شرح مختصر

بر اساس آموخته‌های کلاس و دیتاست زیر، یک Inverted Index با جزییات زیر ایجاد نمایید.

شرح تفصیلی

هدف از این تکلیف، آشنایی دانشجویان با مفاهیمی مانند Term Frequency، Document Frequency و Inverted Index در فرایند بازیابی اطلاعات می‌باشد. امروزه کتابخانه‌های آماده‌ای مانند Apache Lucene وجود دارند که بسیاری از محاسبات و عملیات مورد نیاز در فرایند بازیابی اطلاعات را انجام می‌دهند و خروجی شامل Inverted index را ارائه می‌کنند که قابلیت جستجو و بسیاری عملیات دیگر را دارد. از این کتابخانه‌های آماده به راحتی می‌توان در ایجاد یک سیستم بازیابی اطلاعات و جستجوی وب، استفاده نمود. برای بهره‌برداری از چنین امکاناتی، دانشجو بایستی مفاهیم پایه‌ی درس را به خوبی تمرین کرده باشد. در نتیجه دانشجویان باید فرایند پارس نمودن اولیه‌ی محتوی مجموعه دادگان و عملیات پیش‌پردازش متون و تشکیل ساختار Inverted index را به یک زبان برنامه‌نویسی دلخواه پیاده‌سازی نمایند. مراحل کلی کار به صورت زیر می‌باشد:

۱. تجزیه: قدم اول تجزیه (Parse) فایل‌های دیتاست‌هایی است که در اختیار دارید. برای این کار می‌توانید از کتابخانه‌های موجود برای پارس نمودن فایل‌های XML، مانند SAX Parser یا Lucene استفاده نمایید، یا اینکه پارسر خاص خودتان را پیاده‌سازی نمایید. لذا کارهایی که باید در مرحله‌ی تجزیه‌ی فایل‌های خام مجموعه دیتاست‌ها انجام دهید، شامل موارد زیر است:
   a. تمام فایل‌های مجموعه دیتاست‌ها را به ترتیب پارس نمایید.
   b. قطعه اطلاعات متناظر با هر نظر را می‌توان به عنوان یک سند (Document) متنی فرض کرد.
   c. به هر سند یک شماره DocID اختصاص دهید:
      i. سعی کنید نام فایلی که نظرات از آن استخراج می‌شود در فهرستی در کنار DocIDها ذخیره شود (برای استفاده‌های آتی).
   d. متن موجود در محتوی سند (تگ <TEXT> و <FAVORITE>) را استخراج نمایید.

۲. پیش‌پردازش: پس از استخراج متن خام از فایل‌های دیتاست، نیاز است که برخی عملیات پیش‌پردازش بر آن‌ها اعمال شود. بر اساس آموخته‌های کلاس، عملیات پیش‌پردازش می‌تواند متنوع و به صلاحدید شما انجام شود، اما دستکم باید موارد زیر را انجام دهید:

a. Tokenizing: استخراج کلمات موجود در متن محتوی هر سند و حذف کاراکترهای ناخواسته مانند علایم و ...

b. نرمال‌سازی نمایش تمام کلمات به صورت حروف کوچک انگلیسی

c. اعمال عملیات حذف stop wordها، با استفاده از لیست stop wordهای داده شده در فایل stopword.txt به همراه دیگر فایل‌های مربوط به این تکلیف

d. عملیات Stemming: Stemming را می‌توانید با استفاده از الگوریتم‌ها و پیاده‌سازی‌های موجود از آن‌ها مانند Porter انجام دهید (پیاده‌سازی‌های مختلف از الگوریتم Porter به زبانهای مختلف برنامه‌نویسی وجود دارد که به راحتی قابل جستجو، دسترسی و بهره‌برداریست و نیازی نیست خودتان آن را پیادهسازی نمایید)

۳. ساخت شاخص معکوس (Inverted index): پس از انجام عملیات پیش‌پردازش و استخراج tokenهای نهایی، می‌توان ساختار شاخص معکوس را ایجاد کرد. برای انجام این مرحله، به ساختمان دادهایی که قابلیت جستجو و نمایش نتیجه جستجو را داشته باشد، نیاز است. در حالت کلی، ساختمان داده شاخص معکوس، باید علاوه بر ID اسناد حاوی یک کلمه، به ازای هر سند، تعداد تکرار آن کلمه در سند (TF) را نیز داشته باشد. همچنین برای هر کلمه موجود در شاخص، تعداد اسناد حاوی آن کلمه (DF) نیز محاسبه و در شاخص معکوس ذخیره شود. ساختار دادهای که در پیاده‌سازی از آن استفاده می‌کنید دلخواه است، ولی بهتر است از ساختارهای لیست یا map موجود در زبان‌های برنامه‌نویسی باشد که مدیریت آن‌ها ساده‌تر صورت گیرد.

دیتاست

دادگان در نظر گرفته شده برای این تمرین، بخشی از مجموعه دادگان متنی OpinRankDataset می‌باشد. این این مجموعه دادگان شامل متون جمع‌آوری شده از نظرات کاربران درمورد تعدادی از اتومبیل‌ها در

سال‌های ۲۰۰۷، ۲۰۰۸ و ۲۰۰۹ می‌باشد و از طریق لینک زیر قابل دانلود می‌باشد (حجم فایل ۱۰,۲ مگا بایت).

https://www.dropbox.com/s/ufzyoghgpd8nps1/OpinRankDataset.zip?dl=0

جزییات مربوط به محتوی و ساختار این مجموعه داده‌ها و تگ‌های مختلف آن، در فایل **OpinRankDataset.pdf** به همراه فایل‌های اصلی دیتاست، قابل مشاهده و دسترسی است.

خروجی

دانشجویان باید فرایند کار خود در این تمرین را به صورت یک گزارش مستند نوشته و به همراه فقط کدهای پیاده‌سازی شده ارسال نمایند (و نه فایل پروژه و ...). از جمله نکاتی که باید در گزارش به آن‌ها پرداخته شود شامل موارد زیر است:

۱. معرفی کتابخانه‌های استفاده شده در بخش‌های مختلف کار و توضیح مختصری درمورد نحوه استفاده از این کتابخانه‌ها: مثلاً برای parseکردن فایل‌های مجموعه دادگان و یا عملیات stemming.

۲. تهیه آمار اولیه در قالب یک جدول از تعداد اسناد استخراج شده از مجموعه دادگان، تعداد کل اتومبیل‌های متمایز (از لحاظ برند)، تعداد اتومبیل‌ها در هر سال، تعداد کل نظرات در هر سال و درتمام سالها، تعداد میانگین نظرات برای هر مدل اتومبیل در سال‌های متفاوت و در کل سال‌ها.

۳. تشریح مراحل انجام عملیات پیش‌پردازش متن و جزئیات مختصری از آن. بدیهیست اگر کار اضافه‌ای نسبت به حداقل‌های بیان شده در مستند جاری انجام داده‌اید، باید در این بخش شرح دهید تا در ارزیابی مدنظر قرار گیرد. درنهایت توضیح دهید که فرایند پیش‌پردازشی که انجام داده‌اید چه تاثیری بر فرایند بازیابی خواهد داشت.

۴. تهیه آماری از تعداد کلمات متمایز موجود در کل مجموعه اسناد پردازش شده در قالب جدولی مقایسه‌ای شامل تعداد کل tokenها قبل و بعد از حذف علائم ناخواسته، تعداد tokenها پس از حذف stop wordها و تعداد tokenها پس از stemming.

۵. تشریح ساختارداده‌نهایی شاخص معکوس و جزئیات مربوط به آن (نحوه پیاده‌سازی، عملیات قابل انجام روی این ساختار داده، اطلاعات موجود در آن و ..) همراه با جدولی شامل تعداد کلمات

موجود در شاخص، حداکثر، حداقل و میانگین طول posting list مربوط به کلمات موجود در دیکشنری شاخص معکوس.

۶. جدولی شامل لیست ۲۰ مورد از پرتکرارترین کلمات در کل اسناد به همراه تعداد کل رخدادآن‌ها.

۷. جدولی شامل لیست ۲۰ مورد از کلماتی که در اسناد زیادی رخ داده اند به همراه تعداد اسناد حاوی آن‌ها

نحوه‌ی ارزیابی و تحویل

دانشجویان می‌توانند کار خود را به صورت تیمی (تیم‌های حداکثر سه نفره) انجام دهند. موارد زیر را در یک فایل ZIP، تا تاریخ ۹۷/۲/۵ در ایمیلی با عنوان IR-Hmwrk1 به آدرس ostademajazi@gmail.com ارسال نمایید.

- کدهای پیاده‌سازی همراه با توضیحات مفید مرتبط با کلاس‌ها و متدها به صورت comment (فقط فایل‌های کد برنامه‌ها را ارسال نمایید تا حجم بسته زیاد نشود. بعدا در زمانی مناسب در جلسه کلاس، به صورت حضوری درمورد سورس کد سوال خواهد شد و تمام افراد گروه باید نسبت به کلیت کار آگاهی داشته و به بخشی که خود پیاده‌سازی نموده‌اند تسلط داشته باشند.
- فایل گزارش کار
- یک فایل متنی مستخرج از شاخص معکوس به این صورت که به ازای هر token موجود در دیکشنری شاخص، یک سطر شامل tokenID, tokenString, DocumentFrequency نوشته شده باشد.
- یک فایل متنی متناظر با شاخص معکوس به این صورت که به ازای هر token موجود در شاخص، به ازای اسنادی که این کلمه در آن‌ها ظاهر شده است سطرهایی داشته و در هر سطر اطلاعاتی به فرمت tokenID, docID, TermFrequency نوشته شده باشد. به عنوان مثال اگر یکی از tokenها کلمه Memphis با ID برابر با 7865 باشد که در اسناد شماره 455، 23 و 1385 به

ترتیب به تعداد 3 ،5و12 مرتبه رخ داده باشد، خروجی متناظر آن در فایل موردنظر به صورت سه سطر زیر است:

.....

7865, 23, 5
7865, 455, 3
7865, 1385, 12

.....

- کپی کردن تکلیف سبب لحاظ کردن نمره صفر برای تمام افراد گروه‌های مربوطه خواهد شد.
- مهلت تعیین شده به هیچ وجه تمدید نخواهد شد و عدم ارسال در زمان مقرر (حتی با فاصله یک ساعت دیرتر)، سبب لحاظ کردن نمره صفر برای تکلیف خواهد شد.