« بسمه تعالی »

بازیابی اطلاعات پیشرفته

Modern Information Retrieval

مدرس: دکتر میثم روستائی

دانشگاه مازندران

تمرین اول | تاریخ تحویل: ۱۴۰۲/۰۲/۰۱

تمرین اول درس بازیابی اطلاعات پیشرفته شامل سه بخش می‌باشد. هدف از این تمرین آشنایی با مباحثی همچون تجزیه کردن داکیومنت‌ها (Document Parsing)، پیش‌پردازش (Preprocessing) آن‌ها، آشنایی با مفاهیمی همچون Term Frequency و Document Frequency و در نهایت ایجاد شاخص معکوس (Inverted Index) است. دقت کنید که ابزارهایی نظیر Lucene وجود دارند که به راحتی امکان انجام کل تمرین را با دستورات ساده‌ای به ما می‌دهند اما در اینجا هدف آشنایی با فرآیند بازیابی اطلاعات است در نتیجه سعی در انجام مراحل به صورت دستی خواهیم داشت و هر جا که امکان آن وجود داشته باشد، ابزارهای قابل استفاده معرفی خواهد شد.

دیتاست مورد استفاده در این تمرین، بخشی از OpinRank[1] است که همراه با صورت مسئله ارائه شده است. این مجموعه داکیومنت‌ها شامل متون جمع آوری شده از نظرات کاربران در مورد اتومبیل‌ها بین سال‌های ۲۰۰۷-۲۰۰۹ می‌باشد. در پوشه داده‌ها، سه زیرپوشه مختلف (۲۰۰۷, ۲۰۰۸, ۲۰۰۹) وجود دارد که نشان دهنده سال تولید خودرو است. هر فایل (در این سه پوشه) شامل تمام بررسی‌ها (reviews یا نظرها) برای یک خودرو خاص است. نام فایل نشان دهنده نام خودرو است. در هر فایل خودرو، مجموعه‌ای از بررسی‌ها را در قالب زیر مشاهده خواهید کرد:

```
<DOC>
 <DATE>06/15/2009</DATE>
 <AUTHOR>The author</AUTHOR>
 <TEXT>The review goes here</TEXT>
 <FAVORITE>What are my favorites about this car</FAVORITE>
</DOC>
```

توجه داشته باشید که هر بررسی در داخل یک عنصر <DOC> همانطور که در بالا نشان داده شده است محصور می‌شود و همه اطلاعات قابل استخراج در این عنصر قرار دارند.

بخش اول: تجزیه (Parsing)

گام اول تجزیه فایل‌های دیتاست مسئله است. برای این کار می‌توانید به شیوه‌های متفاوتی عمل کنید. می‌توانید از رویکردهای موجود برای پارس کردن فایل‌های XML نظیر SAX و یا DOM استفاده کنید که در زبان‌های مختلف، کتابخانه‌های متفاوتی برای آن‌ها وجود دارد. و یا اینکه به شیوه پیشنهادی خودتان برای پارس کردن متن استفاده کنید. بنابراین در این گام می‌بایست موارد زیر انجام شود:

- تمام فایل‌های دیتاست را به ترتیب پارس کنید.
- عنصر اطلاعات متناظر با هر بررسی (نظر) را می‌توان به عنوان یک داکیومنت در نظر گرفت.
- به هر داکیومنت، یک شماره اختصاص دهید (DOCID).
  - سعی کنید نام فایلی که بررسی را از آن استخراج می‌کنید در جایی در کنار DOCID‌ها ذخیره کنید زیرا در تمرین‌های بعدی در فرآیند جستجوی پرس‌وجو (Query) به آن نیاز خواهیم داشت.
- متن موجود در تگ‌های TEXT (<TEXT>) و FAVORITE (<FAVORITE>) را استخراج کنید.

[1] Opinion Based Entity Ranking Dataset

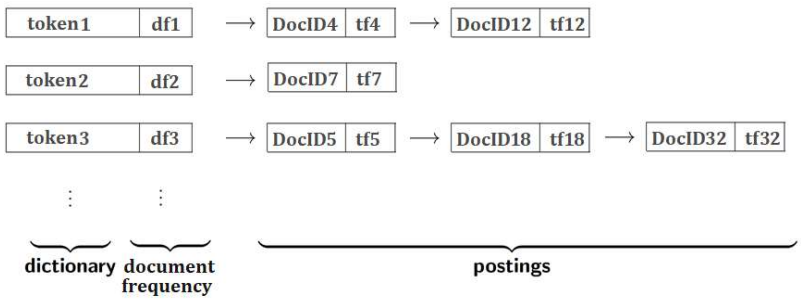
بخش دوم: پیش‌پردازش (Preprocessing)

در این گام به پیش‌پردازش داده‌های استخراج شده خواهیم پرداخت. در واقع جهت استفاده از متن خام استخراج شده می‌بایست چندین مرحله پیش‌پردازش بر روی آن‌ها انجام شود. پیش‌پردازش‌ها می‌تواند متفاوت باشد اما دست کم موارد زیر می‌بایست انجام شود:

- جداسازی توکن‌ها (Tokenization): استخراج توکن‌های موجود در متن هر داکیومنت
- حذف علائم نگارشی و کاراکترهای ناخواسته مانند ؟ و ! و ...
- نرمال سازی نمایش تمام توکن‌ها با تبدیل آن‌ها به حروف کوچک انگلیسی
- حذف کلمات ایست (Stopwords): برای این منظور از لیست ارائه شده در فایل stopwords.txt که همراه صورت مسئله ارائه شده است استفاده کنید.
- عملیات ریشه یابی: برای این منظور می‌توانید از Lemmatization و یا Stemming استفاده کنید. یکی از الگوریتم‌های معروف برای Stemming، الگوریتم Porter است که در کتابخانه‌های مربوط به آن در تمام زبان‌های برنامه‌نویسی موجود و قابل استفاده است.

بخش سوم: شاخص معکوس (Inverted Index)

پس از انجام عملیات پیش‌پردازش و استخراج توکن‌های نهایی، می‌توان شاخص معکوس را ایجاد کرد. هدف از ایجاد شاخص معکوس انجام سریع‌تر فرآیند بازیابی اطلاعات است. در قالب کلی، ساختار شاخص معکوس می‌بایست به این صورت باشد که به ازای هر کلمه، ID داکیومنت‌هایی که شامل آن کلمه هستند به همراه میزان تکرار آن کلمه در آن داکیومنت (TF: Term Frequency) مشخص شود. همچنین برای هر کلمه موجود در شاخص، تعداد داکیومنت‌های حاوی آن کلمه (DF: Document Frequency) نیز محاسبه و در شاخص معکوس ذخیره شود.



- برای پیاده‌سازی شاخص معکوس، می‌توانید از هر ساختمان داده دلخواه استفاده کنید اما پیشنهاد می‌شود برای سادگی، از ساختارهای لیست یا Map بهره ببرید.

نکاتی در مورد انجام تمرین:

خروجی تمرین اول در گام‌های بعدی نیز استفاده خواهد شد. بنابراین بر درستی انجام تمرین اطمینان حاصل کنید. انجام تمرین در قالب گروه‌های حداکثر دو نفره امکان‌پذیر است. پاسخ تمرین را در قالب یک فایل فشرده که شامل (۱) کد برنامه، (۲) گزارش و (۳) خروجی متنی شاخص است را در مهلت تعیین شده در سامانه ارسال بفرمایید. ضمنا تاریخی برای تحویل حضوری تمرین مشخص خواهد شد که در آن هر گروه موظف به پاسخ به سوالات در مورد نحوه کد نویسی و عملکرد هر بخش می‌باشد. همه اعضای گروه باید بر کل هر بخش تسلط کامل داشته باشند.

فایل گزارش می‌بایست شامل موارد زیر باشد:

- بیان شیوه پارس کردن داکیومنت‌ها (معرفی کتابخانه استفاده شده و نحوه استفاده از آن و یا روش تجزیه خودتان)
- معرفی کتابخانه استفاده شده برای ریشه‌یابی
- توضیح مختصری در مورد پیش‌پردازش‌های انجام شده و تاثیر احتمالی بر فرآیند بازیابی.
- ارائه آمارهایی از پردازش نظیر تعداد داکیومنت‌های استخراج شده از دیتاست، تعداد اتومبیل‌ها در هر سال و ...
- تشریح ساختار داده استفاده شده برای پیاده سازی شاخص معکوس و نحوه پیاده سازی آن همراه با اطلاعاتی نظیر تعداد توکن‌ها (اندازه لغت‌نامه)، حداکثر، حداقل و میانگین طول Posting list و ...

خروجی متنی شاخص، یک نمایش متنی از فایل شاخص ایجاد شده است که به ازای هر توکن در دیکشنری، مشخص می‌کند در چه تعداد داکیومنت رخ داده (Document Frequency) و سپس این توکن در چه داکیومنت‌هایی و در هر یک به چه تعداد ظاهر شده داده است. برای شاخص معکوس شکل فوق، نمایش متنی به صورت زیر خواهد بود:

token1: df1

DocID4:tf4

DocID12:tf12

---

token2: df2

DocID7:tf7

---

token3: df3

DocID5:tf5

DocID18:tf18

DocID32:tf32

موفق باشید