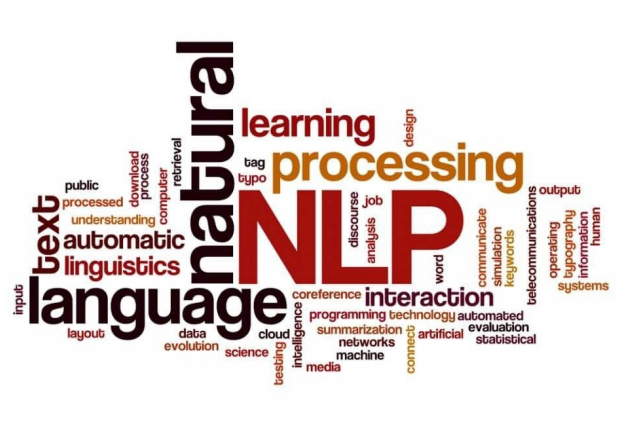
بسم الله الرحمن الرحیم



دانشکده مهندسی برق و کامپیوتر

پردازش زبان‌های طبیعی - تمرین اول

سید مهدی رضوی

استاد : آقای دکتر فیلی

اسفند ماه ۱۴۰۲

فهرست مطالب

فهرست تصاویر

۱ تمرین اول

۱.۱ پیش‌پردازش متن

برای پیش‌پردازش بر روی متن ابتدا بعضی عبارات نگارشی را حذف می‌کنیم تا کلمات هم‌خانواده به راحتی قابل شناسایی باشند.
سپس کلمات را به فرم کوچک آن‌ها تبدیل می‌کنیم و توکن گذاری خواهیم کرد.
در مرحله بعد با استفاده از کتابخانه Porter Stemmer کلمات را هم‌ریشه خواهیم کرد.

۲.۱ document term matrix

برای ساختن این ماتریس به ازای هر سند ما یک بردار به طول اندازه مجموعه واژگان خود خواهیم‌داشت.
بعد به ازای هر سند یک پیمایش بر روی واژگان سند خواهیم داشت و به ازای هر کلمه یک واحد درایه متناظر را اضافه خواهیم‌کرد.
دقت داشته باشید چون که دو کلاس مثبت و منفی در میان داده‌ها داریم ، به ازای هر کدام یک مجموعه واژگان جدا تشکیل می‌دهیم و برای هر کدام نیز یک ماتریس جداگانه تشکیل خواهیم‌داد.
علت این امر محاسبه احتمالات likelihood جدا به ازای هر کلاس خواهد بود.
بردار embedding یا همان احتمالات likelihood برای ساختن مدل Naive Bayes به صورت زیر محاسبه‌خواهدشد :

$$P(term|class) = \frac{count(term, class) + 1}{\sum_{vocabulary}(count(term, class) + 1)}$$

برای ساختن مدل Naive Bayes ما بایستی چند پارامتر مشهور این مدل را تخمین بزنیم. سپس با تخمین این پارامترها قادر به پیش‌بینی نمونه‌های جدید که مدل ندیده‌است ، خواهیم بود.
این پارامترها عبارتند از :

۱. احتمال پسین Prior Probability (چه میزان احتمال دارد که یک نمونه متعلق به یک کلاس باشد.)

۲. احتمال likelihood (به شرط بودن در یک کلاس خاص ، چه میزان احتمال دارد یک پیشامد مانند وقوع یک کلمه اتفاق بیوفتد.)

در مدلی که برای این تمرین پیاده‌سازی کرده‌ام ، ابتدا در تابع init به مقداردهی اولیه این متغیرها پرداخته‌ام.
سپس در تابع fit به ازای نمونه‌های آموزشی به محاسبه هر یک از پارامترهای فوق پرداخته‌ام. چالش ما در این بخش محاسبه احتمال وقوع هر ترم در هر کلاس است. ما برای این کار دو مجموعه واژگان و دو ماتریس document term matrix تشکیل دادیم که در هر کدام به محاسبه P(term | class) خواهیم پرداخت.
در نهایت در تابع predict به ازای نمونه‌های تستی به محاسبه احتمال تعلق هر نمونه به هر کلاس را محاسبه خواهیم کرد و برچسب نمونه را تعیین خواهیم کرد.

$$P(positive|sample) <> P(negative|sample)$$

نتایج ارزیابی ما به ازای این مدل تقریبا به صورت زیر خواهد بود :

precision : 0.6042345276872965

recall : 0.366600790513834

f1Score : 0.45633456334563344

۳.۱ TF-IDF matrix

ماتریس TF-IDF را به صورت حاصل‌ضرب Term Frequency در Inverse Document Frequency تشکیل خواهیم‌داد.
بردار embedding یا همان احتمالات likelihood برای ساختن مدل Naive Bayes به صورت زیر محاسبه‌خواهدشد :

$$P(term|class) = \frac{tfidf(term, class) + 1}{\sum_{vocabulary}(tfidf(term, class) + 1)}$$

در واقع مانند سوال قبل بردار embedding را با استفاده از میانگین تشکیل خواهیم داد.
البته با استفاده از هموارساز add-1-smoothing میانگین احتمالات را برای تشکیل مدل naive bayes را تشکیل می‌دهیم.

برای مدل Naive Bayes این Embedding نیز به مانند مدل قبلی توابع متناظر را پیاده‌سازی خواهیم کرد.
فقط تابع احتمال likelihood ما تغییر خواهد کرد که برای محاسبه آن می‌بایستی از فرمول بالا استفاده کنیم.

نتایج ارزیابی این مدل به صورت زیر خواهد بود : نتایج ارزیابی ما به ازای این مدل تقریبا به صورت زیر خواهد بود :

precision : 0.7142857142857143

recall : 0.004940711462450593

f1Score : 0.009813542688910697

۴.۱ PPMI matrix

برای محاسبه این مرحله ، ابتدا نیاز به تشکیل ماتریس وقوع همزمان کلمات خواهیم داشت. (co-occurence matrix) پس از ساختن این ماتریس ما قادر به ساخت ماتریس PPMI matrix خواهیم بود.

برای برچسب دادن به نمونه‌ها در مدل Naive Bayes در این قسمت ، بایستی که در هر جمله ، میانگین بردار PPMI کلمات آن جمله را محاسبه کنیم.
در نهایت این بردار ماتریس ، بردار Likelihood ما را تشکیل خواهد داد.
در نهایت نیز به پیش‌بینی نمونه‌های جدید خواهیم پرداخت.

precision : 0.63468524567539514
recall : 0.046890927624872576
f1Score : 0.087329505

۵.۱ نتیجه‌گیری

با توجه به این‌که ارزیابی ما با مدل Naive Bayes صورت می‌گیرد و این مدل نیز تا حدی به صورت زیادی به داده‌های آموزش معطوف می‌شود .
در کل معیارهای ذکرشده در این تمرین تقریبا هر بار با تغییر random state تا حدود یک تا دو درصد میزان متغیرهای ارزیابی بالا یا پایین می‌شود.
در کل همانطور که از نام این طبقه‌بند نیز مشخص است ، یک طبقه‌بند احمق است و در اینجا همانطور که از اعداد مشخص است ، با تغییر ماتریس document term به ماتریس TF-IDF که شاخص مناسب‌تری برای ارزیابی میزان مرتبط بودن کلمات در یک جمله است ، شاهد رشد ۱۰ درصدی در دقت بودیم ، اما متاسفانه در ماتریس PPMI که شاخص مناسب‌تری برای ارزیابی میزان وقوع کلمه در یک کلاس است ، ما شاهد بهبود نیستیم.
برای شاخص recall نیز با توجه به این اعداد روش اول یعنی ماتریس document term .از همه روش‌ها مناسب‌تر است اختلاف فاحشی بین این روش با روش‌های دیگر embedding در این تمرین موجود است.
بیشتر به نظر میاد با توجه به ماهیت این ماتریس که متناظر با فرکانس وقوع کلمه است و موارد دیگر خیلی تاثیر کمی دارد ، می‌تواند recall را به خوبی محاسبه کند.