سعید نباتی :

۴- stopword یا کلمات توقف ، در واقع کلماتی هستند که بطور متداول استفاده می‌شوند و موتورهای جستجو از جمله گوگل به گونه ای برنامه نویسی شده اند که این کلمات را هم در هنگام ایندکس کردن صفحات وب و هم در هنگام بازیابی آن نادیده بگیرند . از جمله این کلمات می توان به (از ، به ، و ، را) در زبان فارسی اشاره کرد.

در هنگام پردازش زبان طبیعی ، تمایلی به این وجود ندارد که stopword در پایگاه داده فضا اشغال کند. و یا از زمان ارزشمند پردازش چیزی را به خود اختصاص دهد. به همین دلیل ، می توان این کلمات را به راحتی و با ذخیره سازی لیست کلمات توقف حذف کرد. (تعداد کم این کلمات در متن هیچ تاثیری ندارد.

۱- در ابتدا باید کتابخانه های مورد نیاز یعنی stopword و word_tokenize را از کتابخانه nltk ایمپورت کنیم. که stopword برای شناسایی کلمات توقف و web_tokenize برای تبدیل عبارت به کلمات که ما استفاده می‌شود. ما باید کتابخانه RusterStemmer را که کار آن تبدیل کلمات به پایه هست را نیز ایمپورت کنیم . حالا باید رشته ها را وارد کنیم. این کد در ابتدا رشته ها را tokenize و پس از آن stopword را حذف نموده و بعد از آن با stemming کردن هر رشته ، یکی یکی کلمات را به termlist اضافه می‌کند . بعد از تکمیل termlist برای هر رشته ای دوباره آن را به لیست docs افزوده و پس به سراغ رشته بعدی در corpus می رود. نتیجه نهایی این عمل یک آرایه دو بعدی است که docs ما دارد و شامل termlist های هر document است

۳- Precision : چند درصد از داکیومنت های بدست آمده با کوئری ما مرتبط هستند. (چند Relevant به Retrived هست).

Recall. یعنی از بین تعداد داکیومنت های مرتبطی که باید برایمان بیاورد چند درصد را آورده و چند درصد از Relevant را آورده)

ج ۲-

در ابتدا متن ها را توکن توکن کرده تا تم ها مشخص شوند.

پس از آن باید جدولی مانند جدول روبرو بسازیم

	T_1	T_2
d_1	۱	۲
d_2	۰	۱

اکنون نوبت آن رسیده بردار زیر را بسازیم

$d_1 = (1,2)$

$d_2 = (0,1)$

و در نهایت برای مقایسه شباهت کسینوسی از فرمول زیر استفاده می کنیم.

$$\text{similarity}(A,B) = \frac{A.B}{\|A\| \times \|B\|} = \frac{\sum_1^n A_i \times B_i}{\sqrt{\sum_1^n A_i} \times \sqrt{\sum_1^n B_i}}$$

ج ۲-

از nltk استفاده می کنیم و باید کتابخانه Sentiment Intensity Analyzer را ایمپورت کنیم.

متغیری بنام SIA تعریف و تابع را درون آن قرار می دهیم. حالا اگر ما به SIA یک جمله بدهیم تا آن جمله را برای ما بازیابی کند و چک کند می توانیم تشخیص دهیم آن جمله منفی است یا مثبت.

neg: میزان منفی بودن که عددی بین ۰ تا ۱ است

pos: ~ مثبت ~ ~ ~ ~ ~ ~

neu: میزان خنثی بودن جمله که عددی بین ۰ تا ۱ است.

compound: عددی بین ۱- تا ۱+ است در صورتیکه منفی باشد جمله منفی و اگر مثبت باشد جمله مثبت است

سعید نباتی

۵-

برای رتبه بندی داکیومنت های موجود برای هر کوئری، در نهایت به فرمول TF-IDF میرسیم. یعنی وزنی که قبلا به ترم (TF) و IDF ضرب شده و فرمول زیر به دست آمد:

$$w_{t,i} = \begin{cases} (1+\log_{10} F_{t,i}) \times \log_{10} N/df_t & t \in d_i \\ 0 & otherwise \end{cases}$$

که از این وزن TFidf می‌توان مثلا فاصله اقلیدسی یا کسینوسی را بدست آورد تا بفهمیم کدام داکیومنت ها به هم شبیه تر هستند. به کمک TF-IDF scoring می‌توان امتیاز داکیومنت را به ازای هر کوئری محاسبه کرد.

$$Score(q, d_i) = \sum_{t \in q} w_{t,i}$$