باسمه تعالی

تعریف پروژه درس بازیابی اطلاعات

بهار ۱۴۰۰

مقدمه

هدف از این پروژه ایجاد یک موتور جستجو برای بازیابی اسناد متنی است به گونه‌ای که کاربر پرسمان خود را وارد نموده و سامانه اسناد مرتبط را بازیابی می‌کند. پروژه در سه مرحله تعریف شده است که عبارتند از:

مرحله اول: ایجاد یک مدل بازیابی اطلاعات ساده

مرحله دوم: تکمیل مدل بازیابی اطلاعات و ارائه قابلیت‌های کارکردی پیشرفته‌تر

مرحله سوم(اختیاری): پیاده‌سازی الگوریتم خوشه‌بندی و دسته‌بندی و بازیابی بر اساس خوشه/ دسته

در انجام پروژه به نکات زیر توجه فرمایید:

- پروژه انفرادی است.
- در فاز اول استفاده از کتابخانه‌های آماده برای پردازش متون مجاز نمی‌باشد.
- موارد امتیازی با علامت ستاره ★ مشخص شده است.
- کدهای خود را در کوئرا بارگذاری نمایید.(لینک آن در سایت درس قرار داده می شود).
- کدهای شما (به همراه کدهای دانشجویان ترم‌های گذشته) توسط کوئرا بررسی می شود. درصورت شباهت نمره تمام فازهای پروژه برای دو طرف درگیر صفر خواهد شد.
- مهلت ارسال فاز اول تا پایان روز ۱۰ خرداد، و مهلت ارسال فاز دوم و سوم تا پایان روز ۱۵ تیرماه می‌باشد.
- به ازای هر روز تاخیر در ارسال فاز اول ۲ درصد از نمره‌ی فاز مربوطه کسر می‌شود.
- ارسال فاز دوم و سوم با تاخیر امکان پذیر نخواهد بود.
- تحویل تمامی فازها در تاریخ ۱۸ تیرماه به صورت آنلاین در اسکایپ انجام خواهد شد.

راهنمایی: در صورت نیاز می‌توانید سوالات خود را در خصوص پروژه از تدریسیار درس، از طریق ایمیل زیر بپرسید.

Najmeh.mohammadbagheri77@gmail.com

۱- فاز اول

در این فاز از پروژه به منظور ایجاد یک مدل بازیابی اطلاعات ساده نیاز است تا اسناد شاخص‌گذاری شوند تا در زمان دریافت پرسمان از شاخص معکوس برای بازیابی اسناد مرتبط استفاده شود. به طور خلاصه مراحل انجام این فاز از پروژه به شرح زیر می‌باشد.

۱) استخراج توکن
۲) ساخت شاخص معکوس
۳) پیاده‌سازی ۵ قاعده همسان‌سازی
۴) اعمال یک ایده برای جلوگیری از تغییر داده در بخش همسان سازی
۵) حذف کلمات پرتکرار
۶) پاسخ دهی به پرسمان کاربر

در ادامه به شرح نحوه انجام هریک از مراحل می پردازیم.

۱-۱ شاخص‌گذاری اسناد

برای شاخص‌گذاری اسناد لازم است بخش‌های زیر پیاده‌سازی شوند:

- واکشی اسناد
- استخراج توکن
- همسان‌سازی کلمات
- حذف کلمات پرتکرار

پس از آنکه محتوای تمامی اسناد را به صورت توکن استخراج کردید، توکن‌ها را به صورت یک شاخص معکوس ذخیره کنید. توجه داشته باشید که این شاخص‌گذاری نباید در زمان دریافت پرسمان کاربر انجام شود بلکه باید شاخص از قبل ذخیره شده باشد و در زمان پاسخ گویی به کاربر تنها از آن استفاده کنید. برای آنکه بتوانید درستی عملکرد خود را نشان دهید یک تابع آزمون برای این بخش تعریف نمایید تا با دریافت یک کلمه، یک لیست از شماره اسنادی (خبرهایی) که شامل این کلمه بوده‌اند را به صورت مرتب نمایش دهد. دقت داشته باشید که در شاخص معکوس هم کلمات و هم شماره اسناد باید به صورت مرتب شده باشند.

همان طور که می‌دانید همسان‌سازی کلمات و حذف کلمات پرتکرار برای بهبود عملکرد موتور جستجو الزامی است. برای حذف کلمات پرتکرار می‌توانید لیستی از پرتکرارترین کلمات را استخراج کنید و از آن لیست استفاده نمایید و یا در زمان ساخت شاخص معکوس مکانی پرتکرارترین کلمات را بیابید. برای بخش همسان‌سازی قواعدی را در نظر بگیرید و بر روی توکن‌ها اعمال نمایید. لازم به ذکر است که وجود حداقل ۵ نوع قاعده‌ی همسان‌سازی الزامی است. توجه نمایید حذف "تر" ،"ترین" ، "ها" ، "ات" همگی از نوع قاعده‌ی همسان‌سازی "حذف پسوند" هستند و هر کدام به تنهایی یک قاعده در نظر گرفته نمی‌شود. انواع مختلف همسان‌سازی شامل "حذف پیشوندها" ، "یکسان‌سازی حروف" ، "ریشه‌یابی افعال"، "ریشه‌یابی جمع مکسر"، "یکسان‌سازی کلمات چند جزئی" و موارد مشابه دیگر می‌باشد.

برای ریشه‌یابی می‌توانید باتوجه به مجموعه‌ی اسناد، یک لیست از ریشه‌ی کلمات تهیه کنید و از آن استفاده نمایید. واضح است هرچقدر لیست تهیه‌شده جامع‌تر باشد عملکرد سیستم بازیابی اسناد بهتر خواهد بود. در این پروژه استفاده از حداقل ۲۰ کلمه برای لیست‌ها ضروری است.

این مطلب را نیز در نظر داشته‌باشید که گاهی اوقات همسان‌سازی ممکن است اطلاعاتی را از بین ببرد. به طور مثال اگر دو قاعده حذف "می" از ابتدای فعل‌ها و حذف شناسه مربوط به فعل‌ها را داشته باشیم دو کلمه زیر معادل می شوند.

میدانم ← دان ، میدان ← دان

درنتیجه لازم است حداقل **یک ایده** برای جلوگیری از چنین اشتباهاتی در بخش همسان‌سازی خود به کار ببرید.

★ توجه نمایید پیاده‌سازی هر ایده‌ی بیشتر برای جلوگیری از چنین اشتباهاتی، در صورت منطقی بودن و عملکرد صحیح، دارای نمره‌ی امتیازی خواهد بود.

۱-۲ پاسخ‌دهی به پرسمان کاربر

در این بخش با دریافت پرسمان کاربر باید بتوانید اسناد مرتبط با آن را به صورت دودویی بازیابی نمایید. پرسمان کاربر به دو صورت زیر می تواند باشد:

تک کلمه: تنها کافی است که لیست مربوط به آن را از روی دیکشنری بازیابی نمایید.

چند کلمه: در این بخش لیست فایل‌ها باید بر اساس میزان ارتباط مرتب شده باشد. مرتبط‌ترین سند، سندی است که تمام کلمات را داشته باشد.

۱-۳ مجموعه داده

مجموعه داده مورد استفاده در این پروژه مجموعه‌ای از خبرهای واکشی‌شده از چند وب‌سایت خبری فارسی است که در قالب یک فایل اکسل در اختیار شما قرار خواهدگرفت. لازم است تنها ستون "content" را بعنوان محتوای سند پردازش کنید. شماره‌ی سطر هر خبر را به عنوان id آن سند (خبر) در نظر بگیرید و در زمان پاسخ به پرسمان ورودی id و "url" مربوط به آن خبر را نمایش دهید، تا امکان بررسی صحت عملکرد سیستم وجود داشته‌باشد.

۲- فاز دوم

در این مرحله مدل بازیابی اطلاعات باید بتواند نتایج جستجو را بر اساس ارتباط آنها با پرسمان کاربر رتبه‌بندی کند. مدل بازیابی اطلاعات این کار را با مدل‌سازی اسناد در فضای برداری انجام می‌دهد. به این صورت که برای هر سند یک بردار عددی استخراج می‌شود که بازنمایی آن سند در فضای برداری است. سپس با داشتن یک پرسمان از کاربر ابتدا آن را به فضای برداری برده و سپس با استفاده از یک معیار شباهت مناسب، فاصله‌ی بردار عددی پرسمان را با تمام اسناد در فضای برداری محاسبه کرده و در نهایت نتایج خروجی بر اساس میزان شباهت مرتب‌سازی شوند. همچنین برای افزایش سرعت پاسخگویی مدل بازیابی اطلاعات روش‌های مختلفی به کار گرفته خواهد شد. جزئیات هر بخش به تفصیل در ادامه بیان شده است. توجه نمایید در این فاز می‌توانید بجای استفاده از قوانین همسان‌سازی پیاده‌سازی شده در فاز قبل، از کتابخانه‌ی آماده برای ریشه‌یابی و نرمال‌سازی متن خبر استفاده کنید. زیرا همانطور که می‌دانید هرچقدر عملیات همسان‌سازی (نرمال‌سازی + ریشه‌یابی) دقیق‌تر و جامع‌تر باشد عملکرد مدل بازیابی اطلاعات بهتر خواهد بود.

۲-۱ مدل‌سازی اسناد در فضای برداری

در مرحله قبل پس از استخراج توکن ها اطلاعات به صورت یک دیکشنری ذخیره شدند. در این بخش هدف بر آن است که اسناد در فضای برداری بازنمایی شوند. با استفاده از روش وزن‌دهی $tf-idf$ بردار عددی برای هر سند محاسبه خواهد شد و درنهایت هر سند به صورت یک بردار شامل وزن‌های تمام کلمات آن سند بازنمایی می‌شود. محاسبه‌ی وزن هر کلمه t در یک سند dبا داشتن مجموعه‌ی تمام اسناد D با استفاده از معادله‌ی زیر محاسبه می‌شود:

$$tfidf(t,d,D) = tf(t,d) \times idf(t,D) = \left(1 + \log\left(f_{t,d}\right)\right) \times \log\left(\frac{N}{n_t}\right)$$

که در آن $f_{t,d}$ تعداد تکرار کلمه‌ی t در سند d و n_t تعداد سندهایی است که کلمه‌ی t در آنها ظاهر شده است. توضیحات بیشتر این روش در فصل ۶ کتاب آمده است.

برای آنکه از به کار بردن فضای بیش از حد جلوگیری شود در بازنمایی اسناد به فضای برداری از تکنیک $Index\ elimination$ استفاده نمایید. در واقع به جای آن که برای هر سند یک بردار عددی در نظر بگیرید که بسیاری از عناصر آن صفر هستند می‌توانید وزن کلمات در اسناد مختلف را در همان لیست‌های پست‌ها ذخیره کنید. در زمان پاسخ‌گویی به پرسمان کاربر که در ادامه توضیح داده می‌شود نیز همزمان با جستجوی کلمات در لیست‌های پست‌ها می‌توانید وزن کلمات در اسناد مختلف را نیز واکشی کنید و به این شکل تنها عناصر غیر صفر بردارهای اسناد ذخیره و پردازش می‌شوند.

۲-۲ پاسخ‌دهی به پرسمان در فضای برداری

با داشتن پرسمان کاربر، بردار مخصوص پرسمان را استخراج کنید. سپس با استفاده از معیار شباهت سعی کنید اسنادی را که بیشترین شباهت (کمترین فاصله) را به پرسمان ورودی دارند پیدا کنید. سپس آنها را به ترتیب شباهت نمایش دهید. معیارهای فاصله‌ی مختلف می‌تواند برای این کار در نظر گرفته شود که ساده‌ترین آنها شباهت کسینوسی بین بردارها است که زاویه‌ی بین دو بردار را محاسبه می‌کند. این معیار به صورت زیر تعریف می‌شود:

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

در انتهای کار برای نمایش یک صفحه از نتایج پرسمان فقط کافیست K سندی انتخاب شوند که بیشترین شباهت را به پرسمان دارند. ساده‌ترین راه‌حل برای این کار مرتب‌سازی تمام اسناد براساس شباهتشان با پرسمان است که هزینه زمانی این کار از مرتبه‌ی $O(nlogn)$ است که باتوجه به زیاد بودن تعداد اسناد می‌تواند باعث زیاد شدن شدید زمان پاسخ موتور جستجو شود. برای حل این مسئله از پشته $(heap)$ استفاده کنید و برای نمایش هر صفحه تنها K سند با بیشترین شباهت را از آن بیرون بکشید. توجه کنید که ساختن پشته از مرتبه‌ی زمانی $O(2n)$ و استخراج K سند با بیشترین مقدار از مرتبه‌ی $O(logn)$ است و در مجموع این تکنیک می‌تواند حدوداً مشکل زیاد بودن زمان پاسخ را حل کند. توجه کنید که اسناد با امتیاز صفر نیازی نیست در پشته ریخته شوند. شناسایی این اسناد و حذف آنها را با استفاده از تکنیک $Index\ elimination$ انجام دهید.

۲-۳ افزایش سرعت پردازش پرسمان

با استفاده از تکنیک $Index\ elimination$ تا حدودی مشکل زیاد بودن زمان در مرحله قبل حل می‌شود اما همچنان زمان پاسخگویی برای بسیاری از کاربردها قابل قبول نمی‌باشد. برای آنکه سرعت پردازش و پاسخگویی افزایش یابد روش‌های مختلفی وجود دارند که یکی از آن روش ها استفاده از $Champion\ lists$ می‌باشد که قبل از آنکه پرسمانی مطرح شود و در مرحله پردازش اسناد، یک لیست از مرتبط‌ترین اسناد مربوط به هر $term$ در لیست جداگانه‌ای نگهداری می‌شود. برای پیاده‌سازی این بخش پس از ساخت شاخص معکوس مکانی، $Champion\ list$ را ایجاد کنید و تنها بردار پرسمان را با بردار اسنادی که از طریق جستجو در $Champion\ list$ به دست آورده‌اید مقایسه کنید و K سند مرتبط را به نمایش بگذارید. توضیحات بیشتر این روش در فصل ۷ کتاب آمده است.

توجه۱: می‌توانید وزن دهی $tf-idf$ و ایجاد لیست $Champion$ را با استفاده از شاخص معکوس که در مرحله گذشته پیاده‌سازی کرده‌اید، انجام دهید.

توجه۲: استفاده از پشته برای مرتب‌سازی اسناد، بکاربردن تکنیک$Index\ elimination$ و استفاده از لیست‌های $Champion$ باید به‌گونه‌ای پیاده‌سازی شوند که در زمان تحویل بتوان عملکرد مدل را در دو حالت فعال و غیرفعال بودن هر مولفه بررسی کرد.

۳- فاز سوم (★)

در این بخش از پروژه مقیاس موتور جستجویی که در دو مرحله‌ی گذشته طراحی و پیاده‌سازی شده، بزرگ‌تر می‌شود. اسناد ورودی این بخش باید در موتور جستجو شاخص‌گذاری شده و مورد پرسمان قرار بگیرند. با افزایش حجم اسناد ورودی مقایسه پرسمان با تمام اسناد به صورت کارا و در زمان مناسب امکان‌پذیر نیست. در این فاز برای حل این مسئله می‌خواهیم از خوشه‌بندی استفاده و بردار ویژگی پرسمان را به جای مقایسه با تمام اسناد فقط با اسناد یک (یا چند) خوشه مقایسه کنیم. علاوه بر خوشه‌بندی، دسته‌بندی اخبار نیز در این مرحله از پروژه بایستی پیاده‌سازی شود. به این معنا که هر خبر به یکی ازدسته‌های ورزشی، اقتصادی، سیاسی، سلامت و فرهنگی نگاشت شود تا در هنگام جستجو بتوان مشخص کرد نتایج از کدام دسته‌های خبری باشند. در ادامه به توضیح بیشتر در این خصوص می‌پردازیم.

۳-۱ خوشه‌بندی

در این مرحله می‌خواهیم با استفاده از الگوریتم K-means خوشه‌بندی اسناد را انجام دهید. به منظور بهبود عملکرد الگوریتم خوشه‌بندی می‌توانید چندین بار آن را اجرا و سپس بر مبنای معیار RSS بهترین خوشه‌بندی را انتخاب کنید. بعد از انتخاب یک خوشه‌بندی مناسب، در زمان پاسخ‌گویی به یک پرسمان، ابتدا بردار ویژگی آن را همانند قبل استخراج کنید. سپس شباهت کسینوسی آن را با تمام مراکز خوشه‌ها محاسبه کرده و خوشه با بیشترین شباهت را انتخاب کنید. سپس شباهت کسینوسی بردار پرسمان با تمام سندهای آن خوشه را نیز محاسبه کرده و از میان آنها شبیه‌ترین سندها به پرسمان را انتخاب کنید و به عنوان نتیجه جستجو برگردانید. تمام تکنیک‌هایی که در مراحل قبل پروژه پیاده‌سازی کرده‌اید (همانند index elimination و ...) در این مرحله نیز قابل استفاده هستند.

توجه کنید لزومی بر اینکه فقط یک خوشه را برای جستجو انتخاب کنید وجود ندارد. به این معنی که بعد از محاسبه‌ی شباهت بردار پرسمان با مراکز خوشه‌ها، می‌توانید b مرکز خوشه با بیشترین شباهت را انتخاب کرده و جستجو را در تمام اسناد خوشه‌های مربوط به آنها انجام دهید. این کار خصوصا زمانی موثر است که تعداد خوشه‌ها زیاد باشد و در نتیجه احتمالا تعداد اسناد در یک خوشه کم شده باشد. انتخاب مقدار b و تعداد خوشه‌ها با هم مرتبط هستند و بهترین مقادیر آنها مقادیری است که یک تعادل بین سرعت پاسخ‌گویی و کیفیت نتایج ایجاد کند. ارزیابی دقیق این موضوع مستلزم اندازه‌گیری دقیق زمان پاسخ به پرسمان‌های کاربر و دقت نتایج بازگردانده شده بر روی مجموعه‌ای از پرسمان‌های از قبل آماده شده است. در این پروژه می‌توانید این کار را به صورت شهودی انجام دهید و تنظیم دقیق مقدار b لازم نیست.

توجه: در این قسمت مجاز به استفاده از کتابخانه برای خوشه‌بندی نیستید.

۳-۲ دسته‌بندی

موتور جستجوی طراحی شده در این حالت می‌بایست قابلیت تعیین دسته خبر را در زمان وارد کردن پرسمان به کاربر بدهد. این قابلیت با استفاده از کلمه کلیدی cat ارائه می‌گردد. به عنوان مثال زمانی که کاربر عبارت «استقلال cat:sport» را وارد می‌کند می‌بایست بازیابی در بین اخبار دسته‌ی ورزشی و زمانی که عبارت «استقلال cat:economy» را وارد می‌کند می‌بایست بازیابی در بین اخبار دسته‌ی اقتصادی انجام شود. برای این منظور با استفاده از روش‌های دسته‌بندی اسناد متنی ارائه‌شده در درس، دسته هر خبر را تعیین و ذخیره کنید تا در زمان جستجو بتوان از آن استفاده کرد. دسته‌های خبری مد نظر عبارتند از: ورزشی، اقتصادی، سیاسی، فرهنگی، سلامت.

برای دسته‌بندی اسناد از الگوریتم k نزدیکترین همسایه با مقادیر مختلف k استفاده کنید. در ابتدا باید الگوریتم دسته‌بند را پیاده‌سازی کنید و سپس با استفاده از مجموعه اسنادی که برچسب دارند، اسنادی که برچسب ندارند را برچسب بزنید. سعی کنید یک مقدار مناسب برای k پیدا کنید. برای پیدا کردن k مناسب و ارزیابی عملکرد دسته‌بند خود می‌توانید از روش ارزیابی 10-Fold-Cross-Validation استفاده کنید.

توجه: در این قسمت مجاز به استفاده از کتابخانه برای دسته‌بندی نیستید اما می‌توانید برای ارزیابی 10-Fold-Cross-Validation از کتابخانه استفاده کنید.

۳-۳ ارزیابی موتور جستجو

با استفاده از خوشه‌بندی و با انتخاب مقدار مناسب برای تعداد خوشه‌ها، سرعت پاسخ‌گویی به پرسمان بهبود می‌یابد. در مقابل ممکن است کیفیت نتایج جستجو تحت تاثیر خوشه‌بندی قرار گیرد. برای بررسی این موضوع ۱۰ پرسمان که انتظار دارید نتایج قابل پیش‌بینی داشته باشند را انتخاب کنید. نتایج این ۱۰ پرسمان را در حالت‌های بدون خوشه‌بندی و با خوشه‌بندی از نظر کیفیت نتایج و سرعت پاسخ‌گویی به پرسمان مورد مقایسه قرار دهید. ارزیابی کیفی موتور جستجو براساس ارتباط شهودی پرسمان با اسناد نتیجه انجام می‌شود و تعریف معیار عددی برای این کار نیاز نیست. فقط کافیست به صورت شهودی نتایج را در حالتهای مختلف با هم مقایسه کنید. سعی کنید پرسمان‌هایی که انتخاب می‌کنید هدفمند باشد به طوری که نتایج قابل پیش‌بینی داشته باشند.

موفق باشید!