به نام آنکه آموخت انسان را آنچه نمی دانست

دانشگاه تهران

پردیس دانشکده‌های فنی

دانشکده برق و کامپیوتر

درس پردازش زبان‌های طبیعی

CA2

نظرسنجی

behzad.shayegh@ut.ac.ur

بهزاد شایق بروجنی

810 196 678

اسفند ۱۳۹۸

فهرست

مقدمه

با سلام. امروزه تعداد فیلم‌هایی که در سرتاسر جهان تولید می‌شوند با سرعت بالایی رو به افزایش است. مشخصاً زمانی که انتخاب‌های زیادی داشته باشیم، تصمیم گیری برای ما سخت خواهد بود. به صورت واضحی هر فیلمی که تولید می‌شود ارزش دیدن ندارد و این موضوع مشخص نمی‌شود مگر توسط کسانی که فیلم را مشاهده کرده‌اند. تصور کنید می‌خواهید فیلمی برای تماشا انتخاب کنید و می‌خواهید از نظرات دیگران بهره‌بگیرید، پس باید نظرات حرفه‌ای و غیر حرفه‌ای تمام تماشاچیان تمام فیلم‌ها را مطالعه کنید تا فیلمی مناسب انتخاب کنید. مشخصاً این کار مناسبی نیست. تصور کنید بتوانیم سیستمی برای خواندن نظرات کاربران داشته باشیم که نظرات مثبت و منفی را تشخیص دهد و به این واسطه بتوانیم به فیلم‌ها نمره دهیم. بسیار ساده‌تر خواهد بود که با توجه به امتیاز فیلم‌ها فیلمی را برای تماشا انتخاب کنیم. در این پروژه قصد داریم چنین سیستمی پیاده‌سازی کنیم.

دادگان

دادگان این پروژه، دادگان آماده‌ی movie_reviews موجود در کتابخانه‌ی nltk می‌باشد. این دادگان شامل دو دسته نظرات مثبت و منفی، هر دسته شامل ۱۰۰۰ نظر می‌باشد. هر نظر یک فایل حاوی چند جمله است که جملات در خطوط مجزا مهیا شده‌اند. این دادگان به زبان انگلیسی می‌باشد.

درمورد گزارش

در این پروژه تعداد بالایی آزمون و خطاهای متفاوتی انجام شده که اکثر آن‌ها بی‌فایده بوده‌اند. با توجه به حجم زیاد این آزمون‌ها، این موارد در پیاده‌سازی پروژه (کدهای ضمیمه شده) آورده نشده‌اند و فقط موارد تاثیرگذار باقی‌مانده اند. اما تمام این آزمون‌ها به همراه زادگاه ایده و تحلیل نتیجه هرکدام، در این گزارش آورده شده‌اند و برای هر مورد ذکر شده است که تاثیر آن مورد را باقی گذاشته‌ایم یا خیر. این آزمون‌ها مشمول بخش‌های پیش‌پردازش، استخراج ویژگی و آموزش رده‌بند می‌شوند.

پیش‌پردازش

پیش‌پردازش دادگان از گام‌های مهم پروژه‌های پردازش زبان‌های طبیعی است. این پیش پردازش را به چند بخش تقسیم می‌کنیم :

Normalize

نوشته‌ها گونه‌های نوشتاری بسیار متفاوتی دارند. در ابتدا سعی کردیم عملیات «عادی سازی» را بر روی تمام دادگان خود (اعم از دادگان آموزشی، ارزیابی و تست) اعمال کنیم و متون جایگزین را استفاده کنیم. این تغییرات شامل تبدیل تمام حروف بزرگ به حروف کوچک، جایگزین کردن عبارات کامل بجای مخفف آن‌ها و حذف تمام حروف غیر از الفبا (مانند '-') بود. دلیل این کار این بود که مفهوم یک لغت است که به نوشته مفهوم می‌دهد نه گونه‌ی نوشتاری آن. این کار به بهبودی الگوریتم کمک کرد چرا که با این کار لغات یکسان به هم مربوط می‌شوند. البته تعدادی از این صرف‌نظرها در گام استخراج ویژگی بازگردانده شد که هیچکدام مفید نبودند (جلوتر تشریح خواهیم کرد).

Tokenize

در این پروژه، تمرکز ما بر تاثیر لغات بر مفهوم متن است، پس متون را به لغات تقسیم کرده و لغات را بررسی می‌کنیم.

Not & No

این دو مورد، مشخصاً مواردی هستند که تأثیر مفهومی یک لغت را برعکس می‌کنند، پس بررسی جداگانه‌ی این دو مورد کافی نخواهد بود. پس لغات بعد از این دو مورد را به صورت Not_Word درآوردیم تا در بررسی‌های جلوتر تأثیر این دو مورد نیز لحاظ شود. توجه کنید که این عمل را بدون درنظر گرفتن Stop Wordها انجام دادیم چرا که Not در Not a good man واژه‌ی good را نفی می‌کند نه واژه‌ی a را. این کار در عمل مفید واقع شد.

Stop words

زمانی که با مفهوم لغات کار داریم، مشخصاً stop wordها کمک زیادی نخواهند کرد چرا که از خود مفهوم خاصی ندارند. پس آن‌ها را حذف کردیم و تاثیر مثبتی مشاهده کردیم چرا که دیگر وجود یا عدم وجود آن‌ها باعث گمراهی طبقه‌بند ما نمی‌شد؛ همچین حذف آن‌ها حجم متون را به شدت کاهش داد و باعث سرعت بالاتر الگوریتم شد. البته لازم به ذکر است دو عبارت Not و No خود نیز تاثیر مثبتی داشتند پس از حذف این دو مورد صرف‌نظر کردیم و بازدهی بیشتری مشاهده کردیم.

Stem [Not Used]

تصور می‌شد لغات با ریشه‌های مشترک تأثیر معنایی مشابهی بر محتوای کلی متن دارند اما این اشتراک برای رایانه مشهود نیست. پس تمام لغات تمام دادگان را ریشه‌یابی کرده و از ریشه‌ی لغات بجای آن‌ها استفاده کردیم. به نحو عجیبی این‌کار عملکرد الگوریتم را بدتر کرد و بازدهی کاهش پیدا کرد (حدود ۱ درصد). با اینکه نتوانستیم این نتیجه را توجیه کنیم، با توجه به مشاهده‌ای که داشتیم از استفاده از آن صرف‌نظر کردیم.

Lemma

لغات دارای پیشوندها، پسوندها، شناسه‌ها، عبارات جمع و ... هستند که تصور می‌شد این موارد بر روی تأثیر کلمه بر محتوای معنایی متن بی‌اثرند. پس این موارد را نیز در کل دادگان ساده‌سازی کردیم. البته که این عمل نه تأثیری مثبت بر روی

بازدهی داشت و نه تأثیری منفی اما استفاده از آن را ترجیح دادیم چراکه وجود آن تعداد لغات یکتا و درنتیجه سرعت اجرا را کاهش می‌داد.

Vocabulary

داشتن یک لغت‌نامه کامل از مجموعه دادگان همیشه کارآمد خواهد بود پس از تمام لغات کل دادگان یک لغتنامه به همراه تعداد تکرار هر لغت نیز آماده کردیم.

استخراج ویژگی‌ها

برای اینکه بتوانیم یک رده‌بند را آموزش دهیم، لازم است تعدادی ویژگی را از متن استخراج کنیم. در این بخش ما تعداد زیادی ویژگی متنوع را آزمودیم و آن‌هایی که در عمل مفید بودند را نگه داشتیم. این ویژگی‌ها عبارتند از :

Bag Of Word (BOW) [Not Used]

مجموعه‌ی BOW به فرکانس تکرار هر لغت از لغت‌نامه در هر سند از داده گفته می‌شود. این ویژگی از آن رو ممکن بود تأثیرگذار باشد که کلماتی که در نظرات مثبت تعداد تکرار بیشتری دارند، احتمالا تاثیر مستقیمی بر مثبت بودن نظر دارند. این ویژگی تاثیر بسیار مثبتی داشت اما ویژگی‌های بهتری پیدا شد که تأثیر این ویژگی را پوشش می‌داد، پس از این مورد استفاده نشد.

Limited Bag Of Word (LBOW) [Not Used]

با این استدلال که لغاتی که در کل دادگان به ندرت ظاهر شده‌اند، احتمالا تمام کارایی خود را نشان نداده‌اند و ممکن است تأثیر ناعادلانه‌ای داشته باشند، لغات کم تکرار را از مجموعه BOW حذف کردیم و این عمل در نتیجه تأثیر مثبتی داشت. با آزمون و خطا مشخص شد که بهترین مرز برای حذف لغات در حدود حداقل ۶۰ (قبل از اعمال پیش‌پردازش Not & No این عدد ۱۲۰ بود چرا که با اعمال آن پیش‌پردازش بسیاری از لغات به دو دسته تقسیم شدند) تکرار است. این اعمال حد بر روی سرعت اجرا نیز تأثیر شگرفی داشت چرا که لغات مورد بررسی را بسیار محدود می‌کرد و در نهایت حدود ۲۰۰۰ (با مرز ۱۲۰ تعداد ۱۰۰۰) لغت را بررسی می‌کردیم. با این‌که این ویژگی تأثیر بسیار مثبتی داشت اما ویژگی‌های بهتری پیدا شد که تأثیر این ویژگی را پوشش می‌داد، پس از این مورد نیز استفاده نشد.

Word Existence (WE) [Not Used]

با این استدلال که تکرار یک لغت در یک متن شاید خیلی تأثیر گذار نباشد، از شمردن تعداد لغات صرف‌نظر کردیم و فقط وجود یا عدم وجود لغات را بررسی کردیم. این مورد بسیار تأثیر مثبتی داشت. برای توجیه این موضوع می‌توان به یک مثال اکتفا کرد : فرض کنید در تمام نظرات مثبت لغت good به کرّات و حداقل ۱۰ بار تکرار شده باشد. بدیهی‌ست که این موضوع نشان‌دهنده‌ی تأثیر مثبت این واژه می‌باشد، اما اگر در نظری این واژه فقط یک‌بار تکرار شده باشد این مورد یک کاستی تلقی شده و نظر به عنوان نظر منفی درنظر گرفته خواهد شد، پس بررسی «وجود» بجای «تعداد» تأثیر بهتری داشت. با این‌که این ویژگی از بهترین ویژگی‌ها بود (لازم به ذکر است از ویژگی LBOW بهتر نبود)، باز هم یک ویژگی با تأثیر بهتر یافتیم.

Limited Word Existence (LWE)

با این استدلالی مشابه آنچه برای تبدیل BOW به LBOW استفاده کردیم، WE را به LWE تبدیل کردیم و باز هم شاهد پیشرفت بودیم. این ویژگی بهترین تأثیر را داشت و تنها ویژگی استفاده شده در نسخه‌ی نهایی است.

Positive and Negative Words Count (P&N) [Not Used]

به صورت دستی تعدادی از لغات مثبت بسیار مشهود مانند good یا perfect و همچنین تعدادی از لغات منفی بسیار مشهود را تعریف کردیم و مجموع تعداد حضور این کلمات در هر نظر را به عنوان ویژگی درنظر گرفتیم. این ویژگی تأثیر بسیار خوبی داشت که البته این تأثیر توسط LBOW پوشانده شد، چرا که LBOW تمام این لغات را تشخیص داده (پر تکرار در مثبت‌ها یا منفی‌ها) و تأثیر آن‌ها را اعمال می‌کند. تنها تفاوت استفاده از مجموع تعداد بجای تعریف چند ویژگی بود که آزمون و خطا نشان داد نه تأثیری مثبت و نه تأثیری منفی دارد. پس از این ویژگی نیز صرف‌نظر شد.

Positive and Negative Words Existence (P&N-E) [Not Used]

این ایده حاصل مجموع استدلال‌های WE و P&N بود که تأثیر آن توسط LWE پوشانده شد پس از آن صرف‌نظر کردیم.

Capital Word Existence [Not Used]

با توجه به اینکه بزرگ نوشتن تمام حروف یک کلمه به معنای تأکید بر کلمه است ممکن بود یک ویژگی تأثیرگذار باشد اما آزمون و خطا نشان داد تأثیر شگرفی ندارد. شاید دلیل این باشد که تأکید هم می‌تواند مثبت و هم می‌تواند منفی باشد و بررسی جداگانه آن تأثیر زیادی ندارد.

Sentences, Words, Letters Count [Not Used]

تأثیر تعداد جملات، کلمات و حتی حروف امتحان شد اما هیچ‌کدام هیچ تأثیری نداشتند. این مشاهده می‌تواند به معنای بی‌ارتباطی نتیجه با این ویژگی‌ها باشد.

"?", "!", "...", "!!!", "*", digits Count [Not Used]

تأثیر تعداد تکرار علامت‌های "!"، "!!!"، "؟" و "..." که معمولاً برای بیان احساسات در نوشته‌ها بکار می‌روند آزموده شد و مشاهده شد هیچ تأثیری بر نتیجه ندارند. این نتیجه مشابه Capital Word Existence قابل توجیه است. همچنین تعداد ارقام در متن نیز آزموده شد که تأثیر خاصی نداشت.

Rating Decision [Not Used]

در README دادگان می‌خوانیم :

This section describes how we determined whether a review was positive or negative.

The original html files do not have consistent formats -- a review may not have the author's rating with it, and when it does, the rating can appear at different places in the file in different forms. We only recognize some of the more explicit ratings, which are extracted via a set of ad-hoc rules. In essence, a file's classification is determined based on the first rating we were able to identify.

*- In order to obtain more accurate rating decisions, the maximum rating must be specified explicitly, both for numerical ratings and star ratings. ("8/10", "four out of five", and "OUT OF ****: ***" are examples of rating indications we recognize.)*

- With a five-star system (or compatible number systems): three-and-a-half stars and up are considered positive, two stars and below are considered negative.

- With a four-star system (or compatible number system): three stars and up are considered positive, one-and-a-half stars and below are considered negative.

- With a letter grade system: B or above is considered positive, C- or below is considered negative.

We attempted to recognize half stars, but they are specified in an especially free way, which makes them difficult to recognize. Hence, we may lose a half star very occasionally; but this only results in 2.5 stars in five star system being categorized as negative, which is still reasonable.

پس سعی کردیم موارد گفته شده را به عنوان ویژگی‌های نظرات درنظر بگیریم. از همین رو ویژگی‌هایی نظیر وجود عبارات «*****»، «****» تا «*» و یا وجود «10/10» تا «0/10» یا «5/5» تا «0/5» یا «4/4» تا «0/4» را به ویژگی‌ها اضافه کردیم اما متأسفانه تأثیری مشاهده نشد. شاید به این خاطر بود که تنوع این نوع نمره‌دهی‌ها زیاد بوده و اینکه تعداد کسانی که از هرکدام از آن‌ها استفاده کرده‌اند کم بوده و از نظر تعداد برای آموزش این طبقه‌بند ویژگی‌های مناسبی نبوده‌اند. شاید برای درخت تصمیم‌گیری ویژگی‌های مناسبی محسوب می‌شدند اما چون نتیجه‌ای به همراه نداشتند، آن‌ها را نیز کنار گذاشتیم.

Word to Vector (W2V) [Not Used]

از آنجایی که ما به دنبال تأثیر مفهوم لغات بودیم شاید بردار لغات می‌توانست کمک کند، پس این موضوع را امتحان کردیم. این کار را با پنجره‌های ۲ و ۵ و ۷ و ۱۲ امتحان کردیم، همچنین با Min countهای ۳ و ۷ و ۱۰ و ۶۰ نیز آزمودیم و بردارهایی با اندازه‌های ۵، ۱۰، ۱۵، ۲۰، ۲۵، ۵۰ تولید کردیم و مقادیر داخل بردارها را به عنوان ویژگی استفاده کردیم (بردار هر نظر را میانگین بردار لغاتش تعریف کردیم). این ویژگی تأثیر کمی داشت و دلیل آن نزدیکی دو لغت good و bad از نظر برداریست. درواقع مشکل آنجاست که ما فاصله‌ی لغات را فقط از یک نظر نیاز داریم اما این بردارها از تمام جهات لغات را بررسی می‌کنند. پس نتوانستیم از این ویژگی استفاده کنیم.

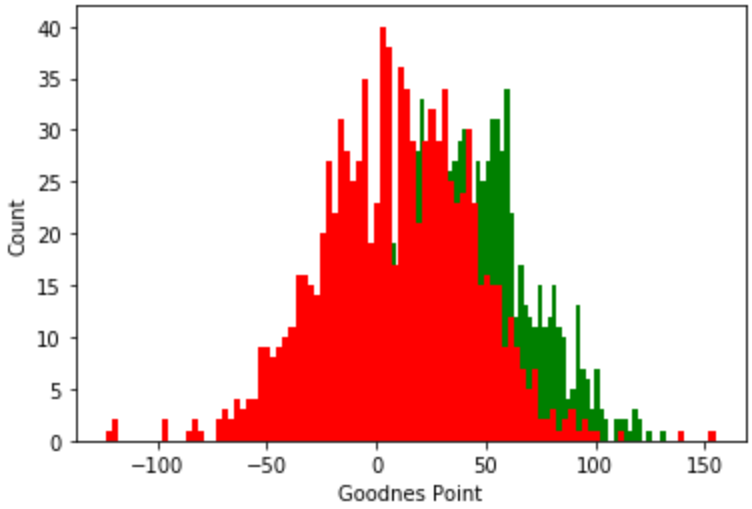
Word to Vector Similarity (W2V-S) [Not Used]

با توجه به مشکلی که در W2V بیان شد، سعی شد تفاوتی که نیاز داریم را از این بردارها استخراج کنیم. برای این منظور، تعدادی از لغات مثبت را در کنار هم و تعدادی از لغات منفی را در کنار هم گذاشت و بردار میانگین هر دسته را محاسبه کردیم. سپس میزان Similarity هر لغت در لغت‌نامه را به هر یک از این دو بردار محاسبه کردیم و میانگین این مقادیر را به ازای لغات هر نظر درنظر گرفتیم و این دو عدد را به عنوان ویژگی‌نظرها درنظر گرفتیم اما تأثیر مثبتی مشاهده نکردیم. پس این تلاش را نیز کنار گذاشتیم.

Output of Linear Regression On Word to Vector (LR-W2V) [Not Used]

در پی تجربه‌ای که از W2V-S حاصل شد تصمیم گرفتیم راهی دیگر برای استخراج تفاوت مدنظر از بردارها بیابیم. پس یک تخمین‌گر Linear Regression آماده کردیم و بردارها را به عنوان ویژگی و تعداد تکرار لغات متناظر با بردارها در نظرات مثبت منهای این تعداد در نظرات منفی ($N_{Positive} - N_{Negative}$)را به عنوان خروجی Linear Regression برای آموزش به آن دادیم. سپس خروجی این تخمین‌گر را بر روی بردار تمام لغات محاسبه کردیم و میانگین این مقادیر را به ازای لغات هر نظر،

به عنوان امتیاز آن نظر درنظر گرفتیم. در این میان محدودیت‌هایی برای لغاتی که تاثیر می‌دهیم نیز اعمال کردیم از جمله تعداد تکرار و تفاوت تکرار در نظرات مثبت و منفی درنهایت با مشاهده‌ی نمودار histogram مربوط به این نتایج شاهد آن هستیم که این ویژگی دارد به‌صورت کارآمدی تفاوت ایجاد می‌کند :



با اینکه این ویژگی تمایز خوبی بین نظرات ایجاد کرد اما تأثیر آن تحت و شعاع ویژگی‌های قبلی قرار گرفت پس بازهم برای ما بی‌فایده بود. شاید اگر بر روی جزعیات آن بیشتر کار می‌کردیم می‌توانست کمک بیشتری بکند اما به دلیل محدودیت زمانی از آن صرف‌نظر کردیم.

آموزش رده‌بند

در بخش قبل، ویژگی‌هایی کارآمد برای طبقه‌بند خود آموزش دادیم. حال وقت آن است که رده‌بندی با آن‌ها آموزش دهیم.
برای این کار قصد داریم از یک طبقه‌بند Naive Bayes استفاده کنیم. پس دو طبقه‌بند MultinomialNB و GaussianNB را امتحان می‌کنیم. GaussianNB را در مواردی که ویژگی‌های پیوسته در دادگان داشتیم (مانند LR-W2V) استفاده شد اما حتی در آن موارد هم مشاهده شد که بهتر است با یافتن یک Threshold آن ویژگی‌ها را نیز به ویژگی‌هایی گسسته تبدیل کرده و از MultinomialNB استفاده کنیم. شاید این به دلیل همیشه وجود داشتن ویژگی‌های گسسته در میان دادگان باشد. پس ما از MultinomialNB استفاده کردیم.

برای تقسیم دادگان به دادگان آموزش و ارزیابی، از روش k-fold validation با shuffle استفاده کردیم.

ارزیابی مدل آموزش‌دیده

با استفاده از رده‌بند آموزش دیده با ویژگی‌های بیان شده، با استفاده از روش k-fold validation به پنج نتیجه رسیدیم که در ادامه علاوه بر همه‌ی آن‌ها، میانگین هر معیار را به عنوان نتیجه اصلی ارزیابی می‌آوریم :

```
{'fit_time': array([0.13245869, 0.15795469, 0.15967321, 0.16152072, 0.16320157]),
 'score_time': array([0.02146816, 0.01993489, 0.02306819, 0.02095079, 0.02186751]),
 'test_accuracy': array([0.8275, 0.8375, 0.85  , 0.815 , 0.83  ]),
```

```
 'test_precision': array([0.84782609, 0.83854167, 0.8287037 , 0.83510638, 0.87096774]),
 'test_recall': array([0.79187817, 0.82564103, 0.88613861, 0.785     , 0.78640777]),
 'test_f1_score': array([0.81889764, 0.83204134, 0.85645933, 0.80927835, 0.82653061])}
```

پس مدل ما درنهایت با معیارهای زیر خروجی خواهد داد:

```
fit_time :            0.15496177673339845
score_time :          0.021457910537719727
test_accuracy :       0.8320000000000001
test_precision :      0.84422911644822
test_recall :         0.815013115816307
test_f1_score :       0.8286414548736858
```

فایل‌های جانبی

به همراه این گزارش، یک پوشه به نام Codes ارائه می‌شود که حاوی فایل‌های زیر است :

NLP_CA2.ipybn : فایل ژوپیتر کد پروژه که شامل تمام مراحل بجز قسمت‌های «Not Used» می‌باشد. البته بصورت استثنا قسمت‌های مربوط به LR-W2V 0 نیز دربر می‌گیرد.

NLP_CA2.html : خروجی قابل نمایش فایل قبل.