Mohammad Javad Ranjbar

810101173

Homework 2

Natural language processing, Spring 2023

سوال ۱:

در این سوال از یک مدل آماده برای فرایند آموزش استفاده شده است.

برای استخراج بردار ویژگی از دو روش استفاده شده است.

روش اول که TF-IDF می‌باشد که در واقع از ضرب دو مقدار TF که مقدار تکرار یک کلمه در یک داکیومنت (جمله) است در IDF که تعداد حضور آن کلمه در داکیومنت‌های مختلف است بدست می‌آید. بنابراین هر جمله یک بردار به طول کلمات دیکشنری خواهد داشت که هر مقدار آن TF-IDF آن کلمه در آن جمله را نشان می‌دهد. در این روش بدلیل اینکه تکرار کلمه در یک جمله اهمیتی ندارد ولی ما آنرا در این روش در نظر میگیریم دقت ۵۴ درصد بدست آمد.

	precision	recall	f1-score	support
HAPPY	0.51	0.88	0.64	650
SAD	0.70	0.24	0.36	740
accuracy			0.54	1390
macro avg	0.60	0.56	0.50	1390
weighted avg	0.61	0.54	0.49	1390

روش دوم که PPMI می‌باشد از لوگاریتم تقسیم تعداد باری که دو کلمه در کنار هم آمده اند بر تعداد تکرار آنها بدست می‌آید. باز هم همانند روش قبل هر جمله معادل یک بردار به طول دیکشنری خواهد بود. برای هر کلمه داخل جمله هم یک بردار به طول دیکشنری وجود خواهد داشت. این بردار نشان دهنده PPMI آن کلمه با تک تک کلمات دیکشنری هست. در نهایت میانگین بردارهای کلمات این جمله به عنوان بردار ویژگی آن کلمه اعلام می‌شود. در این روش دقت ۸۱ درصد بدست آمد.

	precision	recall	f1-score	support
HAPPY	0.86	0.69	0.77	650
SAD	0.77	0.90	0.83	740
accuracy			0.81	1390
macro avg	0.82	0.80	0.80	1390
weighted avg	0.81	0.81	0.80	1390

سوال ۲:

ابتدا داده‌ها را از لینک مورد نظر دریافت می‌کنیم و پیش پردازش‌های مورد نیاز شامل حذف stop wordها و کاراکترهای اضافی، کوچک کردن کاراکترها را انجام می‌دهیم.

سپس با استفاده از فانکشن‌های تعریف شده، نمونه‌های مثبت و منفی رو استخراج می‌کنیم، در این تمرین برای هر کلمه، دو نمونه‌ی مثبت استخراج شده است و با توجه به متن سوال نیز ۸ نمونه‌ی منفی به صورت تصادفی انتخاب می‌کنیم.

حال ماتریس‌های context و وزن را با صد ویژگی به صورت رندوم با مقادیر بین صفر و یک انتخاب می‌کنیم و فرایند آموزش را آغاز می‌کنیم.

مدل را برای ۱۰ ایپاک و با نرخ آموزش ۰/۰۱ آموزش داده‌ایم.

حال مدل به دست آمده را برای کلمات داخل صورت سوال امتحان می‌کنیم:

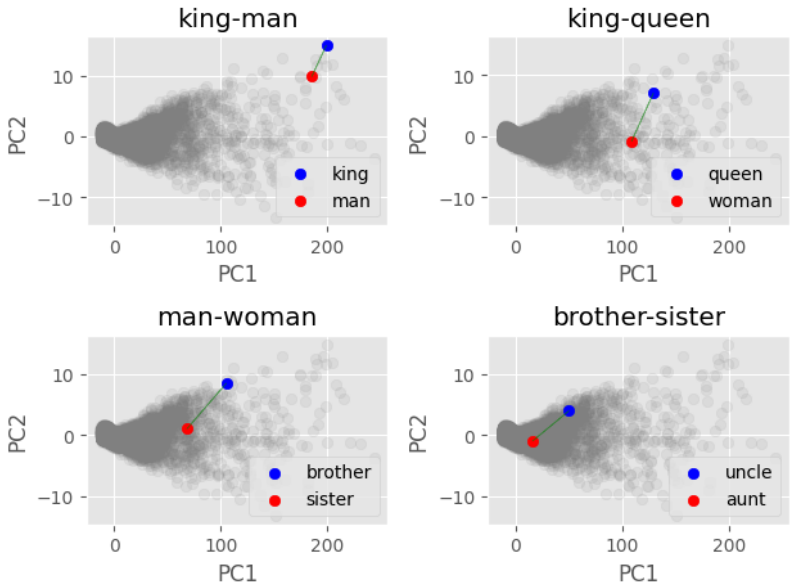
با توجه به مدل به دست آمده مشخص است برای مثال کلماتی مانند queen و woman به یکدیگر نزدیک هستند ولی کلماتی مانند man و text فاصله زیادی نسبت به هم دارند.

علاوه بر این تفاضل بردارهای زیر را رسم می‌کنیم. برای این که این کار را انجام دهیم از PCA برای کاهش بعد به دو استفاده می‌کنیم.

با توجه به ماهیت داده‌ها توقع داریم که برای مثال در نمونه‌های اول ودوم دو بردار با هم موازی باشند یعنی:

$$king - man \;||\; queen - woman$$

نتیجه حاصل از این نمودار به شکل زیر می‌باشد:



نتیجه تقریبا برابر با انتظار ما شد و بردارهای حاصل موازی شدند.

سوال ۳:

الف) ابتدا مانند تمام سوالات قبل متن را load می‌کنیم و عملیات‌های تمیزسازی داده شامل حذف stop wordها و کاراکترهای اضافی، کوچک کردن کاراکترها را انجام می‌دهیم.

سپس وزن‌های glove را load می‌کنیم و برای هر کلمه‌ی متن بردار ویژگی‌ها را استخراج می‌کنیم. سپس مدل lr را از کتابخانه load کرده و 'multi_class='multinomial آموزش می‌دهیم.(برای اینکه از تابع هزینه corss-entropy استفاده کنیم.)

حال نتیجه‌ی این مدل پس از آموزش برای داده‌های آموزش و تست به صورت زیر می‌باشد:

```
              precision    recall  f1-score   support
```

0	0.59	0.31	0.41	543
1	0.72	0.91	0.81	2606
2	0.60	0.40	0.48	1212
accuracy			0.69	4361
macro avg	0.64	0.54	0.56	4361
weighted avg	0.67	0.69	0.66	4361

	precision	recall	f1-score	support
0	0.80	0.26	0.40	61
1	0.68	0.91	0.78	273
2	0.69	0.46	0.55	151
accuracy			0.69	485
macro avg	0.72	0.54	0.58	485
weighted avg	0.70	0.69	0.66	485

ب) مشخص است که با توزیع نامتوازن داده‌ها باعث می‌شود مدل به کلاس‌هایی که داده‌های بیشتری دارند overfit شود. برای مثال کلاس ۱ احتمالا بیشتر از همه کلاس‌ها داده دارد.

ج) با توجه به اینکه ویژگی‌های لزوما از یکدیگر مستقل نیستند. و همینطور این نکته که bayes به همه ویژگی‌ها وزن یکسانی می‌دهد و باعث می‌شود مدل به تعدادی از ویژگی‌هایی که زیاد تکرار شده‌اند حساس شود، به نظر می‌رسد عملکرد مدل باید بدتر شود.

	precision	recall	f1-score	support
0	0.29	0.52	0.37	543
1	0.71	0.75	0.73	2606
2	0.36	0.19	0.25	1212
accuracy			0.57	4361
macro avg	0.46	0.49	0.45	4361
weighted avg	0.56	0.57	0.55	4361

	precision	recall	f1-score	support
0	0.28	0.48	0.35	61
1	0.67	0.78	0.72	273
2	0.48	0.19	0.27	151
accuracy			0.56	485
macro avg	0.47	0.48	0.45	485
weighted avg	0.56	0.56	0.54	485