بسمه تعالی

**بازیابی پیشرفته اطلاعات**

**نیم‌سال اول ۹۷-۹۸**

**مدرس: دکتر حمید بیگی**

**فاز سوم پروژه** **ددلاین: ۱۴ دی ۱۳۹۷**

در فاز سوم پروژه درس، از شما یک سیستم بازیابی اطلاعات بر روی صفحات دیجی کالا خواسته‌شده‌است. برای این سیستم، باید با خزش[1] صفحات دیجی کالا، اطلاعات خواسته شده برای هر صفحه را ذخیره کنید. سپس نمایه صفحات به دست‌آمده را با استفاده از Elasticsearch بسازید و جستجو روی آن بر اساس اطلاعات ذخیره‌شده، پیاده سازی کنید. در نهایت نیز با آدرس صفحات Page Rank را محاسبه کنید.

در این پروژه‌ی رابطه‌ی بین صفحات بر اساس «محصولات مرتبط» در صفحه‌ی هر محصول برقرار می‌شود.

همانطور که بیان شد این فاز از ۴ بخش زیر تشکیل شده که در ادامه به طور مفصل توضیح داده‌شده‌است.

۱- پیاده‌سازی خزنده و به دست آوردن اطلاعات مشخص شده از سایت دیجی‌کالا

۲- نمایه‌سازی اطلاعات با استفاده از Elasticsearch

۳- محاسبه Page Rank

۴- جستجو بر روی صفحات با استفاده از نمایه ساخته‌شده

## بخش اول: پیاده‌سازی خزنده و به دست‌آوردن اطلاعات

برای سایت هدف، ما قسمت محصولات کالای دیجیتال سایت دیجی کالا را انتخاب کرده‌ایم. شما باید با استفاده از خزشگرها یا پارسرهایی مثل lxml ،BeautifulSoup، Scrapy و یا Jsoup[2] صفحات را با پیوند برقرار شده بین آن‌ها در بخش «محصولات مرتبط» خزش کنید و با استفاده از parser، هر صفحه را پردازش کنید و اطلاعات خواسته شده در ادامه را برای هرصفحه‌ی به فرمت خواسته شده ذخیره کنید. برای شروع از این صفحه شروع کنید.

اطلاعاتی که از صفحه باید جمع آوری شوند عبارت اند از:

۱- title: نام محصول

۲- url: آدرس الکترونیکی صفحه

۳- category: دسته‌بندی محصول

۴- expert_summery: نقد و بررسی محصول

[1] crawl

[2] برای استفاده از خزشگرهای دیگر، باید با دستیاران آموزشی هماهنگ شود.

۵- expert_rating: در ادامه‌ی نقد و بررسی محصول، رتبه دهی به محصول از چندین لحاظ مثل «ارزش خرید به نسبت واقعی» یا «طراحی و ظاهر» و یا ... آمده است که در یک لیستی از دیکشنری‌ها با کلید criteria، مقدار عنوان محصول (مثل همین دو مورد نام برده) و با کلید rating، مقدار رتبه کیفی محصول که خوب،معمولی و ... غیره است را ذخیره می‌کنیم.( به هر تعدادی که از این معیارها برای محصول آمده‌است. )

۶-related_product: لیستی از آدرس صفحه محصولات مرتبط. برای هر محصول تنها ۵ آدرس نگه‌دارید.

نمونه فایل json از اطلاعات ذکر شده در فایل product.json آمده‌است. شما نیز همانند این فایل، اطلاعات را استخراج کرده و در فایل‌هایی نگه دارید. برای انتخاب صفحه بعدی نیز از آدرس محصولات مرتبط که برای کل صفحات نگه داشته شده، یکی را انتخاب می‌کنیم.

نکات:

۱- می‌توانید برای انتخابگر عنصر[3] از xpath یا css استفاده کنید.
۲- هر صفحه بیشتر از یکبار پردازش نشود.
۳- باید برای هر قسمت، تنها متن داخل tag ها یا attribute ها استخراج شده باشد.
۴- خزنده تا جای ممکن بدون خطا تمام صفحات خواسته‌شده را پردازش کند.

آن‌چه در این قسمت خواسته شده:

با گرفتن n (تعداد کل صفحاتی که باید پردازش شوند) مجموعه فایل‌های json در فرمت مثال داده‌شده ذخیره شوند. (n حداکثر ۵۰۰۰ است.)

[3] Element selector

## بخش دوم: شاخص گذاری با استفاده از ElasticSearch

حال باید با استفاده از ElasticSearch که ابزاری سطح بالا برای بازیابی اطلاعات است، صفحات دیجی کالا را شاخص گذاری کنیم. برای آشنایی با ElasticSearch به این صفحه مراجعه کنید و راهنمایی راجع به چگونگی استفاده نیز در صفحه آمده‌است. می توانید از واسط هایی که در زبان‌های مختلف برای ارتباط با Elasticsearch ایجاد شده اند، استفاده نمایید.

آن‌چه در این قسمت خواسته شده:

تابعی برای حذف اطلاعات درون ElasticSearch
و همچنین تابعی که آدرس سرور ElasticSearch (مثل localhost:9200) و پوشه‌ی حاوی فایل‌های json بخش قبل را دریافت کند و آن‌ها را وارد ElasticSearch کرده و شاخص گذاری کند.

## بخش سوم: محاسبه Page Rank صفحات

حال باید با استفاده از اطلاعات ذخیره شده در ElasticSearch، برای هر صفحه‌ی خزش شده page rank را محاسبه کنید. همانطور که در بخش‌های قبلی گفته شده است، برای هر صفحه لیستی از صفحاتی که از آن صفحه به آن‌ها لینک وجود دارد، نگهداری می‌شود. بنابراین می‌توانید با استفاده از این اطلاعات page rank را محاسبه نمایید و مقدار آن را به اطلاعات ذخیره شده برای هر صفحه در ElasticSearch اضافه نمایید.

آن‌چه در این قسمت خواسته شده:

با دریافت آدرس ElasticSearch و مقدار $\alpha$ لازم برای محاسبه‌ی page rank، مقدار آن را برای هر صفحه محاسبه نماید و به اطلاعات آن صفحه در ElasticSearch اضافه نماید.

## بخش چهارم: جستجو بر روی صفحات با استفاده از نمایه ساخته‌شده

در این قسمت باید امکان جستجو دودویی در شاخص‌گذاری ساخته شده در بخش قبل، فراهم شود. نوع جستجوی خواسته شده در ورودی داده می‌شود. پرسمان ورودی به ترتیب شامل بخش‌های زیر است:

- شروطی که حتما باید در جواب برقرار باشند
- شروطی که نباید در جواب برقرار باشند
- شروطی که اگر در جواب باشند رتبه‌ی آن را بالا می‌برند

همچنین باید این امکان وجود داشته باشد که بتوان به هر شرط وزن داد. بنابراین شروط ورودی بخش‌ها مختلف در خطوط متمایز به صورت (key, value, weight) داده می‌شوند.

آن‌چه در این قسمت خواسته شده:

تابعی که پرسمان را در 3 خط به عنوان ورودی دریافت می‌کند که هر خط برای یک بخش با شروطی به شکل (key, value, weight) است. این تابع به عنوان خروجی json ای از لیست جواب‌های استخراج شده که پرسمان در آن‌ها صدق می‌کند، برمی‌گرداند.

نکات

1. باید یک واسط کاربری برای تست موارد مختلف مشخص شده، برای تحویل حضوری وجود داشته باشد. واسط کاربری می‌تواند تحت کنسول پیاده سازی شود.
2. زودتر شروع به پیاده‌سازی این فاز از پروژه نمایید. احتمال تمدید این فاز وجود ندارد.
3. برای ارسال پروژه، کد خود را به صورت zip شده در سایت کوئرا بارگذاری نمایید.
4. می‌توانید سوالات و اشکالات خود را در سایت پیاتزا زیر پست مربوط به این تمرین بپرسید.
5. تمام قسمت‌های پروژه را خودتان به تنهایی پیاده‌سازی کنید. در صورت مشاهده تقلب، طبق قوانین دانشکده با شما برخورد خواهد شد.

موفق باشید :)