گزارش پروژه

شایان صورتگر 9731094

1-در این پروژه از کتابخانه hazm استفاده شده است.

بعد از خوانده شدن ستون content ، در یک حلقه ی for هر سند به تابع نورمالایزر داده می شود .

```
12      # Normalizing
13      normalizer = Normalizer()
14      for doc in docs:
15          doc = normalizer.normalize(doc)
```

این عمل برای یکسان سازی کلمات انجام می شود مثلا تمام قرآن ها که به شکل قران هم نوشته می شود به یک شکل واحد در می آیند.همچنین مسائلی مثل نیم فاصله و انواع ی و ک اصلاح می شود.

قرآن می خواند= قران می خواند

سپس هر سند با استفاده از تابع word tokenize ، توکنایز شده و در یک حلقه ، کلمات به ترتیب با تابع stemmer ریشه یابی شده و هر کلمه با doc id و پوزیشن آن ، در token stream قرار می گیرد.

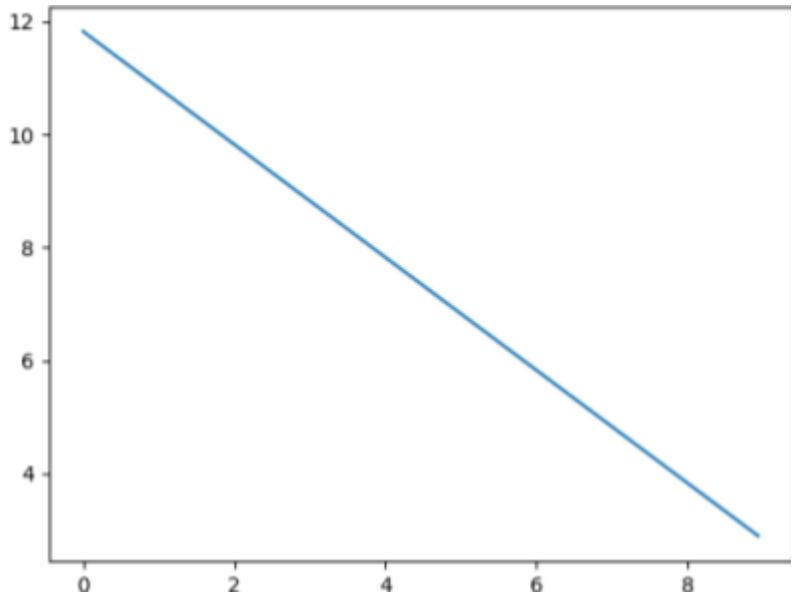
```
21      for docid in range(len(docs)):
22          terms = word_tokenize(docs[docid])
23          for pos in range(len(terms)):
24              term = stemmer.stem(terms[pos])
25              term = terms[pos]
26              token_stream.append([term, [docid, pos]])
27      token_stream.sort(key=lambda x: x[0])
28
```

کتاب ها را آورد =

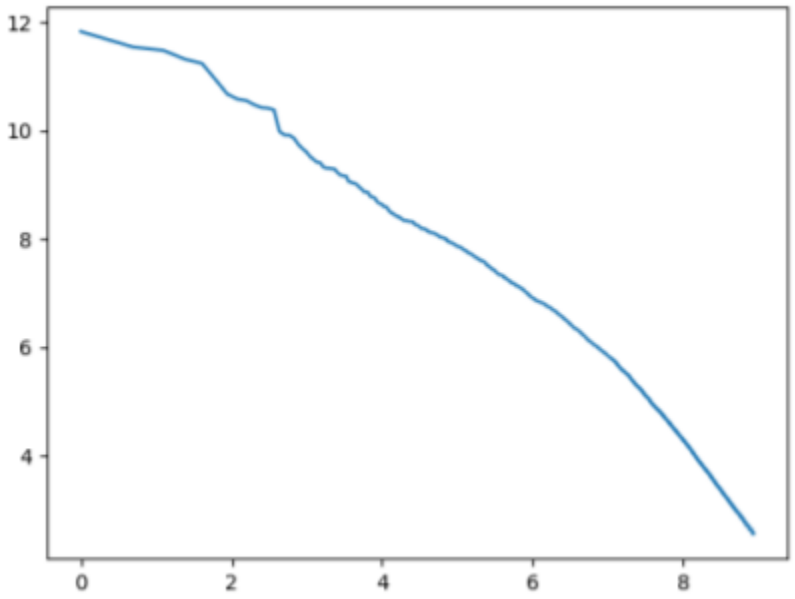
```
[کتاب ,[1,1]]
[را ,[1,2]]
[آورد ,[1,3]]
```

در بخش بعدی هم بعد از محاسبه فرکانس کلمات ، 15 توکن پرتکرار (10 کلمه) حذف شد.دلیل این امر کم اهمیت بودن این کلمات در پرسمان هستند مصل از در به برای و ...

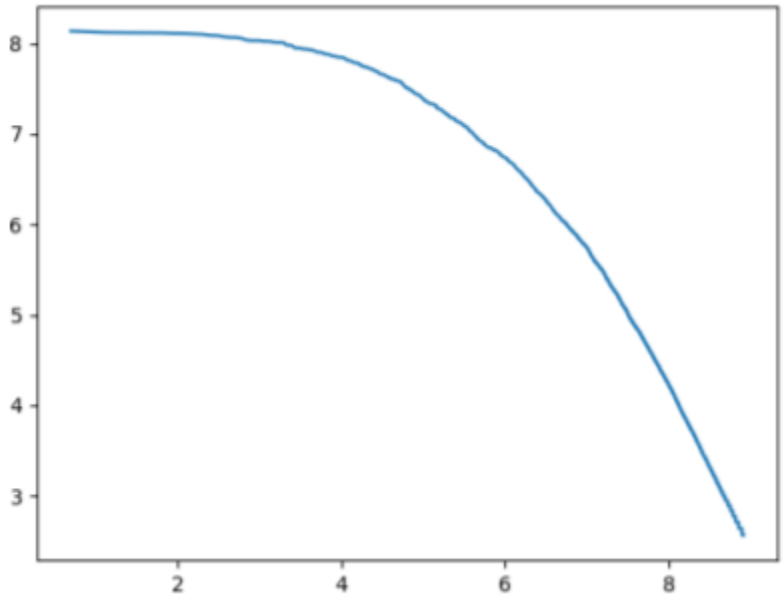
2-طبق قانون زیپف نمودار باید مشابه نمودار زیر باشد:



قبل از حذف stop words



بعد از حذف



می بینیم که در دو حالت قانون برقرار است. برای رسم نمودار ها باید آن ها را لگاریتمی رسم کنیم.

تعداد کلمات با رتبه آنها رابطه توانی دارند. کلماتی که رتبه بالا دارند تعداد دفعات تکرار خیلی زیادی دارند. هر چقدر که جلو تر میرویم تعداد کلمات به شکل نمایی کاهش پیدا میکنند تا جایی که بعضی کلمات به ندرت استفاده میشوند. از آنجایی که این رابطه نمایی است پس با گرفتن لگاریتم به یک رابطه خطی تبدیل میشود و نمودارهای بالا به دست می آید.

-3

تعداد قبل از ریشه یابی(اعداد تقریبی است):

token	word	
8000	150000	500
11000	300000	1000
13000	450000	1500
20000	670000	2000

Log m =1.73

Log t = 1.56

Log 52500 * 1.73 -1.56 = 4200000

بعد از ریشه یابی:

token	word	
7500	150000	500
10500	300000	1000
12500	450000	1500
17500	670000	2000

LogT - 1.55 Log(M) = 1.74

Log(M) = 1.74 * Log(47000) – 1.55 = 6.620000 => M = 4,191,000

در هیچ کدام از حالات اعداد به دست آمده به مقادیر واقعی نزدیک نبودند. یک دلیل آن می تواند صدق نکردن این قانون در فارسی باشد زیرا بسیاری از کلمات از ترکیب ساخته می شوند که در شمارش با زبان انگلیسی فرق می کند و بنابراین در تعداد کل اختلال ایجاد می کنند.

همچنین سایز مجموعه ی ما به نسبت مجموعه هایی دیگر کم است که باعث کاهش دقت می شود.

-4

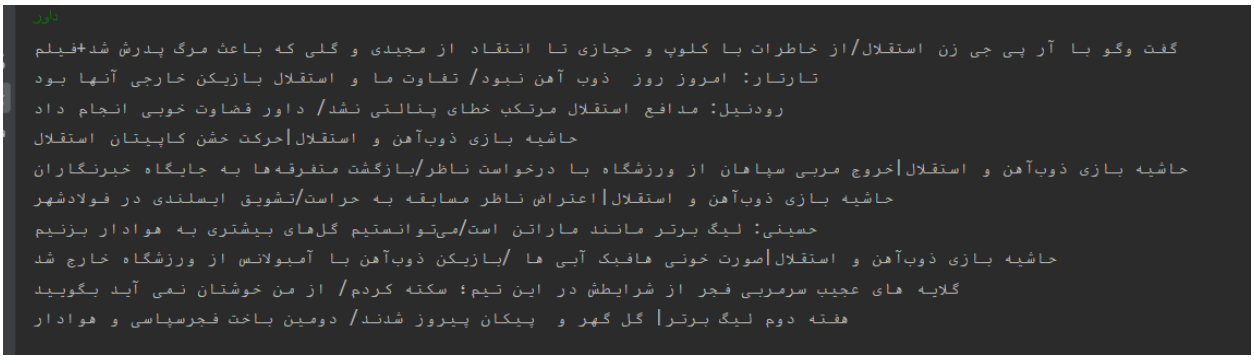
مشکل در ریشه یابی فامیلی ها که نباید ریشه یابی می شدند: مسلمی => مسلم

بسیاری از کلمات به خودی خود معنا دشتند و با اصلشان تفاوت داشتند: نارنجی => نارنج

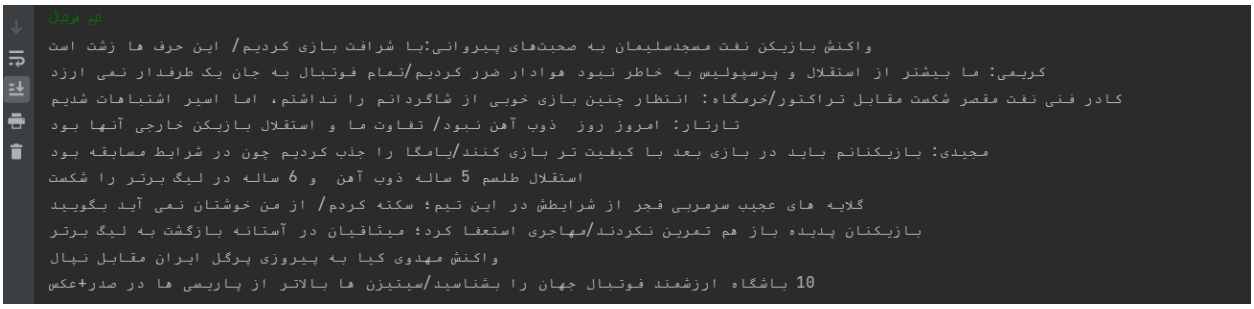
صفات نسبی نباید ریشه یابی می شدند : کاغذی => کاغذ

کلمه ی برترین به بر تبدیل شد که اصلا نباید اتفاق می افتاد

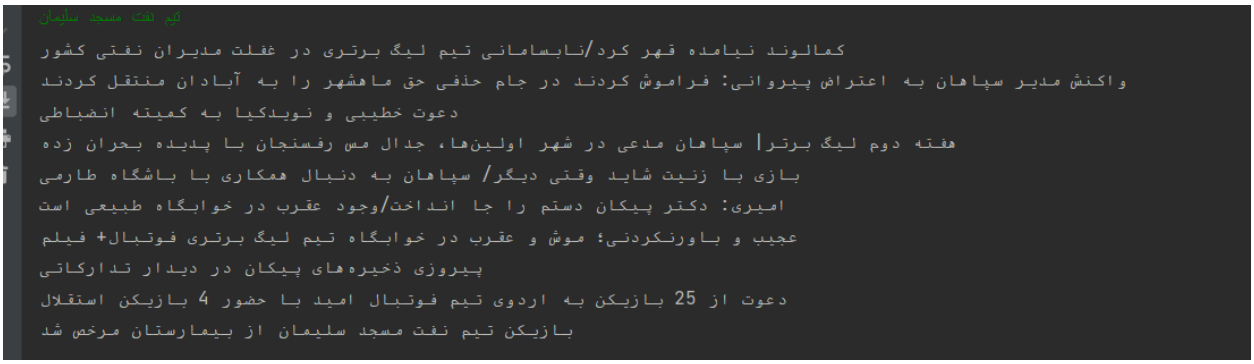
-5



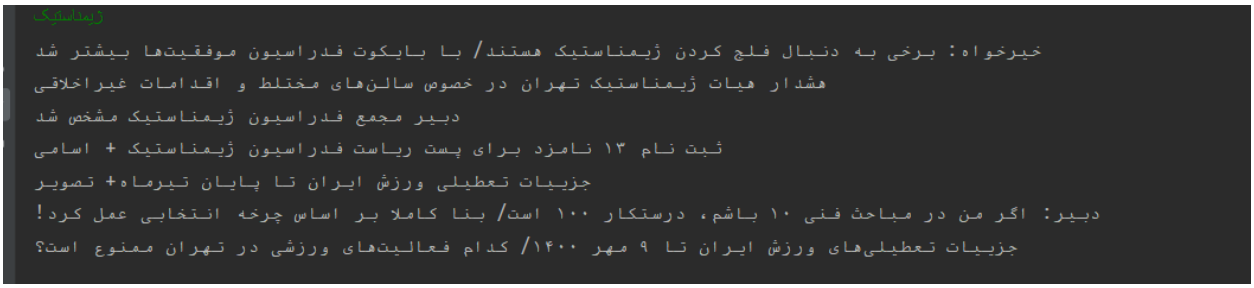
پرسمان داور یک کلمه ای و متداول است و چون اطلاعات زیادی به ما نمی دهد عملا معلوم نیست که نتایج چقدر خوب بوده اند.



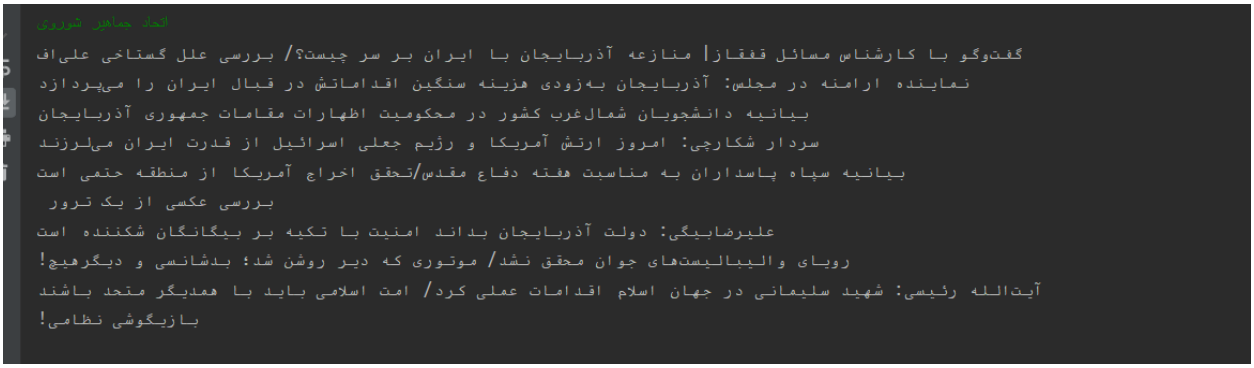
پرسمان داور دو کلمه ای و متداول است و مانند بخش قبل نمی توان ارزیابی درستی از میزان رضایت کاربر داشت.



با وجود اینکه تیم نفت مسجد سلیمان متداول است ولی نسبتا دقت خوبی دارد و حداقل می دانیم تمام خبرها ارتباط خوبی با پرسمان دارند.



چون کلمه ی ژیمناستیک دشوار بوده تمام نتایج برگردانده شده و می توان گفت نهایت تلاشمان را کرده ایم.



اتحاد جماهیر شوروی یک پرسمان دشوار است و می توان گفت که تمام اخبار مربوطه برگردانده شده است.ولی با توجه اینکه کلمه ی اصلی در این پرسمان شوروی است اگر بر روی این کلمه تاکید بیشتری می شد نتایج نامربوط کمتر بود