****

**گزارش فنی پروژه‌ی پایانی بازیابی اطلاعات**

**با موضوع: پیاده‌سازی یک موتور جستجوی ساده با داده‌های وبسایت خبری " خبر آنلاین "**

**مجتبی نبوی**

**شماره دانشجویی: 9753088006**

**استاد راهنما: آقای حسین امیرخانی**

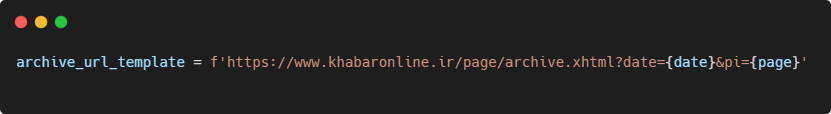
**خرداد 1401**

**چکیده**

در این پروژه یک موتور جستجو‌ی ساده با داده‌های وب سایت خبری "خبر آنلاین" و با زبان پایتون پیاده‌سازی گردید. با توجه به این موضوع که داده‌ها بطور آماده در دسترس نبودند، در ابتدا یک خزنده نوشته شد که داده‌های لازم برای ادامه‌ی فرایند را جمع‌آوری کند. داده‌های جمع‌آوری شده، اخبار 4 ماه ابتدای سال 1400 با تعداد 25411 خبر و دارای 9 ویژگی و با حجمی حدود 100 مگابایت می‌باشد. پس از برخی پردازش‌ها روی این داده‌ها، یک پرس و جو به برنامه داده می‌شود و براساس آن تعدادی اخبار مرتبط بعنوان پاسخ نمایش داده می‌شوند.

**خزش و جمع‌آوری داده‌ها**

برای جمع‌آوری داده‌ها، ابتدا صفحه‌ی آرشیو خبر‌ها را پیدا کرده و در آن بدنبال الگویی برای خزش بودیم. یکی از الگوهای بسیار خوب که از آن در این پروژه استفاده گردید در تصویر 0 در ادامه آمده است.



تصویر 0 - الگوی بدست آمده برای شروع خزش و جمع آوری آدرس خبرها

در این الگو می‌توان با دادن یک تاریخ شمسی با کمک کتابخانه‌ی jdatetime (برای تبدیل تاریخ میلادی به شمسی)، خلاصه‌ی اخبار آن روز را مشاهده و در صفحات آن حرکت کرد. در جمع‌آوری داده‌های این پروژه تمام تاریخ‌های بین 01/01/1400 تا 31/04/1400 استفاده شد و در نهایت، آدرس تمام اخبار آن هر روز با حرکت در تمام صفحات و با کمک کتابخانه‌های requests و beautifulSoup بدست آمد.

در ادامه، به تمام آدرس‌های جمع‌آوری‌شده در مرحله‌ی قبل مراجعه می‌کنیم و داده‌های مورد نیاز خود را از بخش‌های مختلف هر سند، مانند متن و ابرداده‌ها (metadata) با استفاده از کتابخانه‌ی newspaper استخراج و ذخیره می‌کنیم.

با توجه به امکان بروز مشکلات در فرایند جمع‌آوری وذخیره‌ی داده‌ها مانند قطعی برق، اینترنت و پرشدن تمام فضای حافظه‌ی اصلی (ram)، ذخیره‌سازی داده‌ها، پس از جمع‌آوری کامل داده‌های یک روز انجام و پس از آن حافظه‌ی میانجی (buffer) برای داده‌های سایر روز‌ها خالی می‌شود. کد‌های خزنده (crawler) در تصویر 1 آمده است.



تصویر 1 - کد‌ها و توابع خزنده

**نحوه‌ی استفاده از توابع و کدهای خزنده**

همان‌طور که در تصویر 1 مشاهده کردید، برای اعمال مختلف، توابعی در نظر گرفته شده است. این توابع، بطور عادی کاری انجام نمی‌دهند و باید براساس نیاز خود آن‌ها را فراخوانی کنیم. برای شروع عمل خزش و ذخیره‌ی داده‌ها باید بصورت زیر عمل می‌کنیم:



تصویر 2 - نحوه‌ی استفاده از خزنده

**ویژگی داده‌های جمع آوری شده**

با استفاده از خزنده‌ی نوشته شده، تعداد 25411 خبر دارای 9 ویژگی و با حجمی حدود 100 مگابایت، از تاریخ 01/01/1400 تا 31/04/1400 جمع‌آوری شد. در تصویر 3، می‌توان 4 خبر اول را در قالب یک جدول مشاهده کرد که به درک بهتر از داده‌ها کمک می‌کند. هر خبر شامل داده‌های زیر است که با کمک کتابخانه‌ی newspaper استخراج شده‌اند.

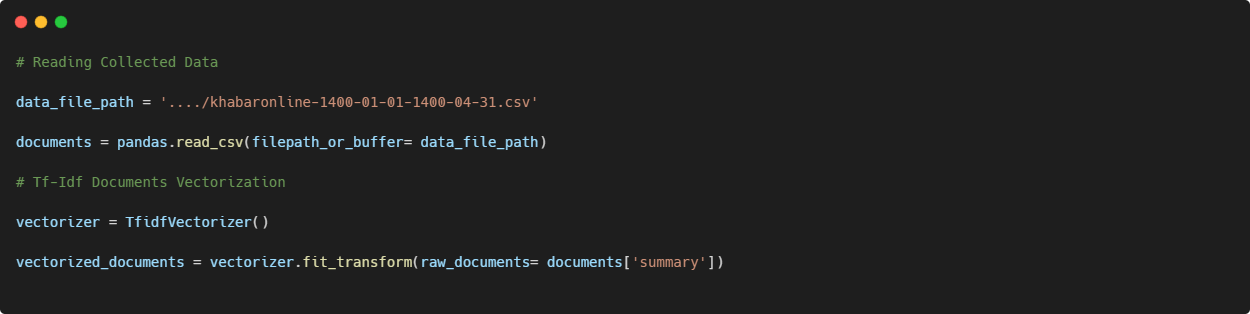
1. آدرس خبر
2. آدرس تصویر خبر
3. شناسه‌
4. عنوان
5. خلاصه
6. متن اصلی
7. تاریخ انتشار
8. تگ‌ها
9. کلمات کلیدی



تصویر 3 - چهار خبر اول از داده‌های جمع‌آوری شده

**تبدیل اسناد خام به ماتریسی از ویژگی‌های TF-IDF و تشکیل واژگان**

در این مرحله با استفاده از داده‌های بدست آمده، واژگان خود را ایجاد کرده و تعداد تکرار هر کلمه در اسناد را نیز بدست می‌آوریم؛ سپس برای هر یک از اسناد، براساس معیار TF-IDF یک عدد را به آن اختصاص می‌دهیم. از این عدد برای برسی میزان شباهت پرس و جوی کاربر و اسناد، توسط معیار Cosine که در یک فضای برداری، زاویه‌ی کسینوسی میان هر یک از اسناد و پرس و جوی کاربر را حساب می‌کند، استفاده می‌شود. برای ایجاد واژگان و حساب‌کردن TF-IDF برای هر سند، از کتابخانه‌ی معروف SciKit-Learn و کلاس TfidfVectorizer استفاده می‌کنیم.



تصویر 4 - خواندن داده‌ها، ایجاد واژگان و برداری‌کردن اسناد

**ویژگی واژگان ایجاد شده بر اساس اسناد**

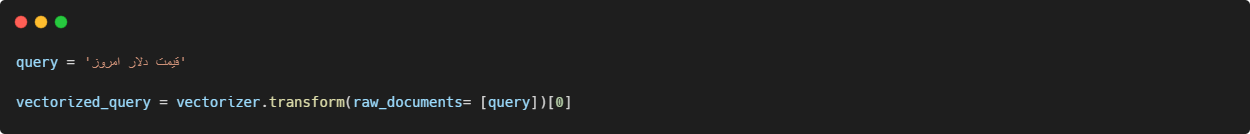
واژگان ایجاد شده در مرحله‌ی قبل دارای 24565 عبارت می‌باشد. در تصویر 5 می‌توان 8 عبارت اول واژگان را مشاهده کرد.



تصویر 5 - پنج عبارت اول در واژگان موتور جستجو

**تبدیل پرس و جوی کاربر به ماتریسی از ویژگی‌های TF-IDF**

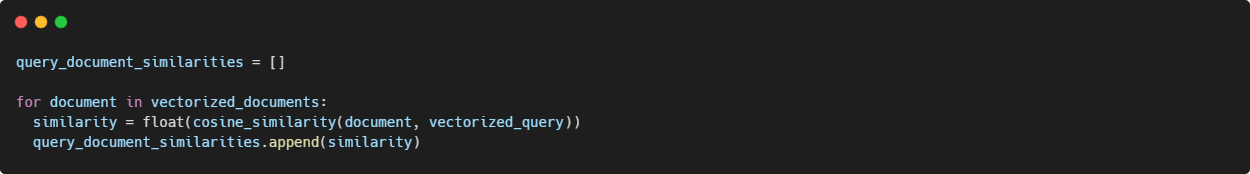
در این مرحله نیز مانند مرحله‌ی قبل و براساس واژگان بدست آمده، برای هر یک از عبارات (term) پرس و جو TF-IDF را حساب کرده و از آن در مرحله‌ی بعد، برای برسی شباهت پرس و جو و اسناد استفاده می‌کنیم.



تصویر 6 - تبدیل پرس و جوی کاربر به بردار با استفاده از واژگان بدست آمده

**محاسبه‌ی میزان شباهت هر سند و پرس و جو**

در این مرحله با استفاده از کلاس cosine\_similarity که در کتابخانه‌ی SciKit-Learn قرار دارد؛ برسی می‌کنیم که هر سند چقدر با پرس و جوی کاربر شباهت دارد. این شباهت در یک فضای برداری و با استفاده از اختلاف زاویه‌ی پرس و جوی کاربر با هر یک از اسناد بدست می‌آید که عددی بین 0 تا 1 داده خواهد بود. هر چقدر این امتیاز (عدد) بیشتر باشد، میزان شباهت نیز بیشتر خواهد شد.



تصویر 7 - محاسبه‌ی میزان شباهت هر سند و پرس و جو

**انتخاب نتایج برتر و نمایش به کاربر**

پس از تمام مراحل قبل، از جمع‌آوری داده تا برسی میزان شباهت پرس و جوی کاربر با اسناد، نوبت آن رسیده است که اسناد با شباهت بیشتر که احتمالا نتایج مورد نظر کاربر نیز هستند را به او نشان دهیم. در مرحله‌ی قبل، ما میزان شباهت هر یک از اسناد را بدست آوردیم و یک امتیاز به آن‌ها اختصاص دادیم؛ اکنون باید اسناد با بیشترین امتیازها را بعنوان پاسخ به کاربر نمایش دهیم. برای این منظور مطابق تصویر 8 عمل می‌کنیم.



تصویر 8 - انتخاب اسناد با امتیاز برتر و نمایش آن به کاربر

همانطور که در خط آخر تصویر 8 دیده‌ می‌شود، پاسخ شامل: میزان شباهت، عنوان و آدرس سند است. تعداد اطلاعات هر سند و نوع نمایش را می‌توان به راحتی تغییر و آن را بهبود داد.

**برخی از نتایج موتور جستجو**

در مثال زیر، پرس و جوی کاربر "قیمت امروز دلار" بوده و 20 نتیجه‌ی زیر بدست آمده‌اند. در میان این 20 سند مرتبط با پرس و جوی کاربر، بیشترین شباهت 0.708 یعنی حدود 70 درصد و کمترین آن 0.427 یا به عبارتی حدود 42 درصد می‌باشد.



تصویر 9 – برخی از نتایج موتور جستجوی نوشته شده با پرس و جوی "قیمت امروز دلار"

**بهینه‌سازی و افزایش سرعت**

برای پیاده‌سازی موتور جستجوی معرفی شده برای وب سایت خبر آنلاین، از ساده‌ترین روش‌ها استفاده شده است و در بسیاری از موارد می‌توان آن را از ابعاد مختلف بهینه کرد. بطور مثال یکی از بهینه‌سازی ها می‌تواند در هنگام برسی میزان شباهت اسناد و پرس و جوی کاربر صورت بگیرد؛ بگونه‌ای که این برسی و اختصاص امتیاز شباهت برای تمام اسناد محاسبه نشود؛ زیرا قسمت بزرگی از اسناد ما به پرس و جوی کاربر ارتباطی ندارند و دخالت آن‌ها در فرایند امتیازدهی بی‌معنی است.

**کد‌ها و داده‌های استخراج شده**

تمام کدهای نوشته شده و داده‌های استخراج شده، برای برسی بیشتر در گیت‌هاب قرار داده شده‌اند که لینک آن در ادامه آمده است.

<https://github.com/mojtabanabavi/Information-Retrieval-Project>