Information Retrieval Project

دکتر محمود نشاطی

سید محمد موسوی جوردی

۹۶۲۴۳۰۶۶

Data

ابتدا فایل ها را تبدیل به csv میکنیم.
از کتابخانه pandas استفاده میکنیم
دیتایی که خوانده شده را به فرمت data frame
تبدیل میکنیم(فرمت رایج در پایتون برای کار با
دیتاهایی که جدولی هستند)

تنها از فایل tagsاستفاده کردیم
از داخلش ۲ تا دیتای اصلی به نام های زیر استخراج
میکنیم که در ادامه توضیحشان میدهیم

Question 1

از ما خواسته شد ۵ تگ مشابه با سه مورد را پیدا کنیم.

یک فایل به نام tags داریم.
داخلش یک id وجود دارد که شماره سوال میباشد.
یک کلمه هم وجود دارد که همان اسم تگ میباشد.
برای مثال:
' ۴c++'
10 'java'
168 'data'
10 'algo'
۱۳ 'java'

این یعنی سوال ۴ تگی به نام c++ دارد
سوال ۱۰ دو تا تگ به نام javaو algo دارد.
از دید دیگر هم یعنی تگ java در دو سوال ۱۰ و ۱۳ وجود دارد

Requirments

در نتیجه نیاز داریم که دو نوع دیتا استخراج کنیم.

۱) برای هر تگ، چه سوالاتی هست که آن تگ داخلشون استفاده شده؟
۲) برای هر سوال، چه تگ هایی وجود دارد؟

این دیتاها به ترتیب در متغیرهای tag2questionو question2tag
که دیکشنری میباشند ذخیره شده اند.(همانطور که میدانیم یک کلید و یک مقدار دارند)

به طور مثال برای tag2question
دیکشنری ما شبیه اینه:
}
'java' : [4, 13],
'algo': [10],
...
}

Approach

حال دیتا مورد نیاز را داریم و به حل سوال میپردازیم.
یک تگ به ما داده شده و میخواهد ۵ تگ مشابهش را پیدا کنیم.

با استفاده از دیکشنری tag2question میبینم این تگی که داده است در چه سوالاتی آمده؟
مثلا ۱۰۰۰ تا سوال وجود دارد که داخلش تگ java میباشد.

حال با استفاده از question2tag میبینم این ۱۰۰۰سوال, خودشون چه تگ های دیگری دارند؟

مثلا مجموعا این ۱۰۰۰ تا سوال شامل ۴۰۰۰ تا تگ میشوند. بین این ۱۰۰۰ تا سوال یکسری تگ ها قطعا مشابه هست و تکرار شده.
تعداد تکرارشان را ثبت میکنیم.
در نهایت نزولی سورت میکنیم
۵ تا تگی که بین این سوالا بیشترین تکرار را داشته باشند
را چاپ میکنیم.

خروجی

```
---------------------------------------------- Question 1 ----------------------------------------------

Top 5 most related tags for intellij-idea are :  ['maven', 'android', 'eclipse', 'spring', 'tomcat']
Top 5 most related tags for jax-rs are :  ['rest', 'jersey', 'web-services', 'json', 'resteasy']
Top 5 most related tags for user-interface are :  ['swing', 'android', 'jframe', 'netbeans', 'jpanel']
```

Accuracy

Related Tags

java × 4715

rest × 2227

jersey × 1979

json × 713

resteasy × 643

web-services × 623

jakarta-ee × 616

cxf × 559

jersey-2.0 × 482

spring × 338

more related tags

برای مثال طبق خود ریکامندر استک اور فلو
برای ورودی intleij-idea پیشنهاد های زیر می‌آید
که مشابه خروجی ما میباشد.
['rest', 'jersey', 'web-services', 'json', 'resteasy']
البته از آنجایی که دیتایی که ما با آن کار کردیم
همان نسخه لحظه ای دیتا نمیباشد، میتوان علت تفاوت این یک مورد بخاط همین باشد.

Question 2

سوال ۲)
باید فاصله ی اقلیدسی ۲ تا تگ رو بدست آوریم
فاصله, عکس similarity میباشد یعنی هرچی فاصله کمتر, مشابهت بیشتر

Approach

ما برای اینکه فاصله ۲ تا تگ رو داشته باشیم, باید یک بردار با سایز ثابت داشته باشیم.
از هرکدام, که از روی اون ۲ تا بردار, فاصله رو حساب کنیم.
اینکه این بردار ویژگی چه باشد مهمه.
یک بردار اندازه تعداد کل سوال ها درست کردیم (۱۸۹۹۵)
بعد میگوییم اگر تگ aدر سوال xباشد, مقدار خونه x را ۱ میکنیم, و گرنه ۰ میذاریم.

بعد برای ۲ تا تگ این ۲ تا بردار رو حساب کردیم و فاصله اقلیدسی رو حساب میکنیم.
. چون خواسته است بین ۰ و ۱ باشد.
جواب, فاصله رو تقسیم بر رادیکال nمیکنیم.

خروجی

```
Data is prepared successfully!

---------------------------------------------- Question 2 ----------------------------------------------

The Euclidean distance between regex and static is:  0.13440075505027466
The Euclidean distance between session and spring is:  0.21434596830020816
The Euclidean distance between nullpointerexception and dependency-injection is:  0.09005574516462167
```

Question 3

باید ده جفت ترین شبیه تگ را پیدا کنیم و از کد قسمت قبل استفاده کنیم

برای هر جفت تگ تقریبا زمان ۱۰ دقیقه نیاز هست و چون باید بین تمام تگ ها مقایسه شود
و برای حل چالش مرتبه سوال که n^۲ شده است.
خروجی به زمان غیر قابل دسترسی برای نمایش نیاز دارد.
تفریبا ۶۸۰۰ سال ☺ با وجود ۱۹۰۰۰ تگ سوال (دو for تو دو در تو)
۱۸۹۹۵^۲ * ۱۰min
البته با جست و جو متوجه شدم که اگر با دیتابیست هایی مانندneo۴j که گرافی هستند کار بکنیم این چالش حل میشود.

```
pair_tag_dist = dict()
for tag1 in tags_set:
    for tag2 in tags_set:
        if tag1 != tag2 and not ((tag1, tag2) in pair_tag_dist) and not ((tag2, tag1) in
            pair_tag_dist):
            distance = cal_euclidean_dist(tag1, tag2)
            pair_tag_dist[(tag1, tag2)] = distance
```

فایل کد ها همراه گزارش ارسال شده است
با تشکر از توجه شما