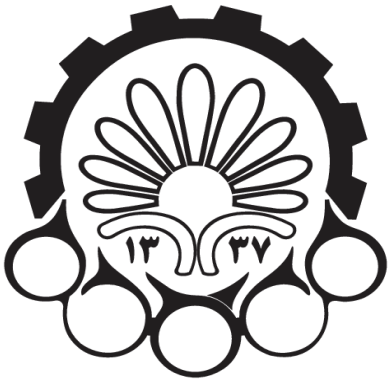

دانشگاه صنعتی امیرکبیر

دانشکده مهندسی کامپیوتر

درس:

بازیابی اطلاعات

تعریف پروژه – مرحله اول

نیمسال دوم

سال تحصیلی ۹۸–۹۹

۱- مقدمه

هدف پروژه‌های درس، پیاده‌سازی عملی الگوریتم‌های آموزش داده شده در بخش‌های مختلف درس است. در هر یک از پروژه‌ها بخش‌هایی از یک موتور بازیابی اطلاعات پیاده‌سازی خواهد شد. جزئیات مربوط به نحوه انجام پروژه در ادامه توضیح داده شده است.

۲- مجموعه داده پروژه

مجموعه داده مورد استفاده در این پروژه یک مجموعه‌ی ۵۰۰۰۰ خبری از خبرهای واکشی شده از چند وب‌سایت خبری فارسی است که در قالب چند فایل csv در اختیار شما قرار خواهد گرفت. هر سطر این فایل حاوی یک خبر خواهد بود. برای هر خبر اطلاعات زیر در فایل مذکور وجود دارد:

- تاریخ و ساعت انتشار خبر
- عنوان خبر
- خلاصه خبر
- متن اصلی خبر
- کلمات کلیدی
- نام خبرگزاری یا سایت انتشار دهنده خبر
- لینک عکس مربوط به خبر

۳- مرحله‌ی اول پروژه

در این مرحله از پروژه شما می‌بایست هر خبر را از مجموعه‌ی اسناد ورودی واکشی کرده و اقدامات لازم را برای ساخت شاخص معکوس انجام دهید و به بررسی قوانین Zipf و Heaps بر روی اسناد ورودی، و مقایسه نتایج در دو حالت گفته شده بپردازید. در ادامه به توضیح بیشتر مراحل انجام پروژه پرداخته می‌شود.

۳-۱ ساخت شاخص معکوس

اولین مرحله در پیاده‌سازی یک موتور جستجو، شاخص‌گذاری مجموعه اسناد و ساخت دیکشنری است. برای شاخص‌گذاری اسناد لازم است بخش‌های زیر پیاده‌سازی شوند:

- واکشی خبر
- استخراج توکن
- نرمال‌سازی (Normalization)
- ریشه‌یابی کلمات (Stemming)
- حذف کلمات پرتکرار (Stop words)

ساخت شاخص معکوس باید در دو حالت زیر انجام شود و در نهایت تاثیر پردازش‌های صورت گرفته بر روی تعداد کلمات منحصر بفرد استخراج شده بررسی شود.

۳-۱-۱ حالت اول

باید ابتدا هر خبر واکشی شود و موارد بی‌معنی مانند اعداد، برچسب‌های html و علائم نگارشی مانند علامت سوال، علامت تعجب، ویرگول و ... با یک فاصله جایگزین شوند. سپس براساس فاصله، استخراج توکن انجام شود. یعنی برای استخراج توکن‌های هر متن، کلمات آن بر اساس فاصله از هم جدا شوند. بعد از استخراج توکن‌ها، کلمات پر تکرار از متن حذف شوند. حذف کلمات پر تکرار بدین صورت است که ابتدا یک لیست (فایل) از این کلمات تهیه کرده و هر توکن از متن خبر را با این لیست مقایسه می‌کنید و اگر در لیست وجود داشت باید حذف شود. این لیست شامل کلماتی مانند "در"، "برای"، "به"، "چون"، "است"، "مانند"، "باید" و ... است. برای تهیه این لیست می‌توانید از لیست‌های موجود بر روی وب استفاده کنید. پس از حذف کلمات پرتکرار با استفاده از توکن‌های باقی‌مانده طبق آنچه در درس بازیابی اطلاعات آموزش داده شده است شاخص معکوس ساخته شود.

۳-۱-۲ حالت دوم

در این مرحله می‌بایست بخش‌های پردازش اسناد شامل نرمال‌سازی، ریشه‌یابی کلمات و استخراج توکن تکمیل شود و بعد از اعمال این پردازش‌ها و حذف کلمات پر تکرار، ساخت شاخص معکوس ساخته شود. جزئیات هر یک از این موارد در ادامه توضیح داده شده است.

۳-۱-۲-۱ نرمال‌سازی متن

این بخش از سامانه، وظیفه‌ی یکسان‌سازی کاراکترها و پردازش مناسب کاراکترهای غیر الفبایی را برعهده دارد. برخی کاراکترهای فارسی دارای تنوع هستند و گاه با نسخه‌های عربی خود جایگزین می‌شوند. به عنوان نمونه کاراکترهای «ك» و «ک» دو شکل نوشتاری از یک کاراکتر هستند. این مسئله برای استخراج دیکشنری و مقایسه کلمات مناسب نیست و باید متن وروی به شکلی استاندارد نرمال شود. در این بخش شما باید یک لیست از تمام کاراکترهای موجود استخراج کنید و در صورت نیاز هر کدام از آنها را به یک کاراکتر استاندارد نگاشت کنید. همچنین کاراکترهای غیر الفبایی مثل اعراب‌ها، انواع نیم‌فاصله، انواع جداکننده، کاراکترهای غیرمعمول، ایموجی‌ها و علامت‌های نشانه‌گذاری مثل نقطه، ویرگول و ... را به طور مناسب برای موتور جستجو پردازش کنید.

۳-۱-۲-۲ استخراج توکن

این بخش از موتور جستجو وظیفه‌ی تکه‌تکه کردن متن ورودی را برعهده دارد به طوری که هر تکه از آن یک کلمه (ترم) با معنی و کامل باشد. در طراحی و پیاده‌سازی این بخش باید نکات زیر مد نظر قرار داده شوند:

- افعال به هر شکلی که در ورودی ظاهر شدند، یک کلمه در نظر گرفته شوند. به عنوان مثال اگر «می‌توانسته‌ام» در ورودی به شکل «می توانسته ام» یا هر شکل دیگری ظاهر شده باشد باید یک کلمه در نظر گرفته شود.

- اگر یک کلمه با علامت‌های جمع به کار رفته بود، یک کلمه در نظر گرفته شود. مثلا «درخت ها» یک کلمه است.
- عبارت‌های ترکیبی پرکاربرد که تکه‌هایشان غالباً در کنار هم ظاهر می‌شوند باید یک تکه در نظر گرفته شوند. به عنوان مثال عبارت‌های «فی‌مابین»، «چنان چه»، «بنا بر این»، «علی ای حال»، «مع ذلک» و ... همه یک کلمه محسوب می‌شوند و به هر شکلی که در سند ظاهر شدند باید یک تکه در نظر گرفته شوند. یک لیست ۲۰ تایی از این عبارات تهیه کنید و برای تشخیص این عبارت‌ها از آن استفاده کنید. (آیا می‌توانید روشی خودکار برای استخراج این نوع عبارات پیشنهاد کنید؟)

۳-۲-۱-۳ ریشه‌یابی کلمات

در این بخش از سامانه کلمات به ریشه‌شان تبدیل می‌شوند تا در ادامه‌ی پردازش از نسخه‌ی ریشه‌ی کلمه استفاده شود تا تفاوت در صرف کلمه باعث نشود یک سند در نتیجه‌ی جستجو ظاهر نشود. به عنوان مثال اگر پرسمان ورودی «روش‌های پختن عدسی» باشد، ما می‌توانیم اسنادی که در آنها کلمه‌های «روش»، «پختم» و «عدس» ظاهر شده‌اند هم در نتایج جستجو بیاوریم. در طراحی و پیاده‌سازی بخش ریشه‌یاب کلمات باید نکات زیر در نظر گرفته شوند:

- فعل‌ها به ساده‌ترین شکل خود تبدیل شوند.
- کلمات جمع به ساده‌ترین شکل خود تبدیل شوند.
- پیشوندها یا پسوندهای چسبیده به کلمات حذف شوند.

مثال‌های جدول زیر را در نظر بگیرید:

ورودی	ریشه	ورودی	ریشه
می‌روم	رو	**درختان**	درخت
گفتند	گفت	**کتابم**	کتاب
می‌خواهید	خواه	**عادلانه‌ترین**	عادلانه
رفته است	رفت است	**جعبه‌ای**	جعبه
شوند	شو	**خانه‌هایمان**	خانه

برای بررسی شهودی عملکرد ریشه یاب پروژه خود، لیست کلماتی که به هرکدام از موارد زیر نگاشت شده است را در گزارش خود بیاورید:

گفت، گو، رود، رو، خواه، سپاس، هنر، شریف، دوست، یاد، توان، شنو، کرد، ساز، دان،

۳-۱-۳ بررسی قوانین Zipf , Heaps

در این بخش باید بررسی کنید که مجموعه اسناد مورد استفاده در این پروژه تا چه حد با قوانین Zipf , Heaps سازگاری دارند. برای این کار می‌توانید به روش زیر عمل کنید. ابتدا دو زیرمجموعه اسناد مثلا ۵۰۰۰ و ۱۵۰۰۰ تایی بصورت تصادفی از مجموعه اسناد داده‌شده انتخاب کنید و با تعداد اسناد و تعداد ترم‌های متمایز آن‌ها ، پارامترهای قانون Heaps را

بیابید. سپس تعداد ترم‌های پیش‌بینی شده توسط این قانون را با اندازه واقعی آن‌ها مقایسه کنید.(این کار را برای هر دو حالت شاخص معکوس انجام دهید و نتیجه را مقایسه کنید.)

همچنین برای قانون zipf پارامتر K را برابر با تعداد تکرار اولین پرتکرارترین کلمه دیکشنری در نظر بگیرید و قانون را برای هر دو حالت شاخص معکوس ساخته شده بررسی کنید. در نهایت نمودارهای این دو قانون را برای هر دو حالت تحلیل کنید و در گزارش خود ضمیمه کنید.

توجه : همانند اسلاید درس محورهای نمودار قانون Zipf را $\log^{cf}$, $\log^{rank}$ و محورهای نموار قانون Heaps را $\log^{T}$, $\log^{M}$ در نظر بگیرید.

۴- نکات پیاده سازی

- در این پروژه محدودیتی برروی زبان برنامه‌نویسی استفاده شده وجود ندارد.
- در قسمت پردازش متن برای بخش ریشه یابی و استخراج توکن استفاده از ابزارهای آماده مانعی ندارد اما برای نرمال‌سازی نباید از ابزار آماده استفاده کنید.
- به دلیل زیاد بودن حجم اسناد ورودی، سعی شود برای ساخت شاخص معکوس از فشرده‌ترین و بهینه‌ترین ساختمان‌داده استفاده شود. می‌توانید در پایتون از دیکشنری و در جاوا از HashMap استفاده کنید. بدین صورت که در ابتدا یک دیکشنری ساخته که در آن کلیدهای دیکشنری، ID کلمات متمایز هست و مقدار هر کلید یک لیست از شماره اسنادی است که کلمه‌ی کلید در آن‌ها وجود دارد.
- خبر هایی که در اختیار شما قرار می‌گیرد علاوه بر متن خبر شامل موارد اضافی مانند تگ‌های html است، پس از واکشی خبر و قبل از هر اقدامی باید این موارد اضافی را از خبرها حذف کنید و تنها بر روی متن اصلی خبر کار کنید.
- برای ساخت شاخص معکوس فقط قسمت متن اصلی خبر را پردازش کنید .

۵- نکات مهم

- پروژه به صورت فردی انجام می‌شود.
- به همراه فایل‌های پیاده‌سازی، یک گزارش کتبی (شامل نحوه پیاده‌سازی کلیه قسمت‌های پروژه، پاسخ به سوالات، تحلیل‌ها و نمودارها) را نیز آپلود کنید.