به نام خدا

گزارش پیشرفت پروژه سوم درس بازیابی پیشرفته اطلاعات

میناقشمی ۸۵۱۰۶۶۰۹ ، مهرانه لیایی ۸۵۱۰۷۳۹۵

این پروژه از ۳ قسمت تشکیل شده است : جستجوی کلمات و عبارات ، اضافه کردن صفحه ی جدید به دیکشنری و رده بندی .

اضافه کردن صفحه ی جدید : برای این کار ،کلاسی داریم به نام wiki url ، این کلاس آدرس یک صفحه ویکی پدیا را گرفته و محتوای آن و همه لینک های ویکی پدیای آن را در می آورد ، برای استخراج محتوای صفحه از HtmlParser استفاده کرده ایم و برای استخراج لینک های صفحه از Regular Expression ها استفاده کرده ایم و نیز لینک های محلی (لینک به همان صفحه) را به درستی درنظر نگرفته ایم . سپس در صورت تکراری نبودن صفحه url این صفحه را در فایلURLs سامانه ذخیره می کنیم . بنابراین سامانه از اضافه کردن صفحات تکراری به سامانه و دیکشنری جلوگیری می کند . برای هر صفحه جدید که به سامانه اضافه می شود یک فایل txt. ایجاد میکنیم که شامل آدرس صفحه و خلاصه صفحه است که در هنگام نمایش نتایج از این خلاصه استفاده می کنیم .خلاصه صفحه ها به صورت استاتیک هنگام اضافه شدن صفحه به سامانه ایجاد می شود و در فایل ذخیره می شود ، روند ایجاد خلاصه به این صورت است که در بین محتویات صفحه ، مکان کلمه title صفحه پیدا شده و ۲۰ کلمه بعد و ۲۰ کلمه قبل از آن به عنوان خلاصه استخراج می شود بدین ترتیب چون title صفحات ویکی پدیا شامل همان موضوع اصلی متن است بنابراین با احتمال خوبی تعدد کلمه مورد جستجو در خلاصه تولید شده بالاست .

سپس به tokenize کردن صفحه می پردازیم و توکن های آن را در دیکشنری ذخیره می کنیم ، ما تنها یک فایل دیکشنری روی دیسک ذخیره می کنیم که هنگام run برنامه همه محتویات این فایل در حافظه ی اجرایی load می شود و هنگام خروج برنامه محتویات دوباره از ram به روی فایل dictionary.txt ذخیره می شود.

فرمت دیکشنری ما به این صورت است که در هر خط آن داریم :

Token : document_id : position1 , position2 , position3 , ….

و به این ترتیب positional index کرده ایم که از آن در phrase query ها استفاده می کنیم.

و در حافظه از یک hashmap برای نگه داشتن دیکشنری استفاده می کنیم .

برای tokenize کردن محتوا ، خودمان کلاس tokenizer نوشتیم و delimiter ها را تعریف کردیم .

برای نگه داشتن position ها از ایده ی pooling استفاده کرده ایم زیرا به جای اینکه هر بار بگوییم new Integer(i) که این باعث ایجاد شی جدید و اتلاف وقت حتی برای i های تکراری می شود ، یک مجموعه از شی های Integer را در pool ذخیره کرده ایم و از آن استفاده می کنیم که این باعث صرفه جویی در زمان می شود.

جستجوی کلمات و عبارات : برای جستجوی عبارات باید در ابتدا و انتهای عبارت " قرار دهیم به این ترتیب سامانه می فهمد یک عبارت جستجو شده است ابتدا با استفاده از position های نگه داشته شده در دیکشنری مستندات مربوطه را پیدا می کنیم و رتبه بندی نتایج را هم بر اساس میزان تکرار عبارت در صفحه انجام میدهیم .

در هنگام جستجوی کلمات ، برای رتبه بندی مستندات از فضای برداری استفاده می کنیم که کلمات فضای برداری مان همه کلمات پرسمان هستند بدین ترتیب یک بردار برای پرسمان و یک بردار برای هر صفحه که حاوی کلمات query است در می آوریم و سپس با استفاده از فرمول tf-idf میزان نزدیکی این دو بردار را محاسبه می کنیم و به این ترتیب اولویت می دهیم به نتایج جستجو

رده بندی : ما از رده بندی استفاده کرده ایم و با استفاده از feature selection ،ویژگی های هر کلاس را پیدا کردیم و سپس با استفاده از این ویژگی ها صفحات جدید را کلاس بندی می کردیم . به این ترتیب که ۵ کلاس داریم به نام های : ادبی و هنری – تاریخی – سیاسی – علمی – ورزشی

در ابتدا باید این کلاسهای را learn میکریم و feature ها را در می آوردیم ،سایز learning collection ما ۳۲۰ صفحه ویکی پدیا بود که دستی صفحات مناسب را پیدا کرده و داده بودیم .

برای انتخاب ویژگی های هر دسته از فرمول زیر استفاده کردیم :

$$I(U;C) = \frac{N_{11}}{N}\log_2\frac{NN_{11}}{N_{1.}N_{.1}} + \frac{N_{01}}{N}\log_2\frac{NN_{01}}{N_{0.}N_{.1}} + \frac{N_{10}}{N}\log_2\frac{NN_{10}}{N_{1.}N_{.0}} + \frac{N_{00}}{N}\log_2\frac{NN_{00}}{N_{0.}N_{.0}}$$

این فرمول را برای هر توکن در دیکشنری و هر کلاس محاسبه می کنیم و در نهایت ۶۰ کلمه ی اول که بالاترین نمره را دارند به عنوان ویژگی های آن کلاس انتخاب می شوند .

سپس برای رتبه بندی صفحه ی جدید از فرمول زیر رفتیم :

$$\log p(c) + \Sigma\,(\log[(number\ of\ feature\ j\ occurance\ in\ class\ c))\ /(number\ of\ all\ terms\ in\ c)] * (number\ of\ feature\ j\ occurance\ in\ document)$$

این فرمول را برای هر کلاس C و هر فیچر j از آن کلاس محاسبه می کنیم و در نهایت همه score های بدست آمده از فیچر های یک کلاس را با هم جمع زده و به عنوان نمره ی آن صفحه ی ویکی در آن کلاس منظور می کنیم و کلاسی که نمره اش بالاتر باشد کلاسی است که صفحه ی ویکی به آن متعلق است .