به نام خدا

پروژه نرم افزار

عنوان:

استخراج ویژگی محصولات و تحلیل احساسات

استاد راهنما:

دکتر سمانه یزدانی

گردآورنده:

ولی احمد رحمانی

بهار ۱۴۰۲

۱ مقدمه

پردازش زبان طبیعی (Natural Language Processing) که مختصراً NLP نیز نامیده می‌شود، یک روش است که برای درک زبان انسان توسط رایانه استفاده می‌شود. این علم یکی شاخه‌های دانش هوش مصنوعی محسوب می‌شود و به رایانه‌ها کمک می‌کند تا با آگاهی از چگونگی استفاده بشر از زبان، زبان انسانی را درک کند. با این وجود که پردازش زبان طبیعی یک دانش پیچیده و دشوار است، تاکنون پیشرفت‌های زیادی در این زمینه انجام شده است. با استفاده از قدرت پردازش زبان طبیعی، رایانه‌ها می‌توانند احساسات موجود در متون را درک کنند. بطور مثال، دیدگاه مثبت یا منفی نسبت به موضوع مورد بحث، دیدگاه منتقدانه، دیدگاه تاییدکننده؛ همچنین زمانی که یک سیستم کامپیوتری بخواهد پیشنهادهایی برای ارائه خدمات بهتر به یک مجموعه کاری یا فرد بدهد نیز می‌توان از پردازش زبان طبیعی بهره‌مند شد. تحلیل احساسات (Sentiment Analysis) یک روش پردازش زبان طبیعی است که برای تعیین مثبت، منفی و یا خنثی بودن داده‌ها استفاده می شود. تجزیه و تحلیل احساسات اغلب بر روی داده‌های متنی انجام می شود تا به کسب و کارها کمک کند احساسات مشتریان را نسبت به برند بسنجند و نیازهای مشتری را درک کنند{۱}. تحلیل احساسات تا به امروز کمک بزرگی به صاحبان شرکت ها کرده است. یکی از داغ ترین موضوعات مورد استفاده در این روزها در تحلیل احساسات نظرات داده شده بر روی محصولات شرکت ها، نه تنها تحلیل احساس متن در نگاه کلی، بلکه استخراج ویژگی های محصولات و تحلیل احساسات بر روی نظراتی که نسبت به آن ویژگی ها داده شده است، می باشد. محتوای تولید شده توسط کاربران در فضای اینترنت، دارای ویژگی هایی ست که متخصصان با استفاده از آنها سعی دارند به دقت بالاتر و اطلاعات بیشتری دست یابند. دو اصطلاح انتخاب ویژگی و استخراج ویژگی مطرح است که در این پروژه اصطلاح دوم، استخراج ویژگی بکار برده شده است و توضیحات هردو در ادامه آمده است. در پروژه ما قصد داریم تا با پیاده سازی یک مدل، ابتدا به استخراج ویژگی های محصولات و سپس به تحلیل احساسات نسبت به آن ویژگی ها بپردازیم. برای استخراج ویژگی از مدلی که توسط نومان محمد علی و همکاران در مقاله‌ی{۲} " :Extracting Prominent Aspects of Online Customer Reviews A Data-Driven Approach to Big Data Analytics" پیشنهاد شده است استفاده می‌کنیم و برای تحلیل احساسات از واژگان استفاده می‌کنیم. هدف بالا بردن دقت هم در استخراج ویژگی ها و هم تحلیل احساسات است. در استخراج ویژگی چون از تعداد ویژگی های مطرح شده توسط کاربران اطلاع نداریم، پس دراینجا مدل بصورت یادگیری بدون ناظر است. چالش هایی که متخصصین امر با آن مواجه هستند، تشخیص دقیق ویژگیِ محصول، تشخیص ویژگی های ضمنی، تشخیص ویژگی های چند کلمه ای و... می‌باشد. برای حل این چالش ها راه حل هایی وجود دارد، هر چند کامل نیستند اما تا حدی خواسته ی

مشتریان را برآورده می سازد. یکی از راه حل ها که به وفور استفاده می‌شود، استفاده از نقش کلمات در جمله است که معمولا کلماتی که بعنوان اسم نقش دارند برای استخراج ویژگی انتخاب می‌شوند. بعضا از جایگاهی که اسم در قرار گرفته استفاده می‌کنند، بدین شکل که یک دنباله با عنصرهای یکسان به جز جایگاهی که اسم در آن قرار دارد را برای آموزش استفاده می‌کنند و مدل را براساس این ترتیب آموزش می‌دهند. روش های موجودِ حال حاضر برای استخراج ویژگی بسیار است و بیشترین آنها دارای دقت های بالا هستند. مقاله‌ی انتخاب شده در بین روش هایی وجود دارد که جز بالاترین دقت هاست و روشی که استفاده شده پتانسیل فراونی برای بهبود یافتن دارد. توضیحات مربوطه به مدل استخراج ویژگی در بخش ۴ بطور مفصل به شرح آن پرداخته شده است. تحلیل احساسات وقتی بطور کلی بر روی یک متن انجام می‌شود، کار مشکلی نیست و طبعا دقت بالا را بهمراه دارد. چراکه تمامی کلمات متن می‌توانند بطور مستقل هم تجزیه و تحلیل شوند. اما زمانی که بخواهیم تحلیل احساسات را در یک زیر مجموعه‌ای از متن انجام دهیم بعلاوه اینکه این زیر مجموعه وابسته به زیر مجموعه‌ی دیگر(ویژگی) در همان متن است، انجام دهیم، کار بسیار سخت می‌شود. گاه در یک متن بیش از یک ویژگی و یک نظر وجود دارد و تشخیص اینکه کدام نظر مربوط به کدام ویژگی است کار را مشکل می‌سازد. از دیگر چالش ها تشخیص احساس ضمنی است که در خود متن بصورت صریح نیامده است. برای حل این چالش ها ابزارهایی مانند تجزیه گر استنفورد (Stanford Parser) وجود دارد که می‌تواند تمام وابستگی های موجود در متن را نشان دهد. در این پروژه، براساس مجموعه داده در دست، برای هر متن ما یک ویژگی داریم، پس چالش مذکور برای مورد ما مطرح نیست. توضیحات مدل پیشنهادی برای تحلیل احساسات در بخش ۴ به آن پرداخته شده است.

۲ پیشینه‌ی پژوهش

تعداد بسیاری پژوهش در زمینه‌ی استخراج ویژگی و سپس تحلیل احساسات انجام شده است. آنهایی که بالاترین دقت را داشته اند انتخاب شده اند و بصورت نزولی مرتب شده و خلاصه آنها در زیر آورده شده است. در اینجا ما فقط به مقدار f1-score بسنده می‌کنیم، چراکه خود نشان دهنده و وابسته به دو معیار دیگر یعنی precision و recall است.

۱. آقای ویشو هو و همکاران در سال ۲۰۱۰ {۱۶} در پژوهش خود به بالاترین دقت در تحقیقات موجود در دست ما رسیده است. دقت بدست آمده ۰.۹۱ است. ایشان بعد از برچسب گذاری نظراتی که حاوی نظر مثبت، منفی یا خنثی بوده را جدا کرده است، چرا که ممکن است یک نظر حاوی ویژگی باشد اما نظر راجع به آن ویژگی داده نشده باشد. سپس به استخراج ویژگی ها، چه صریح و چه ضمنی پرداخته اند و در آخر با حذف کردن داده های نامربوط به ویژگی های منتخب رسیده اند.

۲. آقای دی تیجا سنتوش و همکاران در سال ۲۰۱۶ {۱۷} دقت ۰.۹ را بدست آورده اند. آنها با استفاده از مدل LDA(Latent Dirichlet Allocation) و ترکیب آن با FOT(Feature Ontology Tree) ویژگی ها را استخراج کرده اند.

۳. بولان اوجوکو و همکاران در سال ۲۰۱۲ {۱۸} مانند مورد ۲ دقت ۰.۹ را بدست آورده اند. با این تفاوت که در بخش تحلیل احساسات بیشترین دقت را داشته اند. در تحلیل احساسات به دقت ۰.۸۷ برای f1-score رسیده اند. آنها پس از انجام پیش پردازش کار پیدا کردن ویژگی و کلمات دارای بار معنایی نسبت به ویژگی ها را جداگانه انجام دادند. سپس با استفاده از تجزیه گر استنفورد ارتباط هر کلمه نظر را با ویژگی مربوطه پیدا کرده و یک در قالب نتیجه کار پیشنهاد داده اند که محصول مورد نظر به مشتری پیشنهاد شود یا خیر.

۴. چانگین کوان و فوجی رِن در سال ۲۰۱۴ {۱۹} به دقت ۰.۸۷ دست یافته اند. آنها کار خود را به سه قسمت پیش پردازش، استخراج ویژگی و استخراج نظر تقسیم کرده اند. برای مرحله‌ی پیش پردازش از برچسب گذاری، استخراج اسم ها و عبارت های اسمی و سپس تشخیص موجودیت ها و در آخر حذف کلمات توقف استفاده کرده اند. در مرحله استخراج ویژگی از PMI و TF-IDF استفاده کرده اند. سپس در مرحله‌ی استخراج نظر، نظر مربوطه به هر ویژگی را تشخیص داده و براساس آن نتایج خود را ارائه دادند.

۵. نعمان محمد علی و همکاران در سال ۲۰۲۲ {۲۰} به دقت ۰.۸۶ رسیده‌اند. با این وجود که کار تحلیل احساسات را انجام نداده اند اما در روش خود به روزترین بودند. این مقاله برای پیاده سازی توسط نویسنده انتخاب شده است و توضیحات بطور مفصل در اینجا آمده است.

۶. ین کنگ و لینا ژو در سال ۲۰۱۳ {۲۱} دست یافته اند. آنها در سه مرحله‌ی پیش پردازش، استخراج ویژگی و پاکسازی داده پژوهش خود را انجام دادند. در پیش پردازش ابتدا توکن بندی سپس حذف کلمات توقف و برچسب گذاری و در آخر استخراج ارتباط بین کلمات را انجام دادند. آنها ویژگی های ضمنی و صریح را استخراج کرده وسپس کار پاکسازی داده را انجام دادند.

۷. سیتی روهیده احمد و همکاران در سال ۲۰۱۹ {۲۲} به دقت ۰.۸۲ دست یافتند. آنها بعد از پیش پردازش بر روی داده ها از ترکیب الگوریتم کولونی مورچه و KNN برای استخراج ویژگی بکار گرفتند. ارتباط بین ویژگی ها و کلمات با بار معنایی را پیدا کردند. آنها برای تعبیه‌ی کلمات از مدل TF-IDF استفاده کردند. در آخر مدل خود را ارزیابی کردند.

۳ تجزیه و تحلیل احساسات در سطوح مختلف

شهریار محمدی و اسلام ناظمی در مقاله‌ی {۳} خود تجزیه و تحلیل احساسات را به سه سطح مختلف تقسیم کرده اند که به اختصار در اینجا آورده شده است.

۳/۱ در سطح سند

در این سطح از تجزیه و تحلیل به کل سند بعنوان یک موجودیت واحد نگاه می‌شود که دارای قطبیت مثبت یا منفی ست. یادگیری ماشین (یادگیری با ناظر) و یادگیری براساس واژگان (یادگیری بدون ناظر) از مهمترین رویکردها برای تعیین قطبیت در این سطح است. در رویکرد یادگیری مبتنی بر واژگان، ابتدا در سطح کلمه و سپس در سطح جمله و در آخر از تجمیع مراحل قبل در سطح سند، قطبیت بدست می‌آید. از معایب این سطح این است که برای تعیین قطبیت به یک سند بطور کلی نگاه می‌شود، این درحالیست که یک سند ممکن است دارای جملات مثبت، منفی یا خنثی باشد. بنابراین تعیین قطبیت در سطح جمله بسیار دقیق تر خواهد بود.

۳/۲ در سطح جمله

در این سطح هدف تعیین کردن این است که جمله دارای قطبیت مثبت، منفی یا خنثی است. در اکثر کارهایی که انجام شده است، دیده شده که نتیجه بسیار به گرایش معنایی کلمات در آن جمله وابسته است و امکان دارد که یک کلمه در جمله، با توجه به گرایش احساسی کلمات پیرامون آن، گرایش متفاوتی را داشته باشد.

۳/۳ در سطح ویژگی

آنطور که این سطح قطبیت نظرات را نمایان می‌سازد، دو سطح قبلی این توانایی را ندارند. بجای اینکه بصورت کلی چه سطح سند و چه سطح جمله، به قطبیت نگاه شود، سطح ویژگی مستقیماً خود نظر را دنبال میکند .در اصل در این سطح این مفهوم رسانده می‌شود که هر نظر دارای قطبیت مثبت، منفی یا خنثی است و این نظر نسبت به یک ویژگی بیان شده است. در این سطح، از روشهای مختلفی برای استخراج ویژگی استفاده میشود. روشهایی همچون روشهای آماری (مانند بهره اطلاعاتی، مربع کای و...)، مبتنی بر واژگان، هستی شناسی، روشهای مبتنی بر تکرار و رابطه و... که هریک دارای مزایا و معایب خاص خود است. همانطور که مطرح شد، با دو اصطلاح انتخاب ویژگی و استخراج ویژگی روبرو هستیم. در این بخش تصمیم داریم اندکی به آن بپردازیم و تفاوت آن دو را برای ایجاد درک بیشتر نمایان سازیم.

۳/۳/۱انتخاب ویژگی

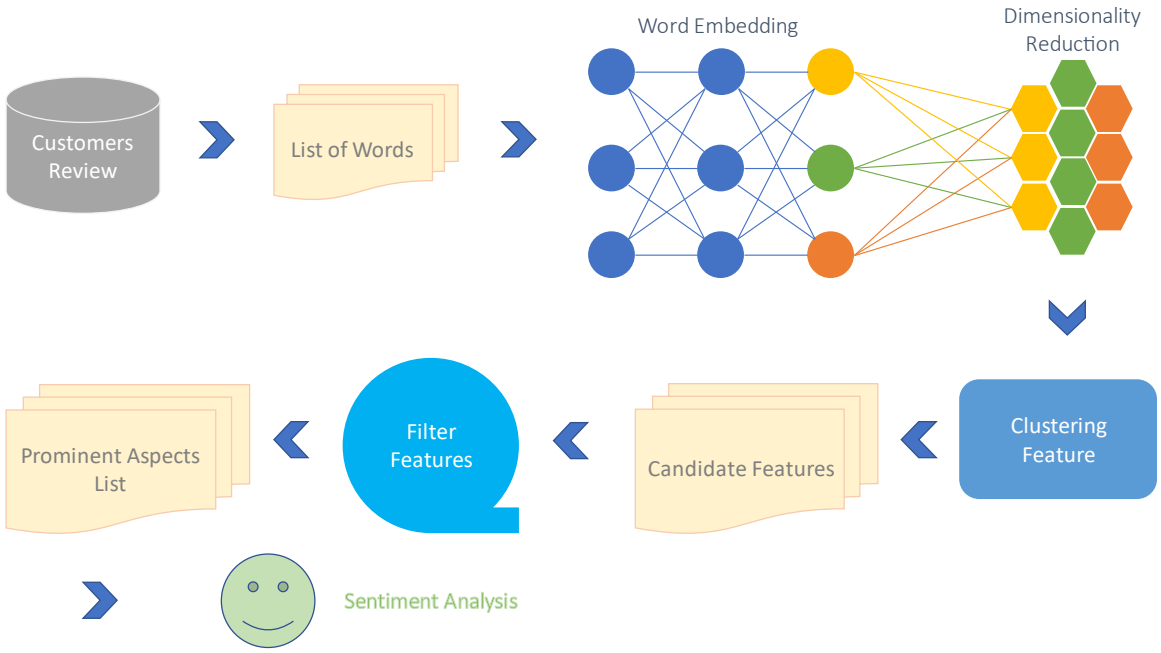
انتخاب ویژگی، همانطور که از اسمش پیداست، انتخاب و گزیدن ویژگی از بین ویژگی‌های موجود در داده هاست. برای مثال فرض بگیرید که یک کار تحلیل احساسات در زبان انگلیسی می‌خواهیم انجام دهیم. این را می‌دانیم که کلمات توقف در تحلیل احساسات نقشی ندارد، یعنی به کار تحلیل احساسات کمکی نمی‌کنند. پس اگر ما کلمات را بعنوان ویژگی در نظر بگیریم، به جز کلمات توقف مابقی کلمات(ویژگی ها) را برای کار تحلیل احساسات انتخاب می‌کنیم. در واقع در اینجا ما ویژگی های تاثیرگذار را انتخاب کرده و از دیگر ویژگی ها صرف نظر می‌کنیم.

۳/۳/۲استخراج ویژگی

استخراج ویژگی، با کمی تفاوت نسبت به انتخاب ویژگی بدین معنی ست که با استفاده از ویژگی‌های موجود در مجموعه داده، ویژگی های جدیدی تولید کرده تا کار مورد نظر به شکل بهتری انجام شود. مثال بالا را در نظر بگیرید؛ برای استخراج ویژگی می‌توان ابتدا نقش کلمات(ویژگی‌ها) را در جمله مشخص کرد و سپس برای تحلیل احساسات فقط از صفات استفاده شود. چراکه صفات بیشترین بار معنایی را نسبت به دیگر اجزا دارند. شاید در اینجا شبهه‌ای پیش آید که باز هم انتخاب ویژگی صورت گرفته است اگرچه چنین نیست. ویژگی‌ها همان کلمات هستند که با بیرون کشیدن(استخراج) نقش آنها و تبدیل آنها بعنوان ویژگی جدید از آن استفاده می‌کنیم. جالب است که اینجا کار استخراج را صریحاً انجام داده‌ایم، در حالیکه شاید بتوان بجای گسترش دادن(Extend) از ترکیب کردن (Synthesize) استفاده کرد. مثلا برای تشخیص اینکه یک شکل مربع است یا مستطیل، از تقسیم طول و عرض (ترکیب) به مقصود دست پیدا کرد.

۴ روش

در اینجا ما به توضیح معماری مدل پیشنهادی {۲} و خود می‌پردازیم و هر قسمت را بطور مفصل شرح خواهیم داد. همانطورکه گفته شد بخش استخراج ویژگی طبق مدل پیشنهادی توسط {۲} و بخش تحلیل احساسات توسط نویسنده پیاده سازی شده است. پیاده سازی در زبان پایتون انجام شده است. در تصویر زیر معماری مدل قابل مشاهده است.



۴/۱ تعبیه کردن کلمات

۴/۱/۱ Word2Vec :

از مشهورترین تکنیک ها Word2Vec می‌باشد که در سال ۲۰۱۳ توسط میکولوف و همکارانش در {۴} معرفی شد. این مدل شامل دو لایه شبکه عصبی ست و یک مجموعه داده را بعنوان ورودی دریافت می‌کند و بعنوان خروجی یک مجموعه داده وکتور شده (هر توکن تبدیل به یک آرایه یک بُعدی می‌شود) را مربوط به داده‌ی ورودی برمی‌گرداند. مدل Word2Vec دو رویکرد Hierarchical Softmax و Negative Sampling را داراست. رویکرد اول با نام Continuous Bag of Word یاد می‌شود که از کلمات اطراف یک کلمه برای پیش بینی آن کلمه استفاده می‌کند. رویکرد دوم با نام Skip-gram یاد می‌شود که برخلاف اولی از کلمه مورد نظر برای پیش بینی کلمات اطراف استفاده می‌کند. آزمایشات نشان داده است که رویکرد دوم نسبت به رویکرد اول عملکرد بهتری دارد{۲}. برای پیاده سازی از کتابخانه gensim با ورژن ۴.۳.۱ استفاده شده است و آرگومان های آن با مقادیر زیر مقداردهی شده است (تمام مقداردهی‌ها برای تمام مدل‌ها طبق {۲} انجام شده است).

- `window=30`
- `min_count=100`
- `sg=1`
- `vector_size=300`
- `workers=8`

*لینک پروژه نرم افزار:

https://github.com/valiahmad/Software-Project

۴/۱/۲ BERT:

مدل BERT(Bidirectional Encoder Representations from Transformers) یکی از بهترین مدل های زبانی ساخته شده براساس تبدیل کننده، معرفی شده توسط تیم گوگل، جیکوب دِولین و همکارانش در سال ۲۰۱۸ است. تبدیل کننده یا ترانسفورمر یک attention mechanism است که تفاوت بین کلمات و زیرکلمات یا اجزای آن را در روابط متنی مشخص می‌سازد. این مدل دو مکانیسم جدای رمزگذار (Encoder) و رمزگشا (Decoder) را در خود جای داده است. اولی (رمزگذار) ورودی را می‌خواند و دومی (رمزگشا) کار پیش بینی را انجام می‌دهد. مدل BERT بصورت تصادفی یک زیر مجموعه از کلمات را در یک جمله با یک mask token جایگرین می‌کند؛ سپس کلمه‌ی مورد نظر را براساس کلمات اطراف آن پیش بینی می‌کند. این مدل دارای چهار بُعد [#layers, #batches, #tokens, #features] است. این مدل یک شبکه عصبی عمیق با ۱۲ لایه برای ورژن Base و ۲۴ لایه برای ورژن Large است. منظور از batch number تعداد جملات ورودی به مدل است، منظور از feature number تعداد ابعاد نمایش داده شده توسط هر توکن است، ۷۶۸ در ورژن Base و ۱۰۲۴ در ورژن Large است{۲}. برای پیاده سازی از کتابخانه transformers با ورژن ۴.۲۸.۱ استفاده شده است. برای بدست آوردن وکتور هر کلمه دامنه‌ی batch حذف شده و ترتیب از [#layers, #batches, #tokens, #features] به [#tokens, #layers, #features] تغییر یافته است و فقط وکتورهای ۴ لایه‌ی آخر به یکدیگر وصل شده است.

۴/۲ کاهش چندین دامنه

۴/۲/۱ SOM:

مدل (Self-Organizing Maps) SOM توسط تیوو کوهِنِن در سال ۱۹۸۲ معرفی شد {۵-۸} ، همچنین بعنوان SOM کوهنن یا شبکه کوهنن شناخته می‌شود. مدل SOM جزء تکنیک های یادگیری بدون ناظر است که وکتورها را با دامنه بزرگ به فضای کوچکتر نگاشت می‌کند. SOM یک نوع از ANN است که با منابع کم می تواند مجموعه داده را مدیریت کند. بطور معمول، SOM برای خوشه بندی و به منظور بصری سازی استفاده می‌شود{۹}. این مدل می‌تواند وکتورهایی با دامنه بزرگ را به دامنه‌ی دو بعدی نگاشت کند. یکی از بهترین مزایای این مدل نسبت به دیگر تکنیک های خوشه بندی این است که SOM توپولوژی داده‌ی اصلی را حفظ می‌کند و این بدین دلیل است که داده ها در فضای دو بعدی بازتابی از آن در فضا با دامنه‌ی بالا می‌باشد. مکانیزم یادگیری داخلی بطور مداوم وزن‌ها را بروز می‌کند تا این که وکتورهای ورودی و خروجی پایدار شوند. به عبارت دیگر، دیگر تکنیک های خوشه بندی مانند K-means خروجی ای تولید می‌کنند که برای بصری سازی کردن سخت است چرا که در فضای دو بعدی نیست {۲}. برای پیاده سازی از کتابخانه susi با ورژن ۱.۲.۲ استفاده شده است. آرگومان های آن به شکل زیر مقداردهی شده است.

- n_rows=80,
- n_columns=80
- init_mode_unsupervised='random'
- n_iter_unsupervised=50000
- train_mode_unsupervised='batch'
- neighborhood_mode_unsupervised='linear'
- learn_mode_unsupervised='min'
- distance_metric='euclidean'
- learning_rate_start=0.5
- learning_rate_end=0.05
- nbh_dist_weight_mode='pseudo-gaussian'
- n_jobs=3
- verbose=0

۴/۲/۲ t-SNE:

مدل t-SNE یک تکنیک آماری موثر برای کاهش دامنه می‌باشد که می‌تواند برای بصری سازی داده های چند دامنه‌ای استفاده شود {۱۰-۱۳}. با توجه به مجموعه داده‌های پیچیده با ابعاد متعدد، t-SNE این داده‌ها و روابط مهم بین بردارها را در فضایی با ابعاد پایین‌تر نگاشت می‌کند و با حفظ ساختار مجموعه داده اصلی، آن را به صورت دو بعدی یا سه بعدی نشان می‌دهد. بعلاوه برای بصری سازی داده ها، قدرت t-SNE برای خوشه بندی در یادگیری بدون ناظر آن را قادر می‌سازد تا در پیش بینی و طبقه بندی به الگوریتم های یادگیری ماشین کمک کند {۲}. برای پیاده سازی از کتابخانه scikit-learn با ورژن ۱.۰ استفاده شده است. آرگومان های آن با مقادیر زیر مقداردهی شده است.

- `n_components=2`
- `perplexity=40`
- `learning_rate=300`
- `early_exaggeration=24`
- `n_iter=2500`
- `n_iter_without_progress=500`
- `min_grad_norm=1e-7`
- `metric='euclidean'`
- `init='pca'`
- `random_state=25`
- `square_distances='legacy'`

۴/۳ خوشه بندی ویژگی ها

بطور کلی الگوریتم های خوشه بندی به شش دسته‌ی اصلی تقسیم می‌شوند:

* خوشه بندی سلسله مراتبی «مبتنی بر اتصال» (مثال، BIRCH، ROCK، DIANA)
* خوشه بندی Partitioning "مبتنی بر مرکز" (مثال، K-means، K-modes)
* خوشه بندی مبتنی بر چگالی (مانند DBSCAN، DENCAST)
* خوشه بندی مبتنی بر توزیع (مثال، مدل های ترکیبی گاوسی، DBCLASD)
* خوشه بندی فازی (مثال، Fuzzy C means، Rough k-means)
* مبتنی بر محدودیت (مثال، درخت تصمیم، جنگل تصادفی، Gradient Boosting)

برای خوشه بندی ما الگوریتم خوشه بندی K-means++ را انتخاب کرده ایم.

K-Means :

دیوید آرتور و سرجی واسیلویتسکی الگوریتم خوشه بندی K-means++ را درسال ۲۰۰۷ معرفی کردند تا بر اشکالات ناشی شده از الگوریتم K-means را برطرف کنند {۱۴}. آنها نتایج شبیه ساز مونته کارلو را ارائه کردند که ثابت می‌کند K-means++ از نسخه استاندارد بهتر عمل می‌کند. این یک الگوریتم خوشه‌بندی بدون ناظر است که داده‌ها را به k تعداد خوشه، جدا می‌کند. مکانیسم کار k-means++ انتخاب نقاط مرکزی اولیه و شروع K-means استاندارد است. این کاری است که k-means++ انجام می دهد تا از خوشه بندی ضعیفی که توسط الگوریتم K-means کلاسیک انجام می شود جلوگیری کند. مراکز K-means++ بین داده ها توزیع می شوند و هزینه کمتری نسبت به مقداردهی اولیه تصادفی مراکز دارند. مقداردهی اولیه در K-means++ با تخصیص تصادفی یک مرکز خوشه شروع می شود و با توجه به مرکز اول، مراکز دیگر را جستجو می کند. بنابراین، الگوریتم تضمین می‌کند که به جای بدست آوردن بهینه محلی به دست آمده توسط الگوریتم قبلی، نتیجه بهینه را پیدا کند. برای پیاده سازی از کتابخانه scikit-learn با ورژن ۱.۰ استفاده شده است. آرگومان ها با مقادیر زیر مقداردهی شده است.

- `n_clusters=150`
- `init='k-means++'`
- `n_init=15`
- `max_iter=400`
- `tol=0.0001`
- `verbose=0`
- `random_state=0`
- `copy_x=True`
- `algorithm='elkan'`

۴/۴ استخراج ویژگیها

با توجه به مجموعه‌ای از خوشه‌های بردارها که کلمات و عبارات تولید شده از مرحله خوشه بندی را نشان می‌دهند، هدف فیلتر کردن نشانه‌های نامربوط و استخراج لیستی از ویژگی ها است. الگوریتم زیر روش انتخاب ویژگی پیشنهادی را نشان می دهد. با شمارش توزیع فراوانی برای اسم ها، صفت ها و افعال فقط با استفاده از تگ های نقش کلمات شروع می شود زیرا این نقش های کلمه اکثر ویژگی ها را در خود جای می‌دهد. مرحله بعدی محاسبه فاصله برای هر خوشه و استخراج فاصله از مرکز خوشه برای هر توکن است. سپس به صورت صعودی آیتم‌ها را در هر خوشه مرتب می‌شود و آیتم‌های برتر، «فهرست ویژگی های برگزیده» را انتخاب می‌کند. تبدیل وکتورها به کلمات معادل گام بعدی است. مرحله بعدی فیلتر کردن لیست با انتخاب فقط کلمه‌های موجود در جملات است. در نهایت، استخراج فراوانی برای کلمات و عبارات واحد از توزیع فراوانی ذکر شده اولیه و حذف مواردی که از آستانه خاصی عبور نکرده‌اند. در نتیجه موارد فیلتر شده فهرست ویژگی ها را تشکیل می‌دهند. الگوریتم ۱ در صفحه بعد، الگوریتم استخراج ویژگی می‌باشد.

Algorithm 1 Feature Selection

Input: (LIST: *Vectors_Clusters*)
(LIST: *Tagged_Sentences*)
Output: (LIST: *Prominent_Aspects*)
Initialization :
Extract $\leftarrow$ *Tagged_Sentences* $\Rightarrow$ "*Nouns, Verbs, Adjectives* : *NVA* "
Count $\leftarrow$ *NVA* $\Rightarrow$ "*Freq_Dist*"
Extract $\leftarrow$ *Vectors_Clusters* $\Rightarrow$ (*LIST* : *Cluster_Labels*)
Calculate $\leftarrow$ *Vectors_Clusters* $\Rightarrow$ "*Distance_Space*"
Create $\leftarrow$ (*LIST* : {*Label* : (*LIST* : *distance*)}) $\Rightarrow$ "*Cluster_Distance*"
for each Label $\in$ *Cluster_Labels* **do**
    Extract $\leftarrow$ *Distance_Space* $\Rightarrow$ "*distance*"
    Append $\leftarrow$ *distance* $\Rightarrow$ "*Cluster_Distance*[*Label*]"
end for
threshold = *min*
Create $\leftarrow$ (*LIST* : *Prominent_Aspects*)
for each Label $\in$ *Cluster_Distance* **do**
    Sort $\leftarrow$ *items* $\in$ "*Cluster_Distance*[*Label*]" $\Rightarrow$ "*sorted_items*"
    Select $\leftarrow$ *sorted_items* $\Rightarrow$ "*top_items*"
    for each item $\in$ *top_items* **do**
        Convert $\leftarrow$ *Vector: item* $\Rightarrow$ "*word_token*"
        token_freq = *Zero*
        if (`Length(`*word_token*`) > One`) **then**
            for each part $\in$ *word_token* **do**
                Get $\leftarrow$ *Freq_Dist* $\Rightarrow$ "*Freq_Dist*[*part*]"
                token_freq = *token_freq* + *Freq_Dist*[*part*]
            end for
        else
            Get $\leftarrow$ *Freq_Dist* $\Rightarrow$ "*Freq_Dist*[*word_token*]"
            token_freq = *Freq_Dist*[*word_token*]
        end if
        if (*token_freq* >= *threshold*) **then**
            Append $\leftarrow$ *word_token* $\Rightarrow$ "*Prominent_Aspects*"
        end if
    end for
end for
return *Prominent_Aspects*

۴/۵ مجموعه داده

از دو مجموعه داده در این آزمایش استفاده شده است که اِسماً مجموعه داده لپ تاپ ها و مجموعه داده رستوران است {۱۵}. این مجموعه داده در دو فرمت xml. و xlsx. موجود می‌باشد. داده ها شامل متن نظر(یک جمله)، ویژگی نامبرده شده در متن، یک شناسه برای یک نظر(شامل چند جمله) و قطبیت است. ستون شاخصه (Category) بعد از ادغام دو مجموعه داده بوجود می‌آید.

SenID	Review	Feature	Polarity	Category
2210	The last two times I ordered from here my food...	flavor	negative	Restaurant
2401	If you're looking for something to fly through...	game	neutral	Laptops
1929	The food was delicious (I had a halibut specia...	service	positive	Restaurant
2766	The food is delicious - from the specials to t...	regular menu-fare	positive	Restaurant
3578	For the people who want great food plus great ...	food	negative	Restaurant

۴/۶ پیش پردازش

مراحل پیش پردازش به ترتیب زیر می‌باشد:

۱. تبدیل حروف بزرگ به حروف کوچک: این کار بدلیل اینکه دقت در پیاده سازی و تشخیص افزایش یابد، انجام می‌شود. بطور مثال کلمات "Screen" و "screen" در واقع یکسان هستند اما در زمان مقایسه در اینطور نیست. پس برای رفع این مشکل تمام حروف بزرگ به حروف کوچک تبدیل می‌شود.

۲. نشانه‌هایی مانند !"#$٪&'()*+,-./:;<=>?@[\\]^_`{|}~ چه در استخراج ویژگی، چه در تحلیل احساسات نقشی ندارند، پس حذف می‌شوند.

۳. اعداد اگرچه ممکن است در استخراج ویژگی نقش داشته باشند اما برای تحلیل احساسات می‌توان آنها را حذف کرد. البته برای استخراج ویژگی هم بنا به شرایط موجود می‌توان تصمیم گیری کرد.

۴. حذف جملاتی که کمتر از ۱۰ حرف دارند. چرا که عملا نمی‌توان بر روی آنها ادامه‌ی پیش پردازش را جلو برد.

۵. توکن بندی کلمات برای مدل Word2Vec: در اینجا برای توکن بندی از کتابخانه‌ی NLTK با ورژن ۳.۸.۱ استفاده شده است. توکن بندی به منظور پیشبرد مراحل بعدی استفاده می شود.

۶. توکن بندی مناسب مدل BERT: برای این مدل باید طبق آنچه سازندگان گفته اند عمل کرد. در اینجا برای این کار BertTokenizer کتابخانه‌ی transformers با ورژن ۴.۲۸.۱ استفاده شده است.

۷. برای هر توکن یک شناسه قرار داده می‌شود تا تغییرات ایجاد شده در طول پیش پردازش کار شناسایی را مشکل نسازد. بدین شکل یک توکن هرچقدر هم تغییر کند، بازهم با شناسه اش قابل پیگیری ست.

۸. برای پیگیری یک ویژگی در یک جمله نیز یک شناسه‌ی ویژگی مخصوص آن قرار می‌دهیم.

۹. غلط های املائی تصحیح می‌شوند تا با مشکل توضیح داده شده در ۱ روبرو نشویم. مثال ۱ را در نظر بگیرید؛ "screen" و "scren" هر دو در اصل یک کلمه هستند، اما در زمان مقایسه نتیجه این را نمی‌گوید. پس با تصحیح "scren" به "screen" مشکل حل خواهد شد.

۱۰. به این دلیل که نوسنده‌ی {۲} فقط از صفات، اسم‌ها و فعل ها استفاده کرده است، پس نیاز است تا به توکن ها برچسب بزنیم. برای اینکار از کتابخانه NLTK استفاده کرده‌ایم.

۱۱. حذف کلمات توقف: این کلمات they, is, are, I, am,... تاثیری در هر دو بخش استخراج و تحلیل ندارند، پس آنها را حذف می‌کنیم.

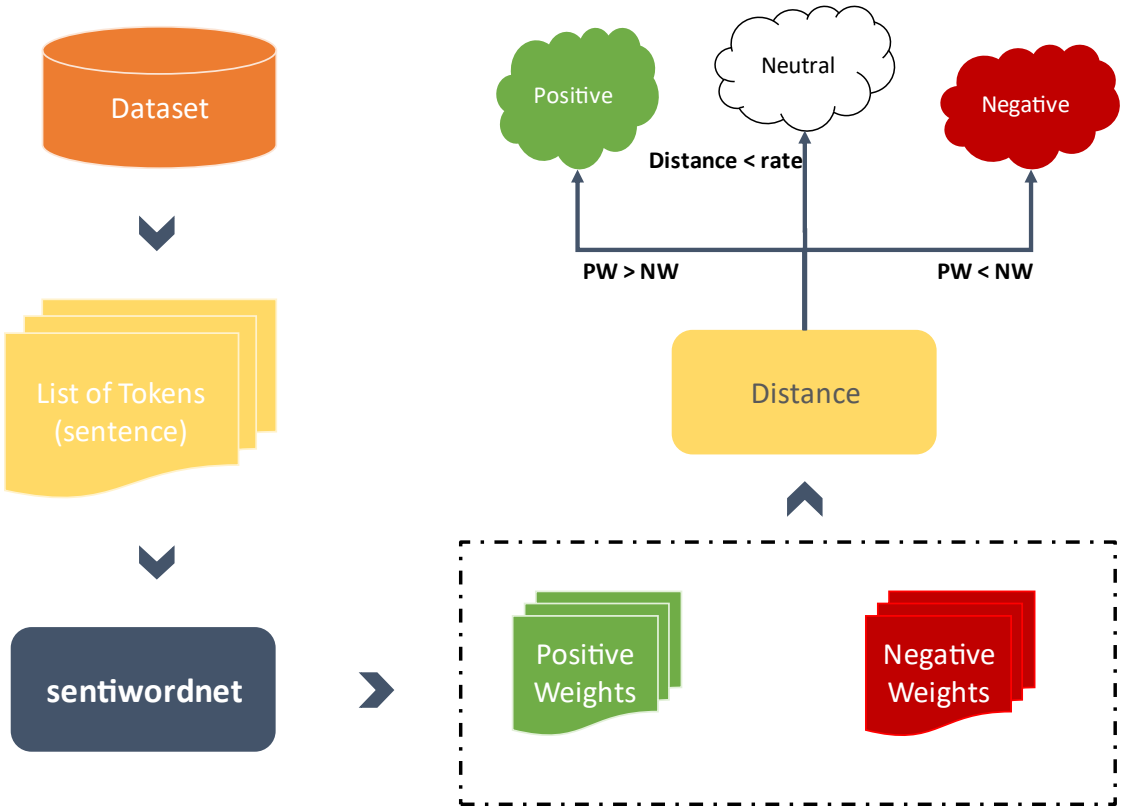
۱۲. ریشه یابی (Stemming): اینکار به این دلیل انجام می‌شود که کلماتی که از یک کلمه واحد ساخته شده‌اند، در واقع یک کلمه در نظر گرفته شوند. مثال ۱ را در نظربگیرید؛ کلمه‌ی "screen" و "screener" هر دو یک ریشه دارند که در کار پردازش بهتر است از ریشه آنها استفاده شود.

۱۳. ریشه یابی (Lemmatization): این مورد هم مانند مورد بالاست اما با این تفاوت که مورد بالا فقط پسوند و پیشوند را حذف می‌کند تا به ریشه برسد و این مورد با جست و جو بدنبال نقش کلمه در جمله به ریشه آن پی می‌برد.

۱۴. همانطور که گفته شد برای مدل BERT باید داده‌ها را آماده سازی کرد سپس بعنوان ورودی به مدل داد. مرحله‌ی BERT-Formatting داده‌ها را طبق دستور العمل سازندگان مدل آماده می‌کند.

۴/۷ تحلیل احساسات

برای تحلیل احساسات، معماری پیشنهاد شده توسط نویسنده، برگرفته از آزمایشی* ست که انجام شده است.



براساس آنچه در قسمت پیش پردازش اتفاق افتاد، در انتها به ازای هر نظر یک لیست از کلمات که حاوی اسم ها، صفت ها و افعال هستند، داریم. با استفاده از این لیست و واژگان قدرتمند sentiwordnet در کتابخانه NLTK، براحتی می‌توانیم به توکن در لیست مورد وزن دهی کنیم. این واژگان برای هر کلمه سه وزن تعیین می‌کند؛ مثبت، منفی و خنثی که ما فقط از وزن های مثبت و منفی استفاده می‌کنیم چرا که وزن خنثی برای کلمات بسیار بیشتر از دو وزن دیگر است. برای حل این چالش راه حلی سنجیده شده که مشکل را حل می‌کند. ابتدا یک مثال از چگونگی وزن دهی این واژگان ارائه می‌دهیم؛ کلمه happy را در نظر بگیرید. واژگان برای این کلمه بصورت زیر وزن های مثبت، منفی و خنثی را تعیین می‌کند.

*این آزمایش در پروژه‌ی درس پردازش زبان طبیعی انجام شده و در لینک زیر موجود است.
https://github.com/valiahmad/Sentiment-Analysis-on-Amazon-Reviews

وزن مثبت: ۰.۸۷۵، وزن منفی: ۰، وزن خنثی: ۰.۱۲۵

همانطور که مشاهده می‌شود جمع سه وزن برابر ۱ می‌باشد. این روش وزن دهی برای همه‌ی کلمات دیگر همین گونه است.

حال که با طریقه‌ی وزن دهی واژگان آشنا شدیم، همه‌ی کلمات را وزن دهی می‌کنیم. برای یک نظر در حال حاضر هر کلمه یا توکن یک وزن مثبت و یک وزن منفی دارد. از میانگین این وزن ها به دو مقدار می‌رسیم که یکی نشان دهنده‌ی وزن مثبت جمله است و دیگری نشان دهنده‌ی وزن منفی آن. چالش اصلی در اینجا معلوم می‌شود، می‌توان گفت که از قدرمطلق تفریق این دو وزن به نتیجه مطلوب می‌رسیم، اما اگر دو وزن به یکدیگر خیلی نزدیک باشند، یعنی مقدار قدرمطلق تفریق آنها نزدیک به صفر باشد، آیا این احتمال وجود ندارد که این نظر یه نظر خنثی ست؟ برای حل این مشکل ما یک نرخ (rate) تعیین کرده ایم. بدین گونه که اگر قدرمطلق تفریق دو وزن مثبت و منفی کمتر از حدآستانه معین شده بود، نظر را برچسب خنثی می‌زنیم. اگر این طور نبود، نظر آن برچسبی می‌گیرد که وزن غالب داشته باشد. بنابراین باید با آزمایش و خطا به این نرخ دست پیدا کرد، سپس نرخ بهینه را برای طبقه بندی برگزید.

۵ ارزیابی

۵/۱ استخراج ویژگی

برای ارزیابی ویژگی های استخراج شده طبق فرمول های زیر عمل می‌کنیم.

دقت (precision) = تعداد ویژگی‌هایی که به درستی تشخیص داده شده تقسیم بر تعداد کل ویژگی‌های تشخیص داده شده.

نرخ یادآوری (recall) = تعداد ویژگی‌هایی که به درستی تشخیص داده شده تقسیم بر تعداد کل ویژگی‌های موجود در مجموعه داده.

چندین مدل برای آزمایش طراحی شده است که به قرار زیر است:

1. W-S-K: Word2Vec → SOM → K-Means++
2. W-t-K: Word2Vec → t-SNE → K-Means++
3. Bb-S-K: BERT-Base → SOM → K-Means++
4. Bl-S-K: BERT-Large → SOM → K-Means++
5. Bb-t-K: BERT-Base → t-SNE → K-Means++
6. Bl-t-K: BERT-Large → t-SNE → K-Means++

طبق آزمایشات انجام شده در {۲} بهترین نتیجه برای مدل Bl-S-K بوده است. که برای داده‌های لپ تاپ ها مقدار k ۱۵۰ و برای داده‌های رستوران مقدار k ۲۰۰ انتخاب شده است.

۵/۲ تحلیل احساسات

در تحلیل احساسات تمرکز فقط بر روی بالا بردن precision و recall است. هرچه مقدار این دو بالاتر باشد، مدل عملکرد بهتر را نشان می‌دهد. پس ما برای ارزیابی مدل خود فقط از این دو بعلاوه f1-score که از این دو بدست می‌آید، استفاده می‌کنیم.

برای ارزیابی تحلیل احساسات فرمول های زیر مطرح است:

- Precision = $\frac{tp}{tp+fp}$
- Recall = $\frac{tp}{tp+fn}$
- F1-score = $\frac{2*Precision*Recall}{Precision+Recall}$

برای پیاده سازی از کتابخانه scikit-learn استفاده شده است و کلاس های مثبت، منفی و خنثی مورد ارزیابی قرار گرفته اند.

۶ نتایج

همانطور که گفته شد بخش استخراج ویژگی قبلا توسط {۲} انجام و ارزیابی شد که نتیجه آن نشان داد که مدل Bb-S-K بهترین دقت را دارد. تمرکز ما در اینجا بروی تحلیل احساسات است، بنابراین نتایج زیر براساس آنچه در بخش ۴ در مورد معماری تحلیل احساسات گفته شد، بدست آمده است.

rate	**0.01**			**0.005**			**0.003**		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	**F1**
Negative	**0.7**	0.06	0.11	0.65	**0.08**	**0.15**	0.62	0.08	**0.15**
Neutral	0.16	**0.14**	**0.15**	**0.22**	0.10	0.14	0.19	0.06	**0.09**
Positive	**0.56**	0.9	0.69	0.55	0.93	**0.69**	0.55	**0.94**	**0.69**

rate	**Precision (mean)**	**Recall (mean)**	**F1-score (mean)**
0.01	0.473	0.367	**0.316**
0.005	**0.474**	**0.372**	0.326
0.003	0.452	0.36	**0.30**

۷ بحث و نتیجه گیری

براساس آنچه در قسمت های قبل گفته شد، تحلیل احساسات با مدل پیشنهادی برای نرخ ۰.۰۰۵ بهترین نتیجه را داشته که این موضوع نشان می‌دهد که قدرمطلق تفریق وزن‌های مثبت و منفی یک نظر هرگاه بیشتر از مقدار آستانه‌ی ۰.۰۰۵ باشند، می‌توانند برای مثبت یا منفی شدن رقابت کنند. اما اگر کمتر از این مقدار باشد، نظر بعنوان یک نظر خنثی انتخاب می‌شود. دلیل این که دقت ها در سطح بالا نیستند این که ما فقط بروی بخشی از داده پردازش را انجام داده‌ایم. یعنی ما فقط افعال، صفات و اسم را در نظر گرفته و مابقی را بعلاوه ساختار جمله کنار زده‌ایم. اگر چه که این معماری ساده است اما می‌توان از آن برای طبقه بندی در سطح چند کلاسه(بیش از سه کلاس) استفاده کرد*.

*نویسنده از این ایده در پروژه مذکور برای بالا بردن دقت در سطح طبقه بندی پنج کلاسه استفاده کرده است.

منابع

1. www.hamyarit.com
2. Ali, N.M.; Alshahrani, A.; Alghamdi, A.M.; Novikov, B. “Extracting Prominent Aspects of Online Customer Reviews: A Data-Driven Approach to Big Data Analytics.” Electronics 2022, 11, 2042.https://doi.org/10.3390/electronics11132042.
3. Mohammadi, S., Nazemi, E., (2021). Sentiment Analysis at the Product Feature Level and Based on Users Gender, Journal of Business Intelligence Management Studies, 10(37), 267-296.
4. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient Estimation of Word Representations in Vector Space. arXiv 2013, arXiv:1301.3781v3.
5. Kohonen, T. Self-Organized Formation of Topologically Correct Feature Maps. Biol. Cybern. 1982, 43, 59–69.
6. Kohonen, T. The Self-Organizing Map. Proc. IEEE 1990, 78, 1464–1480.
7. Kohonen, T. Self-Organizing Maps, 1st ed.; Springer Series in Information Sciences; Springer: Berlin/Heidelberg, Germany, 1995; Volume 30.
8. Kohonen, T. Essentials of the Self-Organizing Map. Neural Netw. 2013, 37, 52–65.
9. Riese, F.M.; Keller, S.; Hinz, S. Supervised and Semi-Supervised Self Organizing Maps for Regression and Classification Focusing on Hyperspectral Data. Remote Sens. 2020, 12, 7.
10. Maaten, L.v.d. Learning a Parametric Embedding by Preserving Local Structure. In Proceedings of the 2009 Artificial Intelligence and Statistics, Clearwater Beach, FL, USA, 16–18 April 2009; pp. 384–391.
11. Maaten, L.V.D. Accelerating t-SNE using Tree-Based Algorithms. JMLR 2014, 15, 3221–3245.
12. Maaten, L.V.D.; Hinton, G. Visualizing Data using t-SNE. JMLR 2008, 9, 2579–2605.
13. Maaten, L.V.D.; Hinton, G. Visualizing Non-Metric Similarities in Multiple Maps. Mach. Learn. 2012, 87, 33–55.
14. Arthur, D.; Vassilvitskii, S. k-means++: The Advantages of Careful Seeding. In Proceedings of the Eighteenth Annual ACM-SIAM Symposium on

Discrete Algorithms, New Orleans, LA, USA, 7–9 January 2007; Volume 1, pp. 1027–1035.
15. https://www.kaggle.com/datasets/charitarth/semeval-2014-task-4-aspectbasedsentimentanalysis
16. W. Hu, Z. Gong and J. Guo, "Mining Product Features from Online Reviews," 2010 IEEE 7th International Conference on E-Business Engineering, Shanghai, China, 2010, pp. 24-29, doi: 10.1109/ICEBE.2010.51.
17. D. T. Santosh, B. V. Vardhan and D. Ramesh, "Extracting Product Features from Reviews Using Feature Ontology Tree Applied on LDA Topic Clusters," 2016 IEEE 6th International Conference on Advanced Computing (IACC), Bhimavaram, India, 2016, pp. 163-168, doi: 10.1109/IACC.2016.39.
18. OJOKOH, B. A. ., & KAYODE, O. . (2012). A FEATURE–OPINION EXTRACTION APPROACH TO OPINION MINING. Journal of Web Engineering, 11(1), 051–063. Retrieved from https://journals.riverpublishers.com/index.php/JWE/article/view/4227
19. Changqin Quan, Fuji Ren, Unsupervised product feature extraction for feature-oriented opinion determination, Information Sciences, Volume 272, 2014, Pages 16-28, ISSN 0020-0255, https://doi.org/10.1016/j.ins.2014.02.063.
20. Ali, N.M.; Alshahrani, A.; Alghamdi, A.M.; Novikov, B. Extracting Prominent Aspects of Online Customer Reviews: A Data-Driven Approach to Big Data Analytics. Electronics 2022, 11, 2042. https://doi.org/10.3390/electronics11132042
21. Kang, Yin and Zhou, Lina, "Extracting Product Features from Online Consumer Reviews" (2013). AMCIS 2013 Proceedings. 2. https://aisel.aisnet.org/amcis2013/IntelligentSystems/GeneralPresentations/2
22. Ahmad, Siti Rohaidah, Bakar, Azuraliza Abu, and Yaakub, Mohd Ridzwan. 'Ant Colony Optimization for Text Feature Selection in Sentiment Analysis'. 1 Jan. 2019 : 133 – 158.