



# مدل جدیدی برای جستجوی عبارت بر اساس کمینه جابه‌جایی وزن‌دار

جواد پاکسیما

دانشکده 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، فاصله بین واژگان پرس‌وجو در سند است. در این مقاله تعریف جدیدی از فاصله بر اساس کمینه جابه‌جایی وزن‌دار<sup>۱</sup> واژگان سند به‌منظور تطبیق بر پرس‌وجو ارائه می‌شود. همچنین با توجه به این‌که بیش‌تر الگوریتم‌های رتبه‌بندی از فرکانس رخداد یک واژه در سند<sup>۲</sup> 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<sup>۳</sup> و معکوس فرکانس سند<sup>۴</sup> با توجه به مفهوم جدید فاصله تعریف‌شده و الگوریتم‌هایی برای محاسبه آن‌ها ارائه شده است. همچنین نتایج الگوریتم پیشنهادی با چند الگوریتم مقایسه شده است که افزایش خوبی را در میانگین دقت 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.

<sup>1</sup> MWRM

<sup>2</sup> Term Frequency

<sup>3</sup> Phrase Frequency

<sup>4</sup> Inverted Document Frequency

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

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.

واژه در هر پرس و جو [5] و مجموعه کل لغات خیلی زیاد است. این در حالی است که در بیش تر الگوریتم های بازیابی اطلاعات به طور معمول برای تعداد اسناد کم و تعداد واژگان پرس و جوی زیاد کارایی خوبی دارند [1].

یک عبارت<sup>۱</sup>، فهرستی از واژگان است که دارای ترتیب مشخصی هستند. به عنوان مثال «موتور جستجو» یک عبارت شامل دو واژه است که واژه نخست آن «موتور» و واژه دوم آن «جستجو» است. طول بیش تر عبارات دو واژه است و کمتر از یک درصد عبارات طولی، بیش تر از شش واژه دارند [6].

در یک کار تحلیلی بر روی یک و نیم میلیون پرس و جو در موتور جستجوی Excite مشخص شد که ۸/۴ درصد از پرس و جوها شامل علامت نقل قول (' یا ") هستند [7]. پژوهشی دیگر نشان می دهد که مقصود کاربران در چهل درصد از پرس و جوهایی که دو واژه یا بیشتر هستند و علامت نقل قول هم ندارند، عبارت بوده است نه واژگان مجزا [8].

روش اصلی برای جستجوی یک عبارت استفاده از نمایه معکوس<sup>۲</sup> است [9]. هر ردیف در نمایه معکوس شامل یک واژه است که فهرستی از سه تایی های مرتب شامل شماره سند، تعداد رخداد واژه در سند و موقعیت هایی که در آن واژه مورد نظر در سند ظاهر شده و به صورت زیر است:

$\langle d, tf_{d,t}, [p_1, p_2, \dots] \rangle$

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 می دهد. فهرست زیر یک نمونه از این سه تایی ها است:

$\langle 101, 4, [5, 11, 25, 77] \rangle$

با استفاده از فهرست نمایه معکوس، زمان پردازش برای شمارش تعداد عبارت به تعداد واژگان پرس و جو و تعداد تکرار واژگان عبارت در سند وابسته است. برای ارزیابی یک عبارت باید ابتدا موقعیت های واژگان پرس و جو استخراج و سپس با ترکیب آن ها شمارش انجام شود.

## ۱- مقدمه

پیدا کردن صفحات دارای کیفیت بالا در وب یکی از مهم ترین وظایف موتورهای جستجو است. مفهوم میزان ارتباط اسناد پیدا شده با پرس و جو وابسته به نظر کاربر است و این موضوع باعث افزایش پیچیدگی الگوریتم های رتبه بندی می شود. نکته دیگر این است که اغلب کاربران ده تا بیست نتیجه نخست را بررسی می کنند [1] در حالی که برای یک پرس و جو ممکن است، میلیون ها صفحه مرتبط وجود داشته باشد. بنابراین موتورهای جستجو باید برای یافتن مرتبط ترین صفحات، الگوریتم مناسب با کارایی بالا ارائه کنند.

بخش رتبه بندی یکی از مهم ترین قسمت های موتورهای جستجو است. رتبه بندی فرآیندی است که در آن کیفیت صفحه توسط موتور جستجو تخمین زده می شود. در حال حاضر دو روش عمده برای رتبه بندی صفحات وب وجود دارد. در روش نخست رتبه بندی بر اساس محتوای اسناد انجام می شود (رتبه بندی سنتی). مدل هایی مانند مدل بولی، مدل احتمالی و مدل فضای برداری جهت رتبه بندی اسناد مبتنی بر محتوا ارائه شده اند [2]. در روش دوم بر اساس گراف و اتصالات وب و میزان اهمیت صفحات رتبه بندی صورت می گیرد.

در رتبه بندی سنتی، موتور جستجو سعی می کند با پیدا کردن میزان ارتباط سند و پرس و جو اسناد را رتبه بندی کند. در این الگوریتم ها برای هر پرس و جو، اسناد با محتوای شبیه تر به واژگان موجود در پرس و جو امتیاز بالاتری را دریافت می کنند. به عنوان مثال الگوریتم های TF-IDF [3]، BM25 [4] دو نمونه از رایج ترین الگوریتم های این نوع رتبه بندی هستند.

وب شامل تعداد زیادی اسناد غیر ساخت یافته است که به هم متصل هستند و یک گراف خیلی بزرگ را ایجاد می کنند. به طور معمول تعداد واژگان پرس و جوها کم بوده (۲/۴)

<sup>1</sup> Phrase

<sup>2</sup> Inverted Index

کرد که بر اساس موقعیت واژگان پرس و جو در سند با غربال اسناد غیر مرتبط می‌تواند نتایج را دقیق‌تر کند. کین برای نخستین بار مفهوم فاصله بین دو واژه را تعریف کرد. طبق تعریف او فاصله عبارت است از تعداد واژگان تطبیق داده نشده بین نخستین و آخرین تطبیق در یک جمله. او هفت دیدگاه مختلف برای استفاده از مجاورت واژگان پیشنهاد داد. در مطالعه وی بهترین نتایج، زمانی به دست می‌آمد که الگوریتم به کمبودن فاصله بین واژگان پرس و جو پاداش می‌داد [14].

هم‌چنین در سال ۱۹۹۱ کرافت<sup>۹</sup> و همکارانش [15] برای نخستین بار روشی برای استفاده از شبکه استنتاج برای واژگان مجاور با عنوان InQuery ارائه دادند. آن‌ها هم‌چنین واژگان با DF بالا را از پرس و جو حذف کردند؛ برای مثال واژه «شهر» را از پرس و جو «اماکن تفریحی شهر شیراز» حذف کردند. آن‌ها دقت خوبی را در صفحات ابتدایی جستجو به دست آوردند؛ ولی در کل، دقت کاهش یافت. این موضوع بعدها در سال ۲۰۰۵ توسط متلزر<sup>۱۰</sup> و کرافت تأیید شد [16]. دنگ<sup>۱۱</sup> و همکارانش کار کرافت را دنبال کردند. آن‌ها بر اساس واژگان ظاهرشونده در پنجره‌ای که محل تمرکز واژگان پرس و جو است و هم‌چنین بازخورد کاربر، امتیاز اسناد را محاسبه می‌کردند [17].

مدل‌های زبانی متنوعی برای وابستگی واژگان پرس و جو ارائه شده است. نخستین بار در سال ۱۹۹۹ سانگ<sup>۱۲</sup> و کرافت [18] مدل تک‌واژه‌ای<sup>۱۳</sup> استاندارد را با درون‌یابی مدل زوج‌واژه‌ای<sup>۱۴</sup> توسعه دادند. در مقیاس آزمایشی کوچک، استفاده از زوج‌واژگان، نتایج را بهبود بخشید. گائو<sup>۱۵</sup> و همکارانش در سال ۲۰۰۴ مدل زبانی وابسته (DLM<sup>۱۶</sup>) 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 پرس و جوها تخمین زدند. در مدل، فقط زوج‌واژگان مورد توجه قرار گرفتند. این الگوریتم در عمل قابل استفاده نبود؛ زیرا استخراج تمام ساختارهای پیوند برای یک پرس و جو در زمان جستجو بسیار زمان‌بر بود [20].

برای رتبه‌بندی اسناد، زمانی که پرس و جو فقط یک واژه باشد، پارامتر TF<sup>۱</sup> و IDF<sup>۲</sup> 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.

به‌طور معمول در موتورهای جستجو برای جستجوی عبارتی که در آن از علامت نقل قول استفاده شده است مشابه جستجوی تک‌واژه عمل می‌شود و الگوریتم‌های متعددی برای آن ارائه شده است [10] و [11]. اما اگر علامت نقل قول 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<sup>۳</sup> تعریف شده است که فرکانس یک عبارت را نشان می‌دهد و الگوریتمی برای محاسبه IDF بر اساس PF ارائه می‌شود.

در ادامه 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 ریچس برگن<sup>۴</sup> (۱۹۷۷) است که به صورت نظری رتبه‌بندی اسناد را بر اساس مجاورت واژگان پرس و جو بررسی کرده است [12]. در روش پیشنهادی وی، میزان ارتباط واژگان پرس و جو بر اساس اطلاعات زوج‌واژه مشخص می‌شد و برای بازیابی از درخت پوشای بیشینه (MST<sup>۵</sup>) استفاده شده است. سال‌ها بعد در سال ۲۰۰۲ نالپاتی<sup>۶</sup> و آلان<sup>۷</sup> ایده خود را در مورد نحوه مشخص کردن میزان ارتباط واژگان پرس و جو ارائه دادند [13]. برای کاهش زمان، آن‌ها پیشنهاد کردند که به جای استفاده از آمار اسناد، از آمار جملات برای ساخت درخت پوشای بیشینه استفاده شود.

در سال ۱۹۹۱ کین<sup>۸</sup> نتایج تجربی‌اش را منتشر و فرض

<sup>9</sup> Croft

<sup>10</sup> Metzler

<sup>11</sup> Dang

<sup>12</sup> Song

<sup>13</sup> Bigram

<sup>14</sup> Unigram

<sup>15</sup> Gao

<sup>16</sup> Dependence Language Model

<sup>1</sup> Term Frequency

<sup>2</sup> Inverse Document Frequency

<sup>3</sup> Phrase Frequency

<sup>4</sup> Rijsbergen

<sup>5</sup> Maximum Spanning Tree

<sup>6</sup> Nallapati

<sup>7</sup> Allan

<sup>8</sup> Keen

پرس‌وجو در متن سند به‌دست آورده شده است و مجموع آن‌ها برای محاسبه امتیاز به کار برده می‌شود.

مدل<sup>11</sup> CRTER 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 محاسبه 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 توسعه پرس‌وجو<sup>12</sup> است. به‌عنوان مثال میائو و همکارانش در سال ۲۰۱۲ از مفهوم واژگان مجاور با استفاده از بازخورد کاربر، توسعه 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' عبارت 'c 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 محاسبه فاصله مؤثر است. شکل (۱) نمایی از این سه بخش و فاصله فرضی بین آن‌ها را نشان می‌دهد.

گروه دیگری از پژوهش‌گران روش‌هایی برای افزودن وابستگی واژگان در مدل‌های احتمالی پیشنهاد دادند. برای مثال رسولف<sup>1</sup> 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 فاصله زوج‌واژگان در یک پنجره پنج‌واژه‌ای محاسبه شده است. هی<sup>۲</sup> و همکارانش در سال ۲۰۱۱ از یک پنجره برای شمارش فرکانس n واژه‌ای<sup>۳</sup> 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.

ایخف و همکارانش با استفاده از ابزاری آماری به نام گپولاس<sup>۴</sup> مدل آماری خود را توسعه دادند [22]. آن‌ها با استفاده گپولاس احتمال هم‌رخدادی واژگان پرس‌وجو را به دست آوردند. مزیت اصلی روش آن‌ها سرعت بالای آن بود. باتچر<sup>۵</sup> 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 نمایه معکوس پیاده‌سازی شده است. در موقع پردازش فهرست موقعیت‌های<sup>۶</sup> واژگان پرس‌وجو، اگر واژه‌ای تغییر کرد، فاصله به‌صورت تجمعی افزوده شود. کار مشابهی توسط تائو<sup>۷</sup> و همکارش در سال ۲۰۰۷ انجام شد [24]. آن‌ها پنج ویژگی مربوط به فاصله 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<sup>۸</sup> و یان<sup>۹</sup> مدل زبانی واژگان مجاور ('PLM') را در سال ۲۰۰۹ پیشنهاد دادند [25]. در روش آن‌ها کمترین فاصله‌ها بین تمام واژگان

<sup>1</sup> Rasolof

<sup>2</sup> He

<sup>3</sup> n-grams

<sup>4</sup> Copulas

<sup>5</sup> Büttcher

<sup>6</sup> Posting List

<sup>7</sup> Tao

<sup>8</sup> Zhao

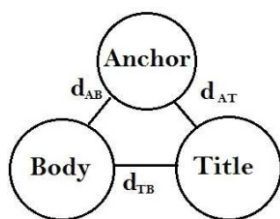
<sup>9</sup> Yun

<sup>10</sup> Proximity Language Model

<sup>11</sup> CRoss TErm Retrieval

<sup>12</sup> Query Expansion

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 محاسبه 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 در نتیجه 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 به صورت '...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. هزینه جابه‌جایی در این حالت برابر با ۳۹۸ است؛ زیرا 'b' به میزان ۱۰۱ جابه‌جایی نیاز دارد تا قبل از 'c' قرار گیرد و '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.

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:

$$s(0) = 1$$

$$\text{If } d_1 > d_2 \text{ then } s(d_1) < s(d_2)$$

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 رابطه لگاریتمی استفاده شده است.

PF: برای مطابقت دادن 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:

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

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.

<sup>۱</sup>QTM: 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 محاسبه فاصله و دیگری محاسبه PF است. همچنین مقدار IDF هم بر اساس تعریف PF ارائه می‌شود. در موقع محاسبه 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.

<sup>۱</sup> Query Maximum Terms

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

$$d_x = w_1(x - p_1) + w_2(x - p_2) + \dots + w_{k+1}(p_{k+1} - x) + \dots + w_n(p_n - x) \quad (2)$$

و  $d_{x+1}$  هزینه جابه‌جایی واژگان برای تمرکز در موقعیت  $x+1$  سند به صورت زیر خواهد بود:

$$d_{x+1} = w_1(x + 1 - p_1) + \dots + w_{k+1}(p_{k+1} - x - 1) + \dots + w_n(p_n - x - 1) \quad (3)$$

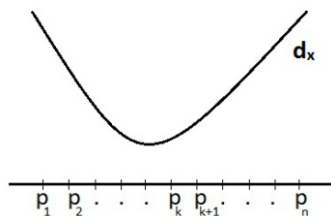
تفاضل  $d_x$  و  $d_{x+1}$  را با  $\Delta_x$  نشان می‌دهیم و به صورت زیر محاسبه خواهد شد:

$$\Delta_x = d_{x+1} - d_x = w_1 + \dots + w_k - w_{k+1} - \dots - w_n \quad (4)$$

رابطه (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 به صورت 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  $d_x$  function

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

$$\varphi_k = d_{p_{k+1}} - d_{p_k} \quad (5)$$

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

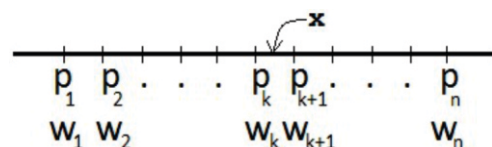
$$\varphi_k = (w_1 + \dots + w_k - w_{k+1} - \dots - w_n) \times (p_{k+1} - p_k) \quad (6)$$

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

هم‌چنین اگر واژگان پرس‌وجو در بیش از یک بخش ظاهر شده باشند، دو موضوع اضافه می‌شود. موضوع نخست، یک فاصله 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 محاسبه فاصله ارائه می‌شود؛ اما برای مشخص کردن فاصله فرضی بین بخش‌ها و وزن واژگان در بخش‌های مختلف باید از یکی از شیوه‌های آموزش استفاده شود. در اینجا از الگوریتم ژنتیک برای محاسبه 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 به صورت تصادفی مقداردهی می‌شوند و سپس با بهینه‌کردن مقدار تابع برازش وزن‌های بهینه به دست می‌آیند. تابع برازشی برابر مقدار معیار دقت  $p@n$  به‌ازای خروجی مدل پیشنهادی است و هدف الگوریتم ژنتیک تنظیم وزن‌های واژگان به‌صورتی است که بتوان مقدار بیشینه  $p@n$  را به دست آورد. مقدار بیشینه 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 پیدا شده در سند نمی‌تواند باشد (موقعیت مکانی محل تجمع واژگان، موقعیت یکی از واژگان پرس‌وجو در سند است).

اثبات: هدف از این لم محدود کردن 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 کلمه پیدا شده 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 ( $w$ )

در الگوریتم (۱) فرض شده که پرس وجو دارای  $m$  واژه است و واژگان عبارت به صورت  $t_1, t_2, \dots, t_m$  هستند. همچنین موقعیت واژه  $t_i$  در سند را  $p_i$  در نظر می گیریم ( $p_i$ ها لزوماً مرتب شده نیستند).

## ۲-۴- الگوریتم محاسبه PF

یکی از الگوریتم هایی که می تواند PF را محاسبه کند الگوریتم جستجوی کامل است. در این روش تمام ترکیب هایی که می توان با استفاده از آن عبارت را ایجاد کرد استخراج می شود و شکلی که مجموع امتیاز آن بالاتر است انتخاب می شود. در [31] و [32] الگوریتم هایی برای استخراج بهترین انطباق های یک عبارت در یک سند ارائه شده که دارای اشکالاتی است. برای مثال در تمام الگوریتم های ارائه شده، در جستجوی 'a b c' دو سند 'b b a c' و 'b a c' یک امتیاز دریافت می کنند؛ در حالی که سند نخست مرتبط تر است. بنابراین در اینجا از جستجوی کامل برای استخراج بهترین تطبیق ها استفاده شده است.

در روش پیشنهادی ابتدا با استفاده از «and query» اسنادی که تمام واژگان عبارت را دارا هستند، استخراج می شوند؛ سپس با استفاده از نمایه معکوس به ازای تمام واژگان عبارت، تمام جایگشت های ممکن از انطباق های عبارت بررسی و سپس برای هر جایگشت جمع فاصله ها محاسبه می شود. پیدا کردن جایگشتی که جمع بیشینه فاصله را تولید کند هدف اصلی است.

الگوریتم (۲) روش محاسبه PF را نشان می دهد. در این الگوریتم متغیر Max برای نگهداری بیشترین مقدار مجموع فاصله مورد استفاده قرار گرفته است. کار اصلی در این الگوریتم 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

$$\begin{aligned} \text{sign}(\varphi_1) &= \text{sign}(w_1 - w_2 - w_3 - \dots - w_n) \\ \text{sign}(\varphi_2) &= \text{sign}(w_1 + w_2 - w_3 - \dots - w_n) \\ \text{sign}(\varphi_3) &= \text{sign}(w_1 + w_2 + w_3 - \dots - w_n) \end{aligned} \quad (7)$$

همان طور که رابطه (۷) نشان می دهد، علامت  $\varphi_k$  در ابتدا منفی است و هر بار مقداری افزایش می یابد و در نهایت با علامت مثبت پایان می پذیرد و بنابراین  $d_x$  در ابتدا نزولی است و سپس صعودی می شود و در نتیجه  $d_x$  سهموی شکل است.

**قضیه ۱:** در مدل MWRM کمترین مقدار  $d_x$  در مرکز ثقل واژگان پرس وجو در سند حاصل می شود.

**اثبات:** طبق لم ۱ نقطه بهینه برای  $d_x$  مکانی از سند است که در بردارنده یکی از واژگان پرس وجو می باشد و طبق لم ۲ مقدار  $d_x$  در ابتدا نزولی و سپس صعودی می شود و بنابراین در لحظه ای که روند نزولی پایان می پذیرد و روند صعودی آغاز می شود، کمترین مقدار  $d_x$  اتفاق می افتد؛ یعنی زمانی که  $\varphi_k$  نزدیک صفر است. اگر  $p_m$  نقطه ی مفروض باشد، داریم:

$$\begin{aligned} \varphi_k &= (w_1 + \dots + w_m - w_{m+1} - \dots - w_n) \cong 0 \\ \Rightarrow w_1 + \dots + w_m &\cong w_{m+1} + \dots + w_n \end{aligned} \quad (8)$$

در نتیجه  $p_m$  نقطه ای است که وزن واژگان دو طرف آن به طور تقریبی مساوی است. به عبارت دیگر بهترین نقطه برای تجمع واژگان سند مرکز ثقل واژگان است.

نتیجه ارزشمند قضیه ۱ کاهش قابل توجه پیچیدگی الگوریتم رتبه بندی بر اساس مدل MWRM است. برای پیاده سازی الگوریتم آن می توان در ابتدا  $\varphi_1$  را محاسبه و سپس با افزودن  $2w_2$  به آن  $\varphi_2$  را محاسبه کرد و همین طور ادامه داد و مقادیر بعدی را به دست آورد. به محض تغییر علامت  $\varphi_k$  از منفی به مثبت،  $p_k$  را به عنوان نقطه بهینه برگرداند.

بر اساس نتایج قضیه و لم های بالا الگوریتم (۱) برای محاسبه 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 محاسبه فاصله  
(Algorithm-1): Distance calculation algorithm

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. رابطه ۱۰ را برای این منظور به صورت زیر تعریف می‌کنیم و در رابطه ۱۱ مقدار  $df$  یک عبارت محاسبه می‌شود:

$$F(d, q) = \begin{cases} 1 & PF \geq 1 \\ PF & PF < 1 \end{cases} \quad (10)$$

$$df_q = \sum_d F(d, q) \quad (11)$$

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

#### Algorithm IDF

```

01: Input Query q
02: Output IDF
03: Assumption
04: q include <t1, t2, ..., tm>
05: N is total document in corpus
06: Begin Algorithm
07: DF ← 0
08: For every document di
09: PF ← calculate PF(q, di)
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 کلیه اسناد، مقدار IDF عبارت را نیز محاسبه کرد. البته مسئله کارایی این الگوریتم‌ها نسبت به روش‌های سنتی برای استخراج TF و IDF بسیار متفاوت خواهد بود.

در روش‌های سنتی، TF هر واژه برای هر سند به سادگی در دسترس است و همچنین IDF‌ها برای هر واژه در زمان نمایه می‌تواند محاسبه و ذخیره شود و در موقع جستجو بدون هیچ پردازشی در اختیار بخش رتبه‌بندی قرار

در الگوریتم (۲) از متغیر  $P_{ij}$  برای مشخص کردن  $i$ -امین موقعیت واژه  $t_i$  پرس‌وجو در سند استفاده شده است. خط ۹ الگوریتم، وظیفه تولید جایگشت‌های مختلف تطابق واژگان عبارت مورد جستجو را با واژگان معادل در سند بر عهده دارد. برای مثال نخستین جایگشت می‌تواند به صورت زیر انتخاب شود:

Phrase 1:  $\langle t_1, P_{11} \rangle, \langle t_2, P_{21} \rangle, \dots, \langle t_n, P_{n1} \rangle$

Phrase 2:  $\langle t_1, P_{12} \rangle, \langle t_2, P_{22} \rangle, \dots, \langle t_n, P_{n2} \rangle$

⋮

برای جایگشت بعدی می‌توان موقعیت  $t_1$  در Phrase 1

را تغییر داد و  $P_{12}$  را در نظر گرفت و به طور مشابه موقعیت  $t_1$  در Phrase 2 را  $P_{11}$  قرار داد و یک جایگشت جدید به دست آورد.

الگوریتم در خط ۱۰ برای تک‌تک عبارات استخراج شده 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 به دلیل محدودبودن QTM خطی است، ولی به خاطر اهمیت ضریب خطی در جستجوی انبوه باید بهینه شود. برای بهینه‌کردن این الگوریتم می‌توان از حرص‌کردن براساس فاصله واژگان استفاده کرد. یکی از روش‌ها برای حرص‌کردن استفاده از همین مفهوم فاصله است. مشخص است که از یک فاصله‌ای بیشتر ارتباط بین واژگان به طور تقریبی قطع می‌شود و با این فرض می‌توان فاصله واژگان را محدود و بخشی از درخت را حرص کرد. به عنوان مثال اگر دو واژه در فاصله 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 تصمیم‌گیری 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 به شکل مناسبی تعریف شود تا به درستی میزان اهمیت عبارت را مشخص کند. IDF پارامتری است که با لگاریتم وارون DF رابطه 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 به صورت رابطه زیر است:

$$idf_t = \log \frac{N}{1 + df_t} \quad (9)$$

**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 می افتد که تک تک واژگان سند در واژگان عبارت باشد پس  $n = \sum_{i=1}^m tf(t_i)$  می باشد. طبق لم ۳ برای این که حاصل ضرب  $tf$  ها بیشینه باشد  $tf(t_i)$  ها باید مساوی باشند. پس در بدترین حالت داریم:

$$tf(t_i) = \frac{n}{m} \quad (۱۳)$$

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

**قضیه ۳:** پیچیدگی زمانی PF در حالت متوسط در  $O(1)$  می باشد.

**اثبات:** برای حالت متوسط باید امید ریاضی  $tf$  ها را پیدا کرد. می توان 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 دو جمله ای است. این موضوع با استفاده از مجموعه داده های TREC2003 و TREC2004 [36] نیز قابل بررسی می باشد. بنابراین داریم  $E(tf) = \mu$  که  $\mu$  میانگین توزیع پواسون می باشد.

اگر فرض کنیم  $tf$  ها از هم مستقل هستند امید ریاضی حاصل ضرب را می توان به حاصل ضرب امید ریاضی ها تبدیل کرد [37]؛ یعنی حد بالایی تعداد جای گشت ها در حالت متوسط را می توان از رابطه (۱۴) به دست آورد:

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

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

## ۵- نتایج ارزیابی

به منظور ارزیابی الگوریتم پیشنهادی از داده آزمایش موتور جستجوی پارسی جو<sup>۱</sup> که بخشی از داده واقعی موتور یاد شده می باشد، استفاده شده است. این داده آزمایش مشتمل بر حدود چهارصد هزار سند صفحه وب و حدود پنجاه پرس و جوی فارسی ارزیابی شده که کمینه، متوسط و بیشینه تعداد واژگان پرس و جوها به ترتیب برابر ۲، ۱۴، ۱۶۹۰۳ و ۸ است. هم چنین از معیار دقت<sup>۲</sup> در مکان  $n$ -م  $(p@n)$  و میانگین متوسط دقت

<sup>۱</sup> <http://www.parsijoo.ir>

<sup>۲</sup> Precision

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

## ۴-۴- پیچیدگی زمانی الگوریتم

در صورتی که بخواهیم یک عبارت را تطبیق دهیم، طبق قضیه ۱ باید مرکز ثقل را بیابیم. اگر برای یافتن مرکز ثقل از مرتب سازی سریع استفاده شود، پیچیدگی زمانی  $O(n \log n)$  است که  $n$  نشان دهنده تعداد مقادیری است که می خواهیم مرکز ثقل آن ها را حساب کنیم. البته الگوریتم های سریع تری برای یافتن میانه نیز وجود دارد که پیچیدگی زمانی آن نسبت به  $n$  خطی است [33] ولی خواهیم دید که تأثیری روی پیچیدگی محاسبه PF ندارند.

همان طور که در بخش ۲ دیدیم، موتورهای جستجو بیشینه تعداد واژه را در پرس و جو مشخص می کنند که ما آن را با مقدار ثابت QTM نشان دادیم. پس پیچیدگی زمانی پیدا کردن فاصله به صورت  $O(QTM * \log(QTM))$  خواهد بود و با توجه به ثابت بودن QTM این پیچیدگی به صورت  $O(1)$  است.

پیچیدگی زمانی محاسبه PF طبق الگوریتم ارائه شده، رابطه مستقیمی با تعداد جایگشت های عبارت در متن دارد. تعداد جایگشت یک پرس و جو  $t_1 t_2 \dots t_m$  به صورت رابطه (۱۲) خواهد بود.

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

برای پیچیدگی زمانی محاسبه PF باید سه حالت بهترین، بدترین و حالت متوسط را به دست آوریم. واضح است که بهترین حالت، زمانی است که  $tf$  تمام واژگان عبارت یک باشد و در نتیجه پیچیدگی زمانی در بهترین حالت  $O(1)$  است. هم چنین بدترین حالت زمانی اتفاق می افتد که متن به طور مطلق شامل فقط واژگان عبارت باشد. می توان ثابت کرد که پیچیدگی زمانی محاسبه PF در حالت میانگین  $O(1)$  خواهد بود. در ادامه با استفاده از لم ۳ و قضایای ۲ و ۳ پیچیدگی زمانی محاسبه PF در بدترین حالت و حالت متوسط ارائه شده است.

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

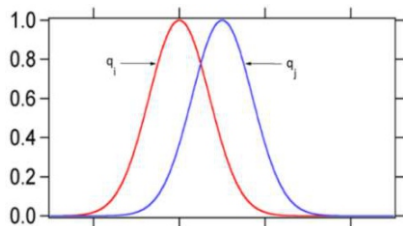
**اثبات:** این لم با استفاده از ضرایب لاگرانژ به سادگی امکان پذیر است [34].

تعاریف جدیدی به صورت زیر برای این دو پارامتر ارائه شده است:

$$tf(q_{ij}; D) = \sum_{m=1}^{tf_i} \sum_{n=1}^{tf_j} \text{Kernel}(\frac{1}{2} \text{dist}(\text{Pos}_m, \text{Pos}_n)) \quad (17)$$

$$df(q_{ij}) = \sum \frac{tf(q_{ij}; D)}{\text{occur}(q_{ij}; D)} \quad (18)$$

$$\text{occur}(q_{ij}; D) = \sum_{m=1}^{tf_i} \sum_{n=1}^{tf_j} 1_{\{\text{Kernel}(\frac{1}{2} \text{dist}(\text{Pos}_m, \text{Pos}_n)) \neq 0\}} \quad (19)$$



(شکل-۴): تقاطع دو واژه  $q_i$  و  $q_j$  بر اساس دو تابع هسته گوسی  
(Figure-4): The intersection of the two words  $q_i$  and  $q_j$  based on two Gaussian core functions

در این روابط تابع  $\text{dist}$  برای محاسبه 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 به عنوان تابع هسته استفاده شده است؛ سپس از توابع  $tf$  و  $df$  جدید در تابع  $\text{BM25}$  استفاده می شود.

جدول (۱) وزن بخش های مختلف سند و مقدار پارامترهایی را که بر اساس مجموعه داده مفروض برای  $\text{BM25F}$  محاسبه شده است نشان می دهد. هم چنین جدول (۲) وزن واژگان در بخش های مختلف سند و فاصله فرضی بین این بخش ها را برای مدل  $\text{MWRM}$  نشان می دهد. برای محاسبه وزن واژگان در بخش های مختلف و فاصله بین آنها از الگوریتم ژنتیک استفاده شده است. هدف تابع برازش الگوریتم ژنتیک، به دست آوردن بالاترین میزان معیار دقت  $p@n$  برای رتبه بندی اسناد با استفاده از مدل پیشنهادی است.

(جدول-۱): پارامترها و وزن های محاسبه شده برای مدل  $\text{BM25F}$   
(Table -1): Parameters and calculated weights for the  $\text{BM25F}$  model

پارامتر	مقدار
$k_1$	1.4
$b_{\text{title}}$	0.1
$b_{\text{body}}$	0.98
$b_{\text{anchor}}$	6
$v_{\text{title}}$	3.6
$v_{\text{body}}$	1
$v_{\text{anchor}}$	1.4

( $\text{MAP}^1$ ) برای مقایسه الگوریتم پیشنهادی با دو الگوریتم دیگر استفاده شده است.

معیار دقت در مکان  $n$ -ام یا  $P@n$  نشان دهنده نسبت تعداد اسناد مرتبط در  $n$  سند نخست رتبه بندی نهایی برای هر پرس و جو به  $n$  است. فرمول  $P@n$  در معادله زیر نشان داده شده است:

$$P@n = \frac{1}{n} \sum_{j=1}^n r_j \quad (15)$$

که  $r_j$  نشان دهنده مرتبط بودن سند  $j$ -ام در رتبه بندی نهایی است.

هم چنین میانگین متوسط دقت یا  $\text{MAP}$  برای هر پرس و جو به عنوان میانگین مقادیر  $P@n$  ها برای همه اسناد مرتبط تعریف می شود:

$$AP = \frac{1}{|D_+|} \sum_{j=1}^N r_j \times P@j \quad (16)$$

$N$  نشان دهنده تعداد کل اسناد؛  $D_+$  نشان دهنده تعداد اسناد مرتبط و  $r_j$  نشان دهنده مرتبط بودن سند  $j$ -ام است؛ متوسط میانگین دقت ( $\text{MAP}$ ) به عنوان میانگین مقادیر  $AP$  همه پرس و جوهای ارائه شده است.

مدل  $\text{MWRM}$  با دو مدل دیگر مقایسه شده است: مدل  $\text{BM25F}$  و مدل  $\text{CRTER}$ . مدل  $\text{BM25F}$  مشابه  $\text{BM25}$  است با این تفاوت که در  $\text{BM25F}$  برای واژگان پیداشده در بخش های مختلف وزن های مختلفی در نظر گرفته می شود؛ در حالی که مقدار پارامترهای مدل برای هر بخش نیز قابل تنظیم است [38]. در مدل  $\text{CRTER}$  تأثیر واژگان بر واژگان مجاور توسط توابعی مثل گوسی، مثلثی، دایره ای و کسینوسی تخمین زده می شود [39]. به این توابع تابع هسته<sup>۲</sup> گفته می شود. هر تابع هسته باید خصوصیات زیر را داشته باشد:

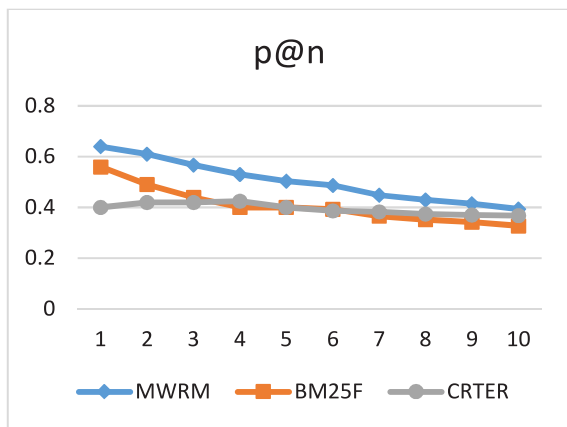
- منفی نباشد؛
- پیوسته باشد؛
- متقارن باشد؛
- یکنواخت باشد؛
- مقدار آن برای فاصله صفر یک باشد.

تقاطع دو واژه  $q_i$  و  $q_j$  زمانی رخ می دهد که این دو واژه در یک سند نزدیک هم ظاهر شوند و در نتیجه تابع هسته آنها نقطه اشتراک دارد. نقطه اشتراک، عددی است بین صفر و یک که برای فاصله های کمتر، بزرگ تر است. شکل (۴) نمونه ای از این تقاطع را نشان می دهد.

پارامترهای  $tf$  و  $df$  در بیش تر مدل ها به طور مستقیم یا غیرمستقیم مورد استفاده قرار می گیرند. در مدل  $\text{CRTER}$

<sup>1</sup> Mean Average Precision

<sup>2</sup> Kernel Function



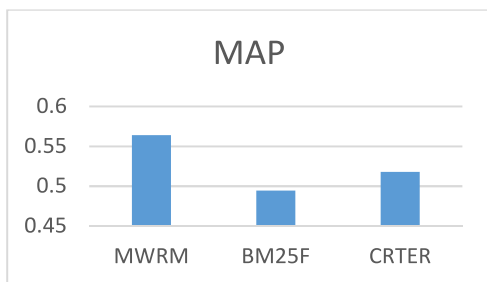
(شکل-۷): مقایسه بین الگوریتم پیشنهادی با الگوریتم BM25F

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 دقت الگوریتم پیشنهادی بیشتر از دو الگوریتم دیگر است.



(شکل-۸): مقایسه 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 در نتیجه مفهوم فاصله تعریف شد که نشان‌دهنده کمیته 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 موزایی‌سازی نیز می‌تواند روشی برای افزایش سرعت محاسبه PF و IDF قابل بررسی باشد.

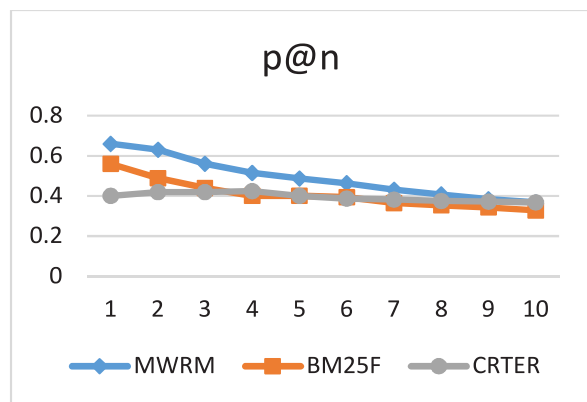
(جدول-۲): فاصله فرضی بین بخش‌های مختلف سند و وزن هر

بخش محاسبه شده 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.

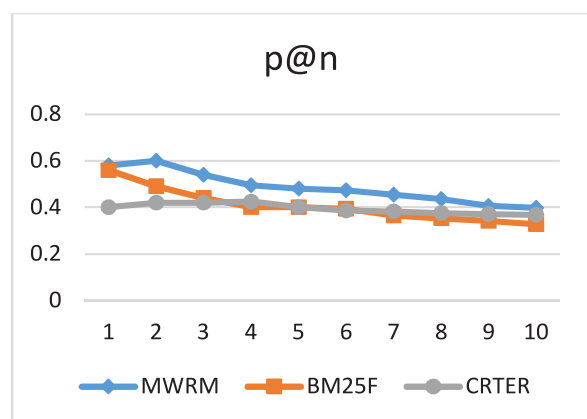


(شکل-۵): مقایسه بین الگوریتم پیشنهادی با الگوریتم BM25F

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

$$s(d) = \frac{1}{d^2}$$

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.

## 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*



**جواد پاکسیما** کارشناسی و کارشناسی ارشد خود را در رشته مهندسی کامپیوتر-گرایش نرم افزار به ترتیب در سال ۱۳۷۴ و ۱۳۷۶ در دانشگاه صنعتی شریف به پایان رسانید

و دکترای تخصصی خود را در رشته کامپیوتر-نرم افزار در سال ۹۶ دریافت کرد. وی در حال حاضر در دانشکده مهندسی کامپیوتر دانشگاه پیام نور به عنوان هیئت علمی مشغول به فعالیت و همچنین مدیر عامل شرکت بشری پرداز است. موضوعات مورد علاقه ایشان شبکه های کامپیوتر و بازیابی اطلاعات است.

نشانی رایانه ایشان عبارت است از:

[paksima@pnu.ac.ir](mailto:paksima@pnu.ac.ir)

- [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.

