بازیابی اطلاعات | Information Retrieval
امیرحسین آقاجری | 40031058

ایجاد شاخص معکوس مکانی (Positional Inverted Index):

شاخص معکوس یک ساختار داده‌ای است که برای بهینه‌سازی جستجو در متون استفاده می‌شود. این شاخص به ما امکان می‌دهد تا به سرعت داکیومنت‌هایی را که شامل یک کلمه خاص هستند، پیدا کنیم. در این پروژه، از شاخص معکوس مکانی (Positional Inverted Index) استفاده شده است که علاوه بر ذخیره‌سازی اطلاعات مربوط به کلمات موجود در داکیومنت‌ها، مکان وقوع هر کلمه در متن را نیز نگهداری می‌کند.

مراحل ساخت شاخص معکوس

1. **بارگذاری داکیومنت‌ها**
ابتدا داکیومنت‌ها از یک فایل JSON بارگذاری می‌شوند. این فایل شامل اطلاعات متادیتای هر داکیومنت مانند شناسه، عنوان، محتوا، تاریخ و برچسب‌ها است. تابع `load_documents` مسئولیت این کار را بر عهده دارد.

2. **پیش پردازش اسناد**
پس از بارگذاری داکیومنت‌ها، مرحله پیش پردازش کردن کلمات انجام می‌شود. در این مرحله، متن هر داکیومنت به توکن‌های مجزا (کلمات) تبدیل می‌شود. این فرآیند با استفاده از تابع `tokenizer` انجام می‌شود که متن را به کلمات جداگانه تجزیه می‌کند. این توکن‌ها شامل کلمات موجود در عنوان، برچسب‌ها و محتوای داکیومنت می‌شوند.

3. **ایجاد شاخص معکوس مکانی**
پس از بارگذاری داکیومنت‌ها، شاخص معکوس مکانی ایجاد می‌شود. این شاخص شامل اطلاعاتی درباره کلمات موجود در هر داکیومنت و مکان وقوع آن‌ها است. برای هر داکیومنت، ابتدا توکن‌های موجود در عنوان و برچسب‌ها با وزن‌های مخصوص به خود استخراج می‌شوند و سپس توکن‌های موجود در محتوای داکیومنت به شاخص اضافه می‌شوند. مکان وقوع هر توکن نیز در شاخص ذخیره می‌شود.

4. **نهایی‌سازی شاخص معکوس**
پس از اضافه کردن تمام توکن‌ها به شاخص، مرحله نهایی‌سازی انجام می‌شود. در این مرحله، وزن‌های مربوط به فراوانی ترم (TF) و معکوس فراوانی داکیومنت (IDF) برای هر توکن محاسبه می‌شوند. این محاسبات باعث می‌شوند تا شاخص نهایی بتواند به صورت بهینه‌تر برای جستجو مورد استفاده قرار گیرد.

5. **ایجاد شاخص معکوس برای کوئری‌ها**
برای هر کوئری ورودی، یک شاخص معکوس مشابه با داکیومنت‌ها ایجاد می‌شود. این شاخص به جستجوی بهتر و دقیق‌تر کمک می‌کند، چرا که امکان مقایسه مستقیم کوئری با داکیومنت‌های موجود را فراهم می‌کند.

6. **استفاده از شاخص معکوس در سیستم جستجو**
شاخص معکوس نهایی شده به همراه داکیومنت‌های بارگذاری شده به عنوان ورودی به سیستم جستجو داده

می‌شود. این سیستم با استفاده از شاخص معکوس می‌تواند به سرعت داکیومنت‌هایی را که شامل کلمات کوئری هستند، پیدا کرده و رتبه‌بندی کند.

ذخیره سازی موقت شاخص معکوس مکانی:

پس از ایجاد شاخص معکوس مکانی، این اطلاعات را به صورت موقت ذخیره می‌کنیم تا در مراحل بعدی اجرای برنامه، نیازی به ساخت دوباره‌ی شاخص معکوس نباشد. این کار باعث می‌شود که برنامه با خواندن اطلاعات از قبل پردازش شده از فایل کش، به طور قابل توجهی سریع‌تر اجرا شود، به‌طوری‌که به تقریباً 89.5 درصد زمان اجرا کاهش یابد (بسته به سخت‌افزار مورد استفاده). همچنین، فضای ذخیره‌سازی موقت تقریباً 76.3 درصد از حجم داده‌های اولیه را شامل می‌شود.

ساختار ذخیره سازی:

```
[
 {
   "term": "AFC",
   "idf": 5.8469095607744315,
   "df": 212,
   "list": [
     {
       "doc_id": "1308",
       "tf": 3.0,
       "lf": 4,
       "list": [70, 86, 122, 163]
     },
     ...
   ]
 },
 ...
]
```

پیش پردازش اسناد:

پیش پردازش محتوای داکیومنت‌ها شامل چند مرحله مهم است:

1. **نرمال‌سازی (Normalization):** در این مرحله، متن ورودی به کمک نرمال‌کنندهٔ Hazm که یکی از ابزارهای پردازش متن فارسی است، نرمال‌سازی می‌شود. این فرایند شامل تبدیل کاراکترهای یونیکد به شکل استاندارد، حذف حروف تکراری و اصلاح کلمات است.
2. **توکن‌سازی (Tokenization):** متن نرمال‌شده به کمک توکنایزر Hazm به تکه‌های جداگانه (توکن‌ها) تقسیم می‌شود که هر توکن یک واحد معنایی (مانند کلمه یا نشانگر) را نمایان می‌کند.
3. **حذف علائم نگارشی (Strip Punctuations):** در این مرحله، علائم نگارشی از هر توکن حذف می‌شود. این علائم ممکن است شامل نقطه، ویرگول، پرانتز ها و علائم دیگر باشند.
4. **حذف اعداد (Strip Numbers):** اگر این گزینه فعال باشد، تمام ارقام و اعداد درون متن حذف می‌شوند تا فقط کلمات معنایی باقی بمانند.

5. **حذف شکلک‌ها (Strip Emoji):** در صورت فعال بودن، تمام شکلک‌ها و نمادهای اموجی از متن حذف می‌شوند.
6. **حذف کلمات پرتکرار (Filter Stopwords):** کلمات پرتکراری که در زبان فارسی اغلب بدون تاثیر معنایی هستند (مانند "و"، "یا" و غیره) حذف می‌شوند.
7. **لماتیزاسیون (Lemmatization):** در نهایت، تمام توکن‌ها به شکل ریشه‌یابی شدهٔ آنها با استفاده از لماتیزر Hazm تبدیل می‌شوند. این فرایند به توکن‌ها کمک می‌کند تا به شکل استاندارد و با یک شکل معنایی مناسب تبدیل شوند.

این مراحل پیش پردازش به ترتیبی که بالا بیان شده، به منظور بهینه‌سازی محتوای متنی برای مراحل بعدی پردازش اطلاعات و استفاده از آنها در سیستم‌های جستجو یا تحلیل متن انجام می‌شود.

لیست ۵۰ ترم پرتکرار قبل از نرمال‌سازی و ریشه یابی:

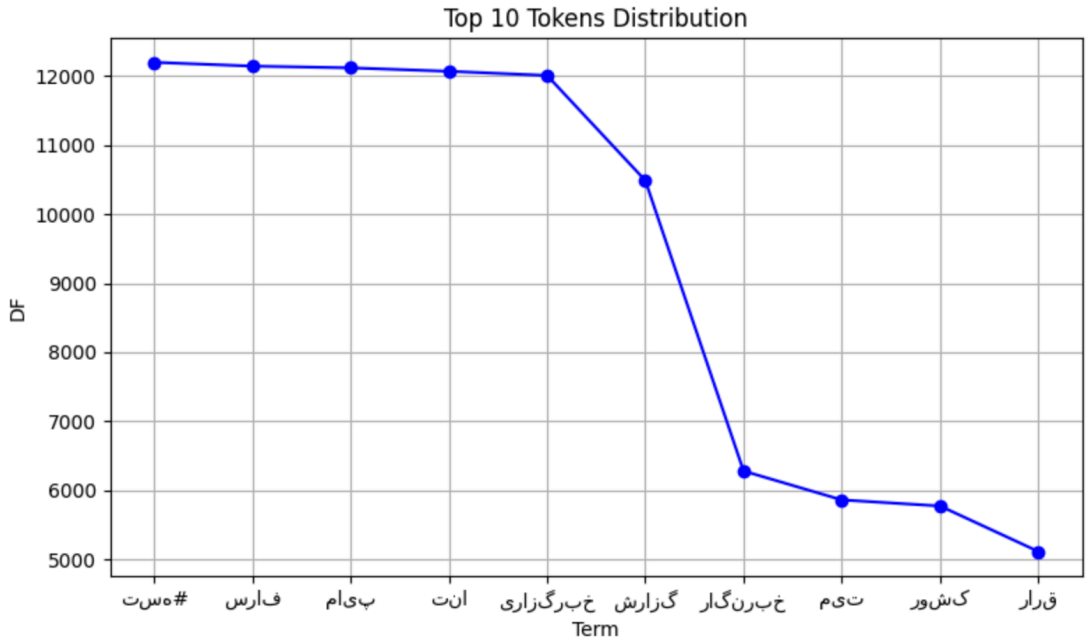
این جدول لیستی از **پرتکرارترین Term های** موجود در اسناد را **قبل از نرمال‌سازی و ریشه یابی** نشان می‌دهد.

دسته اول		دسته دوم		دسته سوم		دسته چهارم		دسته پنجم	
Term	DF	Term	DF	Term	DF	Term	DF	Term	DF
به	12181	گزارش	10433	بر	6175	امروز	4873	کند	4373
فارس	12146	را	10378	یک	6111	دارد	4863	ادامه	4314
پیام	12120	که	9670	خود	5947	باید	4862	حضور	4288
در	12091	است	8752	گفت	5719	شود	4741	بازی	4170
انتهای	12071	برای	8596	داشت	5597	داد	4528	دو	4119
و	12046	کرد	8179	هم	5528	بود	4527	ما	4109
خبرگزاری	12011	شد	7314	تیم	5454	اما	4472	می‌شود	4084
از	11489	تا	6404	قرار	5113	اسلامی	4465	سال	4012
این	11363	وی	6218	آن	5071	اظهار	4436	اینکه	3993
با	11214	خبرنگار	6178	ایران	4903	کشور	4396	افزود	3911

لیست کامل کلمات حذف شده به عنوان stopwords:

فهرست کامل کلماتی که به عنوان کلمات متوقف کننده در حین پردازش متن حذف می‌شوند، در صفحه Hazm قابل مشاهده است. این لیست شامل 389 کلمه است که در زبان رایج و بی‌اهمیت تلقی می‌شوند و معمولاً برای بهبود دقت وظایف تحلیل متن حذف می‌شوند. برخی از متداول‌ترین کلمات متوقف کننده شامل "در", "آن", "و", "که", "به", "از", "برای", "با", "من", "شما" هستند. با حذف این کلمات متوقف کننده، می‌توانیم بر محتوای معنادارتر متن تمرکز کرده و عملکرد وظایفی مانند تحلیل احساسات، مدل‌سازی موضوع و بازیابی اطلاعات را بهبود بخشیم.

نمودار ۱۰ ترم پرتکرار بعد از نرمال‌سازی و ریشه یابی:



روش های پیاده سازی شده برای امتیاز دهی:

در این پروژه، سیستم جستجوی اطلاعات بر اساس ترکیب چند روش مختلف پیاده سازی شده است که به طور خاص شامل موارد زیر می باشند:

۱. مدل امتیازدهی شباهت کسینوسی

این مدل بر اساس شباهت کسینوسی بین بردار نمایش اسناد و بردار نمایش کوئری امتیازدهی می‌کند. برای هر ترم در کوئری وزنی تعیین می‌شود که بر اساس میزان تکرار آن ترم در کوئری محاسبه می‌شود. وزن اسناد نیز بر اساس محاسبات tf-idf محاسبه می‌شود که شامل میزان تکرار آن ترم در مجموعه اسناد و تعداد اسنادی که حاوی این ترم هستند می‌باشد. این دو وزن در نهایت با یکدیگر ضرب می‌شوند تا امتیاز نهایی برای هر اسناد محاسبه شود.

$$cos(a,\ b) = \frac{a.b}{||a||*||b||} = \frac{\sum_{i=1}^{n} a_i * b_i}{\sqrt{\sum_{i=1}^{n} a_i^{\ 2}}\sqrt{\sum_{i=1}^{n} b_i^{\ 2}}}$$

خروجی در بازه $[0,\ 1]$ خواهد بود.

۲. جستجوی عبارت (Phrase Query)

در این روش، سیستم به دنبال جستجوی عبارات پشت سر هم در اسناد می گردد. برای هر عبارت در کوئری، سیستم موقعیت های مختلف ترکیب های ترم ها در اسناد را بررسی می‌کند و اگر عبارت مورد نظر در اسناد یافت شود، به تعداد یافته ها امتیاز مناسب اختصاص می دهد. این امتیاز بر اساس طول و تعداد عبارات پشت سر هم محاسبه می‌شود.

$$phrase\_query(d,\ q) = 1 + \frac{max_{start}(\sum_{i=1}^{n} \delta(positions(t_i, d), start+i-1))}{n = number\ of\ terms\ in\ q} * pqw$$

در اینجا:

- ترم t_i نمایانگر ترم i ام در کوئری می باشد.
- مقدار n نمایانگر تعداد ترم ها در کوئری می باشد.
- تابع $positions(t_i,\ d)$ نشان دهنده مجموعه موقعیت‌های ترم t_i در داکیومنت d است.
- تابع $\delta(positions,\ pos)$ برای یک مجموعه موقعیت $positions$ و یک موقعیت شروع pos بررسی می‌کند که آیا موقعیت pos در مجموعه موقعیت‌ها وجود دارد یا خیر (مقدار ۱ اگر وجود داشته باشد و مقدار ۰ در غیر این صورت).
- تابع max_{start} نمایانگر بزرگ‌ترین مقدار امتیاز برای هر موقعیت شروع است.
- مقدار pqw نمایانگر وزن موثر این تابع است. (`PHRASE_QUERY_WEIGHT`)

خروجی در بازه $[1,\ 1 + pqw]$ خواهد بود.

۳. امتیازدهی بر اساس تاریخ

برای هر اسناد، امتیاز محاسبه می‌شود که بر اساس فاصله زمانی از تاریخ منتشر شدن آن اسناد تعیین می‌شود. فاصله زمانی کمتر به اسناد امتیاز بیشتری می‌دهد، بنابراین اسناد تازه‌تر به ترتیب زمانی بهتر از دیگران رتبه‌بندی می‌شوند.

$$date\_score(d) = \left|\frac{timestamp(date(d)) - timestamp(min)}{timestamp(max) - timestamp(min)}\right| * dsw$$

در اینجا:

- تابع $timestamp(date(d))$ نمایانگر تایم استمپ داکیومنت d است.
- تابع $timestamp(min)$ نمایانگر تایم استمپ قدیمی ترین داکیومنت در میان تمام داکیومنت ها است.
- تابع $timestamp(max)$ نمایانگر تایم استمپ جدیدترین داکیومنت در میان تمام داکیومنت ها است.
- مقدار dsw نمایانگر وزن موثر زمان است. (DATE_WEIGHT)

خروجی در بازه $[0, dsw]$ خواهد بود.

۴. وزن‌دهی در عنوان و تگ‌ها

در سیستم جستجو، وزن‌دهی کلمات در عنوان (title) و تگ‌ها (tags) نسبت به محتوای اصلی داکیومنت‌ها بیشتر است. این وزن‌دهی با توجه به اهمیت و تاثیر بیشتر کلمات در بخش‌هایی از داکیومنت‌ها که معمولاً حاوی اطلاعات مهم و کلیدی‌تری هستند، انجام می‌شود. این موضوع باعث می‌شود که جستجوهای کاربران به دقت بیشتری در داکیومنت‌های مورد نظر خود دسترسی داشته باشند.

این وزن‌ها در هنگام پیش‌پردازش اسناد محاسبه شده‌اند و در tf نهایی تاثیر مستقیم گذاشته‌اند. در ابتدای کار پیش‌پردازش، هر داکیومنت در هر ترم یک وزن داخلی برابر با ۱ دارد. به ازای هر ترم موجود در عنوان این وزن داخلی در TITLE_TOKEN_WEIGHT ضرب می‌شود و به ازای هر ترم موجود در تگ‌ها این وزن داخلی در TAG_TOKEN_WEIGHT ضرب می‌شود. در نهایت tf بعد از محاسبه شدن در این وزن داخلی ضرب می‌شود.

$$tf_{weighted}(t,\ d) = tf_{initial}(t,\ d)\ * tw^{n(t,\ title(d))} * tagw^{n(t,\ \sum_{i=1}^{tags} tag_i(d))}$$

در اینجا:

- تابع $tf_{initial}(t,\ d)$ نمایانگر tf اولیه ترم t در داکیومنت d است.
- تابع $n(t,\ title(d))$ نمایانگر تعداد تکرار ترم t در عنوان داکیومنت d است.
- تابع $n(t,\ \sum_{i=1}^{tags} tag_i(d))$ نمایانگر تعداد تکرار ترم t در مجموعه تگ های داکیومنت d است.

در نهایت می توان شباهت کسینوسی را با ترکیب tf idf به شکل زیر بازنویسی کرد:

$$cos(d,\ q) = \frac{\sum_{t \epsilon q} tf_{initial}(t, q) * tf_{weighted}(t, d) * idf(t)}{||d||}$$

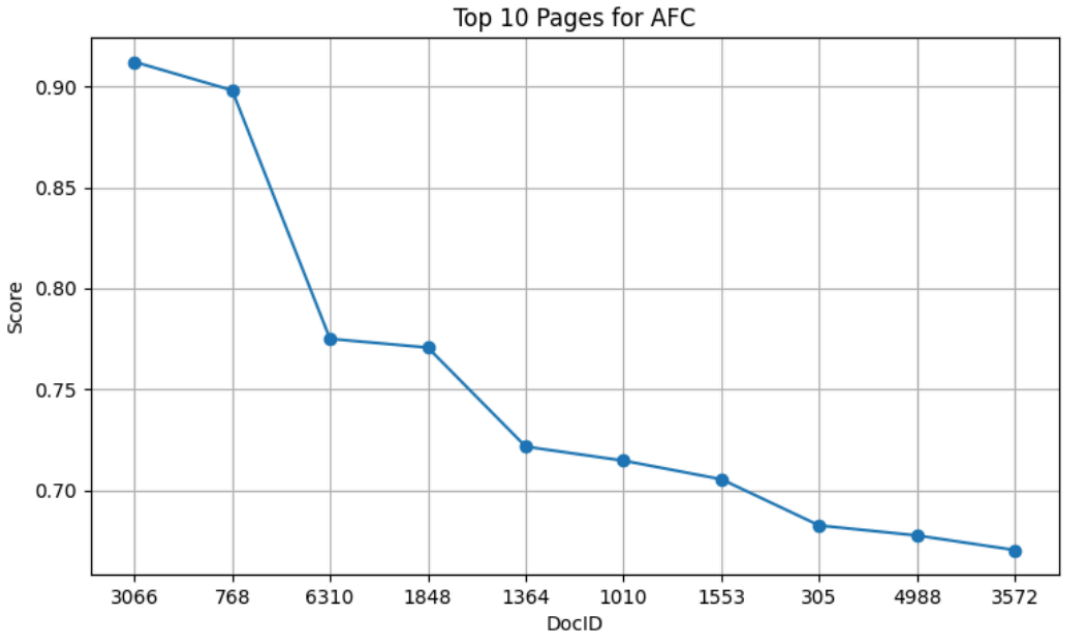
توجه: در پیاده سازی انجام شده از نرمال سازی طول کوئری صرف نظر شده است چرا که تاثیری در انتخاب نهایی ندارد.

در نهایت امتیاز نهایی یک داکیومنت برای یک کوئری مشخص به صورت زیر محاسبه خواهد شد:

$$score(d, q) = cos(d, q) * phrase\_query(d, q) + date\_score(d)$$

بعد از جستجوی هر کوئری یک جدول DocID-Score نیز مشاهده خواهید کرد که میزان شباهت (امتیاز) ۱۰ داکیومنت برتر را به کوئری جستجو شده نشان می دهد.
این امتیاز بازه مشخصی ندارد و هر چقدر بزرگتر شود یعنی شباهت بیشتری نیز داشته است.

برای نمونه با جستجوی عبارت AFC نمودار زیر مشاهده می شود:



افزایش سرعت پردازش کوئری:

- **Champion List**

در سیستم جستجو برای بهبود کارایی و سرعت جستجوها استفاده می‌شود. این لیست شامل اسنادی است که یک ترم خاص در آنها به عنوان مهم‌ترین ترم‌ها یا مدارک شناخته شده است. معمولاً این مدارک بر اساس معیارهایی مانند فراوانی ترم در مدرک، اهمیت محتوا یا موقعیت ترم در مدرک انتخاب می‌شوند. با داشتن Champion List، عملیات جستجو می‌تواند به صورت سریعتر و با کارایی بالاتری انجام شود، زیرا فقط در میان مدارک مهم و تاثیرگذار برای هر ترم جستجو صورت می‌گیرد.
در پیاده سازی نیز موقع ساخت IRData یک آرگومان با نام `champions_list_r` وجود دارد که با ست کردن آن برای هر ترم r سند مهم بر اساس tf.idf ساخته می شود.

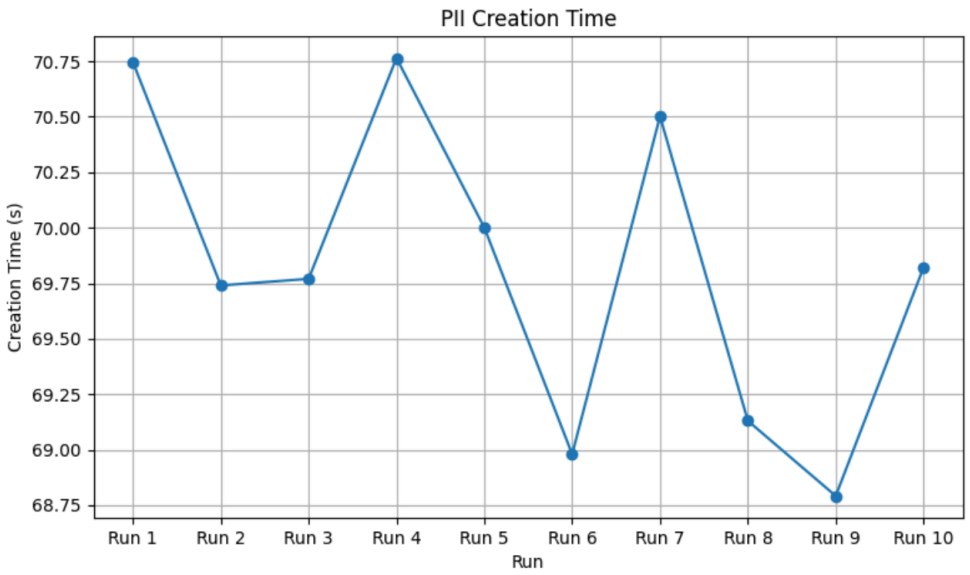
- **Index Elimination**

یک روش است که برای بهبود کارایی و سرعت جستجو در سیستم‌های بازیابی اطلاعات مورد استفاده قرار می‌گیرد. در این روش، اسناد یا مدارکی که شرایط خاصی را نمی‌کنند (مانند فراوانی کم ترم در مدارک یا اعمال شرایط زمانی) از جستجو حذف می‌شوند. این عملیات باعث کاهش حجم اسناد مورد جستجو قرار می‌گیرد و بهبود عملکرد جستجوها را تسریع می‌بخشد. با ست کردن حد آستانه `INDEX_ELIMINATION` می توانیم از این ویژگی نیز در پروژه استفاده کنیم.

بررسی پرفورمنس:

- **پرفورمنس ساخت شاخص مکانی معکوس:**

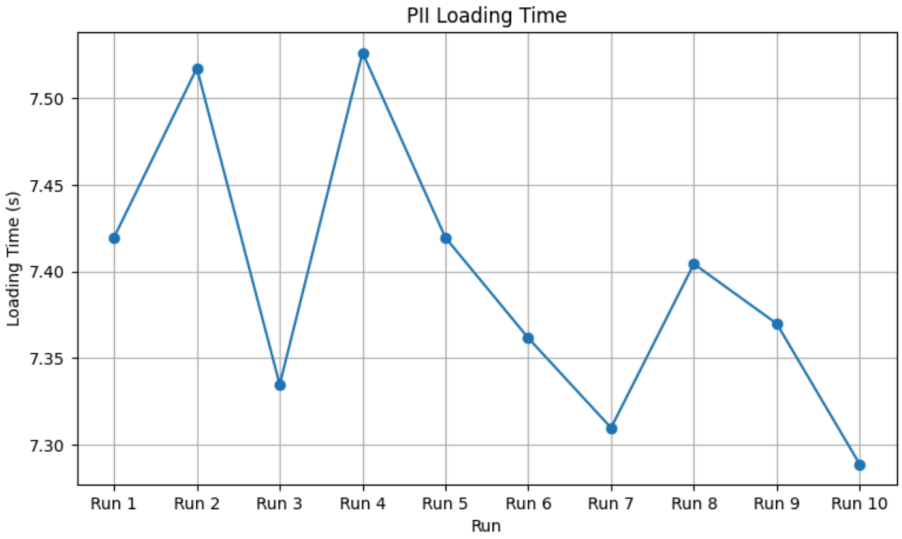
بررسی انجام شده در ده بار اجرا به صورت زیر می باشد:



به طور میانگین 69.8s ساخت شاخص معکوس مکانی طول می کشد.

- **پرفورمنس خواندن شاخص مکانی معکوس از کش:**

بررسی انجام شده در ده بار اجرا به صورت زیر می باشد:

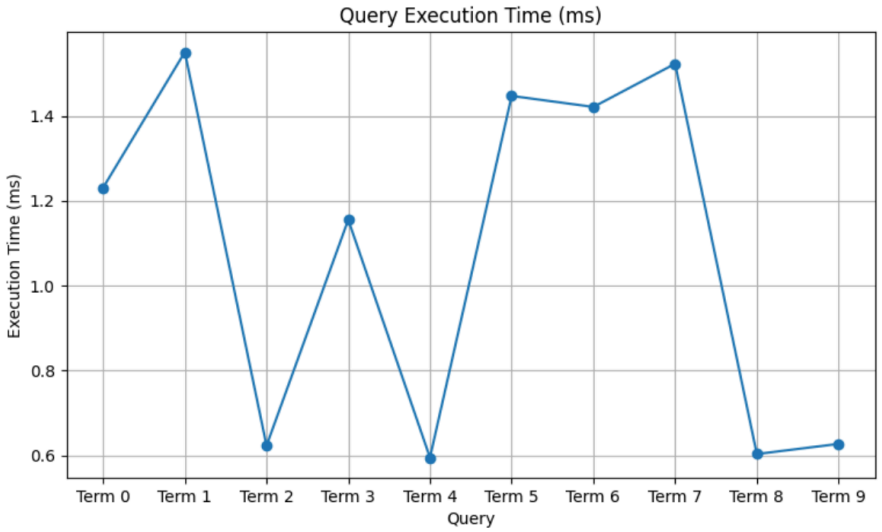


به طور میانگین 7.3s ساخت شاخص معکوس مکانی طول می کشد.

- **پرفورمنس جستجو:**

برای بررسی مدت زمان اجرای جستجو 10 کوئری مختلف که تعداد داکیومنت های قابل توجهی که برای آنها وجود دارد (در دیتای بررسی شده) بررسی شده اند. این کوئری ها به ترتیب عبارت اند از برنامه به ترتیب عبارات جستجوی زیر را ارسال می‌کند: "جام جهانی" - "ایران و عراق" - "طارمی" - "فرهاد مجیدی" - "AFC" - "تست کرونا" - "جشن قهرمانی" - "تیم ملی" - "هنرمندان" و "رئیس جمهور".

خروجی بنچمارک در اجرا های متفاوت شکل خود را حفظ می کند به صورت زیر می باشد:

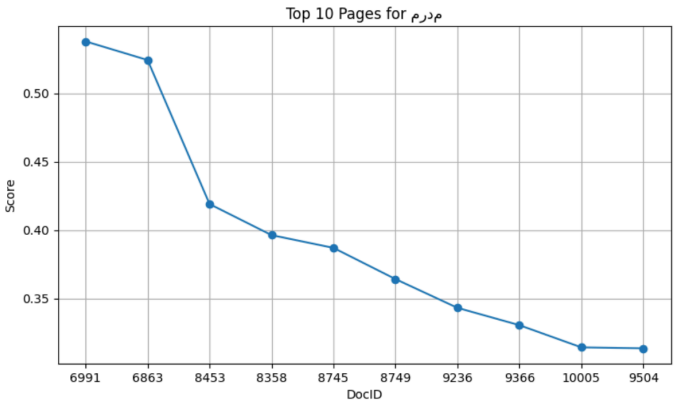


به طور میانگین 1.2ms جستجو کوئری ها طول می کشد.

بررسی صحت درستی جستجو:

با ۴ کوئری مختلف عملکرد جستجو را مورد بررسی قرار میدهیم. در هر کوئری اولین نتیجه را که بیشترین امتیاز را نیز دریافت کرده است تحلیل و بررسی خواهیم کرد.

- **یک پرسمان از کلمات ساده و متداول تک کلمه ای:**
  برای این بخش کلمه "مردم" مورد بررسی قرار گرفته است.



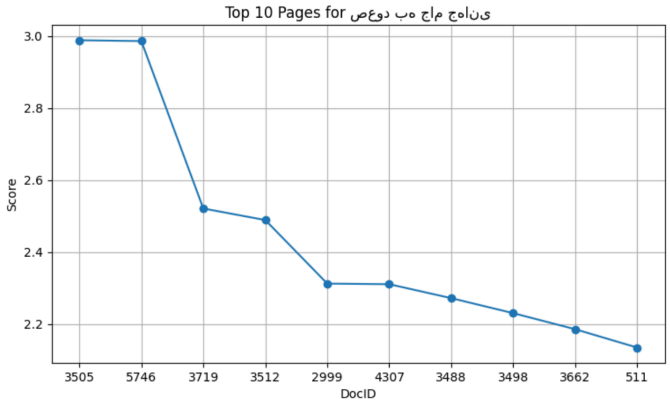
خروجی جستجو:
ID: 6991 | Score: 0.537931842608383
عنوان: تغییر ساعت رسمی کشور زندگی مردم را دچار بی‌نظمی می‌کند
لینک به سایت سند

با توجه به آمار زیر متوجه میشویم که این سند به درستی انتخاب شده است.

- کلمه "مردم" در 3525 سند یافت شده است.
- کلمه "مردم" یک بار در عنوان سند آمده است.
- کلمه "مردم" یک بار در تگ های سند آمده است.
- امتیاز DateScore سند برابر است با 97% که نشان می دهد سند جدیدی است.
- IDF: 1.79
- TF: 5.97
- جملاتی که در آن کلمه "مردم" آمده است:
  - سید البرز حسینی نماینده **مردم** شهرستان خدابنده...
  - ایجاد نوعی بی نظمی در زندگی **مردم** می‌شود.
  - ترافیک مراجعه **مردم** به ادارات را نیز کاهش نداده است.

- **یک پرسمان از عبارات ساده و متداول چند کلمه ای:**

برای این بخش عبارت "صعود به جام جهانی" مورد بررسی قرار گرفته است.



خروجی جستجو:

ID: 3505 | Score: 2.9884161190861445

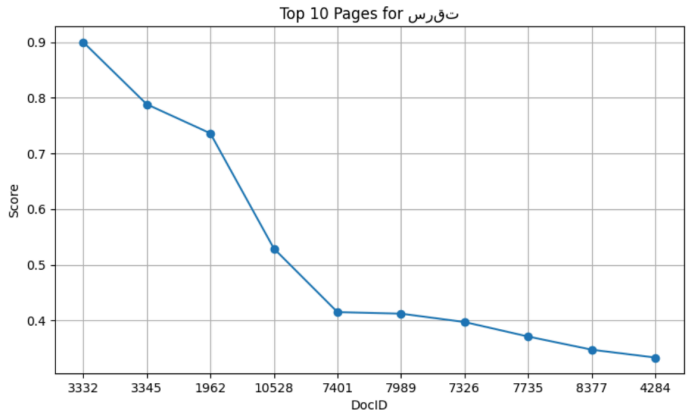
عنوان: امیری: خوشحالم که خیلی زود به جام جهانی صعود کردیم/امیدوارم بردهای تیم ملی ادامه داشته باشد

لینک به سایت سند

با توجه به آمار زیر متوجه میشویم که این سند به درستی انتخاب شده است.

- کلمه "صعود" در 787 سند یافت شده است.
- کلمه "جام" در 1751 سند یافت شده است.
- کلمه "جهانی" در 1488 سند یافت شده است.
- کلمات "صعود" , "جام" و "جهانی" هرکدام یک بار در عنوان سند آمده اند.
- کلمات "صعود" , "جام" و "جهانی" هرکدام یک بار در تگ های سند آمده اند.
- امتیاز DateScore سند برابر است با 68%.
- IDF کلمه صعود: 3.95
- TF کلمه صعود: 5.39
- IDF کلمه جام: 2.80
- TF کلمه جام: 5.39
- IDF کلمه جهانی: 3.03
- TF کلمه جهانی: 5.39
- عبارت "صعود به جام جهانی" در سند آمده است.
- جملاتی که در آن کلمه "مردم" آمده است:
  - قطعی شدن **صعود به جام جهانی** 2022 قطر...
  - توانستیم بازی را ببریم و خیلی زود به **جام جهانی صعود** کنیم.

- **یک پرسمان دشوار و کم تکرار تک کلمه ای:**
برای این بخش کلمه "سرقت" مورد بررسی قرار گرفته است.



خروجی جستجو:
ID: 3332 | Score: 0.9000527321975953
عنوان: ورود پلیس به موضوع سرقت از منزل «فرهاد مجیدی»/ پرونده مراحل قانونی را طی می‌کند
لینک به سایت سند

با توجه به آمار زیر متوجه میشویم که این سند به درستی انتخاب شده است.

- کلمه "سرقت" تنها در 43 سند یافت شده است.
- کلمه "سرقت" یک بار در عنوان سند آمده است.
- کلمه "سرقت" سه بار در تگ های سند آمده است.
- امتیاز DateScore سند برابر است با 70%
- IDF: 8.14
- TF: 10.36
- جملاتی که در آن کلمه "سرقت" آمده است:
  - خبر **سرقت** از منزل فرهاد مجیدی سرمربی تیم فوتبال استقلال
  - متوجه **سرقت** گاو صندوق از منزلش شد
  - خواستار پیگیری **سرقت** از منزلش شد
  - پرونده **سرقت** در دست بررسی بوده

- **یک پرسمان دشوار و کم تکرار چند کلمه ای:**

برای این بخش عبارت "ستاره ایرانی طارمی" مورد بررسی قرار گرفته است.

خروجی جستجو:

ID: 586 | Score: 1.2345798166911153

عنوان: تمجید خبرنگار اماراتی از طارمی؛ او بالاتر از ستاره کره ای است

لینک به سایت سند

با توجه به آمار زیر متوجه میشویم که این سند به درستی انتخاب شده است.

- کلمه "ستاره" در 980 سند یافت شده است.
- کلمه "ایرانی" در 4904 سند یافت شده است.
- کلمه "طارمی" در 234 سند یافت شده است.
- کلمات "ستاره" و "طارمی" هرکدام یک بار در عنوان سند آمده اند.
- کلمات "ستاره" , "ایرانی" و "طارمی" 5 بار در تگ های سند آمده اند.
- امتیاز DateScore سند برابر است با 94% که نشان می دهد سند جدیدی است.
- IDF کلمه ستاره: 3.63
- TF کلمه ستاره: 5.97
- IDF کلمه ایرانی: 3.24
- TF کلمه ایرانی: 2.00
- IDF کلمه طارمی: 5.70
- TF کلمه طارمی: 10.36
- عبارت "ستاره ایرانی" در سند آمده است.
- جملاتی که در آن کلمه "مردم" آمده است:
  - مهدی **طارمی** مهاجم ملی پوش کشورمان
  - این **ستاره ایرانی** شب گذشته برای پورتو مقابل پاسوس ده فریرا یک گل زد
  - درخشش بی نظیر **طارمی** در لیگ پرتغال به تمجید از **ستاره ایرانی** پرداخت
  - آنچه که **طارمی** در این فصل به همراه پورتو در لیگ پرتغال به نمایش گذاشته
  - آمار **طارمی** در این فصل از لیگ پرتغال به شرح زیر است:
  - بالاتر از سون هیونگ مین **ستاره** تاتنهام بهترین بازیکن آسیاست.

جمع بندی:

در دنیای امروزی پر از اطلاعات، بازیابی اطلاعات یکی از مسائل اساسی در علوم کامپیوتر و مهندسی دانش است که برای جستجو، فیلتر کردن، و یافتن اطلاعات مورد نیاز از مجموعه‌ای از اسناد متنی، اساسی است. هدف اصلی این فرآیند، یافتن و بازیابی اسناد مرتبط با کوئری‌های کاربر است. این فرآیند برای سیستم‌های جستجو، موتورهای جستجوی وب، پایگاه‌های داده، و سیستم‌های توصیه‌گر اطلاعاتی بسیار حیاتی است.

در نهایت، استفاده از شاخص معکوس مکانی به‌عنوان یکی از روش‌های پیشرفته جستجو در متون، از اهمیت بسیاری برخوردار است و در بهبود کارایی و دقت سیستم‌های جستجو و بازیابی اطلاعات تأثیر زیادی دارد که در این پروژه تلاش شده است تا حد ممکن یک نسخه کارآمد از آن پیاده سازی شود و مورد بررسی قرار بگیرد.

دریافت سورس کد پروژه