیر قرار داده می شود. این مراحل تا پایان حرکت کاربر بین اسناد برای هر دو سند متوالی مشاهده شده توسط وی انجام می شود. پس از اینکه کاربر سیستم را ترک کرد، احتمال اعمال انتخاب شده مانند روش DLA-FA بروز می شود. پس از این مرحله در صورتیکه در هر یک از اتوماتاهای یادگیر متناظر با اسناد مرور شده توسط کاربر، اعمالی وجود داشته باشند که دارای احتمال کوچکی باشند آن اعمال از مجموعه اعمال آن اتوماتای یادگیر حذف می شوند و احتمال عمل حذف شده بین احتمالهای بقیه اعمال آن اتوماتای یادگیر توزیع می شود تا مجموع احتمالها همچنان ۱ باقی بماند.

۴- رتبه بندی اسناد با استفاده اتوماتای یادگیر توزیع شده

برای حل معادله (۱) نیاز به ایجاد یک مدل ریاضی از ارتباطات بین صفحات در وب می‌باشد. برای این منظور ماتریس L بصورت زیر تعریف می‌شود.

$$L_{ij} = \begin{cases} \dfrac{1}{|O_i|} & \text{if there is a link from page } i \text{ to page } j \\ 0 & \text{otherwise} \end{cases}$$

که O_i تعداد ابرپیوندهای موجود در صفحه i می باشد. با استفاده از این ماتریس معادله (۱) را می‌توان بصورت زیر نوشت.

$$X = L^T X \qquad (۲)$$

که X برداری است که اعضای آن رتبه صفحات می‌باشد. برای حل این معادله که یک دستگاه معادلات خطی می‌باشد از روشهای عددی استفاده می‌شود. روش عددی معمول برای حل این دستگاه، روش گوس-سایدل[1] می‌باشد که در مقایسه با روشهای ژاکوبی[2] و توانی[3]، به تعداد تکرار کمتری برای رسیدن به جوابی با خطای مشخص دارد. روش گوس-سایدل در زیر آمده است.

```
Function Gauss_Seidel()
{
  k=1;
  repeat
      x_i^(k+1) = (1-α) Σ_{j<i} a_ij x_j^(k+1) + α Σ_{j>i} a_ij x_j^(k)   ∀i
      δ = ||x^(k+1) - x^(k)||
  until δ < ε
  returrn x;
}
```

در الگوریتم گوس-سایدل در صورتیکه در دو تکرار متوالی هیچکدام از اعضای بردار جواب، بیشتر از ε تغییر نکرده باشند الگوریتم خاتمه می یابد. در این مقاله مقدار ε برابر 10^{-8} در نظر گرفته شده است.

طبق معادله (۲) برای محاسبه رتبه اسناد ماتریس L بایستی ایجاد شود. در روش ارائه شده بر اساس DLA باید بین اطلاعات موجود در اتوماتاهای یادگیر هر سند و ماتریس L تناظر ایجاد نمود. برای این منظور از بردارهای احتمالات انتخاب اعمال اتوماتاهای یادگیر استفاده می‌شود. ماتریس L ا بصورت زیر تعریف می‌شود.

$$L(i,j) = \begin{cases} P_i(l) & \text{if } A_i(l) = j \\ 0 & \text{otherwise} \end{cases} \qquad (۳)$$

با استفاده از ماتریس L و روش گوس-سایدل، طبق مراحل زیر رتبه اسناد محاسبه میشود.

۱- با استفاده از فرمول (۳) ماتریس L را ایجاد میشود.

۲- رتبه مرحله بعد طبق معادله روبرو محاسبه محاسبه میشود.

$$x_i^{(k+1)} = (1-\alpha)\sum_{j<i} l_{ij} x_j^{(k+1)} + \alpha \sum_{j>i} l_{ij} x_j^{(k)} \quad \forall i$$

۳- خطا مطابق δ مطابق فرمول روبرو محاسبه میشود.

$$\delta = \left\| x^{(k+1)} - x^{(k)} \right\|$$

۴- در صورتیکه خطا از 10^{-8} بیشتر است کنترل به مرحله ۲ منتقل میشود.

۵- x بعنوان بردار رتبه برگردانده میشود.

با اجرای الگوریتم رتبه بندی برروی مجموعه ای از صفحات، رتبه های متناظر با این صفحات بدست می آید. برای مقایسه رتبه های ایجاد شده توسط دو الگوریتم مختلف از معیاری بنام کوریلیشن ترتیب[4] استفاده میشود. کوریلیشن ترتیب بصورت زیر تعریف می شود.

$$\text{Rank Correlation} = 1 - \left[\frac{6 * \sum_{i=1}^{n} D_i^2}{(n*(n^2-1))} \right] \qquad (۴)$$

[1] Gauss-Siedel
[2] Jacobi
[3] Power Method
[4] Rank Correlation

که D_i تفاوت بین رتبه عضو i ام مجموعه رتبه های ایجاد شده توسط دو الگوریتم متفاوت می باشد. عدد بدست آمده که بین صفر و یک می باشد نشاندهنده میزان نزدیک بودن رتبه های دو مجموعه می باشد. هر چه این عدد به ۱ نزدیکتر باشد این دو مجموعه از رتبه ها به یکدیگر شبیه تر می باشند.

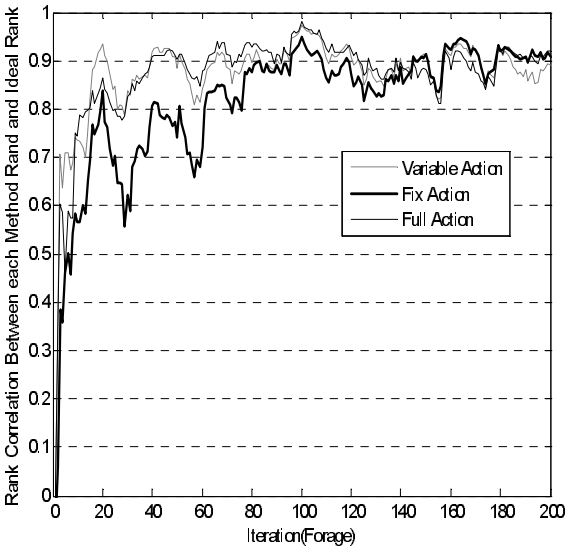
۵- نتایج شبیه سازیها

برای شبیه سازی از مدل ارایه شده در [12] برای تولید مجموعه اسناد و حرکات کاربران استفاده میشود. هر سند با یک بردار محتوا نمایش داده می شود. طول این بردار برابر تعداد موضوع های موجود در سیستم است و هر عضو این بردار میزان ارتباط سند متناظر با آن بردار را با یکی از این موضوعات نشان می دهد. هر یک از موضوعات با یک توزیع احتمالاتی خاصی بین اسناد توزیع شده است. با تغییر پارامتر این توزیع و تغییر تعداد اسناد، میتوان سیستم های اطلاعاتی مصنوعی متفاوتی ایجاد نمود. در این مدل پروفایل علایق کاربران[1]، انگیزه[2] و استراتژی حرکتی[3] آنها از طریق توزیعهای آماری مدل شده اند که با تغییر پارامترهای این توزیعهای آماری می توان کاربرانی با علایق، انگیزه ها و استراتژی های متفاوت ایجاد نمود. برای اطلاعات بیشتر درباره این مدل می توان به [12] مراجعه نمود. در شبیه سازیها، پارامترهای تعداد موضوعات، تعداد کاربران، علایق، انگیزه ها و پارامترهای توزیع آنها در طی شبیه سازی ثابت در نظر گرفته شده اند. با استفاده بردار محتوای هر کدام از اسناد شباهت بین هر دو سند موجود در سیستم محاسبه میشود. ماتریس شباهت بدست آمده بعنوان ماتریس ارتباطات ایده ال بین اسناد در شبیه سازیها به منظور ارزیابی کارایی روشهای پیشنهادی برای رتبه بندی استفاده میشود.

برای ارزیابی کارایی روشهای پیشنهادی، الگوریتمهای رتبه بندی علاوه بر ماتریسهای ارتباطات بدست آمده توسط روشهای پیشنهادی در این مقاله، بر روی ماتریس ارتباطات ایده آل هم اجرا میشود. برای مقایسه مجموعه رتبه های بدست آمده، از کوریلیشن ترتیب بین هر کدام از این مجموعه های بدست و مجموعه رتبه های بدست آمده از ماتریس ارتباطات ایده ال استفاده می شود. برای شبیه سازیها از اتوماتاهای یادگیر توزیع شده متفاوت استفاده شده است که این اتوماتاهای یادگیر توزیع شده عبارتند از:

- اتوماتای یادگیر توزیع شده که در آن از اتوماتاهای یادگیر با ساختار متغییر با تعداد اعمال ثابت (DLA-FA) استفاده شده است
- اتوماتای توزیع شده که در آن از اتوماتاهای یادگیر با ساختار متغییر با تعداد اعمال متغییر(DLA-VA) شده است.
- اتوماتای توزیع شده که در آن از اتوماتاهای یادگیر با ساختار متغییر با تعداد اعمال برابر با تعداد اسناد(DLA-NC) استفاده شده است.

برای شبیه سازی ها شبکه ای از ۵۰ سند در نظر گرفته شده است. در روش DLA-FA طول بردار اعمال ۲۰ در نظر گرفته شد. طول این بردار برای DLA-NC برابر تعداد اسناد یعنی۵۰ و برای روش DLA-VA متغییر می باشد. مراحل یادگیری تا تکرار ۲۰۰ انجام شد. در هر یک از این تکرارها، ارتباطات بدست آمده توسط هر کدام از روشها بعنوان ورودی الگوریتم رتبه بندی در نظر گرفته شد. کوریلیشن ترتیب بین مجموعه رتبه های بدست آمده از هر کدام از این روشها و مجموعه رتبه های بدست آمده از ساختار ایده ال با استفاده از فرمول (۴) محاسبه شده است. نتیجه این مقایسه در شکل ۲ آمده است.



شکل ۲: کوریلیشن ترتیب بین رتبه بندی های بدست آمده با استفاده از سه روش پیشنهادی

همانطور که در شکل ۴ دیده می شود کوریلیشن برای دو روش DLA-FAو DLA-VA تقریبا برابر می باشد. از بین سه روش DLA-FA، DLA-VA و DLA-NC ، روش DLA-VA برای رسیدن به کوریلیشن موردنظر نیاز به تعداد تکرارهای بیشتری دارد. سه روش DLA-FA، DLA-VA و -DLA

[1] Interest Profile
[2] Motivation
[3] Browsing Strategy

NC به ترتیب در تعداد تکرارهای ۴۷ ، ۴۳ و ۴۶ با خطای 10^{-8} به جواب رسیده اند. دو روش DLA-FAو DLA-VA هم از نظر تعداد تکرار لازم برای رسیدن به جواب و هم از نظر میزان کوریلیشن ترتیب بدست آمده نتایج قابل قبولی در مقایسه با روش DLA-NC، که نیاز به حافظه بالایی دارد و برای مواردی که تعداد اسناد زیاد هستند مناسب نمیباشد، تولید کرده اند. به منظور بررسی توانایی قدرت یادگیری روش پیشنهادی در یادگیری رتبه بندی آزمایش دیگری ا نجام شد. در این آزمایش کوریلیشن ترتیب بین رتبه های بدست آمده با استفاده از روشهای DLA-FA، DLA-VA و DLA-NC در تکرارهای ۲۰، ۵۰، ۱۰۰ و ۲۰۰ محاسبه گردید که این نتایج در جدول ۱ آمده است. این نتایج نشان دهنده قابلیت یادگیری هر سه روش میباشد.

جدول ۱: کوریلیشن ترتیب بدست آمده با استفاده از سه روش در تکرارهای مختلف

تکرار / روش	۲۰	۵۰	۱۰۰	۲۰۰
DLA-NC	۰,۸۷۲	۰,۸۸۹	۰,۹۲۶	۰,۹۴۲
DLA-FA	۰,۸۶۴	۰,۸۷۶	۰,۹۱۴	۰,۹۲۷
DLA-VA	۰,۸۵۳	۰,۸۶۴	۰,۹۰۳	۰,۹۰۸

به منظور بررسی ارتباط بین تعداد اسناد و پارامتر توزیع موضوع ها در کتابخانه دیجیتال و تعداد تکرارهای لازم برای محاسبه رتبه اسناد شبیه سازی دیگری انجام شد. در این شبیه سازی، ساختارهای اطلاعاتی بدست آمده با استفاده از هر کدام از روشها در تکرار ۲۰۰۰ بعنوان ورودی الگوریتم محاسبه رتبه اسناد مورد استفاده قرار گرفت. نتایج این شبیه سازی در جدول ۲ آمده است. همانطور که دیده می شود تکرارهای لازم برای محاسبه رتبه اسناد تقریبا از تعداد اسناد در کتابخانه مستقل است.

جدول ۲ : تعداد تکرارهای لازم برای رتبه بندی اسناد با استفاده از اطلاعات بدست آمده از هر کدام از سه روش بر اساس DLA در تکرار ۲۰۰۰ از یادگیری

تعداد اسناد / روش	۱۰	۲۰	۵۰	۱۰۰
DLA-NC	۴۵	۴۳	۴۶	۴۷
DLA-FA	۴۴	۴۶	۴۵	۴۷
DLA-VA	۴۷	۴۶	۴۶	۴۹

۶- نتیجه گیری

در این مقاله ابتدا یک ساختار خود سازمانده مبتنی بر اتوماتای یادگیر توزیع برای مجموعه های بزرگ اسناد پیشنهاد و سپس با استفاده از این ساختار پیشنهادی روشی برای رتبه بندی اسناد ارایه گردید. از مزایای این روش امکان استفاده از آن در کتابخانه های بزرگ و همچنین قابلیت گسترش آن می باشد. کارایی روش پیشنهادی برای رتبه بندی اسناد از طریق مقایسه رتبه بندی ایجاد شده توسط این روش با رتبه بندی ایده آل نشان داده شد..

مراجع

[1] Heylighen, F. "Design of A Hypermedia Interface Translating between Associative and Formal Representations", International Journal of Man-Machine Studies 35, pp.491-515, 2002.

[2]Colley, R. "Web Usage Mining: Discovery and Application of Interesting Patterns from Web Data". Ph.D. Thesis, University of Minnesota, May 2000.

[3] Shahabi, C., Zarkesh, A. M. and Shah, V. "Knowledge Discovery from User's Web-page Navigation", Proceedings 7th IEEE International Conference On Research Issues in Data Engineering, pp. 20-29, 1997.

[4] Bollen, J. "A Cognitive Model of Adaptive Web Design and Navigation: A Shared Knowledge Perspective", Ph.D. Thesis, Vrije Universiteit Brussel, 2001.

[5] Heylighen, F. and Bollen. J. "Hebbian Algorithms for a Digital Library Recommendation System", Proceedings of the International Conference on Parallel Processing Workshops(ICPPW'02) 2002.

[6] W. Teles, Weigang, L. and Ralha, C. "AntWeb — The Adaptive Web Server Based on the Ants' Behavior", Proceedings of IEEE/WIC International Conference on Web Intelligence (WI'03), 2003, pp. 558-564. 1997.

[7] Lakshmivarahan, S. "Learning Algorithms: Theory and Applications", New York: Springer-verlag, 1981.

[8] Meybodi, M. R. and S. Lakshmivarahan, S. , "On a Class of Learning Algorithms which Have Symmetric Behavior under Success and Failure", pp. 145-155. Lecture Notes in Statistics, Berlin: Springer-Verlag, 1984.

[9] Mars, P., Chen, J. R. and Nambir, R. "Learning Algorithms: Theory and Applications in Signal Processing", Control, and Communication", CRC Press Inc., 1996.

[10]Narendra, K. S. and Thathachar, K. S., Learning Automata: An Introduction, New York: Prentice-Hall, 1989.

[11] Heylighen, F. "Mining Associative Meanings from the Web: From Word Disambiguation to the Global Brain", Proceedings of the International Colloquium: Trends in Special Language and Language Technology. pp 15-44. 1995

[12] Liu, J., Zhang, S. and Yang, J. "Characterizing Web Usage Regularities with Information Foraging Agents", IEEE Transactions on knowledge and data engineering, vol. 16, no. 5, may 2004.

[13] S. Brin and L. Page. The Anatomy of a Large-scale Hypertextual Web Search Engine. Computer Networks and ISDN Systems, Vol. 30, pp.107–117, 1998.

[14] L. Page, S. Brin, R. Motwani and T. Winograd. The Page rank Citation Ranking: Bringing Order to the Web. Technical Report, Stanford Digital Libraries SIDL-WP-1999-0120, 1999.

[15] R. B. Yates and E.Davis, "Web Page Ranking Using Link Attributes", Proceedings of International World Wide Web Conference, pp. 328-329, 2004,