
بسم الله الرحمن الرحيم
الرحمن الرحيم الرحمن الرحيم

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

رشته مهندسی کامپیوتر گرایش هوش مصنوعی

پایان‌نامه کارشناسی ارشد

تشخیص احساسات در متن با استفاده از تکنیک‌های هوش مصنوعی

نگارنده: اویس ارغیانی

استاد راهنما

دکتر مرتضی زاهدی

شهریور 95

دانشکده: مهندسی کامپیوتر و فناوری اطلاعات

گروه: مهندسی کامپیوتر- هوش مصنوعی

پایان‌نامه کارشناسی ارشد آقای اویس ارغیانی به شماره دانشجویی: 9201854

تحت عنوان:

تشخیص احساسات در متن با استفاده از تکنیک‌های هوش مصنوعی

در تاریخ ..... توسط کمیته تخصصی زیر جهت اخذ مدرک کارشناسی ارشد مورد ارزیابی و با درجه ..... مورد پذیرش قرار گرفت.

اساتید راهنما	امضاء	اساتید مشاور	امضاء
نام و نام خانوادگی :		نام و نام خانوادگی :	
نام و نام خانوادگی :		نام و نام خانوادگی :	

اساتید داور	امضاء	نماینده تحصیلات تکمیلی	امضاء
نام و نام خانوادگی :		نام و نام خانوادگی :	
نام و نام خانوادگی :			
نام و نام خانوادگی :			
نام و نام خانوادگی :			

این پایان نامه را تقدیم می دارم به

پدر و مادر عزیز و مهربانم

که در سختی ها و دشواری های زندگی همواره یاوری دلسوز، فداکار و پشتیبانی محکم و مطمئن برایم بوده اند.

و همسرم

که سایه مهربانیش سایه سار زندگیم می باشد، او که اسوه صبر و تحمل بوده و مشکلات مسیر را برایم تسهیل نمود.

سپاس و ستایش خداوند را که هر چه هست، همه از مهر اوست.

و سپاسگزارم از اساتید گرامی و بزرگوار

جناب آقای دکتر مرتضی زاهدی که در کمال سعه‌صدر، با حسن خلق و فروتنی، زحمت راهنمایی این پایان‌نامه را بر عهده گرفتند.

و از استادان گرامی، جناب آقای دکتر وحید ابوالقاسمی و آقای دکتر منصور فاتح که زحمت داوری این پایان‌نامه را متقبل شدند، کمال تشکر را دارم.

تعهد نامه

اینجانب اویس ارغیانی دانشجوی دوره کارشناسی ارشد رشته مهندسی کامپیوتر- هوش مصنوعی دانشکده مهندسی کامپیوتر و فناوری اطلاعات دانشگاه صنعتی شاهرود نویسنده پایان‌نامه **تشخیص احساسات در متن با استفاده از تکنیک‌های هوش مصنوعی** تحت راهنمائی آقای دکتر مرتضی زاهدی متعهد می‌شوم.

- تحقیقات در این پایان‌نامه توسط اینجانب انجام شده‌است و از صحت و اصالت برخوردار است.
- در استفاده از نتایج پژوهش‌های محققان دیگر به مرجع مورد استفاده استناد شده‌است.
- مطالب مندرج در پایان‌نامه تاکنون توسط خود یا فرد دیگری برای دریافت هیچ نوع مدرک یا امتیازی در هیچ جا ارائه نشده‌است.
- کلیه حقوق معنوی این اثر متعلق به دانشگاه صنعتی شاهرود می‌باشد و مقالات مستخرج با نام « دانشگاه صنعتی شاهرود » و یا « Shahrood University of Technology » به چاپ خواهد رسید.
- حقوق معنوی تمام افرادی که در به دست آمدن نتایج اصلی پایان‌نامه تأثیرگذار بوده‌اند در مقالات مستخرج از پایان‌نامه رعایت می‌گردد.
- در کلیه مراحل انجام این پایان‌نامه، در مواردی که از موجود زنده (یا بافت‌های آنها) استفاده شده‌است ضوابط و اصول اخلاقی رعایت شده‌است.
- در کلیه مراحل انجام این پایان‌نامه، در مواردی که به حوزه اطلاعات شخصی افراد دسترسی یافته یا استفاده شده‌است اصل رازداری ، ضوابط و اصول اخلاق انسانی رعایت شده‌است.

تاریخ

امضای دانشجو

مالکیت نتایج و حق نشر

- کلیه حقوق معنوی این اثر و محصولات آن (مقالات مستخرج، کتاب، برنامه‌های رایانه‌ای، نرم‌افزارها و تجهیزات ساخته شده) متعلق به دانشگاه صنعتی شاهرود می‌باشد. این مطلب باید به نحو مقتضی در تولیدات علمی مربوطه ذکر شود.
- استفاده از اطلاعات و نتایج موجود در پایان‌نامه بدون ذکر مرجع مجاز نمی‌باشد.

* متن این صفحه نیز باید در ابتدای نسخه‌های تکثیر شده پایان‌نامه وجود داشته باشد.

چکیده

عاطفه جنبه بسیار مهمی از رفتار انسان است که از طریق آن، مردم در جامعه روی هم تأثیر می‌گذارند. تشخیص احساسات برای ارائه نشانه و تعامل بیشتر بین انسان و کامپیوتر است. علاوه بر تشخیص احساسات از روی چهره، حرکات و گفتار، می‌توانیم احساسات را از روی متن نوشته شده نیز تشخیص دهیم. تشخیص احساسات یا Sentiment Analysis شاخه‌ای از علوم کامپیوتر و پردازش زبان طبیعی (NLP) است که می‌تواند به یافتن انگیزه نویسنده کمک کند.

هدف از این تحقیق با عنوان «تشخیص احساسات در متن با استفاده از تکنیک‌های هوش مصنوعی» دسته‌بندی متن‌های احساسی و کشف حالت عاطفی نویسنده متن، در جهت توسعه سیستمی است که به اندازه کافی هوشمند باشد و به تعامل با انسان از جمله احساسات پرداخته و بتواند احساسات کاربر را تشخیص دهد.

مدل پیشنهادی در این تحقیق خصیصه‌های استخراج شده از متن را در دو قالب 1-gram و 2-gram در نظر می‌گیرد. همچنین برای کاهش پراکندگی خصیصه‌ها و در نتیجه افزایش دقت دسته‌بندی و بهبود نتایج، از روش فیلترکردن خصیصه‌ها استفاده می‌شود. مدل پیشنهادی با استفاده از پایگاه‌داده متن‌های احساسی که برای این منظور ساخته شده، آزمایش می‌شود. متن‌های احساسی با استفاده از روش بیز در پنج دسته احساسی غم، شادی، عصبانیت، تنفر و ترس طبقه‌بندی می‌شوند. نتایج این آزمایش‌ها نشان می‌دهد که دقت تشخیص احساس، با استفاده از خصیصه‌های 1-gram و 2-gram بدون فیلترکردن خصیصه‌ها، در بهترین حالت به ترتیب 79.1 درصد و 27.7 درصد می‌باشد. همچنین با فیلترکردن خصیصه‌ها و در نتیجه کاهش پراکندگی آنها، دقت 91.6 درصد و 43 درصد به ترتیب برای خصیصه‌های 1-gram و 2-gram بدست می‌آید.

کلمات کلیدی: تشخیص احساسات، متن احساسی، دسته‌بندی، پراکندگی خصیصه‌ها، فیلترکردن خصیصه‌ها

فهرست مطالب

فهرست شکل‌ها

فهرست جدول‌ها

فصل 1) پیش‌گفتار

1- 1- مقدمه

احساسات[1] بخشی از زندگی انسان است که بیشتر از عوامل دیگر بر روی تصمیم‌گیری تأثیر می‌گذارد. احساسات از دیرباز عاملی کلیدی در نحوه برقراری ارتباط و شکل‌گیری تعاملات بشر بوده‌است. زبان یک ابزار قدرتمند برای برقراری ارتباط و انتقال اطلاعات و همچنین وسیله‌ای برای ابراز احساسات است. عاطفه جنبه بسیار مهمی از رفتار انسان است که از طریق آن، مردم در جامعه روی هم تأثیر می‌گذارند. امروزه با گسترش نوع جدیدی از این ارتباطات یعنی گونه مجازی، تعاملات افراد بیشتر در یکی از قالب‌های تصویری، صوتی و یا تبادل پیام‌های نوشتاری انجام می‌شود.

تشخیص احساس[2] برای ارائه نشانه و تعامل بیشتر بین انسان و کامپیوتر است. از دیرگاه کلمه‌های "ربات[3]" و "کامپیوتر" به سرد، خشک و بی‌احساس بودن شناخته شده‌اند، ولی می‌توان روزی را تصور کرد که این شناخت به کلی متحول شود. اهمیت این مساله جایی مشخص می‌شود که به اطرف خود بنگریم و دریابیم همین حالا هم بسیاری از روابط و برخوردهای ما با هوش مصنوعی است. از کارهای روزمره مثل استفاده از کامپیوتر، تلفن همراه و سایت‌های مختلف گرفته تا حتی کارهای صنعتی و علمی.

علاوه بر تشخیص احساس از روی چهره، حرکات و گفتار، می‌توانیم احساسات را از روی متون نوشته شده نیز تشخیص دهیم. متون اغلب نمایانگر حالت عاطفی نویسنده هستند. بدین ترتیب متن، نه‌تنها کاربردش کم نشده بلکه همچنان جایگاه خود را به عنوان روشی موثر در برقراری ارتباط حفظ کرده‌است. ماهیت احساسات عاطفی نوشتار را می‌توان به راه‌های مختلف تفسیر کرد و آن را با مدل‌های مختلف محاسباتی نشان داد.

---

[1] Sentiment
[2] Sentiment Analysis (SA)
[3] Robot

تشخیص احساس شاخه‌ای از علوم کامپیوتر و پردازش زبان طبیعی[1] است که سعی دارد ماشین و هوش مصنوعی را با احساس و عواطف انسانی آشنا سازد و تشخیص آنها را میسر سازد. بدیهی است که تشخیص احساس می‌تواند این ماشین‌ها را به بشر نزدیک‌تر ساخته و آنها را قادر به کمک‌رسانی هر چه بیشتر به انسان می‌کند.

در این پایان‌نامه قصد داریم به کمک تکنیک‌های هوش مصنوعی روشی ارائه دهیم که حالت روحی نویسنده یک متن را بررسی و تشخیص دهد. اگر بتوانیم سیستمی ارائه کنیم که به اندازه کافی هوشمند باشد و به تعامل با انسان از جمله احساسات بپردازد، می‌تواند احساسات کاربر و در نتیجه تغییر رفتار کاربر را براساس احساساتش با استفاده از این ماشین تشخیص بدهد.

1- 2- تعریف احساس

هنگامی که احساس خاصی به ما دست می‌دهد، در آن لحظه مکث نمی‌کنیم تا آن احساس را معنی کرده و یا در مورد احساس دقیقی که با آن روبرو هستیم، تأمل کنیم. خواه احساس ما غم[2]، شادی[3] و یا عصبانیت[4] باشد، فرقی نمی‌کند، ما فقط آن را حس کرده و با آن روبرو می‌شویم. ما انسان‌ها، مجموعه‌ای از احساسات و عواطف را در طول عمر خود تجربه می‌کنیم که چندین فرم و نوع را شامل می‌شوند. روانشناسان، پنج نوع احساس اولیه و یا مهم را نام برده‌اند: ترس[5]، شادی، تنفر[6]، غم و عصبانیت.

---

[1] Natural Language Processing (NLP)
[2] Sadness
[3] Happiness
[4] Anger
[5] Fear
[6] Disgust

ما همراه با احساسات اولیه، احساسات ثانویه را نیز که واکنشی مستقیم از احساسات اولیه می‌باشند تجربه می‌کنیم. به عنوان مثال، فردی ممکن است پس از تجربه احساس ترس، احساس شرمندگی یا گناه به او دست دهد. روانشناسان همچنین عنوان می‌کنند که انسان نه‌تنها احساسات اولیه و ثانویه را تجربه می‌کند، بلکه احساسات ثالثیه را نیز تجربه خواهد کرد. تعداد این احساسات اولیه همراه با احساسات ثانویه و ثالثیه‌ی وابسته به آنها به بیش از 30 مورد می‌رسد [1].

1- 3- ضرورت انجام تحقیق

امروزه محتواهای متنی دیجیتال با انگیزه‌های مختلفی ایجاد می‌شوند. تشخیص احساس متن می‌تواند به یافتن انگیزه نویسنده کمک کند. موارد استفاده از تشخیص احساس متن در روانشناسی، علوم اجتماعی و ارتباطات است. تشخیص احساس متن می‌تواند مورد علاقه سیاست‌گذاران، اقتصاددانان، محققانِ بازار، تحلیل‌گرانِ سیاست و دانشمندانِ علوم اجتماعی قرار گیرد. سیستم تشخیص احساس متن می‌تواند حس عاطفی ورودی را درک کرده که حائز اهمیت در سیستم‌هایی است که حالت عاطفی در آنها مهم است؛ مثل بازی، آموزش برخط[1]، احراز هویت کاربران، ارتباطات برخط، ساخت ربات‌های هوشمند، بررسی محصول و توسعه برنامه کاربردی احساسات.

تشخیص احساس ماشین‌های فعلی را یک نسل به جلو رانده و باور عمومی بر بی‌احساس بودن ماشین‌ها را کمرنگ‌تر می‌سازد و در پی آن باعث آسان‌تر شدن بسیاری از جهات زندگی انسان نیز خواهد شد. لذا طراحی یک روش خودکار برای تحلیل متن و استخراج نظرات و عقاید موجود در متن ضروری است. در همین راستا تلاش‌های فراوانی صورت گرفته‌است، مثلا در کشور آمریکا 20 تا 30 شرکت به ارائه خدمات تخصصی تحلیل احساس می‌پردازند [2].

[1] Online

1- 4- کاربردهای تشخیص احساس

شبکه اجتماعی[1] را تصور کنید که لحن و احساس شما را از مطالب روزانه‌ای که ارسال کردید تشخیص می‌دهد و شما را با افرادی با حس مشابه در تماس قرار می‌دهد تا با هم بر سر مشکل مشترکتان (مثلا حقوق کم) صحبت کنید! همچنین اگر شبکه اجتماعی، دارای جامعه بزرگی باشد، اطلاعات جمع‌آوری شده از آن برای امور آماری - روان‌شناسی بی‌نظیر خواهد بود. مثلا روانشناسان می‌توانند شروع یک اپیدمی افسردگی در یک منطقه خاص را تشخیص دهند.

برای نمونه ربات‌های تخصیص مشتری که امروزه در فروشگاه‌های زنجیره‌ای بزرگ کاربردی شده‌اند را تصور کنید که احساسات مشتری را تشخیص داده، مشتری عصبانی و پرخاشگر را سریعا به مدیریت ارجاع می‌دهند و مشتریان دیگر را به نوبت به باجه‌های تسویه می‌فرستند. تلفن همراهی را تصور کنید که براساس مکالمات شما با افراد، تشخیص می‌دهد که امروز، روز دشواری داشته‌اید و به‌طور خودکار تماس کسانی که سابقه بیشترین دعوای لفظی دارند را مسدود[2] می‌کند تا شما در آرامش باشید.

تشخیص احساس در زمان حال بیشترین کاربرد را در سایت‌های عرضه انواع کالا دارد. مردم از طریق وبلاگ[3]، احساسات خودشان را با عموم مردمی که ناشناس هستند به اشتراک می‌گذارند. در مورد انواع محصولات، نظرات مشتریان می‌تواند در تصمیم‌گیری آگاهانه در مورد محصول کمک کند. بسیار مهم است که مدیر سایت و شرکت عرضه‌کننده محصول بدانند کدام نقد و بررسی‌ها از یک محصول مثبت هستند و کدام منفی و اینکه شدت منفی یا مثبت بودن چقدر است. کاربر از محصول، کمی ناراضی است یا شدیدا؟

---

[1] Social Network
[2] Block
[3] Weblog

شرکت‌های بزرگی همچون آمازون[1] و گوگل[2] از تشخیص احساس برای بررسی نظرات کاربران استفاده می‌کنند. این شاخه از تشخیص احساس را استخراج عقیده[3] می‌گویند [3]. تشخیص احساس همچنین می‌تواند به تولیدکنندگان برای داشتن نظرات مشتریان در مورد ویژگی‌های ساختن محصول کمک کند. به‌طور کلی نظرسنجی هم برای افراد و هم برای سازمان‌ها و کسب‌وکار مهم است.

1- 5- تاریخچه تحقیق

در زمینه تشخیص احساس، تاکنون روش‌های مختلفی پیشنهاد شده که در دو گروه طبقه‌بندی می‌شوند:

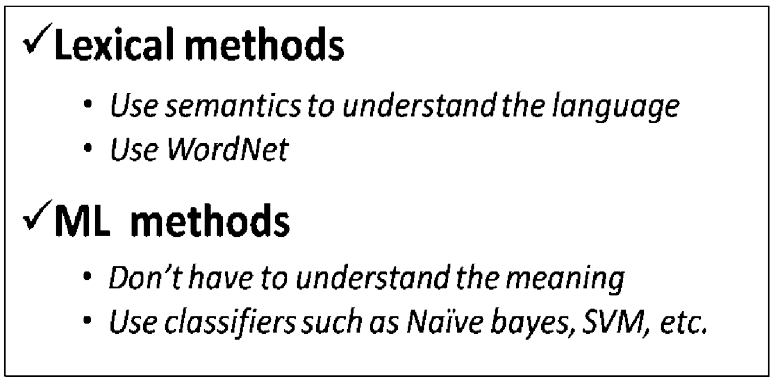
1- روش‌های مبتنی بر یادگیری ماشین[4] و استفاده از دسته‌بندها[5] مانند: ماشین بردار پشتیبان[6]، بیز ساده[7] و بیشترین بی‌نظمی[8]. این روش‌ها نیازی به درنظرگرفتن ساختار گرامری جمله ندارند و باید به‌طور مناسبی با یک مجموعه یادگیری[9]، آموزش[10] داده شوند.

2- روش‌های وابسته به فرهنگ لغات که تکامل یافته‌ی روش لغوی[11] ساده و براساس شباهت معنایی هستند. ضعف مهم این روش‌ها وابستگی به واژه‌نامه‌های لغوی - معنایی مانند وردنت[12] می‌باشد.

---

[1] Amazon
[2] Google
[3] Opinion Mining
[4] Machine Learning (ML)
[5] Classifier
[6] Support Vector Machine (SVM)
[7] Naïve Bayes (NB)
[8] Maximum Entropy (ME)
[9] Learning Set
[10] Train
[11] Lexical
[12] WordNet

در فصل 2 به تفصیل درباره روش‌های فوق بحث خواهد شد. شکل 1-1 تفاوت رویکردهای تحلیل احساس را نشان می‌دهد.



شکل 1- 1 تفاوت رویکردهای تحلیل احساس

1- 6- اهداف تحقیق

تاکنون تحقیقات فراوانی به‌منظور تحلیل احساس در زبان‌های انگلیسی، چینی، عربی و روسی انجام شده‌است. به دلایلی مانند کمبود منابع داده‌ای و همچنین پیچیدگی‌های ذاتی زبان فارسی، کارهای تحقیقی اندکی به‌منظور تحلیل احساس در زبان فارسی انجام شده‌است.

هدف از این تحقیق ارائه روشی برای تحلیل احساس در مجموعه‌ای از متون فارسی می‌باشد، به‌گونه‌ای که متن‌ها در پنج گروه احساسیِ غم، شادی، عصبانیت، تنفر و ترس طبقه‌بندی شوند. از جمله مشکلاتی که برای تحلیل احساس وجود دارد حجم زیاد متون است. همچنین خصیصه[1]های زبان شناختی فراوانی وجود دارد که باید از این میان بهترین خصیصه‌ها را بیابیم و برای مدل‌سازی متون از آنها استفاده کنیم.

[1] Features

در این رساله تلاش بر آن بوده تا بتوانیم از یک الگوریتم با پیچیدگی زمانی کم استفاده کنیم. استفاده از روش‌های معنایی برای انجام این تحقیق مستلزم به‌کارگیری واژه‌نامه فارسی و در صورت عدم وجود، ساخت آن است. ایجاد واژه‌نامه به علت وجود قواعد معنایی متعدد، وقت و هزینه زیادی صرف می‌کند. از طرف دیگر روش‌های مبتنی بر یادگیری ماشین نیازی به قواعد زبان مورد نظر و نیز ساختار جمله ندارند. لذا با توجه به فارسی بودن متون در این تحقیق از روش‌های مبتنی بر یادگیری ماشین استفاده می‌شود.

1- 7- ساختار پایان‌نامه

در فصل دومِ این تحقیق به طبقه‌بندی و بررسی شیوه‌های موجود در تحلیل احساس و بررسی روش‌های نوین ارائه شده در این زمینه پرداخته شده و همچنین کارهای مرتبط و شیوه‌های مورد استفاده در آنها بیان می‌شود. در فصل سوم، روش‌های پیشنهادی برای تحلیل احساس؛ در راستای افزایش دقت و کاهش زمان طبقه‌بندی ارائه شده‌است. در فصل چهارم، روش پیشنهادی، پیاده‌سازی شده و نتایج ارزیابی می‌شود و در فصل پنجم نتیجه‌گیری کلی و برخی از کارهایی که در آینده می‌توان انجام داد بیان شده‌است.

فصل 2) پیشینه تحقیق

2- 1- مقدمه

در این فصل ابتدا به بررسی کارهایی که پیش‌تر انجام شده‌است خواهیم پرداخت و نقاط ضعف و قوت هر کدام را بصورت کوتاه بیان می‌کنیم. هر چند تعداد آثار موجود در زمینه تحلیل احساس در متن، فراوان است ولی در این بخش، تعدادی از تحقیقات مهم را بصورت منتخب بیان خواهیم کرد.

برای مدل‌سازی اسناد، باید مجموعه خصیصه‌های مفیدی از متن استخراج کنیم. مجموعه خصیصه‌های مفید، خصیصه‌هایی هستند که به الگوریتم، برای طبقه‌بندی داده‌ها کمک کنند. این خصیصه‌ها باید به نحوی انتخاب شوند که بهترین مدل ممکن از متن را ارائه دهند و باید توجه کنیم هدف این مدل، تحلیل احساس است؛ لذا باید بیشترین اطلاعات ممکن به‌منظور تحلیل احساس را در اختیار نرم‌افزار طبقه‌بندی قرار دهد.

از جمله مسائلی که در این زمینه وجود دارد حجم زیاد داده‌ها می‌باشد. برای نمونه در صفحه توییتر[1] شخصی جاستین بیبر روزانه 300000 نظر ثبت می‌شود [4]. با توجه به این حجم و تعداد متن‌ها، بردار خصیصه حاصل بزرگ خواهد بود، که این امر مشکلاتی را به همراه دارد. از جمله این مشکلات کاهش کارایی و دقت طبقه‌بندی را می‌توان نام برد، لذا باید از روش انتخاب خصیصه استفاده کرد تا بتوان سودمندترین خصیصه‌ها را از میان هزاران خصیصه استخراج کرد.

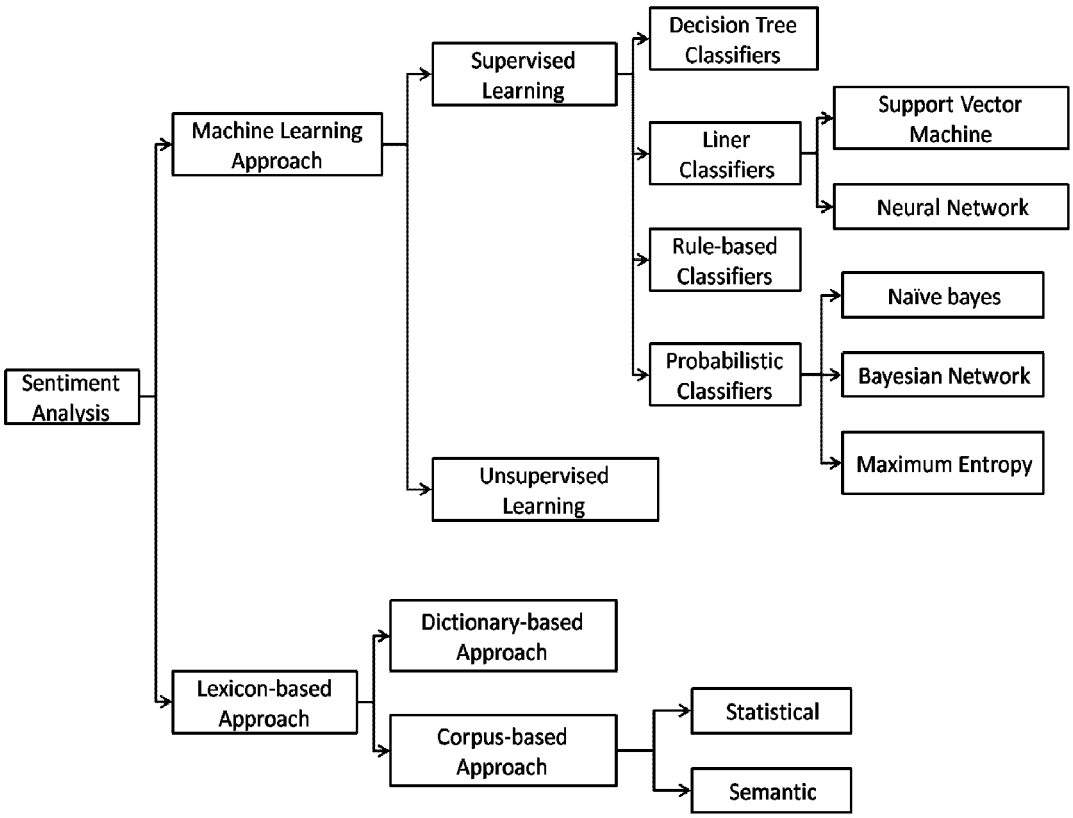
در این قسمت ابتدا تاریخچه تحقیق در زمینه تحلیل احساس در متن را بیان خواهیم کرد، سپس به مجموعه خصیصه‌های N-gram[2] و کارهایی که از آنها استفاده کرده‌اند اشاره می‌شود، پس از آن روش‌های انتخاب خصیصه و تحقیقاتی که از آنها استفاده کرده‌اند بیان خواهد شد.

---

[1] Twitter
[2] A contiguous sequence of **n** items from a given sequence of text or speech.

2- 2- پیشینه تحلیل احساس در متن

سال 2002 پنگ و همکاران تحقیقی را انجام دادند که سرآغاز این راه نامیده می‌شود [5]. هر چند قبل از آن نیز کارهایی انجام شده‌است که به‌طور ضمنی از تحلیل احساسات و عقاید سخن به میان آورده‌اند، ولی اولین بار، پنگ و همکاران به‌طور صریح به تحلیل احساس در متن پرداختند. مهمترین ویژگی این تحقیق ارائه زمینه پژوهشی جدید برای طبقه‌بندی متون بوده‌است. تقسیم‌بندی کلی روش‌های تحلیل احساس در شکل 2-1 نشان داده شده‌است.



شکل 2- 1 دسته‌بندی کلی روش‌های تحلیل احساس

بسیاری از تکنیک‌های تحلیل احساس بر پایه الگوریتم‌های یادگیری نظارت شده[1] هستند، تعدادی روش‌های یادگیری بدون نظارت[2] نیز وجود دارد [6]. در این رساله تمرکز بر روش‌های یادگیری مبتنی بر ناظر است.

2- 3- توضیح خصیصه‌های N-gram

یکی از مهمترین فازهای فرایند تحلیل احساسات و عقاید، مدل‌سازی متون با استفاده از خصیصه‌هایی است که قادرند بخوبی بیان‌کننده صفات اسناد باشند. این رساله بر روی خصیصه‌های N-gram تاکید دارد.

خصیصه‌های N-gram به دو دسته تقسیم می‌شوند:

N-gram ثابت: یک توالی دقیق در سطح کاراکتر یا لغت می‌باشد. مانند 1-gram یا 2-gram.

N-gram متغیر: الگوهایی برای استخراج اطلاعات از متن هستند. مانند < اسم + صفت[3]>

خصیصه‌های N-gram متغیر قادرند مفاهیم پیچیده‌تر زبان شناختی را بیان کنند [7]. علاوه بر این، خصیصه‌های N-POS که ترکیب N تایی از ادات سخن[4] می‌باشند در زمینه تحلیل احساس مورد استفاده قرار می‌گیرند [8]. همچنین N-POS Word ترکیب N تایی از کلمات، به همراه برچسب ادات سخن آنها در برخی تحقیقات به‌کار رفته‌است. تحقیقات اندکی از مدل N-POS Word استفاده کرده‌اند.

---

[1] Supervised
[2] Unsupervised
[3] Noun + Adjective
[4] Post Of Speech (POS)

ویب و همکارانش در سال 2004 به‌منظور کاهش ابهام کلمات در فرایند تحلیل احساس از 3-POS Word استفاده کرده‌اند. با توجه به اینکه خصیصه‌های POS-Tag به همراه خود کلمه می‌تواند باعث کاهش ابهام کلمات شود در نتیجه باعث بهبود دقت ارزیابی و طبقه‌بندی متن‌ها می‌شود [9]. مهمترین دلیل استفاده از 3-POS Word وارد کردن وابستگی به متن در مدل مورد استفاده می‌باشد. لذا اگر بتوان مشکلات ناشی از پراکندگی و افزونگی را مدیریت کرد به نظر می‌رسد استفاده از خصیصه‌های N-gram n>1 به بهبود نتایج کمک زیادی کند. جدول 2-1 مثالی برای هر یک از خصیصه‌های N-gram مطرح شده را نشان می‌دهد.

جدول 2- 1 مجموعه خصیصه‌های N-gram و مثال برای هر خصیصه

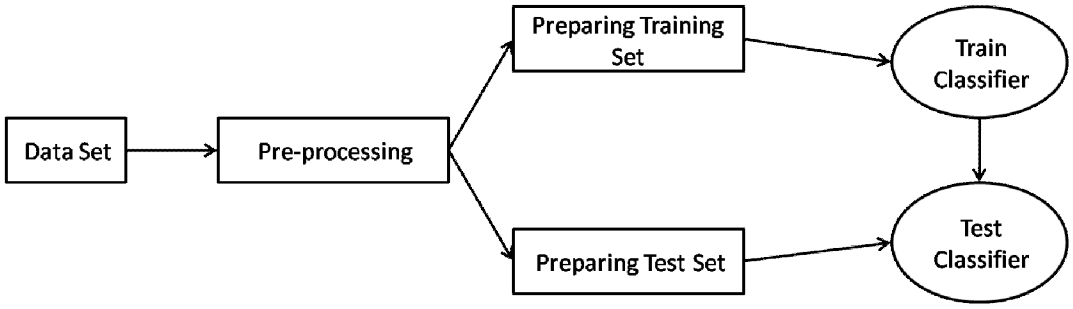
جمله مثال	نوع خصیصه	من عصبانی هستم.
خصیصه‌های N-gram	1-gram	من، عصبانی، هستم
	2-gram	من عصبانی، عصبانی هستم
	3-gram	من عصبانی هستم
خصیصه‌های N-POS	1-pos	FW, VBP,NN
	2-pos	FW VBP,VBP NN
	3-pos	FW VBP NN
خصیصه‌های N-POS Word	3-POS Word	FW VBP NN من عصبانی هستم

2- 4- انواع رویکردها در تحلیل احساس

همان‌طور که گفته شد روش‌های مربوط به تحلیل احساس در دو گروه طبقه‌بندی می‌شوند. در ادامه توضیحات مربوط به هر رویکرد بیان شده و سپس به تعدادی از تحقیقاتی که از این روش‌ها استفاده کرده‌اند اشاره می‌شود.

2-4-1- روش‌های مبتنی بر یادگیری ماشین

دسته‌بندهایی مانند NB، SVM، ME از جمله روش‌هایی هستند که باید به‌طور مناسبی با یک مجموعه یادگیری، آموزش داده شوند. "مناسب" از این لحاظ که اگر مجموعه‌داده[1]های ما دارای موضوعی خاص، مثلا بررسی نقدهای یک هتل باشد؛ مجموعه یادگیری نیز به تبعه باید از همین نوع باشد. استفاده از داده‌های آموزشی نامناسب باعث افت شدیدی در دقت کلاس‌بندی شده و این، اهمیت داده‌های آموزشی مناسب را نشان می‌دهد. شکل 2-2 مراحل اصلی روش‌های مبتنی بر یادگیری ماشین را نشان می‌دهد.



شکل 2- 2 مراحل اصلی روش‌های مبتنی بر یادگیری ماشین

روش بیزِ ساده یک روش قدرتمندِ دسته‌بندیِ مبتنی بر احتمال می‌باشد. در این روش فرض می‌کنیم تعدادی کلاس (دسته) داریم. احتمال وجود داده جدید در هر کدام یک از دسته‌ها را بدست آورده و هر کلاسی که احتمال بیشتری را نتیجه داد، داده خود را به آن کلاس اختصاص می‌دهیم.

پنگ و همکارانش از مجموعه خصیصه‌های 1-gram، 2-gram، صفات، و ترکیبی از این سه نوع مجموعه خصیصه‌ها استفاده کرده‌اند. همچنین برای طبقه‌بندی از الگوریتم‌های SVM، NB و ME بهره گرفته‌اند. روش‌های متفاوتی برای نمایش بردار خصیصه‌ها وجود دارد. پنگ و همکارانش از دو

[1] Data Set

روش فرکانس خصیصه[1] و حضور خصیصه[2] برای نمایش بردار خصیصه‌ها استفاده کرده‌اند. نتایج نشان داد، روش حضور خصیصه نسبت به سایر روش‌های مورد استفاده نتایج بهتری به همراه خواهد داشت. روش‌هایی که آنها برای نمایش بردار خصیصه به‌کار برده‌اند، تاکنون در تحقیقات متفاوت به‌کار گرفته شده‌است. نتایج تحقیق آنها نشان داد خصیصه‌های 1-gram نسبت به سایر خصیصه‌های زبان شناختی، عملکرد بهتری دارند و باعث بهبود طبقه‌بندی می‌شوند. همچنین نتایج طبقه‌بندِ SVM نسبت به سایر الگوریتم‌های طبقه‌بندی دقت بهتری از خود نشان داد. علاوه بر مطالب ذکر شده، آنها مجموعه‌داده‌های بازبینیِ فیلم‌ها را ارائه دادند. این مجموعه‌داده‌ها از سایت IMDB[3] جمع‌آوری شده‌است. مجموعه‌داده بازبینی فیلم‌ها متشکل از 2000 فایل بازبینی فیلم بود که 1000 فایل آن حاوی نظرات مثبتی پیرامون فیلم‌ها و 1000 فایل دیگر نیز حاوی نظرات منفی پیرامون فیلم‌ها بودند. بهترین دقت بدست آمده توسط پنگ و همکارانش به میزان 82.9 درصد با استفاده از 16165 خصیصه 1-gram و در الگوریتم طبقه‌بندی SVM حاصل شده‌است [5].

همچنین نمایش بردار خصیصه ارائه شده در این تحقیق، تاکنون به عنوان یکی از بهترین روش‌های نمایش بردار خصیصه مورد استفاد قرار می‌گیرد. پنگ و همکارانش در این تحقیق بر غیرمفید بودن خصیصه‌های 2-gram و به‌طور کلی خصیصه‌های N-gram n>1 تاکید داشتند.

در تحقیقی که وینسنت و همکارانش سال 2006 انجام داده‌اند خصیصه‌های 1-gram، 2-gram و 3-gram را برای مدل‌سازی اسناد به‌کار بردند [10]. در این تحقیق اسناد متنی به دو دسته حقایق و عقاید دسته‌بندی می‌شوند. اغلب متون حاوی ترکیبی از حقایق و عقاید هستند، بنابراین بیشتر اسناد متنی ترکیبی از متون جهت‌دار (عقاید و نظرات) با متون عینی و واقعی (حقایق) هستند. متون عینی

[1] Feature frequency
[2] Feature presence
[3] Internet Movie Database

و واقعی درون اسناد در واقع همان خصیصه‌های غیرمرتبط با تحلیل احساس هستند چون اطلاعات مفیدی برای الگوریتم یادگیری ماشین در جهت تحلیل احساس موجود در متون را فراهم نمی‌کنند. تعداد زیاد خصیصه‌ها و غیرمرتبط بودن بسیاری از این خصیصه‌ها به تحلیل احساس، مشکلات زیادی را موجب می‌شود. از جمله این مشکلات می‌توان کاهش دقت طبقه‌بندی وکاهش سرعت عملیات طبقه‌بندی را نام برد. بهتر است قسمتی از متن که حاوی حقایق است، در فاز اول از متون حاوی نظرات و عقاید مجزا شود. وینسنت و همکارانش در ابتدا بخش‌هایی از اسناد که عقاید و نظرات را بیان می‌کردند، تشخیص داده، از متن جدا کرده‌اند. آنها با فیلترکردن متون حاوی حقایق از متون احساسی توانستند برای خصیصه‌های 1-gram و 2-gram نتایج بهتری را نسبت به پنگ و همکارانش بدست آورند. همچنین آنها نشان دادند خصیصه‌های N-gram n>1 قادرند وابستگی کلمات موجود در متن را در مدل‌سازی وارد کنند، بنابراین به دقت عملکرد الگوریتم یادگیری ماشین در جهت طبقه‌بندی متون کمک می‌کنند. در این تحقیق دقت حاصله از طبقه‌بندی اسناد با استفاده از خصیصه‌های 1-gram به میزان 87.1 درصد گزارش شده‌است. این میزان نسبت به نتیجه بهترین روش ارائه شده توسط پنگ و همکارانش 5 درصد بهبود یافتـه‌است. همچنین با استفاده از خصیصه‌های 1-gram + 2-gram + 3-gram (ترکیب هر سه نوع خصیصه) فرایند طبقه‌بندی اسناد را با دقت 89.2 درصد انجام داده‌اند. در این تحقیق به بررسی اثر گذاری خصیصه‌های N-gram پرداخته شده‌است.

وینسنت و همکارانش نشان دادند استفاده از خصیصه‌های 2-gram به همراه 1-gram باعث بهبود عملکرد طبقه‌بندی می‌شود. همچنین به این نتیجه دست یافتند که خصیصه‌های 2-gram به‌تنهایی بهبودی در طبقه‌بندی ایجاد نمی‌کنند که دلیل این موضوع نیز پراکندگی خصیصه‌های 2-gram است. بنابراین چنانچه بتوانیم پراکندگی موجود در خصیصه‌های 2-gram را کاهش دهیم می‌توانیم دقت عملکرد این نوع خصیصه‌ها را بهبود دهیم.

گامن در سال 2004 چهار گروه خصیصه را مورد بررسی قرار داد. گروه اول خصیصه‌های N-gram از ترکیب خصیصه‌های 1-gram، 2-gram و 3-gram تشکیل شده‌اند. گروه دوم خصیصه‌های متشکل از ترکیب N-gram و POS بوده‌اند. گروه سوم، خصیصه‌هایی مانند طول جمله، طول عبارات، تعداد کلمات بوده‌اند و گروه چهارم ترکیب سه گروه خصیصه ذکر شده بوده‌اند. تعداد خصیصه‌ها در این روش از 1000 تا 40000 خصیصه بوده‌اند [11]. بهترین دقت حاصله برای طبقه‌بندی متون با استفاده از خصیصه‌های گروه چهارم بدست آمده‌است که نشان می‌دهد ترکیب خصیصه‌ها مدل بهتری از اسناد به‌منظور تحلیل احساس در متن را ارائه می‌دهد. در بهترین حالت دقت طبقه‌بندی 89 درصد گزارش شده‌است. در این تحقیق ترکیب‌های متفاوت از خصیصه‌ها، مورد بررسی قرار گرفت و میزان اثرگذاری آنها بحث شده‌است.

مدل n-gram کاراکترها (N-char) توسط عباسی و همکارانش در سال 2008 مورد استفاده قرار گرفت. مثلا مدل 2-gram عبارت Like بصورت "li ik ke" خواهد بود [12]. در این مدل تعداد بسیار زیاد خصیصه‌ها مشکل‌ساز خواهد بود و استفاده از الگوریتم‌های انتخاب خصیصه به دلیل تعداد بسیار زیاد خصیصه‌ها ما را با مشکل پیچیدگی زمانی روبرو خواهد کرد. استفاده از خصیصه‌های n-char همواره باعث افزونگی و افزایش تعداد خصیصه‌های غیرمفید می‌شود، به این دلیل که همپوشانی بسیار زیادی در خصیصه‌های n-char وجود دارد.

عباسی و همکارانش در سال 2011 مجموعه کاملی از خصیصه‌های N-gram که در کارهای پیشین استفاده شده بود را جمع‌آوری کرده و برای مدل‌سازی اسناد از آنها استفاده کردند [7]. این مجموعه خصیصه‌ها در جدول 2-2 بیان شده‌اند. آنها در این تحقیق با استفاده از طبقه بند SVM به دقت 90 درصد برای طبقه‌بندی مجموعه‌داده‌های بازبینی فیلم‌ها دست یافتند. در مدل ارائه شده که در جدول 2-2 قابل مشاهده است، بسیاری از خصیصه‌ها همدیگر را پوشش می‌دهند لذا باعث تشدید افزونگی در مدل حاصله می‌شوند.

جدول 2- 2 مجموعه کاملی از خصیصه‌های N-gram استفاده شده در [7]

Label	Description	Examples	
N-Char	Character-level n-grams	1-Char	I, L, O, V, E, C, H, O, C, O, L, A
		2-Char	LO, OV, VE, CH, HO, OC, CO, OL
		3-Char	LOV, OVE, CHO, HOC, OCO
N-Word	Word-level n-grams	1-Word	I, LOVE, CHOCOLATE
		2-Word	I LOVE, LOVE CHOCOLATE
		3-Word	I LOVE CHOCOLATE
N-POS	Part-of-speech tag n-grams	1-POS	I, ADMIRE_VBP, NN
		2-POS	ADMIRE_VBP NN
		3-POS	I ADMIRE_VBP NN
N-POSWord	Word and POS tag n-grams	1-POSWord	LOVE ADMIRE_VBP
		2-POSWord	I I LOVE ADMIRE_VBP
		3-POSWord	I I LOVE ADMIRE_VBP CHOCOLATE NN
N-Legomena	Hapax legomena and Dis legomena n-grams	2-Legomena	LOVE DIS
		3-Legomena	I LOVE DIS
N-Semantic	Semantic class n-grams	1-Semantic	SYN-Pronoun, SYN-Affection
		2-Semantic	SYN-Pronoun SYN-Affection
		3-Semantic	SYN-Pronoun SYN-Affection SYN-Candy
IEP-A/E	Information extraction patterns	IEP-A	<possessive> NP, <subj> AuxVP AdjP, <subj> AuxVP Dobj, ActVP <dobj>, ActVP Prep <np>
		IEP-B	<subj> PassVP, InfVP Prep <np>, InfVP <dobj>
		IEP-C	<subj> ActVP
		IEP-D	<subj> ActVP Dobj
		IEP-E	<subj> ActInfVP, <subj> PassInfVP, ActInfVP <dobj>

هر گروه از این خصیصه‌ها دارای تعداد زیادی خصیصه‌های غیرمرتبط با تحلیل احساس موجود در متن هستند، استفاده همزمان از همه این خصیصه‌ها باعث افزایش چشم‌گیر خصیصه‌های غیرمرتبط در نتیجه کاهش اثر گذاری خصیصه‌های مرتبط با تحلیل احساس و در نهایت کاهش دقت طبقه‌بندی می‌شود. آنها برای حل این مشکل یک روش انتخاب خصیصه به نام شبکه ارتباطی خصیصه[1] را ارائه

[1] Feature Relation Network

دادند، که پیچیدگی زمانی بالایی دارد. می‌توان با بهره‌گیری از خصیصه‌های مطلوب‌تر خصیصه‌های افزونه و خصیصه‌های غیرمرتبط را کاهش داد و برای تعیین سودمندی خصیصه‌ها از الگوریتم انتخاب خصیصه ساده‌تر با پیچیدگی زمانی کمتر بهره برد.

همان‌طور که در جدول 2-2 مشاهده می‌شود، مجموعه خصیصه‌های N-gram که برای مدل‌سازی اسناد می‌توان از آنها بهره گرفت بسیار زیاد هستند. هر کدام از این مجموعه خصیصه‌ها، با یک بردار و هزاران خصیصه اسناد را مدل‌سازی می‌کنند. بسیاری از این خصیصه‌ها افزونه و یا با تحلیل احساس غیرمرتبط هستند. برای دستیابی به دقت و سرعت بالاتر در عملیات طبقه‌بندی بهتر است از یک الگوریتم انتخاب خصیصه بهره بگیریم تا بتوانیم سودمندترین خصیصه‌ها را از میان هزاران خصیصه استخراج کنیم و عملیات طبقه‌بندی و تحلیل احساس را با سرعت و دقت بیشتری انجام دهیم.

آگروال و میتال سال 2013 تحقیقی را انجام داده‌اند [13]. در این تحقیق از روش‌های انتخاب خصیصه‌ی سودمندی اطلاعات[1] و حداقل افزونگی - حداکثر وابستگی استفاده شده‌است. همچنین از خصیصه‌های 1-gram و 2-gram و گزیده‌ای از POS-Word در جهت مدل‌سازی متن بهره گرفتند. طبقه‌بندی در این تحقیق بر روی مجموعه‌داده بازبینی فیلم‌ها، دقت بالاتری نسبت به روش ارائه شده توسط عباسی و همکارانش داشت [7]. آنها نشان دادند که روش انتخاب خصیصه حداقل افزونگی - حداکثر وابستگی عملکرد بهتری نسبت به سودمندی اطلاعات دارد. از جمله مشکلات روش ارائه شده توسط آگروال و میتال مجموعه خصیصه‌های مورد استفاده آنها می‌باشد.

تحقیق [13] مدلی ترکیبی از خصیصه‌های N-gram را ارائه داده‌است و این مدل را روی 4 مجموعه‌داده تست کرده تا پایداری روش پیشنهادی خود را بررسی کند. مجموعه خصیصه‌های مورد

[1] Information Gain

استفاده در تحقیق مذکور عبارت بودند از: 1- خصیصه‌های 1-gram 2- خصیصه‌های 2-gram 3- ترکیب این دو مجموعه خصیصه‌ها.

الف- خصیصه‌های 1-gram: در ابتدا بردار خصیصه‌های 1-gram را از متن استخراج کرده و سپس مدل حاصل را با استفاده از الگوریتم‌های یادگیری ماشین طبقه‌بندی کرده‌اند. در این روش آنها به دقت طبقه‌بندی 82.7 درصد در مجموعه‌داده بازبینی فیلم‌ها دست یافتند. در حالت دوم با الگوریتم انتخاب خصیصه حداقل افزونگی – حداکثر وابستگی، خصیصه‌های غیرسودمند را از مدل حاصل در مرحله قبل فیلتر کرده‌اند. در این حالت به حداکثر دقت 89.2 درصد دست یافتند.

ب- خصیصه‌های 2-gram: این مجموعه خصیصه بدون اعمال الگوریتم انتخاب خصیصه از متن استخراج شده‌اند و دقت بدست آمده از طبقه‌بندی 79.2 درصد برای مجموعه‌داده بازبینی فیلم‌ها بوده‌است. در حالت دوم با اعمال الگوریتم انتخاب حداقل افزونگی – حداکثر وابستگی به حداکثر دقت 81.1 درصد دست یافتند.

ج- ترکیب خصیصه‌های 1-gram و 2-gram: این مجموعه خصیصه بدون هیچ‌گونه الگوریتم انتخاب از متن استخراج شده‌اند. در این روش آنها توانستند مجموعه‌داده بازبینی فیلم‌ها را با دقت 87 درصد طبقه‌بندی کنند. در روش دوم با الگوریتم انتخاب خصیصه حداقل افزونگی – حداکثر وابستگی به حداکثر دقت 91.1 درصد دست یافتند و با اعمال الگوریتم انتخاب خصیصه سودمندی اطلاعات به دقت طبقه‌بندی 90.1 درصد دست یافتند.

در [14] برای انتخاب خصیصه از روش مربع کای[1] استفاده شده‌است. آنها بهترین نتیجه خود را با به‌کارگیری طبقه‌بند SVM و حداکثر آنتروپی به صورت ترکیبی بدست آوردند. باید توجه کنیم برای بهبود طبقه‌بندی بهتر است، بتوانیم مدل درستی از اسناد را ارائه دهیم، تا به دقت بالاتری دست

[1] Chi-squared

یابیم. استفاده همزمان و ترکیبی از چند الگوریتم طبقه‌بندی باعث افزایش پیچیدگی زمانی خواهد شد و نهایتا برای مجموعه‌داده‌های متفاوت لزوما باعث افزایش دقت طبقه‌بندی نخواهد شد. استفاده ترکیبی از چند الگوریتم طبقه‌بند برای یک مجموعه‌داده نمی‌تواند راه حلی برای بهبود سرعت و دقت طبقه‌بندی متون باشد. به‌جای استفاده از چند طبقه‌بند می‌توانیم از چند فیلتر انتخاب خصیصه استفاده کرده، یا به دنبال مدل مناسب‌تر برای مدل‌سازی اسناد باشیم.

روش‌های انتخاب خصیصه‌ی تک متغیره نسبت به روش‌های چند متغیره پیچیدگی زمانی کمتری دارند به همین دلیل در بسیاری از تحقیق‌ها از روش‌های تک متغیره استفاده شده‌است.

[15] و [16] برای طبقه‌بندی، از سودمندی اطلاعات استفاده کرده‌اند. تحقیق [17] با اتکا به نتایج [15] و [16] روش سودمندی اطلاعات را برای انتخاب سودمندترین خصیصه‌ها برگزید. [10]، [11] و [18] از روش درست‌نمایی لگاریتمی[1] استفاده کرده‌اند. در بین سه مقاله ذکر شده مقاله [11] به حداکثر دقت 90 درصد دست یافته‌است.

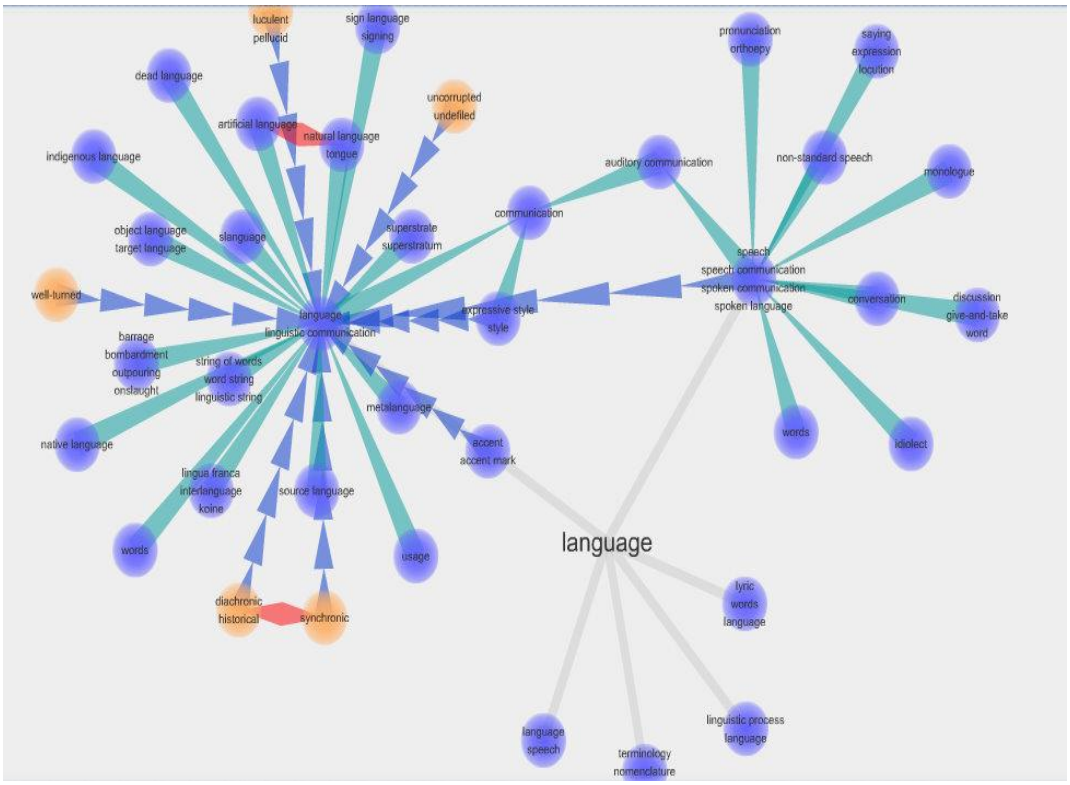
2- 4- 2- روش‌های معنایی- لغوی

برای نمونه یک روش لغوی ساده با استفاده از یک واژه‌نامه، که تمام لغات مربوط به دامنه کاری در آن طبق مثبت یا منفی بودن امتیاز داده شده‌اند، متن را بررسی می‌کند. این روش بسیار ساده بوده و در متن، لغاتی مثل "بد" "زشت" "خوب" "عالی" و غیره را که بار منفی یا مثبت دارند، در نظر گرفته و امتیاز آنها را جمع می‌کند. نتیجه نهایی، امتیاز جمله بدست آمده‌است که اگر مثبت باشد جمله مثبت و اگر منفی باشد جمله را منفی در نظر می‌گیرد. این روش با وجود سادگی به دلیل پیچیدگی ساختار زبانی، بندرت مورد استفاده قرار می‌گیرد. زیرا یک جمله می‌تواند دارای شمار زیادی از کلمات منفی باشد ولی معنی مثبتی داشته باشد. مشکل دیگر عدم توانایی شناختن استعاره‌ها و کنایه‌ها است. این

[1] Log likelihood

مشکل را تا حدودی می‌توان با داشتن یک واژه‌نامه جداگانه برای استعاره‌های متداول و روش‌های تشخیص تشابه حل کرد. ولی در این حالت هم بررسی لغتی به‌تنهایی کافی نیست و باید از روش‌های دیگری مثل درخت دستور زبان[1] استفاده برد [19].

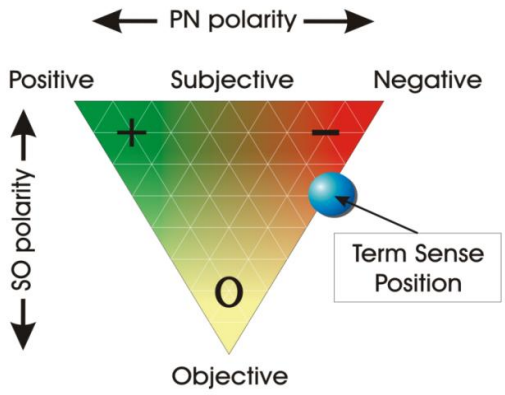
در روش‌های تکامل یافته‌تر از واژه‌نامه‌های لغوی پیشرفته برای تحلیل متن استفاده می‌شود. از جمله این واژه‌نامه‌ها می‌توان به WordNet که برای پژوهش‌های مربوط به پردازش زبان ساخته شده اشاره کرد. این واژه‌نامه با تاکید بر روی کلمه‌های هم‌معنا، تمام کلمات انگلیسی را خوشه بندی کرده و هر کدام از این خوشه‌ها با خوشه‌های دیگر می‌توانند رابطه‌ای مثل تضاد یا تناسب داشته باشند [20]. در شکل 2-3 نمایی از WordNet نمایش داده شده‌است[2].



شکل 2- 3 نمای کلی از WordNet

[1] Grammar Tree
[2] https://www.languagetrainers.co.uk/blog/2009/05/04/visualising-the-wordnet/

همچنین SentiWordNet یک نسخه ویرایش شده از وردنت است که به تمام لغات، دو وزن مثبت و منفی را تخصیص می‌دهد. میزان وزن هر لغت در SentiWordNet دو عدد بین 0 و 1 است [21]. مجموع این وزن‌ها در SentiWordNet را پلاریته کلمه می‌گویند [22]. شکل 2-4 نحوه تخصیص وزن‌ها در SentiWordNet را نشان می‌دهد[1].



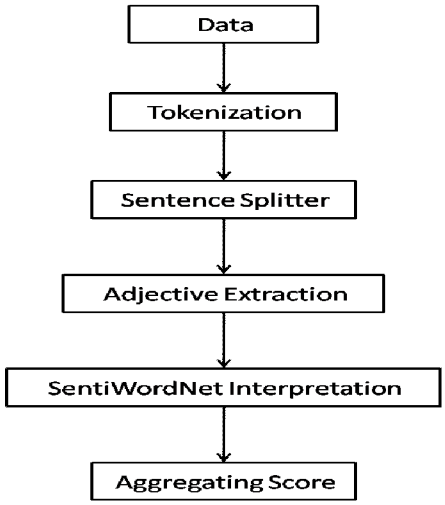
شکل 2- 4 نحوه تخصیص وزن‌ها در SentiWordNet

در برخی از تحقیقات برای مدل‌سازی اسناد از خصیصه‌های N-POS استفاده شده‌است. فی و همکارانش در سال 2004 از خصیصه‌های 1-POS و 2-POS استفاده کرده‌اند و بهترین دقت حاصل از طبقه‌بندی در این تحقیق 86 درصد بوده‌است [23]. آنها الگوهایی نحوی را ارائه دادند که اغلب متون جهت‌دار در این الگوها قرار می‌گیرند. مثلا یکی از الگوهای ارائه شده 2-POS، "اسم و صفت" بوده‌است. آنها ابتدا متن را برچسب گذاری کرده و الگوهای مورد نظر را از متن استخراج کردند. مدلی که در این روش ارائه شده همانند روش وینسنت و همکارانش سعی دارد در ابتدا متن را فیلتر کند و فقط متن جهت‌دار (متن حاوی نظرات مثبت یا منفی کاربران درباره یک موجودیت مشخص) را برای مرحله طبقه‌بندی و مدل‌سازی استفاده کند. اما مشکل این روش آن است که نمی‌توان برای همه

[1] http://ontotext.fbk.eu/sentiwn.html

حالت‌های متنِ جهت‌دار، الگویی ارائه داد و همواره ممکن است یک متن خاص، با الگوهای ارائه شده سازگار نباشد. مدل‌سازی متن با استفاده از الگوهای N-POS حتی نسبت به مدل 1-gram دقت کمتری را برای طبقه‌بندی به همراه داشت. لذا روش مناسبی برای مدل‌سازی اسناد نیست.

سی و گوپتا در سال 2013 مقاله‌ای را ارائه کرده‌اند [24]. آنها به‌جای استفاده از روش‌های انتخاب خصیصه پیچیده، تلاش کرده‌اند ترکیب مناسبی از خصیصه‌ها را جهت دستیابی به دقت بالاتر در عملیات طبقه‌بندی بیابند. همچنین برای کاهش دادن اندازه بردار خصیصه و حذف خصیصه‌های غیرمرتبط از SentiWordNet استفاده کرده‌اند. مجموعه‌داده این تحقیق اندازه کوچکتری نسبت به مجموعه‌داده مورد بررسی در سایر تحقیقات ذکر شده داشته‌است. در تحقیق مذکور با فیلترکردن خصیصه‌هایی که وزن مثبت یا منفی آنها (وزن هر کلمه همان مقداری بین 0 و 1 است که از SentiWordNet استخراج شده‌است) کمتر از 0.5 است، تعداد خصیصه‌ها را کاهش داده‌اند. ویژگی این روش استفاده از ترکیب‌های ساده و متفاوت، همچنین کاهش تعداد خصیصه‌ها با استفاده از SentiWordNet است. مشکل اصلی این روش، مجموعه‌داده مورد استفاده در آن است. این مجموعه‌داده قابلیت نشان دادن پایداری روش را ندارد. مراحل انجام این روش در شکل 2-5 مشاهده می‌شود.



شکل 2- 5 مراحل انجام تحلیل احساس با استفاده از SentiWordNet

دشتبانی و پیله‌ور در سال 2012 تحقیقی انجام داده‌اند [25]. آنها برای تشخیص احساس در متون فارسی از شباهت معنایی کلمات استفاده کرده‌اند. در این تحقیق به کمک دستور زبان فارسی و استفاده از یک متخصص در زمینه ادبیات فارسی 20 دسته احساسی معرفی شده‌است. ابهامات متن با توجه به اینکه لغت ورودی در چه دسته‌های احساسی می‌تواند قرار بگیرد مدیریت خواهند شد. در این تحقیق تمامی کلمات احساسی متن به همراه چگالی عددی و عددی که نشان دهنده میزان شباهت کلمه ورودی به دسته‌های احساسی می‌باشد، در یک فایلِ پیوستِ متن ورودی قرار داده شده و با توجه به آن، متن پردازش می‌شود.

در این مقاله برای ایجاد دسته‌های احساسی از روش ساخت WordNet با اندکی تغییرات استفاده شده‌است. درختی که در فرایند ساخت واژه‌نامه ارائه می‌شود، مبتنی بر لغات احساسی زبان فارسی است. نحوه ساخت درخت به‌گونه‌ای است که گره ریشه شامل تمام کلمات موجود در واژه‌نامه شده و انشعاب از هر گره بر این اساس که کدامیک از لغات می‌توانند در یک دسته قرار بگیرند انجام می‌شود. درخت ساخته شده در این تحقیق شامل 4 سطح بوده که سطح آخر درخت شامل کلماتی است که از نظر احساسی به هم شباهت دارند. سپس برای هر کلمه از متن عدد شباهت آن با دسته‌های احساسی محاسبه شده و با توجه به عدد شباهت هر کلمه به دسته‌های احساسی و چگالی کلمه در متنِ ورودی، احساس غالب متن مشخص خواهد شد.

2- 5- نتیجه‌گیری

با وجود اینکه روش‌های لغوی- معنایی از لحاظ سرعت و قدرت پردازش بسیار سریع‌تر از سایر روش‌ها عمل می‌کنند اما به دلیل میزان خطای بالای آنها و همچنین الزام نوشتن قواعد بسیار برای زبان، باعث شده این روش‌ها بندرت مورد استفاده قرار بگیرند. امروزه از روش‌های یادگیری ماشین به دلیل سادگی و همچنین عدم نیاز به دسترسی به ساختار جمله بیشتر استفاده می‌شود [19]. نتایج بدست

آمده در [13] نشان می‌دهد استفاده از ترکیب روش‌های معنایی و یادگیری ماشین می‌تواند بدون افزایش پیچیدگی، دقت را افزایش دهد.

فصل 3) روش پیشنهادی

3- 1- مقدمه

هدف اصلی این نوشتار ارائه مدلی مناسب برای متن‌ها می‌باشد. قصد داریم مجموعه‌ای از خصیصه‌ها را ارائه داده و با استفاده از آنها به مدلی دست یابیم که با داشتن آن، دیگر نیازی به استفاده کردن از روش‌های پیچیده انتخاب خصیصه نباشد.

در این فصل به توضیح و تشریح روش ارائه شده خواهیم پرداخت. با مطالعه و بررسی مقالات و تحقیقات ارائه شده به این نتیجه دست یافتیم که بهتر است برای تحلیل احساس در متن، از مجموعه خصیصه‌هایی استفاده کنیم که قادر باشند بیشترین اطلاعات لازم برای تحلیل احساس را در اختیار الگوریتم طبقه‌بندی، قرار دهند. به این ترتیب می‌توان دقت طبقه‌بندی متن‌ها را افزایش داد. هدف این رساله ارائه روشی برای تحلیل احساسات و عقاید موجود در متن می‌باشد. به‌گونه‌ای که این تحلیلگر، متن‌های موجود در مجموعه‌داده‌ها را در 5 دسته احساسی طبقه‌بندی کند.

از جمله مشکلات مطرح در پردازش زبان فارسی عدم وجود ابزارهای لازم برای ریشه‌یابی، برچسب‌گذاری POS کلمات و انتخاب خصیصه‌های سودمند است. البته برای بعضی کاربردها ابزارهایی وجود دارد، ولی این ابزارها از دقت کافی برخوردار نیستند. همچنین مشکلی که برای تحلیل احساس وجود دارد ارائه مدلی کامل و مفید برای متن‌ها می‌باشد. برای حل این مشکل مجموعه خصیصه‌های متفاوتی مورد بررسی قرار گرفته‌اند و از این میان تلاش شده مناسب‌ترین خصیصه‌ها را انتخاب کنیم.

مشکل دیگر در تحلیل احساسات و عقاید تعداد زیاد خصیصه‌ها می‌باشد. این مسئله باعث بروز مشکلات دیگری مانند خصیصه‌های افزونه و خصیصه‌های غیرمرتبط می‌شود. لذا باید برای انتخاب‌کردن خصیصه‌های سودمند از میان هزاران خصیصه راهی اندیشیده شود. روشی که اینجا ارائه شده فیلترکردن خصیصه‌ها است. به این ترتیب تعداد خصیصه‌ها و پیچیدگی زمانی کاهش یافته و دقت دسته‌بندی افزایش می‌یابد.

در ابتدا مجموعه‌ای از متن‌ها برای مجموعه‌داده آماده می‌شود. سپس متن‌ها خوانده شده و پیش‌پردازش‌هایی بر روی آنها انجام می‌گیرد. پیش‌پردازش، متن‌ها را برای مراحل بعدی آماده خواهد کرد. در مرحله بعد خصیصه‌های مورد نیاز استخراج شده و فیلترهای اولیه بر روی خصیصه‌ها اعمال می‌شود و در ادامه، سیستم به کمک خصیصه‌های انتخاب شده متن‌ها را دسته‌بندی می‌کند.

3- 2- مجموعه‌داده

برای پیاده‌سازی روش پیشنهادی به منابع داده‌ای و نرم‌افزاری نیاز خواهیم داشت. روش‌هایی که از خصیصه‌های N-gram و روش‌های مبتنی بر یادگیری ماشین استفاده می‌کنند در اولین گام نیازمند مجموعه‌داده‌های مناسب می‌باشند. در این تحقیق مجموعه‌داده، جهت آموزش و آزمون، در مراحل مختلف به‌کار گرفته می‌شود.

برای انجام این تحقیق مجموعه‌داده باید شامل متن‌هایی به زبان فارسی باشد. هر متن باید بتواند با توجه به بار احساسی خود در یکی از 5 دسته احساسی غم، شادی، عصبانیت، تنفر و ترس طبقه‌بندی شود. به علت در دسترس نبودن مجموعه‌داده‌ای با این مشخصات ناگزیر به ساخت آن هستیم.

متن‌های جمع‌آوری شده پس از آماده‌سازی به دو گروه داده‌های آموزشی و داده‌های تست تقسیم می‌شوند. تعداد این متن‌ها باید به اندازه‌ای باشد که بتوان سیستم را به‌طور مناسب با آنها آموزش داده و تست کرد.

3- 3- پیش‌پردازش

هر سند حاوی متون بازبینی، نظرات، احساسات و عقاید کاربران است. همه متن این اسناد در تحلیل احساس مفید واقع نخواهند شد. همچنین قالب و فرمت متن باید به‌گونه‌ای تغییر یابد تا علاوه بر اینکه بتواند مدل مناسبی از متون را ارائه دهد، به قالب ساده و مناسب برای پردازش در مراحل بعدی

تبدیل شود. ورودی این مرحله، مجموعه‌ای از اسناد متنی، حاوی نظرات، احساسات و عقاید کاربران است.

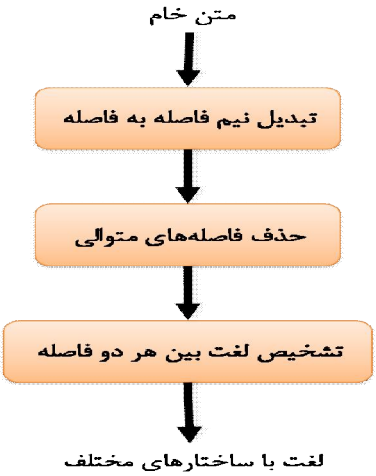
3- 3- 1- نشانه‌گذاری

پیچیدگی‌هایی در ساختار زبان فارسی وجود دارد که فرایند پردازش و تحلیل ماشینی زبان فارسی را نسبت به زبان انگلیسی سخت‌تر می‌کند. از جمله این مشکلات می‌توان به پیشوند و پسوندهای متفاوت کلمات، فاصله و نیم‌فاصله بین کلمات اشاره کرد. کلماتی مانند "می‌پسندم" دارای پیشوند "می" و کلمه "دوستان" دارای پسوند "ان" است. بسیاری از کلمات فارسی در کاربردهای متفاوت با پیشوندها و پسوندهایی همراه می‌شوند. این امر باعث افزایش چشم‌گیر تعداد خصیصه‌ها و همچنین افزایش پراکندگی آنها می‌شود، مثلا کلمات "می‌پسندم"، "پسندید"، "می‌پسندند" هر کدام یک خصیصه خواهند بود، در صورتی که بهتر است همه آنها را به عنوان یک خصیصه "پسندیدن" در نظر گرفت.

یکی از چالش‌های موجود در پردازش ماشینی زبان فارسی، فاصله بین کلمات در متون فارسی بوده‌است. به عنوان مثال "لذت‌بخش" در عبارت "زندگی در طبیعت برای شما لذت‌بخش خواهد بود" یک کلمه مرکب بوده که "لذت" و "بخش" با استفاده از نیم‌فاصله از هم جدا می‌شوند. اگر دو بخش "لذت" و "بخش" مانند "لذت بخش" با یک فاصله از هم جدا شوند دو نشانه مجزا خواهند بود.

برای تشخیص کلمات و استخراج نشانه‌های متن مهمترین جداکننده نشانه‌ها و اصطلاحات، فاصله بین آنها است. لذا باید قبل از انجام پردازش‌های لازم برای تحلیل احساس، واژه‌ها را با استفاده از فاصله مشخص کرده و در مرحله بعدی از آنها به عنوان ورودی استفاده کنیم. در ادامه فرایند، همه متن‌ها به واژه‌های تشکیل دهنده خود تجزیه شده‌اند و بیشتر با واژه‌های هر سند کار خواهیم کرد و دیگر کمتر

به متن خام و بدون پیش‌پردازش نیاز خواهیم داشت. شکل 3-1 فرایند نشانه‌گذاری متن را نشان می‌دهد.

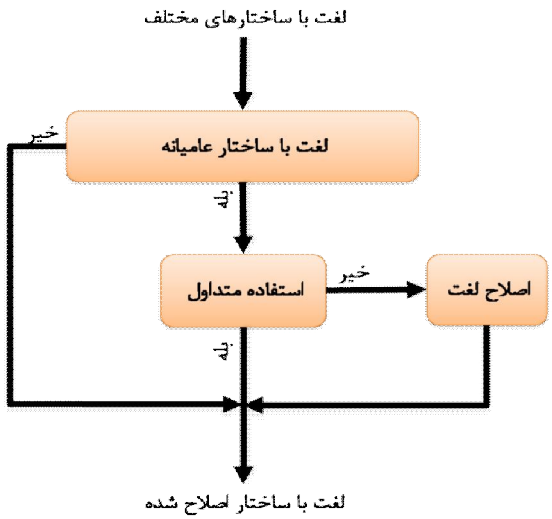


شکل 3- 1 فرایند نشانه‌گذاری متن

3- 3- 2- قالب نوشتاری عامیانه

دست‌نوشته‌ها و جملات احساسی جمع‌آوری شده برای مجموعه‌داده، مانند عبارت "چرا اون ابر سیاه بی کسی رو بخت تیره منه"، امکان دارد از قوانین گرامری و نحوی زبان فارسی پیروی نکنند. علاوه بر اینکه در نوشتار امکان استفاده از کلمات عامیانه وجود دارد، این کلمات خود نیز در متون متفاوت می‌توانند قالب نوشتاری متفاوت داشته باشند. قالب نوشتاری متفاوت کلمات عامیانه باعث افزایش پراکندگی خصیصه‌ها و همچنین کاهش دقت طبقه‌بندی می‌شود. به عنوان مثال عبارات "عالیه"، "عاللللییییییه" و " عالی هس" همه یک مفهوم را بیان می‌کنند. به‌منظور حل‌کردن چالش‌های مطرح شده، در روش‌های مبتنی بر فرهنگ لغات، عملیات پیش‌پردازش بر روی متن انجام می‌گیرد. فرایند پیش‌پردازش، کلمات و عبارات عامیانه را به قالب نرمال و صحیح نحوی و گرامری تبدیل می‌کند.

با توجه به استفاده از روش‌های مبتنی بر یادگیری ماشین در این تحقیق، برای حل این مساله اگر کلمه عامیانه به میزان زیاد و به‌طور متداول در متون استفاده شده باشد، به همان صورت باقی مانده تا سیستم با همان شکل عامیانه واژه آموزش دیده و در مرحله تست بتواند واژه را تشخیص دهد. و اگر کلمه عامیانه یک ایست‌واژه باشد به لیست آنها افزوده شده تا در طول فرایند، از متن‌ها حذف شود. در غیر این صورت واژه عامیانه در فرایند پیش‌پردازش به شکل اصلی خود تبدیل می‌شود. شکل 3-2 فرایند اصلاح ساختار واژه‌ها را نشان می‌دهد.

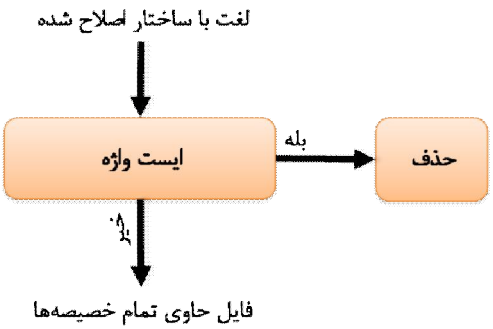


شکل 3- 2 فرایند اصلاح ساختار واژه‌ها

3- 3- 3- حذف ایست‌واژه‌ها

ورودی این مرحله، متنی حاوی احساس است که کلمه‌های آن توسط فاصله مشخص شده‌اند. در این مرحله از فرایند ایست‌واژه‌ها حذف می‌شوند. ایست‌واژه‌ها کلمات و عباراتی هستند که هیچ کمکی به فرایند طبقه‌بندی اسناد، در جهت تحلیل احساس موجود در متن نمی‌کنند. مجموعه ثابت و یکسانی برای ایست‌واژه‌ها وجود ندارد بلکه برای حوزه‌های متفاوت در پردازش زبان طبیعی از ایست‌واژه‌های متفاوتی استفاده شده‌است. به این معنی که شاید یک کلمه در یک حوزه کلیدی محسوب شده و در حوزه‌ای دیگر ایست‌واژه در نظر گرفته شود. در این مرحله از فرایند ابتدا هر سند خوانده شده و پس از

حذف ایست‌واژه‌ها سایر کلمات باقی مانده به مرحله بعدی هدایت می‌شوند. شکل 3-3 فرایند حذف ایست‌واژه‌ها را نشان می‌دهد.



شکل 3- 3 فرایند حذف ایست‌واژه‌ها

3- 4- انتخاب خصیصه

هر متن حاوی کلمه‌های احساسی و غیراحساسی است. همه کلمه‌های متن، برای تحلیل احساس مفید نیستند. در این تحقیق قصد داریم مجموعه مفیدی از خصیصه‌ها را ارائه دهیم. کارها و تحقیقات قبلی که به‌منظور تحلیل احساس در متن به انجام رسیده‌اند نشان داده‌اند که به‌کار گرفتن خصیصه‌های N-gram n>1 می‌تواند وابستگی میان واژه‌ها را بهتر نشان دهد. اما مدل‌سازی متن‌ها با استفاده از خصیصه‌های N-gram n>1 باعث افزایش تعداد خصیصه‌ها و کاهش دقت دسته‌بندی شده و این خود مشکلی بر سر راه استفاده از آنها است. در این رساله پیشنهاد شده‌است با استفاده از فیلترکردن خصیصه‌ها از این مشکلات رها شویم.

3- 4- 1- فیلترکردن خصیصه‌ها

در تحقیقات پیشین، مجموعه خصیصه‌هایی که برای مدل‌سازی متن‌ها ارائه شده به‌گونه‌ای بوده که این مجموعه خصیصه‌ها اطلاعات مفیدی از محتوای اسناد برای فرایند تحلیل احساسات و عقاید مهیا کنند و دقت طبقه‌بندی را افزایش دهند. در اغلب تحقیقات گذشته برای مدل‌سازی متن تنها از خصیصه‌های 1-gram استفاده شده‌است و گاهی ترکیب 1-gram و 2-gram را به‌کار برده‌اند. به این

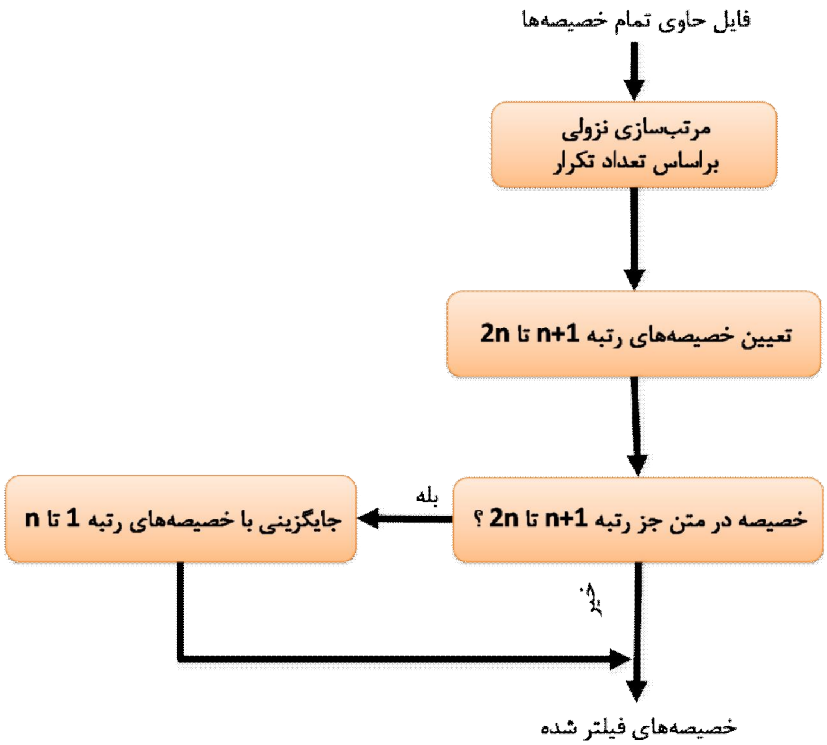
دلیل که خصیصه‌های 2-gram به‌تنهایی عملکرد بدتری نسبت به 1-gram دارند. به‌طور کلی خصیصه‌های N-gram با N بزرگتر، عملکرد بدتری نسبت به خصیصه‌های با N کوچکتر دارند. این عملکرد بد به دلیل غیرمرتبط بودن و غیرمفید بودن این مجموعه خصیصه‌ها نیست، بلکه به علت پراکندگی آنها است. با بزرگتر شدن N پراکندگی این خصیصه‌ها بیشتر می‌شود که خود عاملی بر کاهش دقت طبقه‌بندی و افزایش زمان اجرا خواهد بود. زبان طبیعی ما کلمات هم‌معنای زیادی در خود دارد که قابلیت استفاده به‌جای یکدیگر را دارند، استفاده از کلمات هم‌معنا در عبارات باعث ایجاد این پراکندگی می‌شود. این پراکندگی نه‌تنها برای خصیصه‌های N-gram با $N > 1$ مطرح بوده بلکه برای خصیصه‌های 1-gram نیز مطرح است.

برای استفاده از خصیصه‌های مفید و بهبود دقت طبقه‌بندی می‌توان با استفاده از فیلترکردن خصیصه‌ها، پراکندگی آنها را کاهش داد. برای کاهش پراکندگی، می‌توان خصیصه‌های هم‌معنا را پیدا کرده و به‌جای تمام آنها خصیصه با معنای مشابه را در متن‌ها جایگزین کنیم. اما جستجو و جایگزینی واژه‌های هم‌معنا در مجموعه‌داده مستلزم به‌کارگیری واژه‌نامه‌ی معنایی فارسی می‌باشد. لذا در این تحقیق برای کاهش پراکندگی خصیصه‌ها از روشی آماری استفاده خواهد شد.

برای کاهش پراکندگی خصیصه‌ها باید واژه‌های با معنا و مفهوم مشابه در دسته‌های احساسی تشخیص داده شده و جایگزین شوند. واژه‌هایی که در دسته‌های احساسی زیاد تکرار می‌شوند، به استثنای ایست‌واژه‌ها که در مرحله قبل حذف شده‌اند، حاوی بار احساسیِ مربوط به همان دسته هستند و تا حدودی معنا و مفهوم مشابه را می‌رسانند. می‌توان از این ویژگیِ دسته‌های احساسی برای جایگزینی و کاهش پراکندگی خصیصه‌ها استفاده کرد. مزیت استفاده از این روش عدم نیاز به واژه‌نامه معنایی و پیاده‌سازی آن به کمک آمار می‌باشد.

با توجه به مطالب گفته شده، در این مرحله خصیصه‌های هر دسته براساس تعداد تکرارشان به صورت نزولی مرتب می‌شوند. سپس n خصیصه اول لیست که دارای بیشترین تعداد تکرار هستند جایگزین n

خصیصه بعدی در لیست می‌شوند. برای مثال به ازای n=5، خصیصه‌های رتبه 10، 9، 8 و ... به ترتیب با خصیصه‌های رتبه 1، 2، 3 و ... جایگزین می‌شوند. در شکل 3-4 فرایند فیلترکردن خصیصه‌ها نشان داده شده‌است.



شکل 3- 4 فرایند فیلترکردن خصیصه‌ها

فرایندی که در شکل 3-4 نشان داده شده در هر دسته احساسی به طور مستقل انجام می‌شود. با افزایش n، تعداد خصیصه‌هایی که باید در هر دسته جایگزین شوند بیشتر شده و پراکندگی خصیصه‌ها در آن دسته کاهش می‌یابد. از طرفی با افزایش n، ممکن است مشکل خصیصه‌های مشترک بین دسته‌های احساسی بوجود بیاید.

با افزایش هر چه بیشتر n، خصیصه‌های با تعداد تکرار کمتر برای جایگزینی انتخاب می‌شوند. خصیصه‌های کم‌تکرار ممکن است در بیش از یک دسته احساسی حضور داشته باشند و در حالت عملیاتی مشخص نیست این خصیصه‌ها مربوط به کدام دسته احساسی هستند و برای جایگزینی آنها

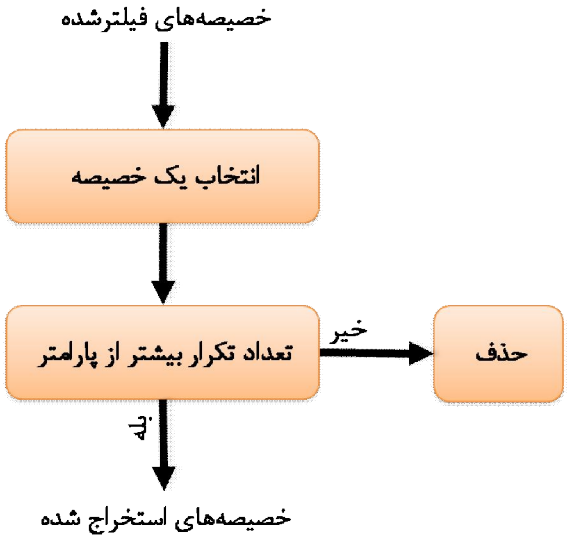
از خصیصه‌های پرتکرار کدام دسته باید استفاده شود. لذا برای رفع این مشکل مقدار n تا اندازه‌ای می‌تواند افزایش یابد که خصیصه‌های مشترک بین دسته‌ها برای جایگزینی انتخاب نشوند یا جایگزینی برای خصیصه‌های مشترک انجام نگیرد.

3- 4- 2- استخراج خصیصه‌ها

این مرحله مهمترین مرحله از فرایند تحلیل احساس می‌باشد. باید مجموعه خصیصه‌هایی را انتخاب کنیم که به خوبی متن‌های موجود در مجموعه‌داده‌ها را مدل‌سازی کنند. همچنین این مدل‌های ایجاد شده سودمندترین اطلاعات برای تحلیل احساس را در خود داشته باشند. توجه داشته باشیم مدل‌های مناسب زیادی برای یک سند می‌توان ارائه داد ولی مهمترین مساله که باید مورد توجه قرار گیرد، این است که کدام یک از این مدل‌ها برای فرایند تحلیل احساس مفید می‌باشد.

مثلا عباسی و همکارانش برای ارائه مدل مناسب از هر سند مجموعه بسیار کاملی از خصیصه‌ها را به‌کار گرفتند [7]. این مجموعه خصیصه‌ها در جدول 2-2 بیان شده‌اند. استفاده از این مجموعه کامل از خصیصه‌ها که بسیاری از آنها با یکدیگر همپوشانی دارند باعث افزایش غیرقابل توجیه تعداد خصیصه‌ها خواهد شد، عباسی و همکارانش برای حل این مشکل از الگوریتم انتخاب خصیصه شبکه ارتباطی خصیصه‌ها استفاده کرده‌اند، این الگوریتم پیچیدگی زمانی بالایی دارد؛ علاوه بر آن افزایش قابل توجهی در دقت طبقه‌بندی نیز حاصل نشده‌است. میتال و آگراوال در سال 2013 مدلی ارائه داده‌اند. آنها در این مدل تنها ترکیبی از خصیصه‌های 1-gram و 2-gram را استفاده کرده‌اند [13]. آنها از الگوریتم انتخاب حداقل افزونگی- حداکثر وابستگی استفاده کردند. این الگوریتم با وجود اینکه پیچیدگی زمانی کمتری نسبت به شبکه ارتباطی خصیصه (ارائه شده در [7]) دارد ولی نسبت به سایر الگوریتم‌های تک متغیره بیان شده در بخش‌های قبل، پیچیدگی زمانی بیشتری دارد.

در این تحقیق استخراج خصیصه از متن‌ها با استفاده از متغیر تعداد تکرار r انجام می‌شود. اگر تعداد تکرار یک واژه در یک دسته احساسی بیشتر از متغیر r باشد، واژه به عنوان خصیصه مفید استخراج می‌شود. مقدار متغیر r تاثیر مستقیمی بر روی نتایج خواهد داشت به همین علت در مرحله تست و با توجه به نتایج، بهترین مقدار برای آن مشخص خواهد شد. شکل 3-5 فرایند استخراج خصیصه را نشان می‌دهد.



شکل 3- 5 فرایند استخراج خصیصه‌ها

3- 5- محاسبه احتمال با روش بیز

به‌طور ساده روش بیز روشی برای دسته‌بندی پدیده‌ها، بر پایه احتمال وقوع یا عدم وقوع یک پدیده است. در این روش با استفاده از احتمال پیشینِ یک رویداد، احتمال تعلق آن به یک دسته (احتمال پسین) محاسبه می‌شود.

رابطه 3-1 فرمول اصلی دسته‌بند بیز است. Px به معنای احتمال x است [26].

(1-3)

$$\hat{y} = argmax\ p(C_k) \prod_{i=1}^{n} p(x_i|C_k) \qquad k\epsilon\{1, \dots, K\}$$

همچنین احتمال پسین از رابطه 3-2 بدست می‌آید.

(2-3)

$$p(C_k|X) = \frac{p(C_k)\, p(X|C_k)}{p(X)} \qquad X = (x_1, \dots, x_n)$$

X در اینجا به معنی داده‌ی جدید و یک متغیر برداریست. C کلاسی است که داده می‌تواند در آن قرار بگیرد یا نگیرد.

فرمول 3-2 را می‌توان به زبان ساده اینطور شرح داد که احتمال قرارگرفتن یک داده در یک دسته برابر تعداد داده‌های مشابه یا یکسان است که قبلا در آن دسته قرار داشته‌اند. ولی از آنجا که داده‌ی ما شامل برداری از خصیصه‌ها است، برای بدست آوردن احتمال وجود داده در دسته مذکور (طبق قانون زنجیره‌ای) به رابطه 3-4 می‌رسیم:

(4-3)

$$\begin{aligned} p(C_k|x_1, \dots, x_n) &= p(C_k)\, p(x_1, \dots, x_n|C_k) \\ &= p(C_k)\, p(x_1|C_k)\, p(x_2, \dots, x_n|C_k, x_1) \\ &= p(C_k)\, p(x_1|C_k)\, p(x_2|C_k, x_1)\, p(x_3, \dots, x_n|C_k, x_1, x_2) \\ &= p(C_k)\, p(x_1|C_K) p(x_2|C_k, x_1) \dots\ p(x_n|C_k, x_1, x_2, x_3, \dots, x_{n-1}) \end{aligned}$$

چون احتمال رخ‌دادن خصیصه‌ها در متن مستقل از هم می‌باشند، می‌توان از رابطه 3-5 استفاده کرد:

(5-3)

$$p(C_k|x_1, \dots, x_n) \propto p(C_k, x_1, \dots, x_n)$$
$$\propto p(C_k)\, p(x_1|C_k)\, p(x_2|C_k)\, p(x_3|C_k) \dots$$
$$\propto p(C_k) \prod_{i=1}^{n} p(x_i|C_k).$$

پس می‌توان گفت احتمال قرارگرفتن داده X در کلاس C از رابطه 3-6 محاسبه می‌شود.

(6-3)

$$p(C_k|x_1, \dots, x_n) = \frac{1}{Z}\, p(C_k) \prod_{i=1}^{n} p(x_i|C_k) \qquad Z = p(X)$$

3- 6- نتیجه‌گیری

همانطور که در فصل 2 بررسی شد نحوه انتخاب خصیصه‌ها و میزان پراکندگی آنها در نتیجه روش، بیشترین تاثیر را دارند. لذا در این تحقیق سعی می‌شود با تمرکز بر روی کاهش پراکندگی و انتخاب خصیصه‌های مفید دقت روش را تا حد امکان بهبود دهیم. پیاده‌سازی روش‌هایی که در این فصل مطرح شد، با استفاده از منابع داده‌ای و نرم‌افزاری یا به صورت دستی (با توجه به عدم دسترسی به منابع معنایی مورد نیاز) در فصل بعد انجام می‌شود. با بررسی نتایج حاصل، عملکرد این روش‌ها ارزیابی شده و بهترین مقادیر برای متغیرهای مورد استفاده در سیستم مشخص می‌شود.

فصل 4) پیاده‌سازی و نتایج

4- 1- مقدمه

در این فصل فرایند پیاده‌سازی روش‌های مطرح شده در فصل قبل تشریح می‌شود. برای انجام این تحقیق، به منابع داده‌ای و نرم‌افزاری نیاز خواهیم داشت. با توجه به امکانات خوبی که زبان جاوا برای کار با متون فراهم کرده، برنامه‌ی مورد نیاز برای تحلیل احساس در متن، به زبان جاوا و با استفاده از محیط Net Beans نوشته می‌شود. همچنین مجموعه‌داده مورد نظر توسط پژوهشگر تهیه خواهد شد. در انتهای هر بخش خلاصه‌ای از نتایج ارائه شده و پس از آن نتایج نهایی و کلی نشان داده شده‌است. در فصل بعد زمینه‌هایی برای کارهای آینده مطرح می‌شود.

4- 2- ساخت مجموعه‌داده

برای بررسی روش‌های پیشنهادی، مجموعه‌داده‌ای نیاز داریم که هر متنِ آن بتواند در یک گروه احساسی دسته‌بندی شود. برای ساخت این مجموعه‌داده از صفحات وب، وبلاگ‌ها، کتاب‌های الکترونیکی و دست‌نوشته‌هایی که حاوی بار احساسی هستند استفاده شده‌است. میزان بار احساسی و نوع احساسی که در هر متن وجود دارد توسط کاربر بررسی شده و اگر متن برای فرایند تحلیل احساس مناسب باشد به مجموعه‌داده‌ها اضافه می‌شود. متن‌های جمع‌آوری شده در این مرحله خام بوده و نیاز به پیش‌پردازش دارند. در جدول 4-1 مشخصات مجموعه‌داده تهیه شده نشان داده شده‌است.

جدول 4- 1 مشخصات مجموعه‌داده

نام دسته	1- غم	2- شادی	3- عصبانیت	4- تنفر	5- ترس	کل
تعداد آموزش	111	95	75	75	58	414
تعداد تست	18	16	12	14	12	72
تعداد کل	129	111	87	89	70	486

4- 3- پیش‌پردازش متن‌ها

در این مرحله هر متن از ابتدا پایش شده تا عملیات پیش‌پردازش بر روی آن انجام گیرد. در مرحله پیش‌پردازش، واژه‌ها با استفاده از نشانه‌گذاری مشخص می‌شوند. معیار تشخیص هر واژه همان‌طور که گفته شد فاصله بین کلمات است. با انجام این فرایند تمام کاراکترهای بین هر دو فاصله به عنوان یک واژه در نظر گرفته می‌شود. جدول 4-2 دو نمونه از نشانه‌گذاری جمله را نشان می‌دهد.

جدول 4- 2 دو نمونه از نشانه‌گذاری جمله

جمله قبل از نشانه‌گذاری	واژه‌های مشخص شده
به انتهای بودنم رسیده‌ام ؛ اما اشک نمی‌ریزم... پنهان شده‌ام پشتِ لبخندی که درد می کند...	به-انتهای-بودنم-رسیده-ام-؛-اما-اشک-نمی-ریزم-.-.-.-پنهان-شده-ام-پشت-لبخندی-که-درد-می-کند-.-.-.
گفته باشم .. من درد می کشم ؛ تو اما چشم‌هایت را ببند سخت است بدانم می بینی ، و بی خیالی...	گفته-باشم-.-.-من-درد-می-کشم-؛-تو-اما-چشم-هایت-را-ببند-سخت-است-بدانم-می-بینی-،-و-بی-خیالی-.-.-.

کلماتی که پس از فرایند نشانه‌گذاری یک متن مشخص می‌شوند ممکن است دارای ساختارهای مختلف باشند. به همین علت در مرحله بعد اگر یک واژه به شکل عامیانه در متن به‌کار رفته باشد تصحیح شده و شکل درست آن جایگزین می‌شود. اما اگر یک واژه به‌طور رایج در متن‌ها به شکلی غیر از شکل اصلی خود استفاده شده باشد همان‌طور باقی مانده تا در روند آموزش، برنامه با همان شکل رایج واژه آموزش داده شود. در جدول 4-3 چند نمونه از واژه‌های عامیانه و شکل اصلی آنها مشاهده می‌شود.

جدول 4- 3 چند نمونه از واژه‌های عامیانه

شکل عامیانه واژه	شکل صحیح واژه	نقش واژه	عملیات
موندن	ماندن	خصیصه	اصلاح
دیگه	دیگر	ایست‌واژه	اصلاح
اونا	آنها	ایست‌واژه	عدم تغییر
رو	را	ایست‌واژه	عدم تغییر
خونه	خانه	خصیصه	اصلاح
آروم	آرام	خصیصه	اصلاح

در هر متن تعداد زیادی واژه وجود دارد که اطلاعات مفیدی برای تحلیل احساس فراهم نمی‌کنند. تعدادی از این کلمه‌ها ایست‌واژه بوده و در مرحله پیش‌پردازش از متن حذف می‌شوند و تعدادی از آنها واژه‌های بدون احساس هستند که در مرحله انتخاب خصیصه، از تاثیر آنها بر روی نتایج تحقیق جلوگیری می‌شود. در جدول 4-4 تعدادی از ایست‌واژه‌ها نشان داده شده‌است.

جدول 4- 4 تعدادی از ایست‌واژه‌ها

اکنون	الان	البته	ای	از	ازجمله
انگار	آورد	باز	حالا	بسیار	بعدا
به	بهتر	بودم	بی	بگیر	تا
می	دارم	دیروز	نیز	همین	فوق
آیا	عنوان	عقب	،	.	پنج
پیدا	چرا	فقط	1	2	دوباره
ضمن	نباید	گردد	چون	نشان	حتی
اینکه	وقتی	ایشان	بعضی	ده	گروهی
نباید	کنم	طی	امروز	تمام	مثل
استفاده	داد	داشته	هست	مردم	چنین

4- 4- انتخاب خصیصه

برای کاهش پیچیدگی زمانی و افزایش دقت دسته‌بندی، خصیصه‌های مفید برای تحلیل احساس به کمک فرایند استخراج خصیصه مشخص می‌شوند. تحقیق [7] مدلی جامع از خصیصه‌های N-gram ارائه داده‌است، ولی تعداد مجموعه خصیصه‌های آن زیاد است و باعث افزایش خصیصه‌های غیر مفید یا افزونه خواهد شد. وجود خصیصه‌ی غیر مفید یا افزونه باعث می‌شود، اثرگذاری سایر خصیصه‌های سودمند و مرتبط با تحلیل احساس کاهش یابد لذا به همان نسبت دقت طبقه‌بندی نیز کاهش می‌یابد، همچنین باعث افزایش حجم بردار خصیصه خواهد شد و در نتیجه ما را با مشکل حافظه مواجه خواهد کرد و سرعت طبقه‌بندی را به شدت کاهش می‌دهد.

نتایج بدست آمده توسط آگروال و میتال نشان از عملکرد مطلوب خصیصه‌های 1-gram و سودمند نبودن خصیصه‌های 2-gram دارد و تصدیق کننده نتایجی است که سال 2002، پنگ و همکارانش به آن دست یافتند. در این رساله تلاش بر آن بوده مجموعه کاهش یافته‌ای از خصیصه‌ها را برای مدل‌سازی اسناد برگزینیم، به گونه‌ای که مدل مناسبی از اسناد را برای تحلیل احساس ارائه دهند. برای بررسی دقت خصیصه‌های مطرح در زمینه تحلیل احساس بر روی مجموعه‌داده‌ای که برای این تحقیق آماده شده‌است از خصیصه‌های 1-gram و 2-gram استفاده می‌شود.

برای افزایش دقت در استفاده از خصیصه‌های n-gram n>1 باید پراکندگی موجود در خصیصه‌ها را کاهش دهیم. برای این منظور روش مطرح شده در فصل قبل انجام شده و خصیصه‌های هم‌معنا در دسته‌های احساسی جایگزین می‌شوند. انتظار می‌رود با فیلترشدن خصیصه‌ها، پراکندگی آنها کم شده و نتایج استفاده از خصیصه‌های 1-gram و 2-gram بهبود یابد. در جدول 4-5 تعدادی از خصیصه‌های پرتکرار دسته‌های احساسی به صورت نزولی مرتب شده‌اند. از این خصیصه‌ها برای جایگزینی و کاهش پراکندگی در دسته‌های احساسی استفاده می‌شود.

جدول 4- 5 کلمات با بیشترین تعداد تکرار در دسته‌های احساسی

1		2		3		4		5	
دلم	37	لبخند	41	خیانت	24	لعنت	44	جن	39
غم	34	شاد	26	آدم	21	لعنتی	34	اومد	37
اشک	32	شادی	26	گند	11	نفرین	27	اتاق	33
درد	26	عشق	24	گرگ	10	عشق	13	ترس	32
گریه	24	زیبا	23	حرف	9	دروغ	12	دست	32
تنهایی	20	دوست	18	پر	8	متنفر	11	پنجره	30
دل	19	زیبایی	15	فحش	7	نفرت	9	صدا	29
مرگ	15	خدا	14	دل	7	نفس	8	بچه	28

همانطور که در جدول 4-5 مشاهده می‌شود، واژه "*دل*" در دو دسته احساسی حضور دارد و اگر مقدار n زیاد باشد مجبور به جایگزینی آن با خصیصه‌های پرتکرار هستیم. اما در حالت عملیاتی که مشخص نیست خصیصه مشترک به کدام دسته تعلق دارد، نمی‌توان جایگزین مناسبی برای آن انتخاب کرد. برای حل این مشکل می‌توان مقدار n را کم در نظر گرفت یا جایگزینی برای خصیصه‌های مشترک را انجام نداده و خصیصه‌های مشترک بدون تغییر باقی بمانند.

در مرحله بعد واژه‌هایی که تعداد تکرار آنها در یک دسته احساسی کمتر از مقدار متغیر r باشد به عنوان داده‌های بدون احساس طلقی شده و در فرایند آموزش از آنها استفاده نمی‌شود. به بیان دیگر اگر یک واژه در متن‌های احساسی به صورت معدود مورد استفاده قرار گرفته باشد می‌توان نتیجه گرفت که واژه بدون احساس است. پس از حذف واژه‌های بدون احساس، اکثر واژه‌هایی که در متن‌ها باقی مانده کلماتی هستند که در متن‌های دارای یک احساس خاص بیشتر استفاده می‌شوند. به بیان دیگر می‌توان از این واژه‌ها به عنوان خصیصه متن‌های احساسی استفاده کرد.

4- 5- محاسبه احتمال و طبقه‌بندی

دسته‌بندی متن‌ها در گروه‌های احساسی با استفاده از خصیصه‌های انتخاب شده در مرحله قبل و به کمک روش بیز انجام می‌شود. در مرحله آموزش، احتمال پیشین هر خصیصه در دسته احساسی خودش محاسبه شده و ثبت می‌شود. در قسمت تست، خصیصه‌های هر متنِ مورد آزمایش استخراج شده و برای هر خصیصه احتمال وقوع (پسین) آن در هر 5 دسته احساسی از میان مقادیر ثبت شده بدست می‌آید. هر متن در مرحله تست ممکن است شامل چندین خصیصه باشد. پس به تعداد خصیصه‌ها مقادیر پنج تایی احتمال وجود دارد. برای محاسبه احتمال تعلق یک متن به دسته احساسی باید مقادیر احتمال مربوط به خصیصه‌ها نظیر به نظیر در هم ضرب شوند. بیشترین مقدار بدست آمده نشان دهنده دسته احساسی مربوط به متن است. در جدول 4-6 مقادیر احتمال مربوط به جمله " *و می روی و ما دور می شویم از هم ، من می مانم و این دل و تنهایی و غم ، تو می روی و ما دلتنگ می شویم با هم ، تو می مانی و این زندگی و یک آسمان بلند .* " از دسته احساسی غم مشاهده می‌شود.

جدول 4- 6 مقادیر احتمال مربوط به یک جمله نمونه

شماره دسته / واژه	1	2	3	4	5
شویم	0.001	0.001	0.001	0.001	0.001
مانم	0.001	0.001	0.001	0.001	0.001
دل	0.210	0.315	0.473	0.001	0.001
غم	0.996	0.001	0.001	0.001	0.001
دلتنگ	0.996	0.001	0.001	0.001	0.001
شویم	0.001	0.001	0.001	0.001	0.001
مانی	0.001	0.001	0.001	0.001	0.001
آسمان	0.996	0.001	0.001	0.001	0.001
حاصلضرب	$2.49 * e^{-5}$	$3.91 * e^{-14}$	$3.45 * e^{-14}$	$3.91 * e^{-17}$	$3.91 * e^{-17}$

4- 6- نتایج

در این بخش نتایج نهایی روش پیشنهادی با توجه به متغیرهای موجود، ارائه می‌شود. تغییر هرکدام از این متغیرها بر روی نتایج تاثیر خواهد داشت. با بررسی این تغییرات، بهترین مقادیر برای متغیرها مشخص شده و نتایج بدست آمده با استفاده از این مقادیر برای متغیرها، به عنوان نتیجه نهایی تحقیق در نظر گرفته می‌شود. در ادامه مقادیر مربوط به جدول‌ها عملکرد روش بر حسب درصد را نشان می‌دهند. جدول 4-7 نتایج تحقیق با خصیصه‌های 1-gram و بدون اعمال فیلتر را نشان می‌دهد.

جدول 4- 7 نتایج تحقیق با خصیصه‌های 1-gram، بدون اعمال فیلتر

شماره دسته / تعداد تکرار r	1	2	3	4	5	کل
1	77.7	68.7	41.6	57.1	100	69.4
2	88.8	81.2	41.6	42.8	100	72.2
3	83.3	93.7	16.6	78.5	100	76.3
4	83.3	93.7	16.6	71.4	100	75.0
5	88.8	93.7	16.6	85.7	100	79.1
6	83.3	87.5	16.6	92.8	100	77.7
7	77.7	93.7	16.6	85.7	100	76.3
8	77.7	87.5	16.6	78.5	100	73.6
9	77.7	87.5	16.6	50.0	100	68.0
10	83.3	87.5	8.3	50.0	100	68.0
میانگین هر دسته	82.1	87.4	20.7	69.2	100	73.5

برای بررسی نتایج در هر دسته، علاوه بر نتایج کلی که در ستون آخر هر جدول قرار دارد دقت روش برای هر دسته به صورت جدا نیز محاسبه شده‌است.

در جدول 4-8 نتایج تحقیق با خصیصه‌های 2-gram و بدون اعمال فیلتر مشاهده می‌شود.

جدول 4- 8 نتایج تحقیق با خصیصه‌های 2-gram، بدون اعمال فیلتر

شماره دسته / تعداد تکرار r	1	2	3	4	5	کل
1	27.7	25.0	8.3	28.5	50.0	27.7
2	0.0	0.0	0.0	14.2	16.6	5.5
3	0.0	0.0	0.0	7.1	8.3	2.7
4	0.0	0.0	0.0	0.0	8.3	1.3
5	0.0	0.0	0.0	0.0	8.3	1.3
6	0.0	0.0	0.0	0.0	8.3	1.3
7	0.0	0.0	0.0	0.0	8.3	1.3
8	0.0	0.0	0.0	0.0	8.3	1.3

همان‌طور که در جدول 4-8 مشاهده می‌شود، افت شدید در نتایج، به علت وجود پراکندگی در خصیصه‌ها بوجود آمده‌است. برای بهبود دسته‌بندی با خصیصه‌های 2-gram و رفع پراکندگی، باید روی خصیصه‌ها فیلتر اعمال شود. برای اعمال فیلتر مقدار n=7 در نظر گرفته می‌شود. جدول 4-9 نتایج تحقیق با خصیصه‌های 1-gram و با اعمال فیلتر را نشان می‌دهد.

جدول 4- 9 نتایج تحقیق با خصیصه‌های 1-gram، با اعمال فیلتر

شماره دسته / تعداد تکرار r	1	2	3	4	5	کل
1	72.2	75.0	50.0	57.1	100	70.8
2	83.3	75.0	50.0	64.2	100	75.0
3	83.3	81.2	50.0	78.5	100	79.1
4	83.3	87.5	50.0	64.2	100	77.7
5	83.3	87.5	66.6	78.5	100	83.3
6	72.2	87.5	75.0	85.7	100	83.3
7	72.2	87.5	75.0	64.0	100	79.1
8	72.2	87.5	75.0	78.5	100	81.9
9	72.2	87.5	75.0	64.2	100	79.1
10	77.7	93.7	50.0	71.4	100	79.1
میانگین هر دسته	77.1	84.9	61.6	70.6	100	78.8

در جدول 4-10 نتایج تحقیق با خصیصه‌های 2-gram و با اعمال فیلتر نشان داده شده‌است.

جدول 4- 10 نتایج تحقیق با خصیصه‌های 2-gram، با اعمال فیلتر

شماره دسته / تعداد تکرار r	1	2	3	4	5	کل
1	38.8	43.7	8.3	50.0	75.0	**43.0**
2	5.5	6.2	0.0	21.4	66.6	18.0
3	0.0	0.0	0.0	7.1	50.0	9.7
4	0.0	0.0	0.0	0.0	41.6	6.9
5	0.0	0.0	0.0	0.0	25.0	4.1
6	0.0	0.0	0.0	0.0	16.6	2.7
7	0.0	0.0	0.0	0.0	16.6	2.7
8	0.0	0.0	0.0	0.0	16.6	2.7

با مقایسه دو جدول 4-8 و 4-10 می‌توان دریافت با فیلترکردن خصیصه‌ها دقت روش با خصیصه‌های 2-gram تاحدی بهبود یافته ولی نسبت به نتایج خصیصه‌های 1-gram اختلاف زیادی دارد. همچنین اعمال فیلتر موجب بهبود عملکرد خصیصه‌های 1-gram نیز شده‌است. اما به علت حساسیت خصیصه‌های N-gram N>1 به پراکندگی، میزان بهبود نتایج در خصیصه‌های 2-gram نسبت به 1-gram بیشتر است.

همان‌طور که در جدول 4-9 مشاهده می‌شود با اعمال فیلتر روی تمام خصیصه‌ها میانگین دقت برای دسته اول از 82.1 به 77.1 درصد و برای دسته دوم از 87.4 به 84.9 درصد کاهش یافته‌است. اما برای دسته سوم از 20.7 به 61.6 درصد و برای دسته چهارم از 69.2 به 70.6 درصد افزایش پیدا کرده‌است. میانگین 20.7 درصد نتیجه بدست آمده برای ستون شماره 3 در جدول 4-7 نشان می‌دهد خصیصه‌ها در این دسته نسبت به سایر دسته‌ها پراکندگی بیشتری دارند. در مرحله بعد اعمال فیلتر

فقط بر روی خصیصه‌های استخراج شده از دسته سوم و در تمام مجموعه‌داده انجام می‌گیرد. جدول 4-11 نتایج استفاده از خصیصه‌های 1-gram با فیلترکردن دسته سوم را نشان می‌دهد.

جدول 4- 11 نتایج استفاده از خصیصه‌های 1-gram با فیلترکردن دسته سوم

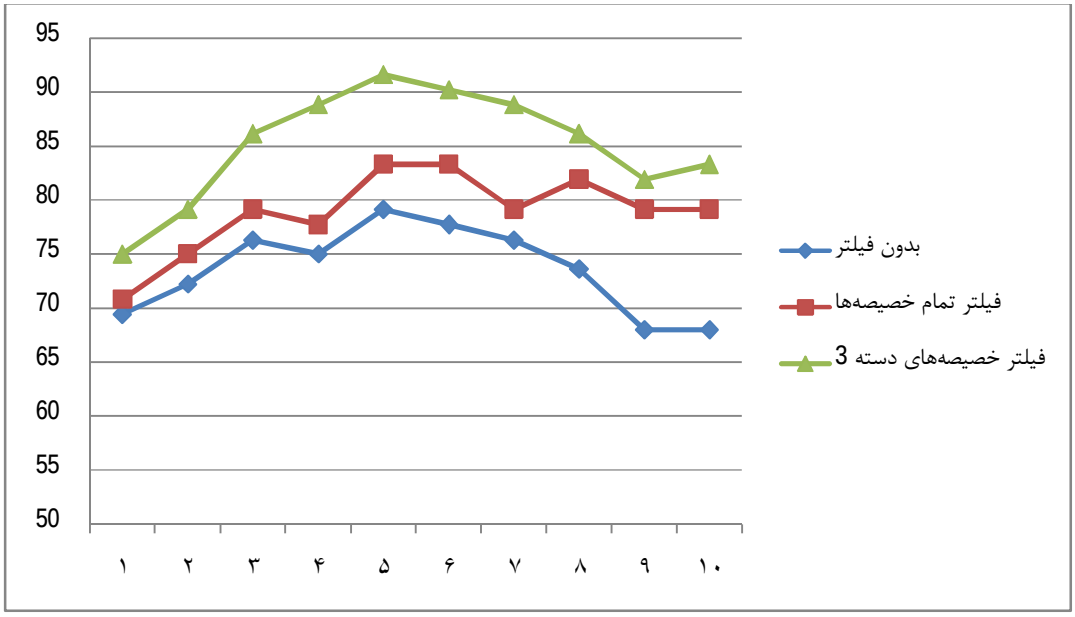
شماره دسته / تعداد تکرار r	1	2	3	4	5	کل
1	77.7	68.7	66.6	64.2	100	75.0
2	88.8	81.2	66.6	57.1	100	79.1
3	83.3	93.7	75.0	78.5	100	86.1
4	88.8	93.7	83.3	78.5	100	88.8
5	94.4	93.7	83.3	85.7	100	**91.6**
6	83.3	87.5	91.6	92.8	100	90.2
7	77.7	93.7	91.6	85.7	100	88.8
8	77.7	87.5	91.6	78.5	100	86.1
9	77.7	87.5	100	50.0	100	81.9
10	83.3	87.5	100	50.0	100	83.3
میانگین	83.27	87.47	84.96	72.1	100	85.0

نتایج استفاده از خصیصه‌های 2-gram با فیلترکردن دسته سوم در جدول 4-12 مشاهده می‌شود.

جدول 4- 12 نتایج استفاده از خصیصه‌های 2-gram با فیلترکردن دسته سوم

شماره دسته / تعداد تکرار r	1	2	3	4	5	کل
1	27.7	25.0	25.0	28.5	50.0	30.5
2	0.0	0.0	16.6	14.2	16.6	8.3
3	0.0	0.0	16.6	7.1	8.3	5.5
4	0.0	0.0	16.6	0.0	8.3	4.1
5	0.0	0.0	16.6	0.0	8.3	4.1
6	0.0	0.0	0.0	0.0	8.3	1.3
7	0.0	0.0	0.0	0.0	8.3	1.3
8	0.0	0.0	0.0	0.0	8.3	1.3

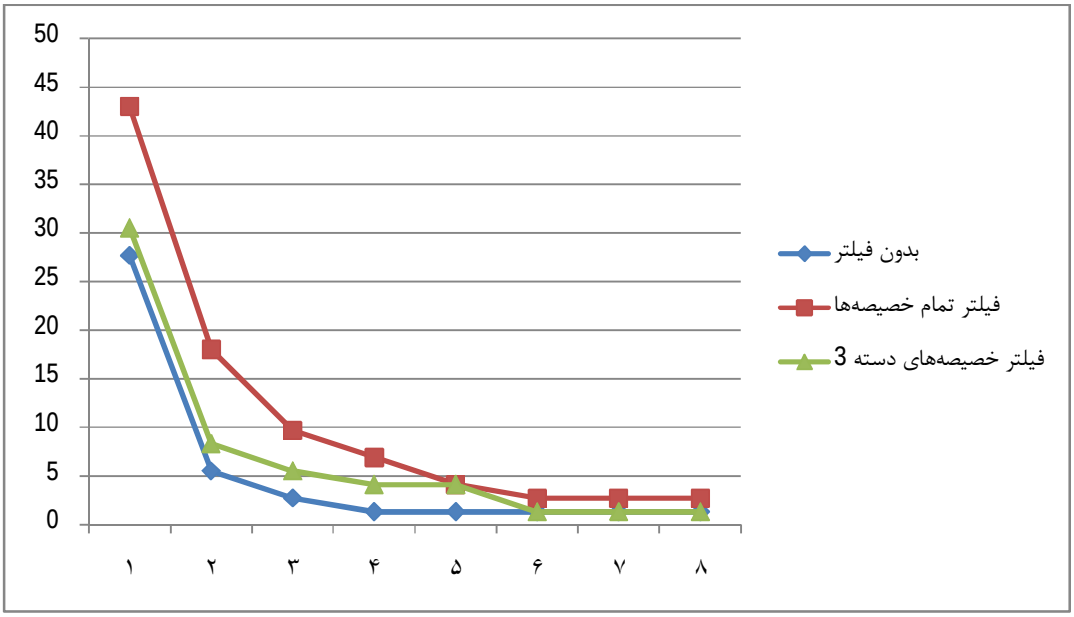
بررسی جدول‌های 4-7، 4-9 و 4-11 در ستون آخر نشان می‌دهد با استفاده از خصیصه‌های 1-gram افزایش مقدار متغیر r از 1 تا 5 به علت حذف خصیصه‌های غیرمفید منجر به بهبود نتایج شده‌است. اما مقادیر بیشتر از 5 برای متغیر r به علت حذف خصیصه‌های مفید دقت دسته‌بندی را کاهش می‌دهد. شکل 4-1 نحوه تاثیر متغیر r بر دقت دسته‌بندی با خصیصه‌های 1-gram را نشان می‌دهد.



شکل 4- 1 نحوه تاثیر متغیر r بر دقت دسته‌بندی با خصیصه‌های 1-gram

با توجه به نمودارهای شکل 4-1 در هر سه مرحله به ازای مقادیر 5 و 6 برای متغیر r بهترین نتایج بدست آمده‌است.

افزایش متغیر r با استفاده از خصیصه‌های 2-gram حساسیت آنها را نسبت به پراکندگی بیشتر کرده و همان‌طور که در ستون آخر جدول‌های 4-8، 4-10 و 4-12 مشاهده می‌شود دقت دسته‌بندی با افزایش r به صورت شدیدی کاهش پیدا می‌کند. در شکل 4-2 نحوه تاثیر متغیر r بر دقت دسته‌بندی با خصیصه‌های 2-gram نشان داده شده‌است.



شکل 4- 2 نحوه تاثیر متغیر r بر دقت دسته‌بندی با خصیصه‌های 2-gram

همان‌طور که در شکل 4-2 مشاهده می‌شود بیشترین دقت با استفاده از خصیصه‌های 2-gram به ازای r =1 بدست می‌آید.

فصل 5) نتیجه‌گیری و پیشنهادات

5- 1- نتیجه تحقیق

در این تحقیق ابتدا با کاهش پراکندگی خصیصه‌ها به صورت آماری و سپس استخراج آنها یک مجموعه خصیصه مفید برای طبقه‌بندی متون در 5 دسته احساسی ارائه شد. سپس با بهره‌گیری از مجموعه خصیصه‌هایی که انتخاب می‌شوند یک مدل مناسب برای متن ایجاد شده و برای طبقه‌بندی این مدل از روش بیز ساده استفاده شده‌است.

مجموعه خصیصه‌های 1-gram دقت بالاتری نسبت به 2-gram داشتند، به این دلیل که حساسیت کمتری به پراکندگی خصیصه‌ها دارند. همچنین نشان دادیم اعمال فیلتر روی خصیصه‌ها موجب بهبود نتایج در هر دو نوع خصیصه و به‌خصوص در خصیصه‌های 2-gram می‌شود.

با توجه به نتایج بدست آمده، روشی که برای اعمال فیلتر روی خصیصه‌ها در این تحقیق به‌کار گرفته شده، می‌تواند در مواقع عدم دسترسی به واژه‌نامه معنایی مورد استفاده قرار گیرد. البته نتیجه مطلوب در این روشِ اعمال فیلتر، مستلزم دقت در مقداردهی به متغیر n می‌باشد.

مقادیر بیشتر از یک برای متغیر r با افزایش حساسیت خصیصه‌های 2-gram نسبت به پراکندگی، باعث افت شدید در نتایج شده اما افزایش مقدار r با خصیصه‌های 1-gram تا حد معینی موجب بهبود دقت سیستم می‌شود.

با توجه به جدول 4-11 بهترین دقت برای روش پیشنهادی در این تحقیق 91.6 درصد بدست آمده‌است. این دقت با خصیصه‌های 1-gram و اعمال فیلتر بر خصیصه‌های دسته سوم و مقدار متغیر r=5 و n=7 حاصل می‌شود.

5- 2- پیشنهاد کارهای آینده

برای افزایش دقت این تحقیق می‌توان از یک روش انتخاب خصیصه مناسب‌تر بهره برد تا بتوان از آن برای بهبود کارایی طبقه‌بندی استفاده کرد. همچنین در این تحقیق به حل مشکل پراکندگی خصیصه‌های N-gram به صورت آماری پرداخته ایم. مساله پراکندگی در خصیصه‌های 2-gram و 3-gram مشکلات بیشتری را به همراه خواهد داشت. برای بهبود عملکرد این روش می‌توان در جهت پیداکردن یک روش مناسب‌تر برای اعمال فیلتر روی خصیصه‌ها تلاش کرد. روش مناسب‌تر برای اعمال فیلتر می‌تواند از واژه‌نامه لغوی- معنایی استفاده کند. لذا بهبود روش انتخاب خصیصه و اعمال فیلتر روی خصیصه‌ها می‌تواند به عنوان کارهای آینده در نظر گرفته شود.

پیوست: کد الگوریتم‌ها

```
package readwritetextfile;//Feature Selection

/**
 *
 * @author oveisarghiany
 */
import java.io.FileReader;
import java.io.FileWriter;
import java.io.BufferedReader;
import java.io.PrintWriter;
import java.io.IOException;

public class ReadWriteTextFile {

private static void doReadWriteTextFile() {

try {

// input/output file names
String inputFileName = "train4.txt";
String inputFileName1 = "stop1.txt";
String outputFileName = "result.txt";

// Create FileReader Object
FileReader inputFileReader = new FileReader(inputFileName);
FileReader inputFileReader1 = new FileReader(inputFileName1);
FileWriter outputFileReader = new FileWriter(outputFileName);

// Create Buffered/PrintWriter Objects
BufferedReader inputStream = new BufferedReader(inputFileReader);
BufferedReader inputStream1 = new BufferedReader(inputFileReader1);
PrintWriter outputStream = new PrintWriter(outputFileReader);



String stop=inputStream1.readLine();
String word="";
int i2;int c2=0;int c3=0;int c6=0;
int [] count=new int[50000];
int [] lenght=new int[50000];
String line;
while ((line = inputStream.readLine()) != null)
{
    String [] c=line.split(" ");c6++;
    for (int i1 = 0; i1 < c.length; i1++)
        {
```

```
        String w = line.substring(0,line.indexOf(" "));c2++;
        line=line.replaceFirst(w+" ", "");
        if (!stop.contains(w+" "))
           {
           c3++;
           if (word.contains(w+",")) {i2=word.indexOf(w+",");count[i2]++;}
           else
{word=word.concat(w+",");count[word.indexOf(w+",")]=1;lenght[word.indexOf(w
+",")]=w.length();}
           }
        }
}
String w1;int c4=0;int c5=0;int t=30;
for (int c1 = 0; c1 < count.length; c1++)
{
   if (count[c1]>t & !(word.isEmpty())){ c4=c4+count[c1];}
}
double p;double p1;
for (int c1 = 0; c1 < count.length; c1++)
{
   if (count[c1]>t & !(word.isEmpty()))
   {
      p=(float)count[c1]/c4*100;
      p1=Math.round(p * 10.0) / 10.0;
      w1=word.substring(c1, c1+lenght[c1]);
      System.out.print(w1+"-"+count[c1]+"--"+p1+"   ");
      outputStream.println(w1+"-"+count[c1]+"-"+p1+"٪");
      c5++;
   }
}
System.out.println();
System.out.println(c6+"-->"+c2+"-->"+c3+"-->"+c4+"-->"+c5);
outputStream.println();
outputStream.println();
outputStream.println("تعداد نمونه: "+c6);
outputStream.println("تعداد کل کلمات: "+c2);
outputStream.println("تعداد بعد از حذف ایست واژه: "+c3);
outputStream.println("بار با احتساب تکرار تعداد کلمات با حداقل تکرار "+(t+1)+": "+c4);
outputStream.println("بار بدون احتساب تکرار تعداد کلمات با حداقل تکرار "+(t+1)+": "+c5);
outputStream.close();
} catch (IOException e) {
System.out.println("IOException:");
e.printStackTrace();
}
}
```

```
package sentiment.analyze;//Calculate the Probability 1-gram

/**
 *
 * @author oveisarghiany
 */
import java.io.FileReader;
import java.io.FileWriter;
import java.io.BufferedReader;
import java.io.PrintWriter;
import java.io.IOException;


public class SentimentAnalyze {

    private static void doSentimentAnalyze(){
    try {
    String inputFileNamestop = "stop1.txt";
    FileReader inputFileReaderstop = new FileReader(inputFileNamestop);
    BufferedReader inputStreamstop = new
BufferedReader(inputFileReaderstop);
    String stop=inputStreamstop.readLine();
    String line;
    int i2;int c4 = 0;int t=4;


    //Analyze train1
    String inputFileName = "train1.txt";
    FileReader inputFileReader = new FileReader(inputFileName);
    BufferedReader inputStream = new BufferedReader(inputFileReader);
    String word1="";
    int [] count1=new int[50000];
    int [] lenght1=new int[50000];
    double [] per1=new double[50000];
    while ((line = inputStream.readLine()) != null)
{
    String [] c=line.split(" ");
    for (int i1 = 0; i1 < c.length; i1++)
        {
        String w = line.substring(0,line.indexOf(" "));
        line=line.replaceFirst(w+" ", "");
        if (!stop.contains(" "+w+" "))
            {
            if (word1.contains(w+",")) {i2=word1.indexOf(w+",");count1[i2]++;}
            else
{word1=word1.concat(w+",");count1[word1.indexOf(w+",")]=1;lenght1[word1.ind
exOf(w+",")]=w.length();}
            }
        }
```

```
}
c4=0;
for (int c1 = 0; c1 < count1.length; c1++)
{
   if (count1[c1]>t & !(word1.isEmpty())){ c4=c4+count1[c1];}
}
for (int c1 = 0; c1 < count1.length; c1++)
{
   if (count1[c1]>t & !(word1.isEmpty())){ per1[c1]=(float)count1[c1]/c4;}
}


   //Analyze train2
   inputFileName = "train2.txt";
   inputFileReader = new FileReader(inputFileName);
   inputStream = new BufferedReader(inputFileReader);
   String word2="";
   int [] count2=new int[50000];
   int [] lenght2=new int[50000];
   double [] per2=new double[50000];
   while ((line = inputStream.readLine()) != null)
{
   String [] c=line.split(" ");
   for (int i1 = 0; i1 < c.length; i1++)
      {
      String w = line.substring(0,line.indexOf(" "));
      line=line.replaceFirst(w+" ", "");
      if (!stop.contains(" "+w+" "))
         {
         if (word2.contains(w+",")) {i2=word2.indexOf(w+",");count2[i2]++;}
         else
{word2=word2.concat(w+",");count2[word2.indexOf(w+",")]=1;lenght2[word2.ind
exOf(w+",")]=w.length();}
         }
      }
}
c4=0;
for (int c1 = 0; c1 < count2.length; c1++)
{
   if (count2[c1]>t & !(word2.isEmpty())){ c4=c4+count2[c1];}
}
for (int c1 = 0; c1 < count2.length; c1++)
{
   if (count2[c1]>t & !(word2.isEmpty())){ per2[c1]=(float)count2[c1]/c4;}
}


   //Analyze train3
   inputFileName = "train3.txt";
```

```
    inputFileReader = new FileReader(inputFileName);
    inputStream = new BufferedReader(inputFileReader);
    String word3="";
    int [] count3=new int[50000];
    int [] lenght3=new int[50000];
    double [] per3=new double[50000];
    while ((line = inputStream.readLine()) != null)
{
    String [] c=line.split(" ");
    for (int i1 = 0; i1 < c.length; i1++)
        {
        String w = line.substring(0,line.indexOf(" "));
        line=line.replaceFirst(w+" ", "");
        if (!stop.contains(" "+w+" "))
            {
            if (word3.contains(w+",")) {i2=word3.indexOf(w+",");count3[i2]++;}
            else
{word3=word3.concat(w+",");count3[word3.indexOf(w+",")]=1;lenght3[word3.ind
exOf(w+",")]=w.length();}
            }
        }
}
c4=0;
for (int c1 = 0; c1 < count3.length; c1++)
{
    if (count3[c1]>t & !(word3.isEmpty())){ c4=c4+count3[c1];}
}
for (int c1 = 0; c1 < count3.length; c1++)
{
    if (count3[c1]>t & !(word3.isEmpty())){ per3[c1]=(float)count3[c1]/c4;}
}

    //Analyze train4
    inputFileName = "train4.txt";
    inputFileReader = new FileReader(inputFileName);
    inputStream = new BufferedReader(inputFileReader);
    String word4="";
    int [] count4=new int[50000];
    int [] lenght4=new int[50000];
    double [] per4=new double[50000];
    while ((line = inputStream.readLine()) != null)
{
    String [] c=line.split(" ");
    for (int i1 = 0; i1 < c.length; i1++)
        {
        String w = line.substring(0,line.indexOf(" "));
        line=line.replaceFirst(w+" ", "");
        if (!stop.contains(" "+w+" "))
            {
```

```
            if (word4.contains(w+",")) {i2=word4.indexOf(w+",");count4[i2]++;}
            else
{word4=word4.concat(w+",");count4[word4.indexOf(w+",")]=1;lenght4[word4.ind
exOf(w+",")]=w.length();}
            }
        }
}
c4=0;
for (int c1 = 0; c1 < count4.length; c1++)
{
    if (count4[c1]>t & !(word4.isEmpty())){ c4=c4+count4[c1];}
}
for (int c1 = 0; c1 < count4.length; c1++)
{
    if (count4[c1]>t & !(word4.isEmpty())){ per4[c1]=(float)count4[c1]/c4;}
}

    //Analyze train5
    inputFileName = "train5.txt";
    inputFileReader = new FileReader(inputFileName);
    inputStream = new BufferedReader(inputFileReader);
    String word5="";
    int [] count5=new int[50000];
    int [] lenght5=new int[50000];
    double [] per5=new double[50000];
    while ((line = inputStream.readLine()) != null)
{
    String [] c=line.split(" ");
    for (int i1 = 0; i1 < c.length; i1++)
        {
        String w = line.substring(0,line.indexOf(" "));
        line=line.replaceFirst(w+" ", "");
        if (!stop.contains(" "+w+" "))
            {
            if (word5.contains(w+",")) {i2=word5.indexOf(w+",");count5[i2]++;}
            else
{word5=word5.concat(w+",");count5[word5.indexOf(w+",")]=1;lenght5[word5.ind
exOf(w+",")]=w.length();}
            }
        }
}
c4=0;
for (int c1 = 0; c1 < count5.length; c1++)
{
    if (count5[c1]>t & !(word5.isEmpty())){ c4=c4+count5[c1];}
}
for (int c1 = 0; c1 < count5.length; c1++)
{
    if (count5[c1]>t & !(word5.isEmpty())){ per5[c1]=(float)count5[c1]/c4;}
```

```
}


//Analyze sentence
   double p1 = 0,p2 = 0,p3 = 0,p4 = 0,p5 =
0,pb1,pb2,pb3,pb4,pb5,pb,m,m1,pz1,pz2,pz3,pz4,pz5,pf,pf1,pf2,pf3,pf4,pf5,t1=
0,t2=0,t3=0,t4=0,t5=0;int iii=1,ii1,ii2,ii3,ii4,ii5;String line1;
   inputFileName = "test.txt";
   inputFileReader = new FileReader(inputFileName);
   inputStream = new BufferedReader(inputFileReader);
   String outputFileName = "result.txt";
   FileWriter outputFileReader = new FileWriter(outputFileName);
   PrintWriter outputStream = new PrintWriter(outputFileReader);
   while ((line = inputStream.readLine()) != null) {

//System.out.println(line);//outputStream.println(line);outputStream.println();outp
utStream.println();
   pz1=1;pz2=1;pz3=1;pz4=1;pz5=1;
   String [] c=line.split(" ");
   for (int i1 = 0; i1 < c.length; i1++){
      String w = line.substring(0,line.indexOf(" "));
      line=line.replaceFirst(w+" ", "");
      if (!stop.contains(" "+w+" "))
         {
         p1=0;p2=0;p3=0;p4=0;p5=0;
         if (word1.contains(w+",")) {ii1=word1.indexOf(w+",");p1=per1[ii1];}
         if (word2.contains(w+",")) {ii2=word2.indexOf(w+",");p2=per2[ii2];}
         if (word3.contains(w+",")) {ii3=word3.indexOf(w+",");p3=per3[ii3];}
         if (word4.contains(w+",")) {ii4=word4.indexOf(w+",");p4=per4[ii4];}
         if (word5.contains(w+",")) {ii5=word5.indexOf(w+",");p5=per5[ii5];}
         //smoothing
         m = Math.max(p1, Math.max(p2, Math.max(p3, Math.max(p4, p5) ) ) );
         if (m==p1) {p1=p1-
4*p1/1000;p2=p2+p1/1000;p3=p3+p1/1000;p4=p4+p1/1000;p5=p5+p1/1000;}
else
         if (m==p2) {p2=p2-
4*p2/1000;p1=p1+p2/1000;p3=p3+p2/1000;p4=p4+p2/1000;p5=p5+p2/1000;}
else
         if (m==p3) {p3=p3-
4*p3/1000;p1=p1+p3/1000;p2=p2+p3/1000;p4=p4+p3/1000;p5=p5+p3/1000;}
else
         if (m==p4) {p4=p4-
4*p4/1000;p1=p1+p4/1000;p2=p2+p4/1000;p3=p3+p4/1000;p5=p5+p4/1000;}
else
         if (m==p5) {p5=p5-
4*p5/1000;p1=p1+p5/1000;p2=p2+p5/1000;p3=p3+p5/1000;p4=p4+p5/1000;}

         //calculate probability
         pb=p1+p2+p3+p4+p5;
```

```
            pb1=p1/pb;pb2=p2/pb;pb3=p3/pb;pb4=p4/pb;pb5=p5/pb;
            if (m==0) {pb1=pb2=pb3=pb4=pb5=0.001;}

pz1=pz1*pb1*10;pz2=pz2*pb2*10;pz3=pz3*pb3*10;pz4=pz4*pb4*10;pz5=pz5*p
b5*10;
            System.out.println(w+" p1= "+Math.round(pb1 * 1000.0) / 1000.0+"
p2= "+Math.round(pb2 * 1000.0) / 1000.0+"    p3= "+Math.round(pb3 * 1000.0) /
1000.0+"    p4= "+Math.round(pb4 * 1000.0) / 1000.0+"    p5= "+Math.round(pb5
* 1000.0) / 1000.0);
            //System.out.println(w);
            //System.out.println(p1+" "+p2+" "+p3+" "+p4+" "+p5);
            //System.out.println();
            //outputStream.println(w+"    p1= "+Math.round(pb1 * 100.0) / 100.0+"
p2= "+Math.round(pb2 * 100.0) / 100.0+"       p3= "+Math.round(pb3 * 100.0) /
100.0+"       p4= "+Math.round(pb4 * 100.0) / 100.0+"       p5= "+Math.round(pb5
* 100.0) / 100.0);
            //System.out.println(w+p1+p2+p3+p4+p5);
            //outputStream.println();




          }
        }
//System.out.println(pz1+"  "+pz2+"  "+pz3+"  "+pz4+"  "+pz5);
m1 = Math.max(pz1, Math.max(pz2, Math.max(pz3, Math.max(pz4, pz5) ) ) );
System.out.print(iii+"--");
if (pz1==pz2 & pz2==pz3 & pz3==pz4 & pz4==pz5)
{System.out.println("unknown");} else{
if (m1==pz1) {{if (1<= iii & iii<=18)
{t1++;}}System.out.println("Sadness");outputStream.println("Sadness");} else
if (m1==pz2) {{if (19<= iii & iii<=34)
{t2++;}}System.out.println("Happiness");outputStream.println("Happiness");}
else
if (m1==pz3) {{if (35<= iii & iii<=46)
{t3++;}}System.out.println("Anger");outputStream.println("Anger");} else
if (m1==pz4) {{if (47<= iii & iii<=60)
{t4++;}}System.out.println("Disgust");outputStream.println("Disgust");} else
if (m1==pz5) {{if (61<= iii & iii<=72)
{t5++;}}System.out.println("Fear");outputStream.println("Fear");}
}

iii++;
//outputStream.println("p1:Sadness   p2:Happiness   p3:Anger   p4:Disgust
P5:Fear");

   }
   System.out.println(t1+"--"+t2+"--"+t3+"--"+t4+"--"+t5+"-------
"+(t1+t2+t3+t4+t5));
```

```
    pf1=t1/18;pf2=t2/16;pf3=t3/12;pf4=t4/14;pf5=t5/12;
    pf=(t1+t2+t3+t4+t5)/72;
    System.out.println(pf1+"---"+pf2+"---"+pf3+"---"+pf4+"---"+pf5);
    System.out.println(pf);
    outputStream.close();
}
catch (IOException e) {
System.out.println("IOException:");
e.printStackTrace();
}



    }
    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
    doSentimentAnalyze();

    }

}
```

```
package readwrite;//Calculate the Probability 2-gram

/**
 *
 * @author oveisarghiany
 */
import java.io.FileReader;
import java.io.FileWriter;
import java.io.BufferedReader;
import java.io.PrintWriter;
import java.io.IOException;
public class ReadWrite {
    private static void ReadWrite(){
    try{
    String inputFileNamestop = "stop1.txt";
    FileReader inputFileReaderstop = new FileReader(inputFileNamestop);
    BufferedReader inputStreamstop = new
BufferedReader(inputFileReaderstop);
    String stop=inputStreamstop.readLine();
    String line;
```

```
    String line1;
    int i2;int t=5;int c4 = 0;


//Analyze train1
    String inputFileName = "train1.txt";
    FileReader inputFileReader = new FileReader(inputFileName);
    BufferedReader inputStream = new BufferedReader(inputFileReader);
    String word1="";
    int [] count1=new int[50000];
    int [] lenght1=new int[50000];
    double [] per1=new double[50000];

//Remove Stop Words 1
    line = inputStream.readLine();
    line1="";
{
    String [] c=line.split(" ");
    for (int i1 = 0; i1 < c.length; i1++)
    {
    String w = line.substring(0,line.indexOf(" "));
    line=line.replaceFirst(w+" ", "");
    if (!stop.contains(" "+w+" "))
    {
    line1=line1.concat(w+" ");
    }
    }
    //Analyze 1

    String [] c1=line1.split(" ");
    String w2;
    for (int i1 = 0; i1 < c1.length; i1++)
        {
        String w1 = line1.substring(0,line1.indexOf(" "));
        line1=line1.replaceFirst(w1+" ", "");
        if (line1.isEmpty()) {w2=".";}  else {w2 = line1.substring(0,line1.indexOf("
"));}
        if (word1.contains(w1+":"+w2+","))
{i2=word1.indexOf(w1+":"+w2+",");count1[i2]++;}
        else
{word1=word1.concat(w1+":"+w2+",");count1[word1.indexOf(w1+":"+w2+",")]=1;l
enght1[word1.indexOf(w1+":"+w2+",")]=w1.length()+w2.length()+1;}

        }
}
c4=0;
for (int c1 = 0; c1 < count1.length; c1++)
{
    if (count1[c1]>t & !(word1.isEmpty())){ c4=c4+count1[c1];}
```

```
}
for (int c1 = 0; c1 < count1.length; c1++)
{
    if (count1[c1]>t & !(word1.isEmpty())){ per1[c1]=(float)count1[c1]/c4;}
}


//Analyze train2
    inputFileName = "train2.txt";
    inputFileReader = new FileReader(inputFileName);
    inputStream = new BufferedReader(inputFileReader);
    String word2="";
    int [] count2=new int[50000];
    int [] lenght2=new int[50000];
    double [] per2=new double[50000];

//Remove Stop Words 2
    line = inputStream.readLine();
    line1="";
{
    String [] c=line.split(" ");
    for (int i1 = 0; i1 < c.length; i1++)
    {
    String w = line.substring(0,line.indexOf(" "));
    line=line.replaceFirst(w+" ", "");
    if (!stop.contains(" "+w+" "))
    {
    line1=line1.concat(w+" ");
    }
    }
    //Analyze 2

    String [] c1=line1.split(" ");
    String w2;
    for (int i1 = 0; i1 < c1.length; i1++)
        {
        String w1 = line1.substring(0,line1.indexOf(" "));
        line1=line1.replaceFirst(w1+" ", "");
        if (line1.isEmpty()) {w2=".";}  else {w2 = line1.substring(0,line1.indexOf("
"));}
        if (word2.contains(w1+":"+w2+","))
{i2=word2.indexOf(w1+":"+w2+",");count2[i2]++;}
        else
{word2=word2.concat(w1+":"+w2+",");count2[word2.indexOf(w1+":"+w2+",")]=1;l
enght2[word2.indexOf(w1+":"+w2+",")]=w1.length()+w2.length()+1;}

        }
}
c4=0;
```

```
for (int c1 = 0; c1 < count2.length; c1++)
{
   if (count2[c1]>t & !(word2.isEmpty())){ c4=c4+count2[c1];}
}
for (int c1 = 0; c1 < count2.length; c1++)
{
   if (count2[c1]>t & !(word2.isEmpty())){ per2[c1]=(float)count2[c1]/c4;}
}


//Analyze train3
   inputFileName = "train3.txt";
   inputFileReader = new FileReader(inputFileName);
   inputStream = new BufferedReader(inputFileReader);
   String word3="";
   int [] count3=new int[50000];
   int [] lenght3=new int[50000];
   double [] per3=new double[50000];

//Remove Stop Words 3
   line = inputStream.readLine();
   line1="";
{
   String [] c=line.split(" ");
   for (int i1 = 0; i1 < c.length; i1++)
   {
   String w = line.substring(0,line.indexOf(" "));
   line=line.replaceFirst(w+" ", "");
   if (!stop.contains(" "+w+" "))
   {
   line1=line1.concat(w+" ");
   }
   }
   //Analyze 3

   String [] c1=line1.split(" ");
   String w2;
   for (int i1 = 0; i1 < c1.length; i1++)
      {
      String w1 = line1.substring(0,line1.indexOf(" "));
      line1=line1.replaceFirst(w1+" ", "");
      if (line1.isEmpty()) {w2=".";}  else {w2 = line1.substring(0,line1.indexOf("
"));}
      if (word3.contains(w1+":"+w2+","))
{i2=word3.indexOf(w1+":"+w2+",");count3[i2]++;}
      else
{word3=word3.concat(w1+":"+w2+",");count3[word3.indexOf(w1+":"+w2+",")]=1;l
enght3[word3.indexOf(w1+":"+w2+",")]=w1.length()+w2.length()+1;}
```

```
        }
}
c4=0;
for (int c1 = 0; c1 < count3.length; c1++)
{
    if (count3[c1]>t & !(word3.isEmpty())){ c4=c4+count3[c1];}
}
for (int c1 = 0; c1 < count3.length; c1++)
{
    if (count3[c1]>t & !(word3.isEmpty())){ per3[c1]=(float)count3[c1]/c4;}
}

//Analyze train4
    inputFileName = "train4.txt";
    inputFileReader = new FileReader(inputFileName);
    inputStream = new BufferedReader(inputFileReader);
    String word4="";
    int [] count4=new int[50000];
    int [] lenght4=new int[50000];
    double [] per4=new double[50000];

//Remove Stop Words 4
    line = inputStream.readLine();
    line1="";
{
    String [] c=line.split(" ");
    for (int i1 = 0; i1 < c.length; i1++)
    {
    String w = line.substring(0,line.indexOf(" "));
    line=line.replaceFirst(w+" ", "");
    if (!stop.contains(" "+w+" "))
    {
    line1=line1.concat(w+" ");
    }
    }
    //Analyze 4

    String [] c1=line1.split(" ");
    String w2;
    for (int i1 = 0; i1 < c1.length; i1++)
        {
        String w1 = line1.substring(0,line1.indexOf(" "));
        line1=line1.replaceFirst(w1+" ", "");
        if (line1.isEmpty()) {w2=".";}  else {w2 = line1.substring(0,line1.indexOf("
"));}
        if (word4.contains(w1+":"+w2+","))
{i2=word4.indexOf(w1+":"+w2+",");count4[i2]++;}
        else
{word4=word4.concat(w1+":"+w2+",");count4[word4.indexOf(w1+":"+w2+",")]=1;l
```

```
enght4[word4.indexOf(w1+":"+w2+",")]=w1.length()+w2.length()+1;}

        }
}
c4=0;
for (int c1 = 0; c1 < count4.length; c1++)
{
    if (count4[c1]>t & !(word4.isEmpty())){ c4=c4+count4[c1];}
}
for (int c1 = 0; c1 < count4.length; c1++)
{
    if (count4[c1]>t & !(word4.isEmpty())){ per4[c1]=(float)count4[c1]/c4;}
}

//Analyze train5
    inputFileName = "train5.txt";
    inputFileReader = new FileReader(inputFileName);
    inputStream = new BufferedReader(inputFileReader);
    String word5="";
    int [] count5=new int[100000];
    int [] lenght5=new int[100000];
    double [] per5=new double[100000];

//Remove Stop Words 5
    line = inputStream.readLine();
    line1="";
{
    String [] c=line.split(" ");
    for (int i1 = 0; i1 < c.length; i1++)
    {
    String w = line.substring(0,line.indexOf(" "));
    line=line.replaceFirst(w+" ", "");
    if (!stop.contains(" "+w+" "))
    {
    line1=line1.concat(w+" ");
    }
    }
    //Analyze 5

    String [] c1=line1.split(" ");
    String w2;
    for (int i1 = 0; i1 < c1.length; i1++)
        {
        String w1 = line1.substring(0,line1.indexOf(" "));
        line1=line1.replaceFirst(w1+" ", "");
        if (line1.isEmpty()) {w2=".";}  else {w2 = line1.substring(0,line1.indexOf("
"));}
        if (word5.contains(w1+":"+w2+","))
{i2=word5.indexOf(w1+":"+w2+",");count5[i2]++;}
```

```
        else
{word5=word5.concat(w1+":"+w2+",");count5[word5.indexOf(w1+":"+w2+",")]=1;l
enght5[word5.indexOf(w1+":"+w2+",")]=w1.length()+w2.length()+1;}

        }
}
c4=0;
for (int c1 = 0; c1 < count5.length; c1++)
{
    if (count5[c1]>t & !(word5.isEmpty())){ c4=c4+count5[c1];}
}
for (int c1 = 0; c1 < count5.length; c1++)
{
    if (count5[c1]>t & !(word5.isEmpty())){ per5[c1]=(float)count5[c1]/c4;}
}
//System.out.println(word1);
//System.out.println(word2);
//System.out.println(word3);
//System.out.println(word4);
//System.out.println(word5);

//Analyze sentence
    double p1 = 0,p2 = 0,p3 = 0,p4 = 0,p5 =
0,pb1,pb2,pb3,pb4,pb5,pb,m,m1,pz1,pz2,pz3,pz4,pz5,pf,pf1,pf2,pf3,pf4,pf5,t1=
0,t2=0,t3=0,t4=0,t5=0;int iii=1,ii1,ii2,ii3,ii4,ii5;
    inputFileName = "test.txt";
    inputFileReader = new FileReader(inputFileName);
    inputStream = new BufferedReader(inputFileReader);
    String outputFileName = "result.txt";
    FileWriter outputFileReader = new FileWriter(outputFileName);
    PrintWriter outputStream = new PrintWriter(outputFileReader);
    while ((line = inputStream.readLine()) != null) {

//Remove Stop Words test
    line1="";

    String [] c=line.split(" ");
    for (int i1 = 0; i1 < c.length; i1++)
    {
    String w = line.substring(0,line.indexOf(" "));
    line=line.replaceFirst(w+" ", "");
    if (!stop.contains(" "+w+" "))
    {
    line1=line1.concat(w+" ");
    }
    }
//find token
    pz1=1;pz2=1;pz3=1;pz4=1;pz5=1;
    String [] c1=line1.split(" ");
```

```
   String w2;
   for (int i1 = 0; i1 < c1.length; i1++){

       String w1 = line1.substring(0,line1.indexOf(" "));
       line1=line1.replaceFirst(w1+" ", "");
       if (line1.isEmpty()) {w2=".";}  else {w2 = line1.substring(0,line1.indexOf("
"));}
           p1=0;p2=0;p3=0;p4=0;p5=0;
           if (word1.contains(w1+":"+w2+","))
{ii1=word1.indexOf(w1+":"+w2+",");p1=per1[ii1];}
           if (word2.contains(w1+":"+w2+","))
{ii2=word2.indexOf(w1+":"+w2+",");p2=per2[ii2];}
           if (word3.contains(w1+":"+w2+","))
{ii3=word3.indexOf(w1+":"+w2+",");p3=per3[ii3];}
           if (word4.contains(w1+":"+w2+","))
{ii4=word4.indexOf(w1+":"+w2+",");p4=per4[ii4];}
           if (word5.contains(w1+":"+w2+","))
{ii5=word5.indexOf(w1+":"+w2+",");p5=per5[ii5];}
           //smoothing
           m = Math.max(p1, Math.max(p2, Math.max(p3, Math.max(p4, p5) ) ) );
           if (m==p1) {p1=p1-
4*p1/100;p2=p2+p1/100;p3=p3+p1/100;p4=p4+p1/100;p5=p5+p1/100;} else
           if (m==p2) {p2=p2-
4*p2/100;p1=p1+p2/100;p3=p3+p2/100;p4=p4+p2/100;p5=p5+p2/100;} else
           if (m==p3) {p3=p3-
4*p3/100;p1=p1+p3/100;p2=p2+p3/100;p4=p4+p3/100;p5=p5+p3/100;} else
           if (m==p4) {p4=p4-
4*p4/100;p1=p1+p4/100;p2=p2+p4/100;p3=p3+p4/100;p5=p5+p4/100;} else
           if (m==p5) {p5=p5-
4*p5/100;p1=p1+p5/100;p2=p2+p5/100;p3=p3+p5/100;p4=p4+p5/100;}

           //calculate probability
           pb=p1+p2+p3+p4+p5;
           pb1=p1/pb;pb2=p2/pb;pb3=p3/pb;pb4=p4/pb;pb5=p5/pb;
           if (m==0) {pb1=pb2=pb3=pb4=pb5=0.001;}

pz1=pz1*pb1*10;pz2=pz2*pb2*10;pz3=pz3*pb3*10;pz4=pz4*pb4*10;pz5=pz5*p
b5*10;
           //System.out.println(w+" p1= "+Math.round(pb1 * 1000.0) / 1000.0+"
p2= "+Math.round(pb2 * 1000.0) / 1000.0+"    p3= "+Math.round(pb3 * 1000.0) /
1000.0+"    p4= "+Math.round(pb4 * 1000.0) / 1000.0+"    p5= "+Math.round(pb5
* 1000.0) / 1000.0);
           //System.out.println(w1+":"+w2+","+pb1+" "+pb2+" "+pb3+" "+pb4+"
"+pb5);
           //System.out.println();
           //outputStream.println(w+"    p1= "+Math.round(pb1 * 100.0) / 100.0+"
p2= "+Math.round(pb2 * 100.0) / 100.0+"       p3= "+Math.round(pb3 * 100.0) /
100.0+"       p4= "+Math.round(pb4 * 100.0) / 100.0+"       p5= "+Math.round(pb5
* 100.0) / 100.0);
```

```
            //System.out.println(w+p1+p2+p3+p4+p5);
            //outputStream.println();




        }
//System.out.println(pz1+"  "+pz2+"  "+pz3+"  "+pz4+"  "+pz5);
m1 = Math.max(pz1, Math.max(pz2, Math.max(pz3, Math.max(pz4, pz5) ) )
);//System.out.println(m1);
System.out.print(iii+"--");
if (pz1==pz2 & pz2==pz3 & pz3==pz4 & pz4==pz5)
{System.out.println("unknown");} else{
if (m1==pz1) {{if (1<= iii & iii<=18)
{t1++;}}System.out.println("Sadness");outputStream.println("Sadness");} else
if (m1==pz2) {{if (19<= iii & iii<=34)
{t2++;}}System.out.println("Happiness");outputStream.println("Happiness");}
else
if (m1==pz3) {{if (35<= iii & iii<=46)
{t3++;}}System.out.println("Anger");outputStream.println("Anger");} else
if (m1==pz4) {{if (47<= iii & iii<=60)
{t4++;}}System.out.println("Disgust");outputStream.println("Disgust");} else
if (m1==pz5) {{if (61<= iii & iii<=72)
{t5++;}}System.out.println("Fear");outputStream.println("Fear");}
}
iii++;
//outputStream.println("p1:Sadness  p2:Happiness  p3:Anger  p4:Disgust
P5:Fear");

    }
    System.out.println(t1+"--"+t2+"--"+t3+"--"+t4+"--"+t5+"-------
"+(t1+t2+t3+t4+t5));
    pf1=t1/18;pf2=t2/16;pf3=t3/12;pf4=t4/14;pf5=t5/12;
    pf=(t1+t2+t3+t4+t5)/72;
    System.out.println(pf1+"---"+pf2+"---"+pf3+"---"+pf4+"---"+pf5);
    System.out.println(pf);




    }catch (IOException e) {
System.out.println("IOException:");
e.printStackTrace();
}
```

```
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        ReadWrite();
    }

}
```

مراجع و منابع

[1] R. Plutchik; (2001) “The Nature of emotions”, **American Scientist**, Vol. 89, pp. 344-350

[2] L. Bing, Z. Lei; (2012) “**Mining Text Data**” Vol. 1, Springer US, USA, pp 415-463.

[3] S. Bethhard, H. Yu, A. Thornton, V. Hatzivassiloglou, And D. Jurafsky; (2006) “**Computing Attitude and Affect in Text: Theory and Applications**” Vol. 20, Springer, Netherlands, pp. 125-141.

[4] M. Ghiassi, J. Skinner, D. Zimbra; (2013) “Twitter brand sentiment analysis: A hybrid system using N-gram analysis and dynamic artificial neural network”, **Expert Systems with Applications**, Vol. 40, pp. 6266–6282.

[5] B. Pang, L. Lee, S. Vaithyanathan; (2002) “Thumbs up? Sentiment Classification using Machine Learning Techniques”, **Empirical Methods in Natural Language Processing (EMNLP)**, pp. 79–86.

[6] P. Bo, L. Lee; (2008) “Opinion Mining and Sentiment Analysis”, **Information Retrieval**, Vol. 2, Nos. 1–2, pp. 1–135.

[7] A. Abbasi, S. France, Z. Zhang, H. Chen; (2011) “Selecting Attributes for Sentiment Classification Using Feature Relation Networks”, **IEEE Transactions on Knowledge and Data Engineering**, pp. 447–462.

[8] P. Turney; (2002) “Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews” 40th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 417-424.

[9] J. Wiebe, T. Wilson, R. Bruce, M. Bell, and M. Martin; (2004) “Learning Subjective Language”, **Computational Linguistics**, vol. 30, no. 3, pp. 277-308.

[10] V. Ng, S. Dasgupta, S.M.N. Arifin; (2006) “Examining the Role of Linguistic Knowledge Sources in the Automatic Identification and Classification of Reviews”, Main Conference Poster Sessions, Association Computational Linguistics (ACL), pp. 611-618.

[11] M. Gamon; (2004) “Sentiment Classification on Customer Feedback Data: Noisy Data, Large Feature Vectors, and the Role of Linguistic Analysis”, Proc. 20th Int’l Conf. Computational Linguistics, pp. 841-847.

[12] A. Ahmed, H. Chen, A. Salem; (2008) “Sentiment Analysis in Multiple Languages: Feature Selection for Opinion Classification in Web Forums”, **ACM Trans. Information Systems**, vol. 26, no. 3, article no. 12.

[13] B. Agarwal, N. Mittal; (2013) “Optimal Feature Selection Methods for Sentiment Analysis”, 14th International Conference on Intelligent Text Processing and Computational Linguistics, Vol-7817, pp. 13-24.

[14] K. Tsutsumi, K. Shimada, and T. Endo; (2007) “Movie Review Classification Based on Multiple Classifier”, Proc. 21st Pacific Asia Conf. Language, Information, and Computation, pp. 481- 488.

[15] C. E. Shannon; (1948) “A Mathematical Theory of Communication” **Bell Systems Technical J**, vol. 27, no. 10, pp. 379-423.

[16] J. R. Quinlan; (1986) “Induction of Decision Trees” **Machine Learning**, vol. 1, no. 1, pp. 81-106.

[17] A. Abbasi, H. Chen, S. Thoms, T. Fu; (2008) “Affect Analysis of WebForums and Blogs Using Correlation Ensembles”, **IEEE Transactions on Knowledge and Data Engineering**, Vol. 20, no. 9, pp. 1168-1180.

[18] J. Yi, T. Nasukawa, R. Bunescu, and W. Niblack; (2003) “Sentiment Analyzer: Extracting Sentiments about a Given Topic Using Natural Language Processing Techniques” Proc. Third IEEE Int’l Conf. Data Mining, pp. 427-434.

[19] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, M. Stede; (2010) “Lexicon-Based Methods for Sentiment Analysis” **Computational Linguistics**, Vol. 37, pp. 267-307.

[20] WordNet A lexical database for English. https://wordnet.princeton.edu

[21] SentiWordNet A lexical resource for opinion mining. http://sentiwordnet.isti.cnr.it

[22] E. Andrea, S. Fabrizio; (2006) “SENTIWORDNET: A Publicly Available Lexical Resource for Opinion Mining”, In Proceedings of the 5th Conference on Language Resources and Evaluation, LREC’06, page 417-422.

[23] Z. Fei, J. Liu, G. Wu; (2004) “Sentiment Classification Using Phrase Patterns”, Proc. Fourth IEEE Int’l Conf. Computer Information Technology, pp. 1147-1152.

[24] C. Priyanka, G. Deepa; (2013) “Identifying the Best Feature Combination for Sentiment Analysis of Customer Reviews”, International Conference on Advances in Computing, Communications and Informatics (ICACCI), India, pp. 102–108.

[25] دشتبانی ش، پیله‌ور ع، (1391) "آنالیز احساسی متون فارسی"، نخستین کنفرانس بین‌المللی پردازش خط و زبان فارسی، ص 1، سمنان.

[26] M. Kirk; (2014) “**Thoughtful Machine Learning**” Vol. 1, O'Reilly Media, USA, pp 51-74.

[27] E. Riloff, S. Patwardhan, and J. Wiebe; (2006) “Feature Subsumption for Opinion Analysis”, Proc. Conf. Empirical Methods in Natural Language Processing, pp. 440-448.

[28] M. Hall, L.A. Smith; (1997) “Feature Subset Selection: A Correlation Based Filter Approach” Proc. Fourth Int’l Conf. Neural Information Processing and Intelligent Information Systems, pp. 855-858.

[29] L. Yu and H. Liu; (2003) “Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution”, Proc. 20th Int’l Conf. Machine Learning, pp. 856-863.

Abstract

Emotion is an important aspect of human behavior through which people in a society affect each other. Sentiment analysis is for providing the signs and more interaction between human and computer. In addition to sentiment analysis from face, gestures and speech, we can recognize the emotions through the written texts. Sentiment analysis is a branch of computer science and Natural Language Processing (NLP) which can help find the author's motivation.

This paper entitled "**Sentiment analysis in text using artificial intelligence techniques**" aims to classify the emotional texts and to discover the emotional mood of the author to develop a system which is smart enough and deals with human including emotions and can recognize the user's emotions.

The proposed model in this paper considers the extracted features in the text in two frames of 1-gram and 2-gram. Feature filtering is used to reduce the feature scattering and consequently increases the classification accuracy and results improvement. The proposed model is tested using the dataset of emotional texts made for this purpose. The emotional texts are categorized in five groups using Bayes method. These five groups are grief, happiness, anger, hatred and fear. The obtained results demonstrate that in the best mode, sentiment analysis accuracy using 1-gram and 2-gram features without filtering are 79.1% and 27.7%, respectively. By filtering these features and therefore by reducing the scattering, accuracy of 91.6% and 43% are obtained for 1-gram and 2-gram features.

Key words: *sentiment analysis, emotional text, classification, scattering of features, filtering of features*

Shahrood University of Technology

Faculty of Computer & IT Engineering

MSc Thesis in Computer Engineering-AI

Sentiment Analysis in Text by using AI techniques

By: oveis arghiani

Supervisor:

Dr Morteza Zahedi

August 2015