بنام خدا - محمد مجتبی صادقی - پروژه سوم هوش مصنوعی - شماره دانشجویی: 8161199483

۱- Stemming و Lemmatization ، دو روش برای normalize کردن کلمات می‌باشند.

در روش Stemming ، چند حرف آخر کلمه حذف می‌شود (حتی اگر حروف باقی مانده دارای معنای خاصی نباشد.)
وقتی dataset بزرگ باشد و به Performance بالا نیاز داشته باشیم استفاده بهتر است

در Lemmatization ، کلمه را می‌یابیم که دارای معنا می‌باشد و base به قواعد زبان بستگی دارد.
نیاز به Lookup table دارد و implement کردن آن سخت‌تر است. زمان اجرا هم بیشتر.

با تفاسیر بالا ، ما از Library هضم و lemmatization برای تبدیل کلمات به ریشه آن‌ها استفاده می‌کنیم.
مثال از هر دو را می‌توانید در قسمت "lemmatization vs stemming" مشاهده کنید.

تابع hazm.Stopwords list ، 389 تا از کلمات رایج را ~~از لیست~~ به ما می‌دهد و ما آن‌ها را از داده‌های خود حذف می‌کنیم.

۲- Bayes rule:

$$P(C|x) = \frac{P(x|C)\,P(x)}{P(x)}$$

مجموعه‌ی text هندز از x_1 تا x_n : x ، $d_1 \dots d_n$: d ، موضوع هندز (topic) : C

P(C|x) :

احتمال اینکه خبر از یک topic خاص باشد، به شرطی که متن خبر x باشد.

که هدف ما یافتن این احتمال می‌باشد.

P(x|C) :

احتمال اینکه متن خبر x باشد، به شرطی که از topic C باشد.

ما P(xi|C) را با می‌بی تعداد accurance هر کلمه محاسبه می‌کنیم.

P(C) : احتمال اینکه خبر از topic C باشد. چون در صورت مسئله گفته شده که از تمام شان به یک اندازه داده شده این احتمال را 1/6 در نظر می‌گیریم.

P(x) :

احتمال اینکه متن خبر x باشد. که برای این کار هم جمع تمام xi های دیده شده را بدست می‌آوریم.

$$P(C|x_1 \dots x_n) = P(C)\, P(x_1|C)\, P(x_2|C) \dots P(x_n|C)$$

بدلیل استفاده از Naive bayes می‌توانیم از نتیجه گیری بالا استفاده نمایم.

۳- مثال : [قسمت رو] و طلا افزایش یافت

من [رو زمین] نشستم

در اینجا bigram کاربست. گاهی کلمات همراه کلمات دیگر معنای Context خاصی را می‌گیرند.

من هم در مدل خود از هردوی unigram و bigram استفاده کردم و به علت کم بودن تعداد راهیت بای bigram ها، به آن ها weight نیز اضافه کردم.

۴- وقتی کلمه x_i در متن train شده وجود نداشته باشد، $P(x_i|C)=0$ می‌شود در نتیجه کل $P(C|x)$ برابر صفر می‌شود و این چیزی نیست که ما می‌خواهیم

۵- برای رفع این مشکل از Laplace Smoothing می‌توان استفاده کرد:

$$P(w'|C) = \frac{f(w', c) + \alpha}{N + \alpha * K}$$

$\alpha = 1$ در نظر می‌گیریم

$f(w', c)$ را تعداد رویدادن w' در c در نظر می‌گیریم

N برابر تعداد کل کلماتی که در c هستند

K : تعداد کل کلمات در این مجموعه‌ی متن یا همان feature ها

با این کار از صفر شدن $P(x_i|C)$ جلوگیری می‌شود.

۶- حوادث: پلیس آگاهی، حادثه، محل حادثه، ...

فرهنگی/هنری: فیلم، ایران، جشنواره‌ی فیلم، ...

فناوری: استفاده، کاربر، هوش مصنوعی، ...

سلامت: رژیم غذا، مصرف، غذا

سیاسی: کشور، ایران، ریاست جمهوری

ورزشی: تیم ملی، تیم، لیگ برتر، ...

۷- Recall & فرض کنید در اینجا، ما توجهی به False Positive ها نداریم،

مثلاً فرض کنید تشخیص دهیم که موضوع تمام جمله‌ها سیاسی است. در این صورت

Recall = 100٪ خواهد شد که معیار چندان خوبی نیست

Percision : در این حکم ما توجهی به False negative ها نمی کنیم.

(تا ۱۰۰)

یعنی فرض کنید تعداد زیادی جنبر سیاسی داریم و مایکل را سیاسی تشخیص میدهیم. Percision = 100% حواهد شد که بازخم؟

تنها این چنان معیار مناسبی نیست.

۸- F_1 از میانگین گیری harmonic استفاده می‌کند. اگر False Positive برای ما هزینه‌ی کمتری داشته باشد، استفاده از Accuracy بهتر است. ولی در عمل وقتی، زمانی که این هزینه‌ها متفاوت هستند استفاده از F_1 اهمیت زیادی پیدا می‌کند.

۹- از میانگین F_1 تمام مدل‌ها استفاده می‌کنیم

macro (macro-F_1): از میانگین F_1 تمام مدل‌ها استفاده می‌کنیم

micro: Accuracy است. $\frac{\text{Correct}}{\text{total}}$ به همان برابر

weighted: به هر موضوع، به اندازه‌ی تعداد جنبرهای در آن مدل، به آن weight اختصاص می‌دهیم. انواع ... weighted-F_1, Percision, ... داریم.

۱۰- نتایج را در جدول مشاهده کنید:

Blue: Adeltive Smoothing

Red: without adeltive smoothing

	Health	Political	Sports	Technology	Art	Accidents	All Classes
Precision	0.195 0.105	0.192 0.001	0.197 0.105	0.196 0.101	0.187 0.001	0.198 0.001	-
Recall	0.197 1	0.192 1	0.198 1.0	0.196 0.16	0.196 1	0.188 0.75	-
F1-score	0.196 0.101	0.192 0.101	0.1978 0.101	0.196 0.103	0.191 0.12	0.193 0.102	-
Accuracy	-	-	-	-	-	-	0.194
Macro Avg	-	-	-	-	-	-	0.194
Micro Avg	-	-	-	-	-	-	0.194
Weighted Avg	-	-	-	-	-	-	0.194

8 Micro ، Macro ، Accuracy رابطه

$$\text{Accuracy} = \frac{\text{total} - \text{wrong}}{\text{total}} = \frac{115 - 60}{115} = 94\%$$

$$\text{Macro } F_1 \text{ average} = \frac{0.96 + 0.92 + 0.978 + 0.96 + 0.91 + 0.93}{6} = 94\%$$

Micro Averages 94%

$$\text{weighted Average} = \frac{168 * 0.96 + 190 * 0.978 + 200 * 0.93 + 177 * 0.96}{115} = 94\%$$

۱۱- همانطور که مشاهده می کنید، این احتمال بسیار بالا وجود دارد که حداقل در یک جمله کلمه ای استفاده شود که در dataBase ما وجود ندارد. به همین خاطر مشاهده می کنیم که با افزودن AdditiveSmoothing، نتایج به طور درخشانی بهتر می شود.

۱۲- می توانید مشاهده کنید. مثلا علل Predict اشتباه:

۱- استفاده از trainingSet کوچک

۲- NaiveBayes قرار دادن کلمات نسبت به هم

۳- در برخی مثال ها از برخی کلماتی استفاده شده که بیشتر در انواع خبر های دیگر کاربرد دارد.

مثلا در در مثال اول، استفاده از لفظ هایی رئیس جمهور، وزیر چند بار امور خارجه، انگلیس، باعث شده که احتمال فرهنگی هنری را سیاسی پیش بینی کند که کار درستی نمیشه کرد!