Search System
آرش محرابی
زهرا کولایی زاده

دیتابیس:
برای این پروژه از دیتابیس MySQL استفاده شده است.
جدول news:

Column	Type
id	int
title_tokens	json
body_tokens	json
title_text	varchar(255)
description	varchar(255)
url	varchar(255)
date	varchar(255)

title_tokens و body_tokens توکن های پیش پردازش شده هستند که به صورت json ذخیره می شوند.
بقیه ستون‌ها بدون پیش پردازش ذخیره می‌شوند. (برای نمایش هنگام retrieve کردن result)

جدول tf_idf:

Column	Type
id	int
news_id	int
token	varchar(100)
tf_idf	varchar(45)

news_id در حقیقت یک foreign key به id در جدول news است.
در این جدول tf_idf را برای هر زوج (news_id, token) حساب و ذخیره می کنیم. پس باید زوج نامبرده UNIQUE باشد.
tf_idf یک عدد double است ولی ترجیح دادم به صورت varchar ذخیره کنم.

کد:
هنگام کد زنی سعی شده از اصول کد نویسی تمیز عمو باب استفاده شود بنابراین نام variable ها و تابع ها تا حد امکان گویا هستند و تابع ها معمولاً کمتر از 5 6 خط هستند و هر کدام یک کار انجام می دهند.

Crawler.py:
این فایل به صفحه news.urmia.ac.ir می‌رود و در هر صفحه همه خبر ها را پیدا می کند. Title و Body آن‌ها را پیش پردازش می‌کند و در نهایت در دیتابیس ذخیره می کند. بعد از اتمام هر صفحه به صفحه بعدی می‌رود و همین کار ها را تکرار می کند.

پیش پردازش:
برای پیش پردازش کردن (نورمالایز کردن، حذف stopword ها و tokenize کردن) از ماژولی که خودم در طول تمرین‌ها دولوپ کردم درست شده که در پوشه normalize قرار دارد.

:tf_idf.py
کار کلی این فایل پر کردن جدول tf_idf است. ابتدا کل خبر ها را بر می‌دارد و بعد df و idf را برای کل توکن های موجود در کل خبر ها حساب می کند، سپس tf_idf را برای هر زوج داکیومنت، توکن حساب می کند.

:db_mysql.py
این فایل یک کلاس DB_MYSQL دارد و هر وقت که بخواهم از کد پایتون به دیتابیس وصل شوم از این کلاس یک object می سازم. این کلاس تابع هایی دارد که با آن‌ها خبر ها و tf_idf ها را در جدولشان درج می کنم.

:preprocess_query.py
این تابع را هر وقت بخواهم چیزی را در PHP پیش پردازش کنم صدا می‌زنم و متنی که قرار است پیش پردازش شود را به صورت argument به این ماژول می دهم. نتیجه پیش پردازش شده آن متن است که به صورت json برگردانده می شود.

:Web Application
قسمت وب اپلیکیشن را با لاراول ساختم.
:Route

Domain	Method	URI	Name	Action	Middleware
	GET|HEAD	/	home	Closure	web
	GET|HEAD	api/user		Closure	api
					auth:api
	GET|HEAD	news	news.index	App\Http\Controllers\NewsController@index	web
	POST	news	news.store	App\Http\Controllers\NewsController@store	web
	GET|HEAD	news/create	news.create	App\Http\Controllers\NewsController@create	web
	GET|HEAD	news/{news}	news.show	App\Http\Controllers\NewsController@show	web
	PUT|PATCH	news/{news}	news.update	App\Http\Controllers\NewsController@update	web
	DELETE	news/{news}	news.destroy	App\Http\Controllers\NewsController@destroy	web
	GET|HEAD	news/{news}/edit	news.edit	App\Http\Controllers\NewsController@edit	web
	GET|HEAD	search	search	App\Http\Controllers\SearchController@search	web

:SearchController
query ای که کاربر وارد می‌کند به این کنترلر می آید. ابتدا کوئری توسط فایل preprocess_query.py پیش پردازش می شود. سپس از جدول tf_idf سطر هایی که شامل توکن های کوئری باشند برداشته می شود.
سپس امتیاز TF-IDF برای هر داکیومنت مرتبط حساب می شود. سپس داکیومنت ها (news) بر اساس امتیازشان مرتب می‌شوند و نتایج به result view می‌روند تا نمایش داده شوند.

:NewsController
در این کنترلر عملیات CRUD روی News بر اساس RestAPI کنترل می شود.
Index پنج خبر آخر را بر می گردند.

Store خبری که کاربر وارد کرده را validate می کند. سپس اگر مشکلی نداشت پیش پردازش می‌کند و در نهایت در جدول news ذخیره می کند.

Update خبری که کاربر وارد کرده را validate می کند. سپس اگر مشکلی نداشت پیش پردازش می‌کند و در نهایت به جای news قبلی ذخیره می کند.

Destroy بر اساس آی دی خبر را پیدا کرده و از جدول news حذف می کند.

Create, show, edit ها تنها view برای انجام این کار ها را بر می گردانند.