# مدل جدیدی برای جستجوی عبارت بر اساس کمینهٔ جابه‌جایی وزن‌دار

**جواد پاکسیما**

دانشکدهٔ مهندسی کامپیوتر، دانشگاه پیام‌نور، ایران

## چکیده

**بر اساس پژوهش‌های انجام‌شده روی موتورهای جستجو، بیش‌تر پرس‌وجوهای کاربران بیش از یک واژه است. برای پرس‌وجوهای با بیش از یک واژه دو مدل می‌توان ارائه داد. در مدل نخست فرض می‌شود واژگان پرس‌وجو مستقل از یکدیگر هستند و در مدل دوم محل و ترتیب واژگان وابسته فرض می‌شود. آزمایش‌ها نشان می‌دهد که در بیش‌تر پرس‌وجوها بین واژگان وابستگی وجود دارد. یکی از پارامترهایی که می‌تواند وابستگی بین واژگان پرس‌وجو را مشخص کند، فاصلهٔ بین واژگان پرس‌وجو در سند است. در این مقاله تعریف جدیدی از فاصله بر اساس کمینه جابه‌جایی وزن‌دار[1] واژگان سند به‌منظور تطبیق بر پرس‌وجو ارائه می‌شود. هم‌چنین با توجه به این‌که بیش‌تر الگوریتم‌های رتبه‌بندی از فرکانس رخداد یک واژه در سند[2] برای امتیازدهی به اسناد استفاده می‌کنند و برای پرس‌وجو با بیش از یک واژه تعریف روشنی از این پارامتر وجود ندارد. در این مقاله پارامترهای فرکانس رخداد یک عبارت[3] و معکوس فرکانس سند[4] با توجه به مفهوم جدید فاصلهٔ تعریف‌شده و الگوریتم‌هایی برای محاسبهٔ آن‌ها ارائه شده است. همچنین نتایج الگوریتم پیشنهادی با چند الگوریتم مقایسه شده است که افزایش خوبی را در میانگین دقّت نشان می‌دهد.**

**واژگان کلیدی: موتور جستجو، رتبه‌بندی، فاصله، وابستگی واژگان، فرکانس عبارت (PF)**

# A novel model for phrase searching based-on Minimum Weighted Relocation Model

**Javad Paksima**

Department of Computer, University of Payam Noor, Iran

**Abstract**

**Finding high-quality web pages is one of the most important tasks of search engines. The relevance between the documents found and the query searched depends on the user observation and increases the complexity of ranking algorithms. The other issue is that users often explore just the first 10 to 20 results while millions of pages related to a query may exist. So search engines have to use suitable algorithms with high performance to find the most relevant pages.**

**The ranking section is an important part of search engines. Ranking is a process in which the web page quality is estimated by the search engine. There are two main methods for ranking web pages. In the first method, ranking is done based on the documents' content (traditional rankings). Models, such as Boolean model, probability model and vector space model are used to rank documents based on their contents. In the second method, based on the graph, web connections and the importance of web pages, ranking process is performed. Based on researches on search engines, the majority of user queries is more than one term. For queries with more than one term, two models can be used. The first model assumes that query terms are independent of each other while the second model considers a location and order dependency between query terms.**

[1] MWRM [2] Term Frequency [3] Phrase Frequency [4] Inverted Document Frequency

• تاریخ ارسال مقاله: ۱۳۹۶/۸/۲۵ • تاریخ آخرین بازنگری: ۱۳۹۷/۹/۲۲ • تاریخ پذیرش: ۱۳۹۷/۱۰/۱۹





[ DOI: 10.29252/jsdp.15.4.71 ]

**Experiments show that in the majority of queries there are dependencies between terms. One of the parameters that can specify dependencies between query terms is the distance between query terms in the document. In this paper, a new definition of distance based on Minimum Weighted Displacement Model (MWDM) of document terms to accommodate the query terms is presented. In the Minimum Weighted Displacement Model (MWDM), we call the minimum number of words moving a text to match the query term by space.**

**In addition, because most of the ranking algorithms use the TF (Term Frequency) to score documents and for queries more than one term, there is no clear definition of these parameters; in this paper, according to the new distance concept, Phrase Frequency and Inverted Document Frequency are defined. Also, algorithms to calculate them are presented. The results of the proposed algorithm compared with multiple corresponding algorithms shows a favorable increase in average precision.**

**Keywords: Search engine, Ranking, Distance, Phrase Frequency.**

## ۱- مقدمه

پیداکردن صفحات دارای کیفیت بالا در وب یکی از مهم‌ترین وظایف موتورهای جستجو است. مفهوم میزان ارتباط اسناد پیداشده با پرس‌وجو وابسته به نظر کاربر است و این موضوع باعث افزایش پیچیدگی الگوریتم‌های رتبه‌بندی می‌شود. نکتهٔ دیگر این است که اغلب کاربران ده تا بیست نتیجهٔ نخست را بررسی می‌کنند [1] درحالی‌که برای یک پرس‌وجو ممکن است، میلیون‌ها صفحهٔ مرتبط وجود داشته باشد. بنابراین موتورهای جستجو باید برای یافتن مرتبط‌ترین صفحات، الگوریتم مناسب با کارایی بالا ارائه کنند.

بخش رتبه‌بندی یکی از مهم‌ترین قسمت‌های موتورهای جستجو است. رتبه‌بندی فرآیندی است که در آن کیفیت صفحه توسط موتور جستجو تخمین زده می‌شود. در حال حاضر دو روش عمده برای رتبه‌بندی صفحات وب وجود دارد. در روش نخست رتبه‌بندی بر اساس محتوای اسناد انجام می‌شود (رتبه‌بندی سنتی). مدل‌هایی مانند مدل بولی، مدل احتمالی و مدل فضای برداری جهت رتبه‌بندی اسناد مبتنی بر محتوا ارائه شده‌اند [2]. در روش دوم بر اساس گراف و اتصالات وب و میزان اهمیت صفحات رتبه‌بندی صورت می‌گیرد.

در رتبه‌بندی سنتی، موتور جستجو سعی می‌کند با پیدا کردن میزان ارتباط سند و پرس‌وجو اسناد را رتبه‌بندی کند. در این الگوریتم‌ها برای هر پرس‌وجو، اسناد با محتوای شبیه‌تر به واژگان موجود در پرس‌وجو امتیاز بالاتری را دریافت می‌کنند. به‌عنوان مثال الگوریتم‌های TF-IDF [3]، BM25 [4] دو نمونه از رایج‌ترین الگوریتم‌های این نوع رتبه‌بندی هستند.

وب شامل تعداد زیادی اسناد غیر ساخت‌یافته است که به هم متصل هستند و یک گراف خیلی بزرگ را ایجاد می‌کنند. به‌طورمعمول تعداد واژگان پرس‌وجوها کم بوده (۲/۴ واژه در هر پرس‌وجو [5]) و مجموعهٔ کل لغات خیلی زیاد است. این در حالی است که در بیش‌تر الگوریتم‌های بازیابی اطلاعات به طور معمول برای تعداد اسناد کم و تعداد واژگان پرس‌وجوی زیاد کارایی خوبی دارند [1].

یک عبارت[1]، فهرستی از واژگان است که دارای ترتیب مشخصی هستند. به‌عنوان مثال «موتور جستجو» یک عبارت شامل دو واژه است که واژه نخست آن «موتور» و واژه دوم آن «جستجو» است. طول بیش‌تر عبارات دو واژه است و کمتر از یک درصد عبارات طولی، بیش‌تر از شش واژه دارند [6].

در یک کار تحلیلی بر روی یک و نیم میلیون پرس‌وجو در موتور جستجوی Excite مشخص شد که ۸/۴ درصد از پرس‌وجوها شامل علامت نقل‌قول (' یا ") هستند [7]. پژوهشی دیگر نشان می‌دهد که مقصود کاربران در چهل درصد از پرس‌وجوهایی که دو واژه یا بیشتر هستند و علامت نقل‌قول هم ندارند، عبارت بوده است نه واژگان مجزا [8].

روش اصلی برای جستجوی یک عبارت استفاده از نمایهٔ معکوس[2] است [9]. هر ردیف در نمایهٔ معکوس شامل یک واژه است که فهرستی از سه‌تایی‌های مرتب شامل شماره سند، تعداد رخداد واژه در سند و موقعیت‌هایی که در آن واژه مورد نظر در سند ظاهرشده و به‌صورت زیر است:

$<d, tf_{d,t}, [p_1, p_2, \ldots.]>$

که در آن d شناسه سندی است که شامل واژه t و $tf_{d,t}$ تعداد رخداد t در سند d است و $p_i$ موقعیت t در سند را نشان می‌دهد. فهرست زیر یک نمونه از این سه‌تایی‌ها است:

<101, 4, [5, 11, 25, 77]>

با استفاده از فهرست نمایهٔ معکوس، زمان پردازش برای شمارش تعداد عبارت به تعداد واژگان پرس‌وجو و تعداد تکرار واژگان عبارت در سند وابسته است. برای ارزیابی یک عبارت باید ابتدا موقعیت‌های واژگان پرس‌وجو استخراج و سپس با ترکیب آن‌ها شمارش انجام شود.

[1] Phrase
[2] Inverted Index



[ DOI: 10.29252/jsdp.15.4.71 ]

برای رتبه‌بندی اسناد، زمانی که پرس‌وجو فقط یک واژه باشد، پارامتر TF[1] و IDF[2] مهم‌ترین پارامترها برای رتبه‌بندی هستند. این پارامترها در روش‌های TF-IDF و BM25 استفاده شده‌اند. اگر بخواهیم از این روش‌ها برای جستجوی عبارت استفاده شود، باید پارامترهای مشابهی برای عبارت نیز تعریف شود.

به‌طورمعمول در موتورهای جستجو برای جستجوی عبارتی که در آن از علامت نقل‌قول استفاده شده است مشابه جستجوی تک‌واژه عمل‌می‌شود و الگوریتم‌های متعددی برای آن ارائه شده است [10] و [11]. اما اگر علامت نقل‌قول نباشد، بهتر است، بیش‌ترین امتیاز برای اسنادی باشد که بیش‌ترین تکرار عبارت با همان ترتیب و با کم‌ترین فاصله در آن صورت گرفته است. به همین منظور در این مقاله پارامتری مشابه TF به نام PF[3] تعریف‌شده است که فرکانس یک عبارت را نشان می‌دهد و الگوریتمی برای محاسبه IDF بر اساس PF ارائه می‌شود.

در ادامهٔ مقاله در بخش ۲ کارهای مرتبط انجام‌شده توضیح داده می‌شود؛ در بخش ۳ اصطلاحات استفاده‌شده، تعریف خواهد شد. پس از آن در بخش ۴ الگوریتم پیشنهادی برای محاسبه PF و IDF معرفی شده و در بخش ۵ مقایسه‌ای بین الگوریتم پیشنهادی و دو الگوریتم دیگر انجام گرفته است.

## ۲- کارهای مرتبط

از قدیمی‌ترین کارها در مورد وابستگی واژگان، کار ریجس برگن[4] (۱۹۷۷) است که به‌صورت نظری رتبه‌بندی اسناد را بر اساس مجاورت واژگان پرس‌وجو بررسی کرده است [12]. در روش پیشنهادی وی، میزان ارتباط واژگان پرس‌وجو بر اساس اطلاعات زوج‌واژه مشخص می‌شد و برای بازیابی از درخت پوشای بیشینه (MST[5]) استفاده شده است. سال‌ها بعد در سال ۲۰۰۲ نالاپاتی[6] و آلان[7] ایدهٔ خود را در مورد نحوهٔ مشخص‌کردن میزان ارتباط واژگان پرس‌وجو ارائه دادند [13]. برای کاهش زمان، آن‌ها پیشنهاد کردند که به‌جای استفاده از آمار اسناد، از آمار جملات برای ساخت درخت پوشای بیشینه استفاده شود.

در سال ۱۹۹۱ کین[8] نتایج تجربی‌اش را منتشر و فرض کرد که بر اساس موقعیت واژگان پرس‌وجو در سند با غربال اسناد غیر مرتبط می‌تواند نتایج را دقیق‌تر کند. کین برای نخستین‌بار مفهوم فاصلهٔ بین دو واژه را تعریف کرد. طبق تعریف او فاصله عبارت است از تعداد واژگان تطبیق داده‌نشده بین نخستین و آخرین تطبیق در یک جمله. او هفت دیدگاه مختلف برای استفاده از مجاورت واژگان پیشنهاد داد. در مطالعهٔ وی بهترین نتایج، زمانی به‌دست می‌آمد که الگوریتم به کم‌بودن فاصلهٔ بین واژگان پرس‌وجو پاداش می‌داد [14].

هم‌چنین در سال ۱۹۹۱ کرافت[9] و همکارانش [15] برای نخستین‌بار روشی برای استفاده از شبکهٔ استنتاج برای واژگان مجاور با عنوان InQuery ارائه دادند. آن‌ها هم‌چنین واژگان با DF بالا را از پرس‌وجو حذف کردند؛ برای مثال واژهٔ «شهر» را از پرس‌وجوی «اماکن تفریحی شهر شیراز» حذف کردند. آن‌ها دقّت خوبی را در صفحات ابتدایی جستجو به‌دست آورند؛ ولی درکل، دقّت کاهش یافت. این موضوع بعدها در سال ۲۰۰۵ توسط متلزر[10] و کرافت تأیید شد [16]. دنگ[11] و همکارانش کار کرافت را دنبال کردند. آن‌ها بر اساس واژگان ظاهرشونده در پنجره‌ای که محل تمرکز واژگان پرس‌وجو است و هم‌چنین بازخورد کاربر، امتیاز اسناد را محاسبه می‌کردند [17].

مدل‌های زبانی متنوعی برای وابستگی واژگان پرس‌وجو ارائه شده است. نخستین‌بار در سال ۱۹۹۹ سانگ[12] و کرافت [18] مدل تک‌واژه‌ای[13] استاندارد را با درون‌یابی مدل زوج‌واژه‌ای[14] توسعه دادند. در مقیاس آزمایشی کوچک، استفاده از زوج‌واژگان، نتایج را بهبود بخشید. گائو[15] و همکارانش در سال ۲۰۰۴ مدل زبانی وابسته (DLM[16]) را پیشنهاد دادند [19]. در این مدل آن‌ها اتصال بین توزیع مدل زبانی تک‌واژه‌ای و اسناد شامل واژگان مجاور را در پنجره‌ای به طول سه تعریف کردند. آن‌ها با استفاده از مجموعه‌ای از اسناد، بیش‌ترین شباهت پیوندها را برای واژگان متوالی پرس‌وجوها تخمین زدند. در مدل، فقط زوج‌واژگان مورد توجه قرار گرفتند. این الگوریتم درعمل قابل استفاده نبود؛ زیرا استخراج تمام ساختارهای پیوند برای یک پرس‌وجو در زمان جستجو بسیار زمان‌بر بود [20].

[1] Term Frequency
[2] Inverse Document Frequency
[3] Phrase Frequency
[4] Rijsbergen
[5] Maximum Spanning Tree
[6] Nallapati
[7] Allan
[8] Keen

[9] Croft
[10] Metzler
[11] Dang
[12] Song
[13] Bigram
[14] Unigram
[15] Gao
[16] Dependence Language Model




[ DOI: 10.29252/jsdp.15.4.71 ]

گروه دیگری از پژوهش‌گران روش‌هایی برای افزودن وابستگی واژگان در مدل‌های احتمالی پیشنهاد دادند. برای مثال رسولف[1] و ساوی در سال ۲۰۰۳ روش BM25 را بر اساس وابستگی واژگان توسعه دادند [21]. آن‌ها تابعی از فاصله را جایگزین پارامتر TF کردند. در روش آن‌ها فاصلهٔ زوج‌واژگان در یک پنجرهٔ پنج‌واژه‌ای محاسبه شده است. هی[2] و همکارانش در سال ۲۰۱۱ از یک پنجره برای شمارش فرکانس n واژه‌ای[3] در یک سند استفاده کردند و BM25 را تغییر دادند که از فرکانس برای محاسبه امتیاز استفاده کند. معیار فاصله در مدل پیشنهادی آن‌ها، دست‌کم تعداد واژه‌ای بود که یک دنباله از واژگان شامل تمام واژگان پرس‌وجو را از هم جدا می‌کرد [20]. در الگوریتم پیشنهادی از این ایده استفاده شده است و پارامتر فرکانس عبارت (PF) تعریف می‌شود تا به جای TF در الگوریتم‌های رتبه‌بندی مورد استفاده قرار گیرد.

ایخُف و همکارانش با استفاده از ابزاری آماری به نام کُپولاس[4] مدل آماری خود را توسعه دادند [22]. آن‌ها با استفاده کُپولاس احتمال هم‌رخدادی واژگان پرس‌وجو را به دست آوردند. مزیت اصلی روش آن‌ها سرعت بالای آن بود.

باتچر[5] و همکارانش در سال ۲۰۰۶ مدل تجمعی را پیشنهاد دادند که به‌طور مجزا امتیاز مجاورت پرس‌وجو را برای هر واژه محاسبه می‌کند [23]. الگوریتم آن‌ها به‌طور خاص براساس فایل‌های نمایهٔ معکوس پیاده‌سازی شده است. در موقع پردازش فهرست موقعیت‌های[6] واژگان پرس‌وجو، اگر واژه‌ای تغییر کرد، فاصله به‌صورت تجمعی افزوده شود. کار مشابهی توسط تائو[7] و همکارش در سال ۲۰۰۷ انجام شد [24]. آن‌ها پنج ویژگی مربوط به فاصلهٔ واژگان را به‌صورت تجمعی محاسبه و به‌عنوان وزن واژگان در روش‌های مبتنی بر محتوا استفاده کردند. بهترین نتیجه زمانی به‌دست می‌آمد که معیار وزن بر اساس کمینه‌کردن فاصله بین تمام واژگان پرس‌وجو در نظر گرفته شود.

بر اساس نتایج تائو و زای، زائو[8] و یان[9] مدل زبانی واژگان مجاور (PLM[10]) را در سال ۲۰۰۹ پیشنهاد دادند [25]. در روش آن‌ها کمترین فاصله‌ها بین تمام واژگان پرس‌وجو در متن سند به‌دست آورده شده است و مجموع آن‌ها برای محاسبه امتیاز به کار برده می‌شود.

مدل CRTER[11] توسط زائو و همکارانش پیشنهاد شده است [26]، [27]. آن‌ها برای تعریف وابستگی از تقاطع واژگان پرس‌وجو استفاده کردند. واژگان نزدیک‌تر وزن بیش‌تری را دریافت می‌کنند. برای محاسبهٔ وزن، توابعی برای هر واژه در نظر گرفته و سپس محل تقاطع دو واژه مجاور به‌عنوان وزن با رابطه BM25 ترکیب می‌شود.

مطالب مرتبط دیگری هم مطرح است؛ برای مثال جستجوی عبارت به‌طور دقیق یعنی زمانی که کاربر با استفاده از علائم نقل‌قول عبارت را مشخص می‌کند. موضوع دیگر توسعهٔ پرس‌وجو[12] است. به‌عنوان مثال میائو و همکارانش در سال ۲۰۱۲ از مفهوم واژگان مجاور با استفاده از بازخورد کاربر، توسعهٔ پرس‌وجو را بهبود بخشیدند [28].

## ۳- اصطلاحات استفاده‌شده

**فاصله:** کمینه تعداد جابه‌جایی واژگان یک متن را برای تطبیق بر عبارت پرس‌وجو فاصله می‌نامیم. برای مثال اگر در سند ‘a c b’ عبارت ‘a b’ جستجو شود، فاصله یک خواهد بود؛ زیرا b موجود در سند باید یک جابه‌جایی داشته باشد تا بر ‘a b’ منطبق شود (برای سادگی، هر یک از حروف لاتین را یک واژه فرض کرده‌ایم. یعنی سند بالا دارای سه واژه a و c و b است). جستجوی عبارت ‘b a’ در سند بالا به دو جابه‌جایی برای تطبیق نیاز دارد و بنابراین فاصله برابر با دو می‌شود. در حالتی که فاصله صفر باشد، مفهوم آن تطابق کامل بدون نیاز به جابه‌جایی است. به‌عنوان مثال جستجوی ‘c b’ در سند بالا فاصله صفر را نتیجه می‌دهد.

البته این اصطلاح در روش‌های مختلفی که در بخش قبلی بیان شد، تعاریف متفاوتی دارد. برخی از پژوهش‌گران کمینه فاصله بین زوج‌واژگان را در نظر گرفته‌اند و برخی جمع فاصله و روش‌های دیگر برای محاسبه فاصله به‌کار برده‌اند.

در مدل MWRM سند وب به سه بخش بدنه، عنوان و متن پیوند تقسیم می‌شود. در این مدل برای واژه در بخش‌های مختلف، وزن‌های مختلفی در نظر گرفته شده و در جابه‌جایی واژگان از یک بخش به بخش دیگر وزن آن‌ها نیز در محاسبهٔ فاصله مؤثر است. شکل (۱) نمایی از این سه بخش و فاصلهٔ فرضی بین آن‌ها را نشان می‌دهد.

[1] Rasolofo
[2] He
[3] n-grams
[4] Copulas
[5] Büttcher
[6] Posting List
[7] Tao
[8] Zhao
[9] Yun
[10] Proximity Language Model
[11] CRoss TErm Retrieval
[12] Query Expansion




[ DOI: 10.29252/jsdp.15.4.71 ]

**تابع s(d):** تابع s(d) تابعی است که فاصله را به امتیاز تبدیل می‌کند. ورودی این تابع فاصله است. این تابع باید با افزایش فاصله کاهش یابد و همچنین در صفر برابر یک باشد. یعنی:

$s(0) = 1$

If $d1 > d2$ then $s(d1) < s(d2)$

به‌عنوان مثال $s(d)=1/(d+1)$ ویژگی‌های بالا را دارد و در ارزیابی عبارت قابل استفاده است [29]. یا در [21] امتیاز به‌صورت عکس مربع فاصله محاسبه شده است یا در [24] از یک رابطهٔ لگاریتمی استفاده شده است.

**PF:** برای مطابقت‌دادن تعداد عبارات در یک سند با روش‌هایی که از TF برای محاسبه امتیاز استفاده می‌کنند پارامتری به نام PF تعریف می‌شود که مخفف Phrase Frequency است. TF همواره یک عدد صحیح است ولی PF می‌تواند یک عدد اعشاری باشد. PF به‌صورت زیر قابل محاسبه است:

$$PF = \sum_{i=1}^{n} s(d_i) \qquad (1)$$

که در آن n تعداد عبارت پرس‌وجویی است که در سند پیدا شده و $d_i$ مقدار فاصله برای عبارت i ام است. در حالت بهینه ترکیب عبارات باید به‌گونه‌ای انتخاب شود که PF بیشینه باشد.

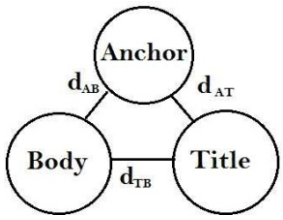
**QTM**[1]: بیشتر موتورهای جستجو برای پرس‌وجوهای ورودی بیشینه تعداد واژه را مشخص می‌کنند. برای مثال در موتور جستجوی گوگل بیشینه تعداد واژگان برابر ۳۲ است [30]. این متغیر بیشینه تعداد واژه مجاز در پرس‌وجو را مشخص می‌کند.

## ۴- الگوریتم پیشنهادی

در الگوریتم پیشنهادی دو موضوع را دنبال می‌کنیم، یکی محاسبهٔ فاصله و دیگری محاسبهٔ PF است. همچنین مقدار IDF هم بر اساس تعریف PF ارائه می‌شود. در موقع محاسبهٔ فاصله فرض می‌کنیم که در سند فقط یک‌بار واژگان عبارت ظاهر می‌شوند و سپس مکانی را برای جمع‌شدن واژگان می‌یابیم که جابه‌جایی‌ها کمینه باشد. در بخش دوم، الگوریتمی با استفاده از روش جستجوی کامل ارائه می‌دهیم که تمام ترکیب‌های عبارات را استخراج کند و ترکیبی را که شامل بیشترین امتیاز می‌شود، پیدا کند تا از آن برای محاسبه PF استفاده شود.

[1] Query Maximum Terms

قابل ذکر است که در بخش ۱، فرض فقط یک‌بار مشاهده واژگان در سند برای کاهش پیچیدگی است و با استفاده روش جستجوی کاملی که در بخش ۲ بیان می‌شود تمام تکرارهای واژگان عبارت در سند در نظر گرفته می‌شوند.




**(شکل-۱): نمایی از بخش‌های مختلف سند و فاصله فرضی بین این بخش‌ها**

**(Figure-1): An overview of the different parts of the document and the hypothetical distance between these sections**

### ۴-۱- الگوریتم محاسبهٔ فاصله

در موقع انطباق یک عبارت در یک سند بر اساس موقعیت‌ها و وزن‌هایی که واژگان پرس‌وجو در سند دارند، فاصله محاسبه می‌شود. در اینجا فاصله برابر با کمترین جابه‌جایی وزن‌دار واژگان پرس‌وجو در سند برای ساختن عبارت پرس‌وجو است. برای مثال اگر در متن ’a d f c d b e‘ عبارت ’a b c‘ را جستجو کنیم؛ در یک حالت (با فرض وزن مساوی برای واژگان) می‌توان a را ثابت در نظر گرفت و b و c را جابه‌جا کرد که در این حالت تعداد جابه‌جایی برای b چهار است و برای c یک است و درنتیجه مجموع جابه‌جایی (فاصله) پنج می‌شود؛ اما اگر c را ثابت بگیریم، مجموع جابه‌جایی چهار می‌شود؛ زیرا b به سه جابه‌جایی نیاز دارد تا قبل از c قرار گیرد و a تنها به یک جابه‌جایی نیاز دارد تا قبل از b قرار گیرد.

به‌عنوان مثال دیگر، اگر پرس‌وجو ’a b c‘ باشد و در سند مفروض a در مکان ۱۰۰ با وزن دو و b در مکان ۳۰۰ با وزن دو و c در مکان ۲۰۰ با وزن سه قرار گرفته باشد، یعنی سند به‌صورت ’a….c…b….‘ باشد، با استفاده از لم ۱ و لم ۲ و قضیه یک ثابت می‌شود. اگر مرکز ثقل ثابت باشد و واژگان دیگر جابه‌جا شوند، کم‌ترین جابه‌جایی صورت گرفته است؛ یعنی در مثال بالا باید c ثابت باشد و a و b جابه‌جا شوند تا پرس‌وجو ساخته شود. هزینهٔ جابه‌جایی در این حالت برابر با ۳۹۸ است؛ زیرا b به میزان ۱۰۱ جابه‌جایی نیاز دارد تا قبل از c قرار گیرد و با توجه به وزن دو، هزینه جابه‌جایی آن ۲۰۲ است. واژه a نیز به ۹۸ جابه‌جایی نیاز دارد تا قبل از b قرار گیرد و با توجه به وزن آن هزینه جابه‌جایی آن ۱۹۶ و در مجموع هزینه ساخت پرس‌وجو ۳۹۸ است.



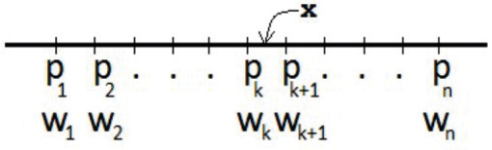

[ DOI: 10.29252/jsdp.15.4.71 ]

هم‌چنین اگر واژگان پرس‌وجو در بیش از یک بخش ظاهر شده باشند، دو موضوع اضافه می‌شود. موضوع نخست، یک فاصلهٔ فرضی است که باید بین بخش‌ها لحاظ و دیگر این‌که برای هر واژه در هر بخش باید وزن خاصی در نظر گرفته شود. برای این‌که حجم محاسبات کاهش یابد فرمول‌هایی برای محاسبهٔ فاصله ارائه می‌شود؛ اما برای مشخص‌کردن فاصلهٔ فرضی بین بخش‌ها و وزن واژگان در بخش‌های مختلف باید از یکی از شیوه‌های آموزش استفاده شود. در اینجا از الگوریتم ژنتیک برای محاسبهٔ فاصله‌ها و وزن‌های بهینه استفاده شده است. روش کار به این صورت است که وزن‌های واژگان به‌عنوان ژن‌های الگوریتم ژنتیک در نظر گرفته شده‌اند. در ابتدا وزن‌های واژگان به‌صورت تصادفی مقداردهی می‌شوند و سپس با بهینه‌کردن مقدار تابع برازش وزن‌های بهینه به‌دست می‌آیند. تابع برازشی برابر مقدار معیار دقّت **p@n** به‌ازای خروجی مدل پیشنهادی است و هدف الگوریتم ژنتیک تنظیم وزن‌های واژگان به‌صورتی است که بتوان مقدار بیشنه **p@n** را به‌دست آورد. مقدار بیشنهٔ آن برابر یک است.

در ادامه ثابت می‌شود که مرکز ثقل وزن و موقعیت واژگان عبارت در متن بهترین مکان برای انتخاب محور جابه‌جایی‌ها است و باید تمام واژگان را در آنجا جمع و فاصله را بر اساس تعداد این جابه‌جایی‌ها محاسبه کرد.

**لم ۱:** محل تجمع واژگان سند در مدل MWRM موقعیت‌های بینابینی واژگان پیداشده در سند نمی‌تواند باشد (موقعیت مکانی محل تجمع واژگان، موقعیت یکی از واژگان پرس‌وجو در سند است).

**اثبات:** هدف از این لم محدودکردن نقاطی است که ممکن است، به‌عنوان محل تجمع واژگان در نظر گرفته شود. برای اثبات، فرض می‌شود واژگان پرس‌وجو در سند بر اساس موقعیتشان در سند مرتب شده‌اند. $p_i$ موقعیت i-ام واژه در سند با وزن $w_i$ فرض شده است (لزوماً واژه i-ام پرس‌وجو بر موقعیت i-ام کلمهٔ پیداشده در سند منطبق نیست). شکل (۲) نمایی از یک تطبیق واژگان پرس‌وجو در سند را نشان می‌دهد.




**(شکل-۲): موقعیت واژگان پرس‌وجو در سند(p) و وزن آن‌ها(w)**
**(Figure 2): The position of the query words in the document (p) and their weight (w)**

اگر $d_x$ هزینهٔ جابه‌جایی واژگان برای تمرکز در موقعیت x سند فرض شود، رابطه آن به‌صورت زیر خواهد بود:

$$d_x = w_1(x-p_1) + w_2(x-p_2) + \cdots + w_{k+1}(p_{k+1}-x) + \cdots + w_n(p_n-x) \quad (۲)$$

و $d_{x+1}$ هزینهٔ جابه‌جایی واژگان برای تمرکز در موقعیت x+1 سند به‌صورت زیر خواهد بود:

$$d_{x+1} = w_1(x+1-p_1) + \cdots + w_{k+1}(p_{k+1}-x-1) + \cdots + w_n(p_n-x-1) \quad (۳)$$

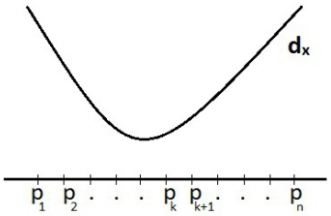
تفاضل $d_x$ و $d_{x+1}$ را با $\Delta_x$ نشان می‌دهیم و به‌صورت زیر محاسبه خواهد شد:

$$\Delta_x = d_{x+1} - d_x = w_1 + \cdots + w_k - w_{k+1} - \cdots - w_n \quad (۴)$$

رابطهٔ (۴) نشان می‌دهد که $\Delta_x$ برای نقاط بین موقعیت‌های متوالی واژگان پرس‌وجو در سند، مقدار ثابتی است و بنابراین $d_x$ یک پاره‌خط و برای پاره خط کمترین مقدار یا ابتدای آن $(p_k)$ یا انتهای آن $(p_{k+1})$ است.

**لم ۲:** تابع $d_x$ به طور تقریبی سهموی شکل است و شکلی مشابه شکل (۳) دارد.

**اثبات:** طبق لم ۱ مشخص شد که $d_x$ در نقاط بینابینی به‌صورت پاره‌خط است و بنابراین شکل $d_x$ فقط بستگی به مقدار آن‌ها در $p_i$ ها دارد.




**(شکل-۳): شکل تقریبی تابع $d_x$**
**(Figure-3): The approximate form of the dx function**

برای نمایش تغییرات $d_x$ در $p_k$-های متوالی $\varphi_k$ را به‌صورت زیر تعریف می‌کنیم:

$$\varphi_k = d_{p_{k+1}} - d_{p_k} \quad (۵)$$

کافی است نشان دهیم $\varphi_k$ در ابتدا روندی نزولی دارد و سپس صعودی می‌شود. با جاگذاری d در رابطهٔ (۲) مقدار $\varphi_k$ به‌صورت زیر ساده می‌شود:

$$\varphi_k = (w_1 + \cdots + w_k - w_{k+1} - \cdots - w_n) \times (p_{k+1} - p_k) \quad (۶)$$

با توجه به این‌که در رابطه (۶) ضریب دوم همواره مثبت است، علامت $\varphi_k$ فقط وابسته به ضریب نخست است. برای پیداکردن روند $\varphi_k$ علامت چند جمله نخست آن را به‌دست می‌آوریم.


[ DOI: 10.29252/jsdp.15.4.71 ]

$$\begin{aligned} \text{sign}(\varphi_1) &= \text{sign}((w_1 - w_2 - w_3 - \cdots - w_n) \\ \text{sign}(\varphi_2) &= \text{sign}(w_1 + w_2 - w_3 - \cdots - w_n) \\ \text{sign}(\varphi_3) &= \text{sign}(w_1 + w_2 + w_3 - \cdots - w_n) \end{aligned} \quad (۷)$$

همان‌طور که رابطه (۷) نشان می‌دهد، علامت $\varphi_k$ در ابتدا منفی است و هر بار مقداری افزایش می‌یابد و در نهایت با علامت مثبت پایان می‌پذیرد و بنابراین $d_x$ در ابتدا نزولی است و سپس صعودی می‌شود و درنتیجه $d_x$ سهموی شکل است.

**قضیه ۱**: در مدل MWRM کمترین مقدار $d_x$ در مرکز ثقل واژگان پرس‌وجو در سند حاصل می‌شود.

**اثبات:** طبق لم ۱ نقطه بهینه برای $d_x$ مکانی از سند است که دربردارندهٔ یکی از واژگان پرس‌وجو می‌باشد و طبق لم ۲ مقدار $d_x$ در ابتدا نزولی و سپس صعودی می‌شود و بنابراین در لحظه‌ای که روند نزولی پایان می‌پذیرد و روند صعودی آغاز می‌شود، کمترین مقدار $d_x$ اتفاق می‌افتد؛ یعنی زمانی که $\varphi_k$ نزدیک صفر است. اگر $p_m$ نقطه‌ی مفروض باشد، داریم:

$$\begin{aligned} \varphi_k = (w_1 + \cdots + w_m - w_{m+1} - \cdots - w_n) &\cong 0 \\ \Rightarrow\ w_1 + \cdots + w_m &\cong w_{m+1} + \cdots + w_n \end{aligned} \quad (۸)$$

درنتیجه $p_m$ نقطه‌ای است که وزن واژگان دو طرف آن به‌طورتقریبی مساوی است. به‌عبارت دیگر بهترین نقطه برای تجمع واژگان سند مرکز ثقل واژگان است.

نتیجهٔ ارزشمند قضیه ۱ کاهش قابل توجه پیچیدگی الگوریتم رتبه‌بندی بر اساس مدل MWRM است. برای پیاده‌سازی الگوریتم آن می‌توان در ابتدا $\varphi_1$ را محاسبه و سپس با افزودن $2w_2$ به آن $\varphi_2$ را محاسبه کرد و همین‌طور ادامه داد و مقادیر بعدی را به‌دست آورد. به‌محض تغییر علامت $\varphi_k$ از منفی به مثبت، $p_k$ را به‌عنوان نقطه بهینه برگرداند.

بر اساس نتایج قضیه و لم‌های بالا الگوریتم (۱) برای محاسبهٔ فاصله ارائه شده است.

```
Algorithm Distance
01: Input Query q, Text T
02: Output Distance
03: Assumption
04: q include <t1, t2,…, tm>
05: T include <(t1,p1), (t2,p2),… , (tm, pm)>
06: Begin Algorithm
07: sort <p1, p2, …, pm>
08: Mid← Center of mass of <p1, p2, …, pm>
09: For i=1 to m
10: move (ti, pi) to Mid +i-1
11: Distance ← Distance +abs(Mid +i-1-pi) * wi
12: Return Distance
```

**(الگوریتم-۱): الگوریتم محاسبهٔ فاصله**

**(Algorithm -1): Distance calculation algorithm**

در الگوریتم (۱) فرض شده که پرس‌وجو دارای m واژه است و واژگان عبارت به‌صورت $t_1,t_2,\ldots,t_m$ هستند. همچنین موقعیت واژه $t_i$ در سند را $p_i$ در نظر می‌گیریم ($p_i$ها لزوماً مرتب‌شده نیستند).

## ۲-۴- الگوریتم محاسبهٔ PF

یکی از الگوریتم‌هایی که می‌تواند PF را محاسبه کند الگوریتم جستجوی کامل است. در این روش تمام ترکیب‌هایی که می‌توان با استفاده از آن عبارت را ایجاد کرد استخراج می‌شود و شکلی که مجموع امتیاز آن بالاتر است انتخاب می‌شود. در [31] و [32] الگوریتم‌هایی برای استخراج بهترین انطباق‌های یک عبارت در یک سند ارائه شده که دارای اشکالاتی است. برای مثال در تمام الگوریتم‌های ارائه‌شده، در جستجوی ‘a b c’ دو سند ‘b b b a c’ و ‘b a c’ یک امتیاز دریافت می‌کنند؛ درحالی‌که سند نخست مرتبط‌تر است. بنابراین در اینجا از جستجوی کامل برای استخراج بهترین تطبیق‌ها استفاده شده است.

در روش پیشنهادی ابتدا با استفاده از «and query» اسنادی که تمام واژگان عبارت را دارا هستند، استخراج می‌شوند؛ سپس با استفاده از نمایه معکوس به‌ازای تمام واژگان عبارت، تمام جایگشت‌های ممکن از انطباق‌های عبارت بررسی و سپس برای هر جایگشت جمع فاصله‌ها محاسبه می‌شود. پیداکردن جایگشتی که جمع بیشینهٔ فاصله را تولید کند هدف اصلی است.

الگوریتم (۲) روش محاسبه PF را نشان می‌دهد. در این الگوریتم متغیر Max برای نگهداری بیشترین مقدار مجموع فاصله مورد استفاده قرار گرفته است. کار اصلی در این لگوریتم تولید تمام ترکیبات عبارت مورد جستجو در متن است.

```
Algorithm PF
01: Input Query q, Document d
02: Output PF
03: Assumption
04: q is <t1, t2,…, tm>
05: d includes <t1,(P11,P12,…)> , <t2,(P21,P22,…)>
,…, <tm,(Pm1,Pm2,…)>
06: Begin Algorithm
07: Max← 0
08: Repeat
09: Make a new permutation
10: S← Calculate Distance of all phrases
11: If S>Max Then
12: Max←S
13: Until no new permutation
14: Return Max
```

**(الگوریتم-۲): الگوریتم محاسبه PF برای یک عبارت**

**(Algorithm-2): An algorithm of calculating PF for a phrase**





[ DOI: 10.29252/jsdp.15.4.71 ]

در الگوریتم(۲) از متغیر $P_{ij}$ برای مشخص‌کردن j-اُمین موقعیت واژه i-اُم پرس‌وجو در سند استفاده شده است. خط ۹ الگوریتم، وظیفهٔ تولید جایگشت‌های مختلف تطابق واژگان عبارت مورد جستجو را با واژگان معادل در سند بر عهده دارد. برای مثال نخستین جایگشت می‌تواند به‌صورت زیر انتخاب شود:

Phrase 1: $<t_1,P_{11}>,<t_2,P_{21}>,\ldots,<t_n,P_{n1}>$
Phrase 2: $<t_1,P_{12}>,<t_2,P_{22}>,\ldots,<t_n,P_{n2}>$
⋮

برای جایگشت بعدی می‌توان موقعیت $t_1$ در Phrase1 را تغییر داد و $P_{12}$ را در نظر گرفت و به‌طور مشابه موقعیت $t_1$ در Phrase 2 را $P_{11}$ قرار داد و یک جایگشت جدید به‌دست آورد.

الگوریتم در خط ۱۰ برای تک‌تک عبارات استخراج‌شده در هر جایگشت مقدار کمینه جابه‌جایی را محاسبه و جمع امتیازهای حاصل را در متغیر S ذخیره می‌کند. این الگوریتم با این‌که در بدترین حالت به‌دلیل محدودبودن QTM خطی است، ولی به‌خاطر اهمیت ضریب خطی در جستجوی انبوه باید بهینه شود. برای بهینه‌کردن این الگوریتم می‌توان از حرص‌کردن براساس فاصلهٔ واژگان استفاده کرد. یکی از روش‌ها برای حرص‌کردن استفاده از همین مفهوم فاصله است. مشخص است که از یک فاصله‌ای بیشتر ارتباط بین واژگان به‌طورتقریبی قطع می‌شود و با این فرض می‌توان فاصله واژگان را محدود و بخشی از درخت را حرص کرد. به‌عنوان مثال اگر دو واژه در فاصلهٔ هزار از یکدیگر قرار گرفته باشند، امتیاز قابل توجهی را تولید نمی‌کنند و بنابراین می‌توان بررسی آن حالت را ادامه نداد.

روش دیگر برای بهینه‌کردن الگوریتم بالا پردازش آماری پرس‌وجوها و استخراج مرز تصمیم گیری براساس tf و idf است؛ یعنی برای idf های پایین tf های کم، پردازش نشود.

## ۳-۴- محاسبه IDF

در موقع جستجوی یک عبارت پارامتر، IDF نیز باید به‌شکل مناسبی تعریف شود تا به‌درستی میزان اهمیت عبارت را مشخص کند. IDF پارامتری است که با لگاریتم وارون DF رابطهٔ مستقیمی دارد و DF نشان‌گر تعداد اسنادی است که شامل عبارت مورد نظر است. هر چه تعداد اسناد کمتر باشد، اهمیت عبارت بیشتر است. یکی از مشهورترین تعاریف برای IDF به‌صورت رابطهٔ زیر است:

$$idf_t = \log\frac{N}{1+df_t} \qquad (۹)$$

که در آن N تعداد کل اسناد است و $df_t$ تعداد سندی را نشان می‌دهد که شامل واژه t هستند.

اگر بخواهیم به شکل مشابه، IDF را برای یک عبارت تعریف کنیم، باید DF را برای یک عبارت محاسبه کنیم. برای محاسبه DF یک عبارت می‌توان اسناد را به دو دسته تقسیم کرد:

- اسنادی که PF بزرگ‌تر یا مساوی یک دارند؛
- اسنادی که PF کوچک‌تر از یک دارند.

واضح است که در شمارش اسناد شامل عبارت، اسنادی که PF بزرگ‌تر یا مساوی یک دارند، باید شمارش شوند؛ اما اسنادی که PF کمتر از یک دارند هم باید شمارش شوند؛ ولی با یک ضریب کمتر از یک. رابطهٔ ۱۰ را برای این منظور به‌صورت زیر تعریف می‌کنیم و در رابطهٔ ۱۱ مقدار df یک عبارت محاسبه می‌شود:

$$\boldsymbol{F}(d,q) = \begin{cases} 1 & PF \geq 1 \\ PF & PF < 1 \end{cases} \qquad (۱۰)$$

$$df_q = \sum_d F(d,q) \qquad (۱۱)$$

حال با استفاده از df به‌سادگی می‌توان از رابطه (۹) مقدار idf را محاسبه کرد. الگوریتم (۳) مراحل محاسبهٔ IDF را نشان می‌دهد.

Algorithm IDF
01: Input Query q
02: Output IDF
03: Assumption
04: q include $<t_1, t_2, \ldots, t_m>$
05: N is total document in corpus
06: Begin Algorithm
07: DF ← 0
08: For every document $d_i$
09: PF ← calculate PF(q, $d_i$)
10: If PF>=1 Then
11: DF ← DF + 1
12: Else
13: DF ← DF + PF
14: IDF ← Log(N/(1 + DF))
15: Return IDF

**(الگوریتم-۳): الگوریتم محاسبه IDF برای یک عبارت**
**(Algorithm-3): An algorithm of calculating an IDF for a phrase**

با یک تغییر ساده می‌توان الگوریتم (۳) را در الگوریتم (۲) یعنی الگوریتم محاسبه PF ادغام کرد؛ یعنی می‌توان در موقع محاسبه PF برای کلیهٔ اسناد، مقدار IDF عبارت را نیز محاسبه کرد. البته مسئلهٔ کارایی این الگوریتم‌ها نسبت به روش‌های سنّتی برای استخراج TF و IDF بسیار متفاوت خواهد بود.

در روش‌های سنّتی، TF هر واژه برای هر سند به‌سادگی در دسترس است و هم‌چنین IDF ها برای هر واژه در زمان نمایه می‌تواند محاسبه و ذخیره شود و در موقع جستجو بدون هیچ پردازشی در اختیار بخش رتبه‌بندی قرار




[ DOI: 10.29252/jsdp.15.4.71 ]

داده شود؛ اما در این روش این دو پارامتر باید در موقع جستجو با یک پردازش محاسبه شود که نسبت به روش سنّتی از نظر زمانی کندتر خواهد بود.

## ۴-۴- پیچیدگی زمانی الگوریتم

درصورتی‌که بخواهیم یک عبارت را تطبیق دهیم، طبق قضیه ۱ باید مرکز ثقل را بیابیم. اگر برای یافتن مرکز ثقل از مرتب‌سازی سریع استفاده شود، پیچیدگی زمانی O(nlogn) است که n نشان‌دهندهٔ تعداد مقادیری است که می‌خواهیم مرکز ثقل آن‌ها را حساب کنیم. البته الگوریتم‌های سریع‌تری برای یافتن میانه نیز وجود دارد که پیچیدگی زمانی آن نسبت به n خطی است [33] ولی خواهیم دید که تأثیری روی پیچیدگی محاسبهٔ PF ندارند.

همان‌طور که در بخش ۲ دیدیم، موتورهای جستجو بیشینه تعداد واژه را در پرس‌وجو مشخص می‌کنند که ما آن را با مقدار ثابت QTM نشان دادیم. پس پیچیدگی زمانی پیداکردن فاصله به‌صورت O(QTM*log(QTM)) خواهد بود و با توجه به ثابت‌بودن QTM این پیچیدگی به‌صورت O(1) است.

پیچیدگی زمانی محاسبه PF طبق الگوریتم ارائه‌شده، رابطهٔ مستقیمی با تعداد جایگشت‌های عبارت در متن دارد. تعداد جایگشت یک پرس‌وجو $t_1t_2\dots t_m$ به‌صورت رابطه (۱۲) خواهد بود.

$$\boldsymbol{T(m)=\prod_{i=1}^{m} tf(t_i)} \qquad (۱۲)$$

برای پیچیدگی زمانی محاسبهٔ PF باید سه حالت بهترین، بدترین و حالت متوسط را به‌دست آوریم. واضح است که بهترین حالت، زمانی است که tf تمام واژگان عبارت یک باشد و درنتیجه پیچیدگی زمانی در بهترین حالت O(1) است.

هم‌چنین بدترین حالت زمانی اتفاق می‌افتد که متن به‌طور مطلق شامل فقط واژگان عبارت باشد. می‌توان ثابت کرد که پیچیدگی زمانی محاسبهٔ PF در حالت میانگین O(1) خواهد بود. در ادامه با استفاده از لم ۳ و قضایای ۲ و ۳ پیچیدگی زمانی محاسبهٔ PF در بدترین حالت و حالت متوسط ارائه شده است.

**لم ۳:** اگر $n=\sum_{i=1}^{m} a_i$ ثابت باشد، آن‌گاه $\prod_{i=1}^{m} a_i$ زمانی بیشینه است که $a_i$ها مساوی باشند.

**اثبات:** این لم با استفاده از ضرایب لاگرانژ به‌سادگی امکان‌پذیر است [34].

**قضیه ۲:** پیچیدگی زمانی PF در بدترین حالت $O(n^{QTM})$ است که n تعداد واژگان سند است.

**اثبات:** مشخص است بدترین حالت زمانی اتفاق می‌افتد که تک‌تک واژگان سند در واژگان عبارت باشد پس $n=\sum_{i=1}^{m} tf(t_i)$ می‌باشد. طبق لم ۳ برای این‌که حاصل‌ضرب tf ها بیشینه باشد $tf(t_i)$ها باید مساوی باشند. پس در بدترین حالت داریم:

$$tf(t_i)=\frac{n}{m} \qquad (۱۳)$$

با توجه به این‌که حداکثر مقدار m برابر با QTM می‌باشد بنابراین پیچیدگی زمانی محاسبه PF در بدترین حالت $O(n^{QTM})$ می‌باشد.

**قضیه ۳:** پیچیدگی زمانی PF در حالت متوسط در O(1) می‌باشد.

**اثبات:** برای حالت متوسط باید امید ریاضی tf ها را پیدا کرد. می‌توان اثبات کرد که توزیع هر واژه در هر سند دارای توزیع پواسون است [35] و بدیهی است که تعداد تکرار هر واژه در هر سند یک توزیع دوجمله‌ای است. این موضوع با استفاده از مجموعه داده‌های TREC2003 و TREC2004 [36] نیز قابل بررسی می‌باشد. بنابراین داریم $E(tf)=\mu$ که $\mu$ میانگین توزیع پوآسون می‌باشد.

اگر فرض کنیم tf ها از هم مستقل هستند امید ریاضی حاصل‌ضرب را می‌توان به حاصل‌ضرب امید ریاضی‌ها تبدیل کرد [37]؛ یعنی حد بالایی تعداد جای‌گشت‌ها در حالت متوسط را می‌توان از رابطه (۱۴) به دست آورد:

$$\boldsymbol{E}\left[\prod_{i=1}^{m} tf(t_i)\right]=\prod_{i=1}^{m} E(tf(t_i))=\mu^m \le \mu^{MTQ} \qquad (۱۴)$$

و با توجه به ثابت بودن $\mu$ و QTM پیچیدگی زمانی محاسبه PF در حالت میانگین O(1) خواهد بود.

## ۵- نتایج ارزیابی

به‌منظور ارزیابی الگوریتم پیشنهادی از داده آزمایش موتور جستجوی پارسی‌جو[1] که بخشی از داده واقعی موتور یادشده می‌باشد، استفاده شده است. این داده آزمایش مشتمل بر حدود چهارصدهزار سند صفحه وب و حدود پنجاه پرس‌وجوی فارسی ارزیابی‌شده که کمینه، متوسط و بیشینه تعداد واژگان پرس‌وجوها به‌ترتیب برابر ۲، ۳,۱۶۹۰۱۴ و ۸ است. هم‌چنین از معیار دقّت[2] در مکان n-ام (p@n) و میانگین متوسط دقّت

[1] http://www.parsijoo.ir
[2] Precision




[ DOI: 10.29252/jsdp.15.4.71 ]

(MAP[1]) برای مقایسه الگوریتم پیشنهادی با دو الگوریتم دیگر استفاده شده است.

معیار دقّت در مکان n-ام یا P@n نشان‌دهندهٔ نسبت تعداد اسناد مرتبط در n سند نخست رتبه‌بندی نهایی برای هر پرس‌وجو به n است. فرمول P@n در معادلهٔ زیر نشان داده شده است:

$$P@n = \frac{1}{n}\sum_{j=1}^{n} r_j \qquad (۱۵)$$

که $\mathbf{r_j}$ نشان‌دهندهٔ مرتبط‌بودن سند j-ام در رتبه‌بندی نهایی است.

هم‌چنین میانگین متوسط دقّت یا MAP برای هر پرس‌وجو به‌عنوان میانگین مقادیر P@n-ها برای همهٔ اسناد مرتبط تعریف می‌شود:

$$AP = \frac{1}{|D_+|}\sum_{j=1}^{N} r_j \times P@j \qquad (۱۶)$$

N نشان‌دهندهٔ تعداد کل اسناد؛ $\mathbf{D_+}$ نشان‌دهندهٔ تعداد اسناد مرتبط و $\mathbf{r_j}$ نشان‌دهندهٔ مرتبط‌بودن سند j-ام است؛ متوسط میانگین دقّت (MAP) به‌عنوان میانگین مقادیر AP همهٔ پرس‌وجوهای ارائه شده است.

مدل MWRM با دو مدل دیگر مقایسه شده است: مدل BM25F و مدل CRTER. مدل BM25F مشابه BM25 است با این تفاوت که در BM25F برای واژگان پیداشده در بخش‌های مختلف وزن‌های مختلفی در نظر گرفته می‌شود؛ درحالی‌که مقدار پارامترهای مدل برای هر بخش نیز قابل تنظیم است[38]. در مدل CRTER تأثیر واژگان بر واژگان مجاور توسط توابعی مثل گوسی، مثلثی، دایره‌ای و کسینوسی تخمین زده می‌شود [39]. به این توابع تابع هسته[2] گفته می‌شود. هر تابع هسته باید خصوصیات زیر را داشته باشد:

- منفی نباشد؛
- پیوسته باشد؛
- متقارن باشد؛
- یکنواخت باشد؛
- مقدار آن برای فاصله صفر یک باشد.

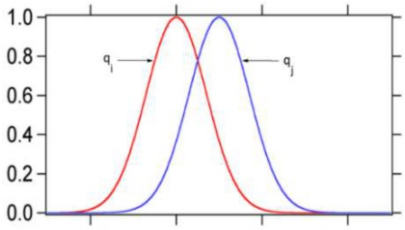
تقاطع دو واژه $\mathbf{q_i}$ و $\mathbf{q_j}$ زمانی رخ می‌دهد که این دو واژه در یک سند نزدیک هم ظاهر شوند و در نتیجه تابع هسته آن‌ها نقطه اشتراک دارد. نقطهٔ اشتراک، عددی است بین صفر و یک که برای فاصله‌های کمتر، بزرگ‌تر است. شکل (۴) نمونه‌ای از این تقاطع را نشان می‌دهد.

پارامترهای tf و df در بیش‌تر مدل‌ها به‌طورمستقیم یا غیرمستقیم مورد استفاده قرار می‌گیرند. در مدل CRTER تعاریف جدیدی به‌صورت زیر برای این دو پارامتر ارائه شده است:

$$\boldsymbol{t}f(q_{ij};D) = \sum_{m=1}^{tf_i}\sum_{n=1}^{tf_j} \mathrm{Kernel}\left(\frac{1}{2}\mathrm{dist}(Pos_m, Pos_n)\right) \qquad (۱۷)$$

$$\boldsymbol{d}f(q_{ij}) = \sum \frac{tf(q_{ij};D)}{occur(q_{ij};D)} \qquad (۱۸)$$

$$\boldsymbol{o}ccur(q_{ij};D) = \sum_{m=1}^{tf_i}\sum_{n=1}^{tf_j} 1\{\mathrm{Kernel}\left(\frac{1}{2}\mathrm{dist}(Pos_m, Pos_n)\right) \neq 0\} \qquad (۱۹)$$




**(شکل-۴): تقاطع دو واژه $q_i$ و $q_j$ بر اساس دو تابع هسته گوسی**

**(Figure-4): The intersection of the two words qi and qj based on two Gaussian core functions**

در این روابط تابع dist برای محاسبهٔ فاصله بین دو واژه در سند مورد استفاده قرار گرفته است. در آزمایشی که در [27] انجام شده بهترین نتایج زمانی حاصل شده که از تابع مثلثی به‌عنوان تابع هسته استفاده شده است؛ سپس از توابع tf و df جدید در تابع BM25 استفاده می‌شود.

جدول (۱) وزن بخش‌های مختلف سند و مقدار پارامترهایی را که بر اساس مجموعه‌دادهٔ مفروض برای BM25F محاسبه شده است نشان می‌دهد. هم‌چنین جدول (۲) وزن واژگان در بخش‌های مختلف سند و فاصله فرضی بین این بخش‌ها را برای مدل MWRM نشان می‌دهد. برای محاسبه وزن واژگان در بخش‌های مختلف و فاصله بین آن‌ها از الگوریتم ژنتیک استفاده شده است. هدف تابع برازش الگوریتم ژنتیک، به‌دست‌آوردن بالاترین میزان معیار دقّت p@n برای رتبه‌بندی اسناد با استفاده از مدل پیشنهادی است.

**(جدول-۱): پارامترها و وزن‌های محاسبه‌شده برای مدل BM25F**

**(Table -1): Parameters and calculated weights for the BM25F model**

| پارامتر | مقدار |
|---|---|
| $k_1$ | **1.4** |
| $b_{title}$ | **0.1** |
| $b_{body}$ | **0.98** |
| $b_{anchor}$ | **6** |
| $v_{title}$ | **3.6** |
| $v_{body}$ | **1** |
| $v_{anchor}$ | **1.4** |

[1] Mean Average Precision

[2] Kernel Function





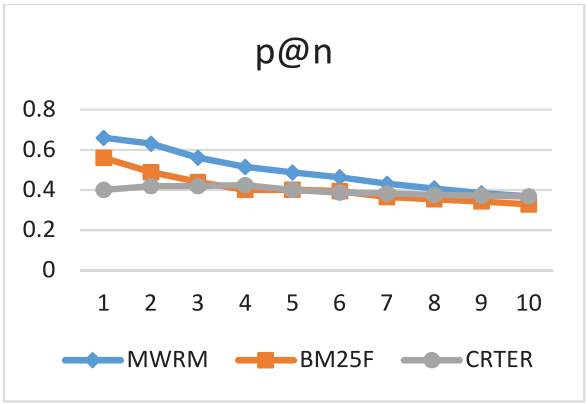

[ DOI: 10.29252/jsdp.15.4.71 ]

(جدول-۲): فاصله فرضی بین بخش‌های مختلف سند و وزن هر بخش محاسبه‌شده برای مدل MWRM

(Table-2): The hypothetical distance between the different document sections and the weight of each computed part for the MWRM model

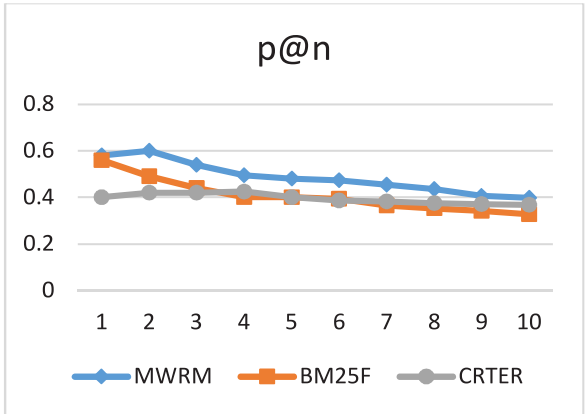
| پارامتر | مقدار |
|---|---|
| $d_{AB}$ | 1 |
| $d_{TB}$ | 0 |
| $d_{AT}$ | 5 |
| $V_{title}$ | 6.6 |
| $V_{body}$ | 1 |
| $V_{anchor}$ | 10.3 |

شکل‌های (۵)، (۶) و (۷) مقایسه‌ای بین سه الگوریتم جستجو را نشان می‌دهد. در این سه شکل از سه تابع متفاوت برای **s(d)** استفاده شده است.



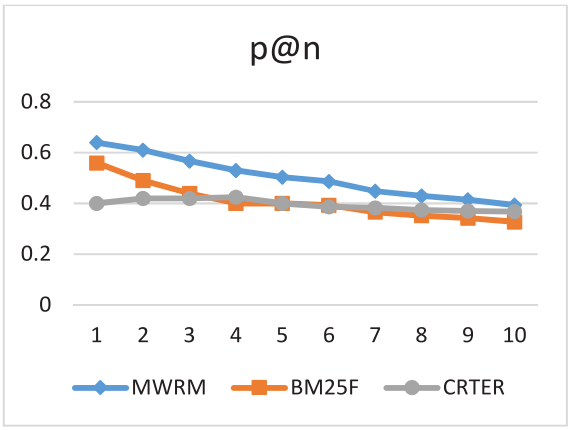

(شکل-۵): مقایسه بین الگوریتم پیشنهادی با الگوریتم BM25F و CRTER با استفاده از معیار P@n در حالت $s(d)=\frac{1}{d}$

(Figure-5): Comparison of the proposed algorithm with the BM25F and CRTER algorithms using the P@n criterion in $s(d)=\frac{1}{d}$




(شکل-۶): مقایسه بین الگوریتم پیشنهادی با الگوریتم BM25F و CRTER با استفاده از معیار P@n در حالت $s(d)=\frac{1}{d^2}$

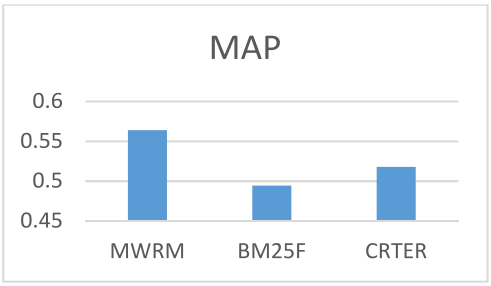
(Figure -6): Comparison of the proposed algorithm with the BM25F and CRTER algorithms using the P@n criterion in $s(d)=\frac{1}{d^2}$




(شکل-۷): مقایسهٔ بین الگوریتم پیشنهادی با الگوریتم BM25F و CRTER با استفاده از معیار P@n در حالت $s(d)=\frac{1}{d\sqrt{d}}$

(Figure-7): Comparison of the proposed algorithm with the BM25F and CRTER algorithms using the P@n criterion in $s(d)=\frac{1}{d\sqrt{d}}$

شکل (۸) نیز نشان می‌دهد که در حالت متوسط با استفاده از معیار MAP دقّت الگوریتم پیشنهادی بیشتر از دو الگوریتم دیگر است.




(شکل-۸): مقایسهٔ بین الگوریتم پیشنهادی و دو الگوریتم دیگر با استفاده از معیار MAP

(Figure-8): Comparison between the proposed algorithm and the other two algorithms using the MAP benchmark

## ۶- نتیجه‌گیری و کارهای آینده

در این مقاله یکی از موضوعات موتورهای جستجو یعنی جستجوی عبارت بدون علامت نقل‌قول بررسی شد. در جستجوی یک عبارت در یک متن، ممکن است، واژگان عبارت در جاهای مختلف متن موجود باشند و درنتیجه مفهوم فاصله تعریف شد که نشان‌دهندهٔ کمینهٔ جابه‌جایی برای تطبیق عبارت بود و با استفاده از فاصله برای محاسبه PF و IDF الگوریتم‌هایی ارائه و بین الگوریتم پیشنهادی و دو الگوریتم مشابه انجام شد.

به‌عنوان چشم‌انداز آینده می‌توان برای افزایش سرعت محاسبه PF و IDF الگوریتم‌های سریع‌تری ارائه داد. هم‌چنین موازی‌سازی نیز می‌تواند روشی برای افزایش سرعت محاسبهٔ PF و IDF قابل بررسی باشد.




[ DOI: 10.29252/jsdp.15.4.71 ]

## 7- References ۷- مراجع


[1] A. Z. Bidoki, "Effective Web Ranking and Crawling(in persian)," University of Tehran, 2009.

[2] R. Baeza-Yates and B. Ribeiro-Neto, "Modern information retrieval," *New York*, vol. 9, p. 513, 1999.

[3] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing and Management*, vol. 24, no. 5, pp. 513–523, 1988.

[4] S. E. Robertson, *Overview of the Okapi projects*, vol. 53, no. 1. MCB UP Ltd, 1997, pp. 3–7.

[5] Y. Zhang and A. Moffat, "Some Observations on User Search Behaviour.," *Austr. J. Intelligent Information Processing Systems*, vol. 9, no. 2, pp. 1–8, 2006.

[6] D. Bahle, H. Williams, and J. Zobel, "Compaction techniques for nextword indexes," in *String Processing and Information Retrieval, International Symposium on*, 2001, p. 33.

[7] H. E. Williams, J. Zobel, and D. Bahle, "Fast phrase querying with combined indexes," *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 4, pp. 573–594, 2004.

[8] A. Doucet and H. Ahonen-Myka, "An efficient any language approach for the integration of phrases in document retrieval," *Language resources and evaluation*, vol. 44, no. 1–2, pp. 159–180, 2010.

[9] I. H. Witten, A. Moffat, and T. C. Bell, *Managing gigabytes: compressing and indexing documents and images*. Morgan Kaufmann, 1999.

[10] D. Bahle, "Efficient Phrase Querying," *School of Computer Science and Information Technology, Royal Melbourne Institute of Technology*, 2003.

[11] A. Fellinghaug, "Phrase searching in text indexes," no. June, p. 137, 2008.

[12] C. J. van Rijsbergen, "A theoretical basis for the use of co-occurrence data in information retrieval," *Journal of documentation*, vol. 33, no. 2, pp. 106–119, 1977.

[13] R. Nallapati and J. Allan, "Capturing term dependencies using a language model based on sentence trees," in *Proceedings of the eleventh international conference on Information and knowledge management*, 2002, pp. 383–390.

[14] E. M. Keen, "The use of term position devices in ranked output experiments," *Journal of Documentation*, vol. 47, no. 1, pp. 1–22, 1991.

[15] W. B. Croft, H. R. Turtle, and D. D. Lewis, "The use of phrases and structured queries in information retrieval," in *Proceedings of the 14th annual international ACM SIGIR conference on Research and development in information retrieval*, 1991, pp. 32–45.

[16] D. Metzler and W. B. Croft, "A Markov random field model for term dependencies," in *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, 2005, pp. 472–479.

[17] E. K. F. Dang, R. W. P. Luk, and J. Allan, "A context-dependent relevance model," *Journal of the Association for Information Science and Technology*, 2015.

[18] F. Song and W. B. Croft, "A general language model for information retrieval," in *Proceedings of the eighth international conference on Information and knowledge management*, 1999, pp. 316–321.

[19] J. Gao, J.-Y. Nie, G. Wu, and G. Cao, "Dependence language model for information retrieval," in *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, 2004, pp. 170–177.

[20] B. He, J. X. Huang, and X. Zhou, "Modeling term proximity for probabilistic information retrieval models," *Information Sciences*, vol. 181, no. 14, pp. 3017–3031, 2011.

[21] Y. Rasolofo and J. Savoy, *Term proximity scoring for keyword-based retrieval systems*. Springer, 2003.

[22] C. Eickhoff, A. P. de Vries, and T. Hofmann, "Modelling Term Dependence with Copulas," in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2015, pp. 783–786.

[23] S. Büttcher, C. L. A. Clarke, and B. Lushman, "Term proximity scoring for ad-hoc retrieval on very large text collections," in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, 2006, pp. 621–622.

[24] T. Tao and C. Zhai, "An exploration of proximity measures in information retrieval," in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, 2007, pp. 295–302.

[25] J. Zhao and Y. Yun, "A proximity language model for information retrieval," in *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, 2009, pp. 291–298.

[26] J. Zhao, J. X. Huang, and B. He, "CRTER: using cross terms to enhance probabilistic information retrieval," in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, 2011, pp. 155–164.




[ DOI: 10.29252/jsdp.15.4.71 ]

[27] J. Zhao, J. X. Huang, and Z. Ye, "Modeling term associations for probabilistic information retrieval," *ACM Transactions on Information Systems (TOIS)*, vol. 32, no. 2, p. 7, 2014.

[28] J. Miao, J. X. Huang, and Z. Ye, "Proximity-based rocchio's model for pseudo relevance," in *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, 2012, pp. 535–544.

[29] C. L. A. Clarke, G. V. Cormack, and E. A. Tudhope, "Relevance ranking for one to three term queries," *Information Processing & Management*, vol. 36, no. 2, pp. 291–311, 2000.

[30] J. Klekota, F. P. Roth, and S. L. Schreiber, "Query Chem: a Google-powered web search combining text and chemical structures," *Bioinformatics*, vol. 22, no. 13, pp. 1670–1673, 2006.

[31] K. Sadakane and H. Imai, "Text Retrieval by using k-word Proximity Search," in *Database Applications in Non-Traditional Environments, 1999.(DANTE'99) Proceedings. 1999 International Symposium on*, 1999, pp. 183–188.

[32] X. Lu, A. Moffat, and J. S. Culpepper, "On the cost of extracting proximity features for term-dependency models," in *CIKM 2015*, 2015, pp. 293–302.

[33] M. Blum, R. W. Floyd, V. Pratt, R. L. Rivest, and R. E. Tarjan, "Time bounds for selection," *Journal of computer and system sciences*, vol. 7, no. 4, pp. 448–461, 1973.

[34] R. Courant, *Differential and integral calculus*, vol. 2. John Wiley & Sons, 2011.

[35] S. E. Robertson and S. Walker, "Some for Simple Effective Approximations to the 2 – Poisson Model Probabilistic Weighted Retrieval," *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 232–241, 1994.

[36] H. Zaragoza, N. Craswell, M. J. Taylor, S. Saria, and S. E. Robertson, "Microsoft Cambridge at TREC 13: Web and Hard Tracks.," in *TREC*, 2004, vol. 4, p. 1.

[37] R. Duda O., P. Hart E., and D. Stork G., *Pattern Classification*. 2000.

[38] S. Robertson and H. Zaragoza, *The probabilistic relevance framework: BM25 and beyond*. Now Publishers Inc, 2009.

[39] J. Zhao and J. X. Huang, "An enhanced context-sensitive proximity model for probabilistic information retrieval," in *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, 2014, pp. 1131–1134.

**جواد پاکسیما** کارشناسی و کارشناسی ارشد خود را در رشتهٔ مهندسی کامپیوتر-گرایش نرم‌افزار به‌ترتیب در سال ۱۳۷۴ و ۱۳۷۶ در دانشگاه صنعتی شریف به پایان رسانید و دکترای تخصصی خود را در رشته کامپیوتر-نرم‌افزار در سال ۹۶ دریافت کرد. وی در حال حاضر در دانشکده مهندسی کامپیوتر دانشگاه پیام‌نور به عنوان هیئت علمی مشغول به فعالیت و هم‌چنین مدیر عامل شرکت بشری‌پرداز است. موضوعات مورد علاقهٔ ایشان شبکه‌های کامپیوتر و بازیابی اطلاعات است.

نشانی رایانهٔ ایشان عبارت است از:

**paksima@pnu.ac.ir**



[ DOI: 10.29252/jsdp.15.4.71 ]