



دانشگاه تهران
دانشکده‌گان فارابی
دانشکده‌ی مهندسی
گروه مهندسی کامپیوتر

تحلیل احساسات و تشخیص نظرات هرز در بستر توئیتر

نگارش:

مهدی دهقانی - محمد رشیدخان مفرد

استاد راهنما:

دکتر کاظم فولادی قلعه

گزارش پروژه برای دریافت درجه‌ی کارشناسی
در رشته‌ی مهندسی کامپیوتر

شهریور ۱۴۰۱

اللَّهُمَّ احْمِنا

صورت جلسه داوری

قدردانی

از خدای منان سپاسگزاریم که همواره دست‌گیر ما بوده است. از کمک‌های استاد راهنما خود، جناب آقای دکتر فولادی که در طی این پروژه همیشه با صبوری به سوالات ما پاسخ می‌دادند و با راهنمایی‌های خوبشان راهگشای حل مسائلی که در طول این تحقیق به وجود می‌آمد، بودند، کمال تشکر و قدردانی را داریم. از جناب آقای دکتر قاسمی و جناب آقای مهندس توحیدی بسیار متشکریم که در راه انجام این پروژه از مشاوره‌های ارزشمند ایشان بهره برده‌ایم و اگر کمک‌ها و راهنمایی‌های ایشان نبود این پروژه به سرانجام نمی‌رسید. از جناب آقای مهندس عرب نیز بابت کمک‌های بی‌دریغ‌شان متشکریم. همچنین از خانواده‌های عزیزمان که در طول زندگی همیشه مشوق و حامی ما بوده‌اند ممنونیم.

مهدی دهقانی - محمد رشیدخان مفرد

شهریور ۱۴۰۱

چکیده

گسترش روزافزون استفاده از اینترنت باعث شده است بسیاری از خدمات به صورت برخط و با استفاده از اینترنت ارائه شوند. تقریباً همه سرویس‌های اینترنتی بخش بسیار مهمی با عنوان نظرات کاربران دارند. این قسمت جهت به اشتراک گذاشتن نظرات و تجربیات کاربران با یکدیگر است. تعداد زیادی از افراد تحت تاثیر این نظرات هستند و مشتریان اطلاعات موجود در بخش نظرات کاربران را به عنوان داده‌هایی واقعی در نظر می‌گیرند و با توجه به این نظرات اقدام به خرید و یا عدم خرید یک محصول می‌کنند. محتوای تولیدشده توسط کاربران درباره برندها یا شرکت‌ها و یا سازمان‌ها یک منبع مهم از کلان‌داده (Big data) است که می‌تواند به اطلاعات ارزشمندی تبدیل شود. روزانه بسیاری از کالاها توسط کاربران نظردهی و امتیازدهی می‌شوند و مدیران نیز علاقه‌ی شدیدی به نظرات این اطلاعات بی‌درنگ (Real-time) دارند تا به کمک آن کیفیت تصمیمات خود را بهبود ببخشند. همینطور مدیران و مسئولان سازمان‌ها و ارگان‌ها با رویت نظرات مردم درباره اقدامات ایشان و اخبار روز می‌توانند سیاست‌های بهتری را برای سازمان خود اتخاذ کنند.

چالش اصلی در این پروژه استخراج داده‌های متنی مورد اطمینان تولیدشده توسط کاربران می‌باشد تا در نهایت بتوانیم آن‌ها را از نظر احساسات تحلیل کنیم و بسته به حس آن متن، امتیازی به آن داده شود. همچنین تشخیص نظرات هرز از نظرات واقعی و مفید نیز چالش دیگری است که در این پروژه مورد بررسی قرار می‌گیرد.

برای جمع‌آوری داده‌های تویتر باید از Crawler استفاده نمود که با ورودی دادن Query‌های متناسب با محصولات مورد نظر، نظرات کاربران را در رابطه با آن‌ها، استخراج کند. برای این منظور از ¹tweetf0rm crawler و برای پردازش داده‌های بدست آمده از ²tweeta و ³Hazm استفاده شده است.

تحلیل احساسات متن، یک روش هوشمند پردازش زبان طبیعی است که می‌توان به وسیله‌ی آن میزان مثبت یا منفی بودن یک متن را استخراج کرد. علی‌رغم پتانسیل بالای زبان فارسی، چالش‌هایی وجود دارد که باعث عدم توسعه‌ی یک مدل مناسب برای تحلیل احساسات در زبان فارسی شده است. در این پروژه با استفاده از مدل یادگیری عمیق Transformers [1] و همچنین Fine-tune کردن مدل از پیش آموزش داده شده ParsBERT [2] (یک مدل زبانی تک‌زبانه بر اساس معماری BERT گوگل [3] است. این مدل بر روی مجموعه‌های بزرگ فارسی با سبک‌های نوشتاری مختلف از موضوعات متعدد (مانند علمی، رمان، اخبار) با بیش از ۳/۹ میلیون سند، ۷۳ میلیون جمله و ۱/۳ میلیارد کلمه از قبل آموزش داده شده است)، مدلی برای تحلیل احساسات ارائه شده است. با آموزش و ارزیابی این مدل بر روی مجموعه داده SentiPers [4] که شامل نظرات وبسایت دیجی‌کالا⁴ در پنج دسته‌بندی [-2, -1, 0, 1, 2] است، پس از متوازن‌سازی داده‌ها دقت ۹۴/۹ درصد به دست آمد.

¹ <https://github.com/bianjiang/tweetf0rm>

² <https://github.com/bianjiang/tweeta>

³ <https://github.com/sobhe/hazm>

⁴ Digikala.com

به دلیل تاثیر زیادی که نظرات کاربران بر روی مشتریان دیگر دارد، این محیط به مکانی مناسب برای افرادی سودجو تبدیل شده است که با هدف تبلیغ و یا تخریب یک محصول و یا برند، اقدام به تولید نظرات غیر واقعی کنند که در این پروژه به این نوع نظرات، نظرات هرز می‌گوییم. این تهدید باعث شده است که تحقیقات زیادی برای شناسایی این گونه نظرات انجام شود [۵]. در این پروژه مدلی برای شناسایی نظرات هرز با استفاده از یادگیری عمیق پیاده‌سازی شده است. به منظور عدم وابستگی به فراداده‌ها و قابلیت شناسایی نظرات هرز تکی، مدل به گونه‌ای پیاده‌سازی شده است که تنها به متن نظر و برچسب آن برای آموزش نیاز داشته باشد. به این منظور نیز از مدل یادگیری عمیق Transformers و همچنین Fine-tune کردن مدل از پیش آموزش داده شده ParsBERT، مدلی برای تشخیص نظرات هرز ارائه شده است. با آموزش و ارزیابی این مدل بر روی ترجمه فارسی مجموعه داده fake-reviews-dataset^۱ که شامل نظرات واقعی و هرز به زبان انگلیسی است، پس از متوازن‌سازی داده‌ها دقت ۹۵/۷ درصد به دست آمد، همچنین برای مجموعه داده Digikala که تنها مجموعه داده این حوزه برای زبان فارسی است، دقت ۹۰/۵ درصد به دست آمد.

در نهایت نظرات افراد به همراه امتیاز آن نظر و تشخیص هرز یا واقعی بودن آن، در قالب یک داشبورد طراحی شده است. همچنین اطلاعات کلی آماری نیز در داشبورد قابل ملاحظه است که به افراد دید بهتری جهت تصمیم‌گیری می‌دهد.

واژگان کلیدی تحلیل احساسات، تشخیص نظرات هرز، ParsBERT، Transformers، یادگیری عمیق، طبقه‌بندی متن، Crawler

^۱ <https://osf.io/3vds7>

فهرست مطالب

فصل ۱: مقدمه.....	۱
۱_۱ خزنده (CRAWLER) چیست؟.....	۱
۲_۱ تحلیل احساسات چیست؟.....	۳
۳_۱ تشخیص نظرات هرز چیست؟.....	۶
۴_۱ اهداف پروژه.....	۸
۵_۱ نوآوری های پروژه.....	۹
۶_۱ ساختار گزارش.....	۱۰
فصل ۲: خزنده (CRAWLER).....	۱۳
۱_۲ مدل استفاده شده و نحوه عملکرد.....	۱۳
۲_۲ پس پردازش داده های جمع آوری شده.....	۱۵
فصل ۳: تحلیل احساسات.....	۱۹
۱_۳ مقدمه.....	۱۹
۲_۳ کارهای مرتبط.....	۲۰
۳_۳ مدل پیشنهادی.....	۲۱
۱_۳_۳ Transformers.....	۲۲
۱_۱_۳_۳ معماری رمزگذار-رمزگشا.....	۲۲
۲_۱_۳_۳ توجه چند سر.....	۲۳
۲_۳_۳ ParsBERT.....	۲۴
۳_۳_۳ پیش پردازش.....	۲۵
۴_۳_۳ جاسازی کلمات.....	۲۵
۵_۳_۳ معماری مدل.....	۲۵
۴_۳ نتایج تجربی.....	۲۶
۱_۴_۳ مجموعه داده.....	۲۷
۲_۴_۳ تنظیمات مدل.....	۲۷
۳_۴_۳ نتایج.....	۲۸
۵_۳ جمع بندی.....	۲۹

فصل ۴: تشخیص نظرات هرز ۳۱

۱_۴ مقدمه ۳۱

۲_۴ کارهای مرتبط ۳۲

۳_۴ مدل پیشنهادی ۳۳

۴_۴ نتایج تجربی ۳۴

۱_۴_۴ مجموعه داده‌ها ۳۴

۲_۴_۴ تنظیمات مدل ۳۵

۳_۴_۴ نتایج ۳۷

۵_۴ جمع بندی ۳۸

فصل ۵: استفاده از مدل‌های آموزش داده شده روی مجموعه داده ۳۹

۱_۵ نحوه برچسب گذاری ۳۹

فصل ۶: داشبورد ۴۱

۱_۶ رابط کاربری ۴۱

۲_۶ نحوه نصب و اجرای کد ۴۲

۳_۶ نحوه نمایش متن ۴۳

۴_۶ استفاده از ابزارک‌ها ۴۳

۵_۶ بخش‌های برنامه ۴۴

۱_۵_۶ دید کلی ۴۴

۲_۵_۶ جستجو ۴۶

۳_۵_۶ کتابخانه‌های دیگر ۴۶

۶_۶ نتیجه‌گیری ۴۶

فصل ۷: خلاصه، نتیجه‌گیری و کارهای آینده ۴۹

۱_۷ خلاصه ۴۹

۲_۷ نتیجه‌گیری ۵۱

۳_۷ کارهای آینده ۵۲

مراجع ۵۵

فصل ۱

مقدمه

۱-۱ خزنده (Crawler) چیست؟

خزنده وب یا وب کراولر^۱، یک برنامه و نرم افزار کامپیوتری است که با نام‌هایی نظیر عنکبوت وب، ربات عنکبوتی، ربات وب یا به زبان ساده کراولر (خزنده) نیز شناخته می‌شود. این ربات، توسط یک موتور جستجو مورد استفاده قرار می‌گیرد تا صفحات وب و محتوای موجود در شبکه گسترده جهانی (world wide web یا همان اینترنت) را نمایه‌بندی^۲ کند.

نمایه‌بندی، یک فرایند اساسی است. چرا که به کاربران کمک می‌کند نتایج مرتبط با پرسش و سوالات خود را در عرض چند ثانیه پیدا کنند. نمایه‌بندی جستجو اینترنتی، می‌تواند با نمایه‌بندی کتاب مقایسه شود. به طور مثال، اگر صفحات پایانی یک کتاب را باز کنید، بخشی با عنوان نمایه^۳ پیدا می‌کنید. این بخش، شامل یک لیست از عنوان‌هاست که به ترتیب حروف الفبا آورده شده‌اند. در مقابل هر یک از عناوین این لیست، شماره صفحه‌ای که در کتاب به آن عنوان اشاره شده، نوشته شده است.

همین قضیه برای نمایه جستجو اینترنتی نیز صادق است، اما بجای نوشتن شماره صفحه، موتور جستجو، لینک‌هایی را به شما نشان می‌دهد که می‌توانید به کمک آن‌ها، پاسخ درخواست‌ها و جستجو خود را پیدا کنید.

تفاوت مهمی که بین نمایه‌بندی جستجو اینترنتی و نمایه‌بندی یک کتاب وجود دارد، این است که اولی، پویا است؛ بنابراین می‌تواند تغییر کند. اما دومی، همواره ثابت و غیرقابل تغییر است.

اگر عبارت «فرق زمین با مریخ چیست» را تایپ کنید و سپس، دکمه Enter را فشار دهید، موتور جستجو، یک لیست از صفحه‌های مرتبط با این عبارت را نشان می‌دهد. معمولاً، سه مرحله طی می‌شود تا اطلاعات مربوط به جستجوهای کاربران به آنها نمایش داده شود:

- یک خزنده وب، روی محتوای وبسایت‌ها می‌خزد. (اصطلاحاً، آنها را کراول (Crawl) می‌کند.)

¹ Web Crawler

² Indexing

³ Index

- برای موتور جستجو، یک نمایه تولید می‌کند.
- الگوریتم‌های جستجو، مرتبط‌ترین صفحات را رتبه‌بندی می‌کنند.

امروزه موتورهای جستجو زیادی وجود دارند مانند Google، Bing، Yahoo!، DuckDuckGo، Baidu، Yandex و بسیاری دیگر. هر کدام از این موتورها، ربات خزنده مخصوص به خود را برای نمایه‌بندی صفحات بکار می‌برند.

این ربات‌ها، فرایند خزیدن (Crawling) را از محبوب‌ترین وبسایت‌ها آغاز می‌کنند. هدف اول ربات‌های وب، این است که مفهوم اصلی محتوای هر صفحه را منتقل کنند. از این رو، ربات‌های وب، کلمات این صفحه‌ها را جستجو می‌کنند و یک لیست کاربردی از این کلمات تهیه می‌کنند. این لیست، دفعه بعد که می‌خواهید اطلاعاتی راجع به درخواست‌ها و پرسش‌های خود پیدا کنید، توسط موتور جستجو مورد استفاده قرار می‌گیرد.

تمام صفحات موجود در اینترنت، توسط ابرپیوندها^۱ به یکدیگر متصل هستند. بنابراین، ربات‌های خزنده، می‌توانند این لینک‌ها را کشف کرده و آن‌ها را تا صفحات بعدی دنبال کنند. ربات‌های وب، تنها زمانی متوقف می‌شوند که محل تمامی مطالب و وبسایت‌های متصل به هم را پیدا کنند. سپس، اطلاعات ثبت شده را برای نمایه جستجو ارسال می‌کنند که بر روی سرورهای سراسر دنیا ذخیره شده است. تمامی این فرایند، شبیه به یک تار عنکبوت واقعی است که در آن، همه چیز در هم تنیده است.

فرایند خزیدن، بلافاصله پس از این که صفحات، نمایه‌بندی‌شدند، متوقف نمی‌شود. موتورهای جستجو به صورت متناوب، از خزنده‌های وب استفاده می‌کنند تا متوجه شوند آیا تغییری در صفحات رخ داده است یا نه. اگر تغییری رخ داده باشد، نمایه تولید شده توسط موتور جستجو، بر اساس آن بروز رسانی می‌شود.

مدل خزنده پیاده‌سازی شده در این پروژه نیز همانند خزنده‌های وب است و در بستر توییتر کار می‌کند به این صورت که با ورودی دادن Query‌های مرتبط با محصولات یا برندهای مورد نظر داده‌های مربوط به آن‌ها را جمع‌آوری می‌کند. این Query‌ها به این صورت است که یک سری قطعه‌های خاص که احتمالاً در متن نظر کاربران روی یک محصول یا برند خاص یا مواردی نظیر آن، می‌آید را مشخص می‌کنیم و مدل مربوطه تمام نظراتی که آن قطعه‌های خاص در آن‌ها وجود دارد را جمع‌آوری می‌کند (به عنوان مثال #همراه_اول و #MyMci برای برند همراه اول). در انتها پس از جمع‌آوری این داده‌ها، از بین تمام ویژگی‌های موجود (مانند زمان توییت، متن توییت، آیدی کاربر، آیدی توییت، نام کاربر و ...)، ویژگی‌هایی که برای تصمیم‌گیری مهم‌تر هستند را جدا می‌کند. همچنین این Crawler می‌تواند روی یک سرور فعال شود تا داده‌های مربوط به Query‌های ورودی را به صورت آنلاین^۲ جمع‌آوری کند.

¹ Hyperlinks

² Real-Time Streaming

در فصل دوم با خزنده پیاده‌سازی شده بیشتر آشنا خواهیم شد.

۲_۱ تحلیل احساسات چیست ؟

تحلیل احساسات^۱ یک روش پردازش زبان طبیعی است که برای تعیین مثبت، منفی و یا خنثی بودن داده‌ها استفاده می‌شود. تجزیه و تحلیل احساسات اغلب بر روی داده‌های متنی انجام می‌شود تا به کسب و کارها کمک کند احساسات مشتریان را نسبت به برند بسنجند و نیازهای مشتری را درک کنند.

تحلیل احساسات نظر مشتری را نسبت به جنبه‌های مختلف خدمت یا کالایی که خریداری کرده، می‌سنجد. این متن می‌تواند یک جمله، یک کامنت و حتی امتیازی باشد که در فرم نظرسنجی از مشتری دریافت شده. به تجزیه و تحلیل احساسات گاهی نظرکاوی^۲ و یا هوش مصنوعی احساسات^۳ هم گفته می‌شود.



¹ Sentiment Analysis

² Opinion mining

³ Emotion AI

از آنجایی که این روزها مشتریان افکار و احساسات خود را بیش از پیش بیان می‌کنند، تجزیه و تحلیل احساسات در حال تبدیل شدن به ابزاری اساسی برای نظارت و درک این احساسات است. تجزیه و تحلیل خودکار بازخورد مشتری، مانند نظرات در پاسخ‌های نظرسنجی و کامنت‌ها در شبکه‌های اجتماعی، به برندهای تجاری اجازه می‌دهد تا بدانند چه چیزی مشتری را خوشحال و یا ناامید می‌کند. در نتیجه می‌توانند محصولات و خدمات خود را متناسب با نیازهای مشتریان تنظیم کنند. به عنوان مثال، استفاده از تجزیه و تحلیل احساسات برای تحلیل خودکار بیش از ۴۰۰۰ نظر درباره محصول شما می‌تواند به شما کمک کند رضایت مشتری را از قیمت‌گذاری و خدمات مشتری خود بسنجید.

حالت تشخیص احساسات، تجزیه و تحلیل احساسات برای شناسایی احساسات، مانند خوشحالی، ناامیدی، عصبانیت، ترس، غم و غصه و غیره است. بسیاری از سیستم‌های تشخیص احساسات از واژگان (به عنوان مثال لیستی از کلمات و احساساتی که منتقل می‌کنند) یا الگوریتم‌های پیچیده یادگیری ماشین استفاده می‌کنند.

تجزیه و تحلیل احساس بسیار مهم است زیرا به کسب و کارها کمک می‌کند تا به سرعت دیدگاه کلی مشتریان خود را نسبت به محصولات درک کنند. با مرتب‌سازی خودکار احساسات متون، مکالمات شبکه‌های اجتماعی و بقیه موارد، می‌توانید سریع‌تر و دقیق‌تر تصمیم بگیرید.

تخمین زده شده است که ۹۰٪ از داده‌های جهان پراکنده و یا به عبارت دیگر غیر سازمان یافته است. هر روز حجم عظیمی از داده‌های تجاری بدون ساختار ایجاد می‌شود (ایمیل‌ها، تیکت‌های پشتیبانی، چت آنلاین سایت، مکالمات و نظرات در شبکه‌های اجتماعی، فرم‌های نظرسنجی، مقالات، اسناد و غیره). اما تجزیه و تحلیل این احساسات بدون داشتن متدولوژی خاصی بسیار زمان‌بر و دشوار است.

تجزیه و تحلیل احساس یکی از سخت‌ترین کارها در پردازش زبان طبیعی است زیرا حتی درک برخی احساسات برای انسان‌ها هم سخت است. برخی از چالش‌های اصلی تحلیل احساسات عبارتند از:

• ذهنیت و لحن

متن دو نوع دارد: ذهنی یا واقعی. بسیاری از متون واقعی حاوی احساسات صریح نیستند، در حالی که در متون ذهنی احساسات به طور صریح بیان می‌شوند. برای مثال فرض کنید شما قصد دارید احساسات دو متن زیر را تجزیه و تحلیل کنید:

❖ این محصول خوب است.

❖ این محصول قرمز است.

اکثر مردم می‌گویند که احساس برای اولی مثبت و برای دومی خنثی است، با تمام بخش‌های جمله (صفت، فعل و برخی از اسم‌ها) نباید با توجه به نحوه ایجاد احساسات یکسان رفتار کرد. در مثال‌های بالا، صفت خوب،

صفتی ذهنی و رنگ قرمز، صفتی واقعی است. طبقه بندی صفات ذهنی بسیار ساده تر است زیرا احساسات در آن صریح تر بیان می شود.

• متن و تضادها

وقتی افراد جمله ای را بیان می کنند، شما خیلی راحت تر احساسات و معانی موجود در آن را متوجه می شوید. اما اگر همین جملات نوشته شوند، درک مفهوم و تجزیه و تحلیل احساسات بسیار دشوار است. به پاسخ های زیر در یک نظرسنجی توجه کنید:

❖ همه چیز

❖ هیچ چیز!

فرض کنید پاسخ های بالا به این سوال داده شده اند:

شما در مورد این رویداد چه چیزی را دوست داشتید؟

پاسخ اول مثبت و پاسخ دوم منفی خواهد بود.

حال، تصور کنید که سوال این است:

چه چیزی را در مورد این رویداد دوست نداشتید؟

پاسخ ها کاملاً معنی متضاد و متفاوتی پیدا می کنند. در نتیجه درک زمینه ای که پاسخ ها با توجه به آن

داده شده برای هوش مصنوعی ضروری است. چنین چیزی نیازمند الگوریتم های دقیق پیش پردازش^۱

داده و پس پردازش^۲ است.

• گوشه کنایه و متون طعنه آمیز

وقتی صحبت از کنایه، طعنه و وارونه گویی می شود، مردم احساسات منفی خود را با استفاده از کلمات مثبت ابراز می کنند، که بدون درک کامل از زمینه گفتگو، تشخیص آن برای ماشین ها دشوار است.

به عنوان مثال، به برخی پاسخ های احتمالی این سوال توجه کنید: آیا از تجربه خرید خود با ما لذت بردید؟

❖ آره، خیلی!

❖ دوبار!

این پاسخ ها با علامت تعجب می تواند منفی باشد، مسئله این است که هیچ نشانه متنی وجود ندارد که به یادگیری ماشین برای درک این موضوع کمک کند؛ در نتیجه هوش مصنوعی ممکن است این پاسخ ها را به عنوان خنثی و یا مثبت دسته بندی کند.

¹ Pre-processing

² Post-processing

• مقایسه‌ها

چگونگی درک مقایسه‌ها در تجزیه و تحلیل احساسات، چالش دیگری است که قابل حل است. به متن‌های زیر توجه کنید:

❖ این محصول رقیب نداره!

❖ این محصول از مدل قدیمی‌ای که داشتم بهتره.

❖ این محصول از هیچی بهتره.

طبقه‌بندی متن اول به هیچ سرنخ از زمینه متنی نیاز ندارد، واضح است که مثبت است. طبقه‌بندی متن دوم و سوم کمی دشوارتر است. آن‌ها را به عنوان خنثی، مثبت یا حتی منفی طبقه‌بندی می‌کنید؟ یک بار دیگر، زمینه می‌تواند در تحلیل تغییر ایجاد کند.

مدل پیاده‌سازی شده با استفاده از Transformers و یادگیری عمیق و استفاده از مکانیزم‌های جدید توجه^۱ قابل به تحلیل برخی از چالش‌های این زمینه است و تا حد قابل قبولی این چالش‌ها را حل می‌کند. برای آموزش و ارزیابی این مدل از مجموعه داده SentiPers استفاده شده است.

در فصل سوم با مدل پیاده‌سازی شده و نتایج آن بیشتر آشنا خواهیم شد.

۳_۱ تشخیص نظرات هرز چیست؟

تقریباً می‌توان گفت در جامعه امروز اکثر افراد، استفاده از خدمات اینترنتی را به نحوی تجربه کرده‌اند. یکی از مهمترین منابع برای انتخاب یک سرویس و یا خرید یک محصول، نظرات کاربران است. اگرچه تمام ارائه دهندگان سرویس‌های مختلف برای محصول خود انواع توضیحات را می‌نویسند و مزایا و معایب محصولات مختلف را با هم مقایسه می‌کنند، اما معمولاً مشتریان اعتماد بیشتری به نظرات کاربرانی که قبلاً از این سرویس یا محصول استفاده کرده‌اند و در سایت نوشته می‌شود، دارند. نظرات کاربران از اهمیت ویژه‌ای برخوردار است زیرا تاثیر بسیار زیادی بر روی مشتریان بالقوه می‌گذارد و با اطلاعاتی که در اختیار کاربران قرار می‌دهد باعث می‌شود یک کاربر تصمیم به استفاده یا عدم استفاده از یک سرویس بگیرد.

تاثیر این نظرات فقط بر روی کاربران نیست و شرکت‌ها و کارخانه‌های مختلف نیز از این نظرات استفاده می‌کنند و با اطلاعاتی که از این نظرات استخراج می‌کنند، محصولات خود را مطابق با نظرات مشتریان تغییر داده و اصلاح می‌کنند.

¹ Attention

بنابراین نظرات کاربران اهمیت بالایی دارد و اطمینان از واقعی بودن این نظرات نیز اهمیت زیادی دارد زیرا ممکن است افرادی با هدف های مختلف سعی در ثبت نظراتی داشته باشند که نظر واقعی آنها نباشد و مقصود دیگری از ثبت آن داشته باشند که این گونه نظرات باید شناسایی و حذف گردند. با توجه به اهمیت این نظرات و تاثیر چشمگیری که می تواند در فروش یک محصول بگذارد، افراد سودجویی وجود دارند که سعی می کنند با ایجاد نظرات غیر واقعی، باعث تبلیغ و یا تخریب یک برند یا محصول شوند. به این گونه نظرات، نظر هرز گفته می شود. این نظرات توسط اشخاصی که تجربه شخصی استفاده از محصول را ندارند و یا به نوعی ذینفع هستند نوشته می شود و باعث گمراه شدن کاربران می گردد [۶].

ارائه دهندگان سرویس های مختلف همواره سعی در تشخیص نظرات هرز دارند تا اعتماد کاربران به این نظرات مخدوش نشود و بتوانند به عنوان یک منبع معتبر برای انتخاب یک محصول از این نظرات استفاده کنند. اما با گسترش استفاده از اینترنت و خدمات اینترنتی، نظرات هرز نیز افزایش می یابد زیرا نفع زیادی را نصیب برخی از افراد سودجو می کند.

شناسایی نظرات هرز در سال های اخیر بیشتر مورد توجه قرار گرفته است زیرا همچنان که روش های شناسایی نظرات هرز پیشرفت می کنند، نویسندگان نظرات هرز نیز از روش های جدیدتری برای نوشتن نظرات هرز به گونه ای که شناسایی نشوند استفاده می کنند و نظری ثبت می کنند که واقعی به نظر برسد.

نوشتن نظر هرز تا حدی برای این افراد سودجو مهم و درآمدزاست که کسب و کارهایی برای ثبت نظر هرز به وجود آمده و سعی می کنند الگوریتم های شناسایی نظرات هرز را دور بزنند و نظرات را با استفاده از تغییرات مختلف، به گونه ای ثبت کنند که سیستم تشخیص دچار خطا شده و نظرات هرز آنها حذف نشوند.

در حال حاضر سایت هایی که مخصوص تبادل نظر هستند، مثل Yelp.com، برای شناسایی این گونه نظرات و حذف آنها از الگوریتم های خودش استفاده می کند. البته در سایت های داخلی الگوریتم های خوبی برای شناسایی این گونه نظرات برای زبان فارسی وجود ندارد و سرویس های ایرانی برای رفع این گونه مشکلات ناچارند محدودیت هایی برای ثبت نظر اعمال کنند که تعداد نظرات هرز را کاهش دهند.

بهترین مدل های شناسایی نظرات هرز از روش های یادگیری ماشین برای پیاده سازی مدل خود استفاده می کنند و به وسیله داده های موجود، سعی در ایجاد مدلی برای شناسایی این گونه نظرات دارند و از ویژگی های مختلف نظر، نویسنده نظر و ... برای آموزش مدل استفاده می شود و مقالات و پیاده سازی های مختلفی در این حوزه وجود دارد که هر کدام نقاط قوت و ضعف خود را دارند.

به طور کلی با توجه به توضیحات داده شده راجع به نظرات هرز، دو نوع نظر هرز وجود دارد. به نظراتی که با ذکر نکات مثبت یک محصول و القا کردن حس خوب از یک محصول، قصد تبلیغ یک محصول را دارند، نظرات هرز مثبت و به نظراتی که با برجسته کردن نقاط ضعف و القا کردن حس بد از خرید یک محصول، قصد تخریب یک محصول را دارند، نظرات هرز منفی گفته می شود.

در مسئله شناسایی نظرات هرز نوشتن مدلی که بتواند جامع باشد و برای حوزه‌های مختلف به خوبی کار کند کار بسیار دشواری است و پیاده‌سازی یک مدل در این حوزه به شدت وابسته به مجموعه داده‌ای است که مدل بر اساس آن آموزش می‌بیند. علاوه بر این شناسایی نظر هرز بسیار دشوارتر از مسائل دیگر دسته‌بندی متن مثل تحلیل احساسات، شناسایی متن توهین آمیز و ... می‌باشد به این دلیل که در دیگر مسائل با بررسی کلمات می‌توان تا حد خوبی به کلاس مورد نظر رسید اما در مسئله شناسایی نظرات هرز اینگونه نیست.

مسئله شناسایی نظرات هرز زمانی دشوارتر می‌شود که ما از فراداده‌ها (نام کاربر، نام کالا، زمان ثبت و ...) نیز استفاده نکنیم و فقط از متن نظر و ویژگی‌هایی که از متن می‌توان استخراج کرد استفاده کنیم و با استفاده از همین ویژگی‌های محدود، نظرات هرز را مشخص کنیم.

مدل پیاده‌سازی شده با استفاده از Transformers و یادگیری عمیق و استفاده از مکانیزم‌های جدید توجه قابل به تحلیل برخی از چالش‌های این زمینه است و تا حد قابل قبولی این چالش‌ها را حل می‌کند. برای آموزش و ارزیابی این مدل از مجموعه داده‌های [۷] fake-review-dataset و Digikala استفاده شده است.

در فصل چهارم با مدل پیاده‌سازی شده و نتایج آن بیشتر آشنا خواهیم شد.

۴_۱ اهداف پروژه

این پروژه با هدف ارائه مدل‌هایی برای تحلیل احساسات و تشخیص نظرات هرز در زبان فارسی برای داده‌های Crawl شده در بستر توئیتر با استفاده از یادگیری عمیق و مدل Transformers است. در این پروژه سعی می‌کنیم با استفاده از جدیدترین و کارآمدترین روش‌های یادگیری عمیق و نیز استفاده از روش‌های نوینی که در مسائل پردازش زبان طبیعی وجود دارد، روش‌هایی را ارائه دهیم که به شکل بهینه و با دقت مناسب بتوانند داده‌های مورد نیاز را Crawl کنند، احساسات نظرات را تحلیل کنند و نظرات هرز را از نظرات واقعی جدا کنند و این نظرات را طبقه‌بندی کنند.

در این پروژه سعی شده است که از کمترین ویژگی‌های ممکن در دسترس برای تحلیل احساسات و شناسایی نظرات هرز استفاده شود. علت این امر این است که در مجموعه داده‌هایی که برای تحلیل احساسات یا شناسایی نظرات هرز استفاده می‌شوند، معمولاً تعداد ویژگی‌های کمی وجود دارند و یا اگر در تعدادی از این مجموعه‌ها تعداد ویژگی‌ها بیشتر است، با هم تفاوت دارند و هرکدام به سلیقه خود ویژگی‌های خاصی را جمع‌آوری کرده‌اند و در نتیجه کار را برای ارائه مدلی که بتواند با همه ی این مجموعه داده‌ها قابل آموزش باشد دشوار است.

مسئله دیگری که در این پروژه مورد بررسی قرار گرفته است، تحلیل احساسات و شناسایی نظرات هرز در زبان فارسی با استفاده از یادگیری عمیق و مدل Transformers است که در این زمینه تاکنون تحقیق‌های کمی صورت گرفته است و بسیاری از مواردی که وجود دارند از روش‌های قدیمی‌تر و یا روش‌های ضعیف‌تر یادگیری ماشین برای تحلیل احساسات و شناسایی نظرات هرز استفاده شده است. در این پروژه سعی شده است که مدل ارائه شده به گونه‌ای پیاده‌سازی شود که نه تنها بر روی مجموعه داده‌های مختلف قابل آموزش باشد بلکه بتوان با تغییراتی اندک آن را برای زبان‌های مختلف نیز استفاده کرد. این مسئله باعث می‌شود پیچیدگی‌ها و زمان صرف شده برای پیاده‌سازی کاهش یابد و از طرف دیگر وابستگی مدل به فراداده‌های مختلف را کاهش می‌دهد. در نتیجه می‌توان با آموزش مدل بر روی مجموعه داده‌های مختلف، مدلی با دقت بیشتر را ارائه داد.

در مدل پیشنهادی تنها از داده‌های مربوط به نظر، برای یادگیری مدل استفاده شده است و در واقع مدل ما مبتنی بر نظر است. در واقع تمام ویژگی‌هایی که برای آموزش مدل استفاده می‌شود از متن نظر استخراج شده است و وابستگی فقط به متن نظر دارد. این موضوع باعث می‌شود که مشکلی که برای کمبود مجموعه داده‌های آموزشی برچسب‌دار در حوزه شناسایی نظرات هرز وجود دارد تا حد زیادی برطرف شود و مدل به راحتی برای تمامی مجموعه داده‌های این حوزه قابل آموزش باشد.

هدف اصلی این پروژه، رتبه‌بندی محصولات یا برندها یا مواردی نظیر آن براساس تحلیل احساسات و شناسایی نظرات هرز بر روی نظرات کاربران فارسی زبان تویتر (یا بسترهایی شبیه به آن) که درباره آن‌ها نظر داده‌اند، با استفاده از یادگیری عمیق و مدل Transformers می‌باشد که تابحال تحقیق یا پروژه‌ای در این باره صورت نگرفته است. همچنین در انتها داشبوردی طراحی شده است که نظرات بررسی شده را همراه با امتیاز مربوط به تحلیل احساسات و واقعی یا غیر واقعی بودن آن نظر را به صورت آماری نشان می‌دهد.

۵_۱ نوآوری‌های پروژه

هدف انجام این پروژه ارائه‌ی مدلی برای تحلیل احساسات و تشخیص نظرات هرز در زبان فارسی برای داده‌های Crawl شده در بستر تویتر با استفاده از یادگیری عمیق و مدل Transformers جهت رتبه‌بندی محصولات یا برندها یا مواردی نظیر آن است. در این زمینه با توجه به کارها و پیشینه تحقیقاتی که در این حوزه وجود دارد و با توجه به بررسی چالش‌های مختلف در این حوزه تصمیم به پیاده‌سازی مدلی شد که کمترین مقدار ممکن از ویژگی‌ها را برای آموزش بخواهد و در عین حال کارایی و عملکرد قابل قبولی داشته باشد. با توجه به این مسئله در پیاده‌سازی مدلی که در این پروژه بر روی آن کار کرده‌ایم، سعی شده است که مدل به گونه‌ای پیاده‌سازی شود که برخی از چالش‌های موجود در زمینه‌های تحلیل احساسات و شناسایی نظرات هرز و همچنین Crawl داده‌های مورد نیاز از تویتر را پوشش دهد.

مدلی که پیاده‌سازی کرده‌ایم مبتنی بر متن نظر است، یعنی مدل به گونه‌ای پیاده‌سازی شده است که برای آموزش آن فقط به مجموعه داده‌ای شامل متن نظر و برچسب آن نیاز داریم. با توجه به کمبود داده‌های آموزشی و تعداد ویژگی‌های کم در این داده‌های آموزشی در حوزه تحلیل احساسات و به خصوص شناسایی نظرات هرز که یک چالش اساسی است، استفاده از ویژگی‌های کم باعث می‌شود بتوانیم مدل را بر روی همه مجموعه داده‌های موجود آموزش دهیم و تا حدی این مشکل را برطرف سازیم.

استفاده از فراداده‌های زیاد مثل ویژگی‌های نویسنده نظر، ویژگی‌های محصول و ... برای آموزش مدل باعث می‌شود مشکلی به نام مشکل شناسایی نظرات تکی - یعنی نظراتی که نویسنده آن فقط همان یک نظر را نوشته است - به وجود بیاید. اما به دلیل اینکه ما در آموزش مدل خود از این فراداده‌ها استفاده نمی‌کنیم، مشکل نظرات تکی نیز خود به خود حل می‌شود و در شناسایی آن‌ها مدل دچار مشکل نمی‌شود.

مورد دیگری که به عنوان کاری جدید در این پروژه انجام می‌شود، تحلیل احساسات و شناسایی نظرات هرز با استفاده از یادگیری عمیق و مدل Transformers برای زبان فارسی است. به دلیل معماری قابل تعمیمی که مدل ما دارد، با اندکی تغییر در مراحل پیش پردازش و همچنین استفاده از مدل‌های چندزبانه BERT مانند XLM-RoBERTa [۸] یا مواردی نظیر آن، می‌توانیم برای هر زبان دیگری نیز این مدل را پیاده‌سازی کنیم.

در نهایت نوآوری خاص این پروژه، استفاده از روش‌های یادگیری عمیق و Transformers برای تحلیل احساسات و شناسایی نظرات هرز روی داده‌های Crawl شده تویتر جهت رتبه‌بندی محصولات یا برندها یا مواردی نظیر آن است که تابحال تحقیق یا پروژه‌ای نظیر آن وجود نداشته است.

۱_۶ ساختار گزارش

ساختار این گزارش به صورتی است که دارای شش فصل می‌باشد. فصل اول که فصل مقدمه است شامل تعاریفی از مواردی که پیاده‌سازی شده‌اند، می‌باشد و در ادامه آن اهداف انجام این پروژه و نوآوری‌هایی که در آن به کار گرفته شده است را به صورت کلی بیان می‌کند.

فصل دوم مدل پیاده‌سازی شده برای Crawler و نحوه جمع‌آوری داده‌ها را در بستر تویتر براساس Query‌هایی که به آن ورودی می‌دهیم، بیان می‌کند. این Query‌ها یک سری قطعه‌های خاص هستند که احتمالاً در متن نظر کاربران روی یک محصول یا برند خاص یا مواردی نظیر آن، می‌آیند. Crawler پیاده‌سازی شده نظرات مرتبط با این Query‌ها را جمع‌آوری

می‌کند و از بین ویژگی‌های جمع‌آوری شده هر نظر، آن‌هایی را که برای تصمیم‌گیری اهمیت دارند را جدا می‌کند تا در مراحل بعدی برای تحلیل احساسات و تشخیص نظرات هرز استفاده شوند.

فصل سوم مدل پیاده‌سازی شده برای تحلیل احساسات روی مجموعه داده SentiPers را بیان می‌کند. مدل پیاده‌سازی شده از پنج دسته‌بندی $[-2, -1, 0, 1, 2]$ برای طبقه‌بندی نظرات استفاده می‌کند که 2- نشان دهنده حالت Furious، 1- نشان دهنده حالت Angry، 0 نشان دهنده حالت Neutral، 1 نشان دهنده حالت Happy و 2 نشان دهنده حالت Delighted است.

فصل چهارم مدل پیاده‌سازی شده برای تشخیص نظرات هرز روی مجموعه‌های داده fake-review-dataset و Digikala را بیان می‌کند. مدل پیاده‌سازی شده از دو دسته‌بندی $[0, 1]$ برای طبقه‌بندی نظرات استفاده می‌کند که 0 نشان دهنده حالت غیرواقعی یا هرز بودن نظر و 1 نشان دهنده حالت واقعی یا غیر هرز بودن نظر است.

فصل پنجم استفاده از مدل‌های آموزش داده شده و نتایج آن را روی مجموعه داده جمع‌آوری شده از توییتر (یا می‌تواند هر مجموعه داده‌ای شبیه به آن باشد) نشان می‌دهد.

فصل ششم داشبورد پیاده‌سازی شده برای نشان دادن نتایج را بیان می‌کند. در این داشبورد مدل‌های آماری جهت تصمیم‌گیری بهتر و همچنین امکان انتخاب حالت‌های مختلف مانند فیلتر نظرات هرز، فیلتر کردن بر اساس امتیاز و قابلیت جستجو بر حسب یک کلمه خاص در نظرات، قرار داده شده است.

فصل هفتم نیز فصل خلاصه و نتیجه‌گیری است و در این فصل به بحث در مورد نقاط ضعف و قوت پروژه پرداخته شده است و همچنین ارائه پیشنهادهایی برای کارهای آینده بیان می‌شود.

فصل ۲

خزنده (Crawler)

۱_۲ مدل استفاده شده و نحوه عملکرد

برای جمع‌آوری داده‌های به روز در هر بستری نیاز به Crawler متناسب با آن بستر داریم. Crawler ای که در این پروژه استفاده شده است tweetf0rm^۱ است که کاملاً مختص توییت‌هاست. برای استفاده از این Crawler ابتدا باید یک برنامه در dev توییت‌ها ساخت تا به API Twitter^۲ دسترسی پیدا کرد. در ادامه توییت‌ها توکن‌های مربوط به دسترسی برنامه مورد نظر به Twitter API را ارائه می‌دهد که شامل APP Key، APP Secret، Oauth Token و Oauth Token Secret است. با قرار دادن این توکن‌ها در config مربوط به خزنده tweetf0rm، این خزنده به داده‌های توییت دسترسی پیدا می‌کند.

```
{
  "apikey": {
    "i0mf0rmer01": {
      "app_key": "APP_KEY",
      "app_secret": "APP_SECRET",
      "oauth_token": "OAUTH_TOKEN",
      "oauth_token_secret": "OAUTH_TOKEN_SECRET"
    }
  }
}
```

این خزنده برای جمع‌آوری توییت‌ها یک سری کلمات کلیدی مثل کلماتی که با # شروع می‌شوند یا حتی بدون آن را ورودی می‌گیرد و تمام توییت‌هایی که عین آن ورودی‌ها را داشته باشند جمع‌آوری می‌کند همچنین تنظیماتی نظیر مشخص کردن مکان مربوط به توییت‌ها مثل شهر یا کشور خاص دارد که با تنظیم کردن geo می‌توان به نتیجه مورد نظر دست یافت و تنظیمات دیگری مثل تنظیم کردن user timelines، تنظیم کردن آیدی‌های خاص برای کاربرانی که فقط توییت‌های آن کاربران را می‌خواهیم، تنظیم کردن آیدی مربوط به توییت‌ها (آیدی توییت‌ها براساس زمان توییت است) که می‌خواهیم از آن آیدی به

^۱ <https://github.com/bianjiang/tweetf0rm>

^۲ <https://developer.twitter.com/en/docs/twitter-api>

بعد از جمع‌آوری کند با تنظیم متغیر `since_id` و ... دارد. در انتها برای ورودی‌های مورد نظر و تنظیمات مربوطه، یک فایل Query به فرمت json در مسیر `test_data` قرار می‌دهیم تا دقیقاً برطبق آن Query، توییت‌ها را جمع‌آوری کند.

نمونه Query (بخش `querystring` همان `terms` است منتها فقط در یک خط):

```
1 {
2   "keyword_list_0": {
3     "geocode": null,
4     "terms": [
5       "\"cervarix\"",
6       "\"cervical cancer\"",
7       "\"cervical #cancer\"",
8       "\"#cervical cancer\"",
9       "\"cervicalcancer\"",
10      "\"#cervicalcancer\""
11    ],
12    "since_id": 1,
13    "querystring": "\"(\\cervarix\\) OR (\\cervical cancer\\) OR (\\cervical #cancer\\) OR (\\#cervical cancer\\) OR (\\cervicalcancer\\) OR (\\#cervicalcancer\\)\""
14  },
15   "keyword_list_1": {
16     "geocode": [
17       "dona_ana_nm",
18       "32.41906196127472,-106.82334114385034,51.93959956432837mi"
19     ],
20     "querystring": "\"(\\cancer #cervical\\) OR (\\cancercervical\\) OR (\\#cancercervical\\)\"",
21     "since_id": 0,
22     "terms": [
23       "cancer #cervical",
24       "cancercervical",
25       "#cancercervical"
26     ]
27   }
28 }
29 }
```

اما توییت‌ها تنها اجازه جمع‌آوری توییت‌ها تا حدود ۱۰ روز اخیر را می‌دهد. این خزنده قابلیت این را دارد که تا جایی که توییت‌ها اجازه می‌دهد توییت‌ها را جمع‌آوری کند و پس از آن به صورت آنلاین، توییت‌هایی که از همان لحظه به بعد اضافه شوند و با Query ورودی هم‌خوانی داشته باشند را نیز جمع‌آوری می‌کند یعنی به صورت Online Streaming عمل می‌کند. بنابراین می‌توان سروری را برای Stream توییت‌های جدید همواره در حال اجرا نگه داشت تا داده‌ها به روز باشند و همچنین افزایش یابند.

تنظیماتی که برای این پروژه استفاده شده فقط شامل کلمات کلیدی و شروع از توییت با آیدی ۰ است و مواردی مانند موقعیت مکانی و ... قرار داده نشده‌اند. همچنین برای ارزیابی پروژه، سه اپراتور بزرگ ایران یعنی ایرانسل، همراه اول و رایتل را در نظر گرفتیم تا نظرات مربوط به آن‌ها را از توییت‌ها جمع‌آوری می‌کنیم و پس از پس پردازش آن‌ها، مدل‌های تحلیل احساسات و تشخیص نظرات هرز را روی آن‌ها اجرا می‌کنیم.

نمونه Query‌های استفاده شده برای سه اپراتور ایرانسل، همراه اول و رایتل (بخش `querystring` به دلیل بزرگ بودن

کامل مشخص نیست):

ایرانسل (`search_irancell.json`):

```

1  "keyword_list_0":{
2    "geocode":null,
3    "terms":{
4      "iran":{
5        "iran":{
6          "iran":{
7            "iran":{
8              "iran":{
9                "iran":{
10               "iran":{
11                "iran":{
12               "iran":{
13                "iran":{
14               "iran":{
15                "iran":{
16               "iran":{
17                "iran":{
18               "iran":{
19                "iran":{
20               "iran":{
21                "iran":{
22               "iran":{
23                "iran":{
24               "iran":{
25                "iran":{
26               "iran":{
27                "iran":{
28               "iran":{
29                "iran":{

```

همراه اول (search_hamrahaval.json):

```

1  "keyword_list_0":{
2    "geocode":null,
3    "terms":{
4      "hamrah":{
5        "hamrah":{
6          "hamrah":{
7            "hamrah":{
8              "hamrah":{
9                "hamrah":{
10               "hamrah":{
11                "hamrah":{
12               "hamrah":{
13                "hamrah":{
14               "hamrah":{
15                "hamrah":{
16               "hamrah":{
17                "hamrah":{
18               "hamrah":{
19                "hamrah":{
20               "hamrah":{
21                "hamrah":{
22               "hamrah":{
23                "hamrah":{
24               "hamrah":{
25                "hamrah":{

```

رایتل (search_rightel.json):

```

1  "keyword_list_0":{
2    "geocode":null,
3    "terms":{
4      "rightel":{
5        "rightel":{
6          "rightel":{
7            "rightel":{
8              "rightel":{
9                "rightel":{
10               "rightel":{
11                "rightel":{
12               "rightel":{
13                "rightel":{
14               "rightel":{
15                "rightel":{
16               "rightel":{
17                "rightel":{
18               "rightel":{
19                "rightel":{
20               "rightel":{
21                "rightel":{
22               "rightel":{
23                "rightel":{
24               "rightel":{
25                "rightel":{

```

۲_۲ پس پردازش داده‌های جمع‌آوری شده

پس از قرار دادن Query های موردنظر و با اجرای خزنده، تویت‌های مطابق با این Query ها تا زمانی که تویت‌ر اجازه بدهد، جمع‌آوری می‌شود و پس از آن خزنده به حالت Streamer می‌رود و تویت‌های جدیدی که اضافه شوند را جمع‌آوری می‌کند اما بدون وجود سرور باید این مرحله از پردازش را قطع کرد.

نمونه دستور برای اجرای خزنده بر روی Query های مربوط به سه اپراتور ایرانسل، همراه اول و رایتل:

```

1 python twitter_tracker.py -c config.json -cmd search -o data/search_query -cc test_data/search_hamrahaval.json -wait 5
2 python twitter_tracker.py -c config.json -cmd search -o data/search_query -cc test_data/search_irancell.json -wait 5
3 python twitter_tracker.py -c config.json -cmd search -o data/search_query -cc test_data/search_rightel.json -wait 5
4

```

خروجی‌های مربوط به پردازش هر Query در مسیر `data\search_query` و در پوشه مربوط به تاریخ اجرای خزنده قرار می‌گیرند. به طور مثال اگر خزنده در تاریخ 07/19/2022 اجرا شود، خروجی‌ها در پوشه 20220719 قرار می‌گیرند. خروجی هر Query در یک فایل متنی با نام هش شده نسبت به تنظیمات و زمان اجرا، ذخیره می‌شود که این نام هش شده را در زمان اجرا نمایش می‌دهد.

هر خط از فایل‌های خروجی شامل یک توییت مطابق با Query ورودی با تمام ویژگی‌های نظیر آن توییت مانند زمان ساخته شدن توییت، آیدی توییت، آیدی کاربر، متن توییت، مشخصات retweet و بسیاری ویژگی‌های دیگر به فرمت json است. اما ما به همه این ویژگی‌ها نیاز نداریم همچنین مشکلاتی نظیر زبان‌های غیر فارسی، وجود mentionها و ... وجود دارد که نیاز به پس پردازش دارند.

نمونه فایل خروجی (طول هر خط به دلیل تعداد زیاد ویژگی‌ها، بسیار زیاد است و به همین خاطر کامل مشخص نیستند):

```
1 "created_at": "Tue Jul 19 19:22:42 +0000 2022", "id": "154947481170666499", "id_str": "154947481170666499", "full_text": "RT @heyedamir: @b6f1@b6f8@b633@b627@b644 @b676@b66cc@b634 @b627@b642@b627@b6cc #b627@b633@b631@b66",
2 "created_at": "Tue Jul 19 19:12:36 +0000 2022", "id": "154947254949206532", "id_str": "154947254949206532", "full_text": "@heyedamir: @b62a@b627@b62a@b627@b645 @b62a@b627@b62a@b627@b645 @b62a@b627@b62a@b627@b645 @b62a@b627@b62a@b627@b645",
3 "created_at": "Tue Jul 19 19:15:41 +0000 2022", "id": "1549473068151700353", "id_str": "1549473068151700353", "full_text": "@heyedamir: @b648@b627@b6cc @b686@b647 @b641@b627@b62c@b639@b647@b62c@b6cc@b627@b6cc@b631@b648@b627@b6cc@b631@b648@b627@b6cc",
4 "created_at": "Tue Jul 19 19:13:32 +0000 2022", "id": "1549472524106432512", "id_str": "1549472524106432512", "full_text": "@imgapedram: @b628@b644@b6cc@b68c @b627@b6cc@b631@b627@b6cc@b631@b644 @b627@b632 @b647@b645@b645",
5 "created_at": "Tue Jul 19 19:11:18 +0000 2022", "id": "154947196296387360", "id_str": "154947196296387360", "full_text": "@41d\\n@b631@b627@b647@b62c@b627@b646 @b62f@b627@b634@b6cc @b628@b642@b6cc@b6cc@b631@b648 @b627@b6cc@b631@b648 @b627@b6cc",
6 "created_at": "Tue Jul 19 19:11:12 +0000 2022", "id": "1549471938694766602", "id_str": "1549471938694766602", "full_text": "@41d\\n@b631@b627@b647@b62c@b627@b646 @b62f@b627@b634@b6cc @b628@b642@b6cc@b6cc@b631@b648 @b627@b6cc@b631@b648 @b627@b6cc",
7 "created_at": "Tue Jul 19 19:11:09 +0000 2022", "id": "1549471923599556617", "id_str": "1549471923599556617", "full_text": "@1\\d\\n@b63d@b634 @b62a@b648@b633@b639@b6cc @b641@b631@b627@b646@b6cc@b631 56 @b62f@b631 @b62",
8 "created_at": "Tue Jul 19 19:05:56 +0000 2022", "id": "1549470614485975041", "id_str": "1549470614485975041", "full_text": "RT @mduq: @b645@b646 @b646@b645@b6cc@b641@b647@b645@b645 @b688@b631@b627 @b648@b642@b62a@b6",
9 "created_at": "Tue Jul 19 19:02:15 +0000 2022", "id": "1549469687473207874", "id_str": "1549469687473207874", "full_text": "RT @theRealPejmann: AUTO RECYCLE BIN , TRASH \\n @rancel11TUES", "truncated": false, "display_text_range": [0, 43], "entities": {"",
10 "created_at": "Tue Jul 19 19:01:47 +0000 2022", "id": "1549469566308208754", "id_str": "1549469566308208754", "full_text": "AUTO RECYCLE BIN , TRASH \\n @rancel11TUES", "truncated": false, "display_text_range": [0, 43], "entities": {"",
11 "created_at": "Tue Jul 19 19:59:37 +0000 2022", "id": "1549469023754149889", "id_str": "1549469023754149889", "full_text": "RT @heyedamir: @b6f1@b6f8@b633@b627@b644 @b676@b66cc@b634 @b627@b642@b627@b6cc #b627@b633@b631@b66",
12 "created_at": "Tue Jul 19 18:57:13 +0000 2022", "id": "1549468421041143888", "id_str": "1549468421041143888", "full_text": "@b6cc@b647 @b62f@b648@b633@b62a@b6cc @b627@b627@b634@b62a@b645 @b62a@b648 @b62f@b648@b631@b648",
13 "created_at": "Tue Jul 19 18:53:20 +0000 2022", "id": "1549467443227142977", "id_str": "1549467443227142977", "full_text": "RT @heyedamir: @b6f1@b6f8@b633@b627@b644 @b676@b66cc@b634 @b627@b642@b627@b6cc #b627@b633@b631@b66",
14 "created_at": "Tue Jul 19 18:51:53 +0000 2022", "id": "154946707830931264", "id_str": "154946707830931264", "full_text": "RT @heyedamir: @b6f1@b6f8@b633@b627@b644 @b676@b66cc@b634 @b627@b642@b627@b6cc #b627@b633@b631@b66",
15 "created_at": "Tue Jul 19 18:46:34 +0000 2022", "id": "1549465738993225733", "id_str": "1549465738993225733", "full_text": "RT @mduq: @b645@b646 @b646@b645@b6cc@b641@b647@b645@b645 @b688@b631@b627 @b648@b642@b62a@b6",
16 "created_at": "Tue Jul 19 18:37:34 +0000 2022", "id": "154946347268064192", "id_str": "154946347268064192", "full_text": "@heyedamir: @b634@b631@b649@b62a @b627@b641 @b631@b6cc@b642 @b627@b6cc@b6cc @b627@b643@b645@b645@b62",
17 "created_at": "Tue Jul 19 18:37:24 +0000 2022", "id": "1549463431302787088", "id_str": "1549463431302787088", "full_text": "@heyedamir: @b6cc@b649@b6cc @b627@b632 @b627@b6cc@b631@b627@b646 @b627@b6cc@b631@b627@b646 @b627@b6cc",
18 "created_at": "Tue Jul 19 18:33:18 +0000 2022", "id": "1549462331985148352", "id_str": "1549462331985148352", "full_text": "RT @heyedamir: @b6f1@b6f8@b633@b627@b644 @b676@b66cc@b634 @b627@b642@b627@b6cc #b627@b633@b631@b66",
19 "created_at": "Tue Jul 19 17:58:12 +0000 2022", "id": "1549453568637980849", "id_str": "1549453568637980849", "full_text": "@bavanz27 @b627@b644 @b627@b646 @b62a@b648 @b636@b627@b645 \\n@b627@b6cc@b646@b62a@b627 @b6",
20 "created_at": "Tue Jul 19 17:27:51 +0000 2022", "id": "1549454592550828545", "id_str": "1549454592550828545", "full_text": "@bavanz27 @b627@b644 @b627@b646 @b62a@b648 @b636@b627@b645 \\n@b627@b6cc@b646@b62a@b627 @b6",
21 "created_at": "Tue Jul 19 17:23:43 +0000 2022", "id": "15494542374237147136", "id_str": "15494542374237147136", "full_text": "@bavanz27 @b627@b644 @b627@b646 @b62a@b648 @b636@b627@b645 \\n@b627@b6cc@b646@b62a@b627 @b6",
22 "created_at": "Tue Jul 19 17:21:37 +0000 2022", "id": "15494541844953620481", "id_str": "15494541844953620481", "full_text": "RT @mduq: @b645@b646 @b646@b645@b6cc@b641@b647@b645@b645 @b688@b631@b627 @b648@b642@b62a@b6",
23 "created_at": "Tue Jul 19 17:04:46 +0000 2022", "id": "154944012026488711", "id_str": "154944012026488711", "full_text": "RT @mduq: @b645@b646 @b646@b645@b6cc@b641@b647@b645@b645 @b688@b631@b627 @b648@b642@b62a@b6",
24 "created_at": "Tue Jul 19 17:02:50 +0000 2022", "id": "1549439634773516288", "id_str": "1549439634773516288", "full_text": "RT @heyedamir: @b645@b646 @b646@b645@b6cc@b641@b647@b645@b645 @b688@b631@b627 @b648@b642@b62a@b6",
25 "created_at": "Tue Jul 19 17:02:02 +0000 2022", "id": "1549439452348080448", "id_str": "1549439452348080448", "full_text": "RT @mduq: @b645@b646 @b646@b645@b6cc@b641@b647@b645@b645 @b688@b631@b627 @b648@b642@b62a@b6",
26 "created_at": "Tue Jul 19 16:58:49 +0000 2022", "id": "1549438439753535568", "id_str": "1549438439753535568", "full_text": "@b645@b646 @b646@b645@b6cc@b641@b647@b645@b645 @b688@b631@b627 @b648@b642@b62a@b6",
27 "created_at": "Tue Jul 19 16:32:11 +0000 2022", "id": "154943191933302784", "id_str": "154943191933302784", "full_text": "@bavanz27 @b627@b644 @b627@b646 @b62a@b648 @b636@b627@b645 \\n@b627@b6cc@b646@b62a@b627 @b6",
28 "created_at": "Tue Jul 19 16:08:22 +0000 2022", "id": "1549423913716834384", "id_str": "1549423913716834384", "full_text": "RT @bavanz27 @b627@b644 @b627@b646 @b62a@b648 @b636@b627@b645 \\n@b627@b6cc@b646@b62a@b627 @b6",
29 "created_at": "Tue Jul 19 16:07:50 +0000 2022", "id": "154942327242950530", "id_str": "154942327242950530", "full_text": "@bavanz27 @b627@b644 @b627@b646 @b62a@b648 @b636@b627@b645 \\n@b627@b6cc@b646@b62a@b627 @b6",
30 "created_at": "Tue Jul 19 15:58:36 +0000 2022", "id": "15494236611425793", "id_str": "15494236611425793", "full_text": "@bavanz27 @b627@b644 @b627@b646 @b62a@b648 @b636@b627@b645 \\n@b627@b6cc@b646@b62a@b627 @b6",
31 "created_at": "Tue Jul 19 15:58:41 +0000 2022", "id": "1549420860989136896", "id_str": "1549420860989136896", "full_text": "@bavanz27 @b627@b644 @b627@b646 @b62a@b648 @b636@b627@b645 \\n@b627@b6cc@b646@b62a@b627 @b6",
32 "created_at": "Tue Jul 19 15:47:15 +0000 2022", "id": "1549420785663614978", "id_str": "1549420785663614978", "full_text": "@bavanz27 @b627@b644 @b627@b646 @b62a@b648 @b636@b627@b645 \\n@b627@b6cc@b646@b62a@b627 @b6",
33 "created_at": "Tue Jul 19 15:46:58 +0000 2022", "id": "1549420445514710785", "id_str": "1549420445514710785", "full_text": "@bavanz27 @b627@b644 @b627@b646 @b62a@b648 @b636@b627@b645 \\n@b627@b6cc@b646@b62a@b627 @b6",
34 "created_at": "Tue Jul 19 15:45:34 +0000 2022", "id": "1549420190697488384", "id_str": "1549420190697488384", "full_text": "@bavanz27 @b627@b644 @b627@b646 @b62a@b648 @b636@b627@b645 \\n@b627@b6cc@b646@b62a@b627 @b6",
35 "created_at": "Tue Jul 19 15:44:45 +0000 2022", "id": "1549420083152024065", "id_str": "1549420083152024065", "full_text": "@bavanz27 @b627@b644 @b627@b646 @b62a@b648 @b636@b627@b645 \\n@b627@b6cc@b646@b62a@b627 @b6",
36 "created_at": "Tue Jul 19 15:42:57 +0000 2022", "id": "1549419538098414880", "id_str": "1549419538098414880", "full_text": "@bavanz27 @b627@b644 @b627@b646 @b62a@b648 @b636@b627@b645 \\n@b627@b6cc@b646@b62a@b627 @b6",
37 "created_at": "Tue Jul 19 15:42:42 +0000 2022", "id": "1549418648382964549", "id_str": "1549418648382964549", "full_text": "@bavanz27 @b627@b644 @b627@b646 @b62a@b648 @b636@b627@b645 \\n@b627@b6cc@b646@b62a@b627 @b6",
38 "created_at": "Tue Jul 19 15:37:40 +0000 2022", "id": "154941819982960827", "id_str": "154941819982960827", "full_text": "@bavanz27 @b627@b644 @b627@b646 @b62a@b648 @b636@b627@b645 \\n@b627@b6cc@b646@b62a@b627 @b6",
39 "created_at": "Tue Jul 19 15:14:34 +0000 2022", "id": "1549412388247670784", "id_str": "1549412388247670784", "full_text": "@bavanz27 @b627@b644 @b627@b646 @b62a@b648 @b636@b627@b645 \\n@b627@b6cc@b646@b62a@b627 @b6",
40 "created_at": "Tue Jul 19 15:09:37 +0000 2022", "id": "154941114102966912", "id_str": "154941114102966912", "full_text": "@bavanz27 @b627@b644 @b627@b646 @b62a@b648 @b636@b627@b645 \\n@b627@b6cc@b646@b62a@b627 @b6",
41 "created_at": "Tue Jul 19 15:04:32 +0000 2022", "id": "1549407218671292416", "id_str": "1549407218671292416", "full_text": "RT @r1a1135: #b645@b646 @b646@b645@b6cc@b641@b647@b645@b645 @b688@b631@b627 @b648@b642@b62a@b6",
42 "created_at": "Tue Jul 19 14:51:32 +0000 2022", "id": "1549406589836380293", "id_str": "1549406589836380293", "full_text": "@bavanz27 @b627@b644 @b627@b646 @b62a@b648 @b636@b627@b645 \\n@b627@b6cc@b646@b62a@b627 @b6",
43 "created_at": "Tue Jul 19 14:51:34 +0000 2022", "id": "1549406589836380293", "id_str": "1549406589836380293", "full_text": "@bavanz27 @b627@b644 @b627@b646 @b62a@b648 @b636@b627@b645 \\n@b627@b6cc@b646@b62a@b627 @b6",
44 "created_at": "Tue Jul 19 14:48:28 +0000 2022", "id": "1549402341242870593", "id_str": "1549402341242870593", "full_text": "@bavanz27 @b627@b644 @b627@b646 @b62a@b648 @b636@b627@b645 \\n@b627@b6cc@b646@b62a@b627 @b6",
45 "created_at": "Tue Jul 19 14:17:44 +0000 2022", "id": "15493988010774528", "id_str": "15493988010774528", "full_text": "@bavanz27 @b627@b644 @b627@b646 @b62a@b648 @b636@b627@b645 \\n@b627@b6cc@b646@b62a@b627 @b6",
46 "created_at": "Tue Jul 19 14:13:46 +0000 2022", "id": "154939708799205738", "id_str": "154939708799205738", "full_text": "@bavanz27 @b627@b644 @b627@b646 @b62a@b648 @b636@b627@b645 \\n@b627@b6cc@b646@b62a@b627 @b6",
47 "created_at": "Tue Jul 19 14:04:31 +0000 2022", "id": "1549394768088086753", "id_str": "1549394768088086753", "full_text": "@bavanz27 @b627@b644 @b627@b646 @b62a@b648 @b636@b627@b645 \\n@b627@b6cc@b646@b62a@b627 @b6",
48 "created_at": "Tue Jul 19 13:57:48 +0000 2022", "id": "154939208635920208", "id_str": "154939208635920208", "full_text": "@bavanz27 @b627@b644 @b627@b646 @b62a@b648 @b636@b627@b645 \\n@b627@b6cc@b646@b62a@b627 @b6",
49 "created_at": "Tue Jul 19 13:56:52 +0000 2022", "id": "154939208635920208", "id_str": "154939208635920208", "full_text": "@bavanz27 @b627@b644 @b627@b646 @b62a@b648 @b636@b627@b645 \\n@b627@b6cc@b646@b62a@b627 @b6",
50 "created_at": "Tue Jul 19 13:53:20 +0000 2022", "id": "154939134336208897", "id_str": "154939134336208897", "full_text": "@bavanz27 @b627@b644 @b627@b646 @b62a@b648 @b636@b627@b645 \\n@b627@b6cc@b646@b62a@b627 @b6"
```

برای پس پردازش فایل‌های خروجی خزنده tweetf0rm می‌توان از کتابخانه tweeta^۱ استفاده کرد که مختص خروجی‌های همین خزنده پیاده‌سازی شده است. این کتابخانه متدهایی نظیر دریافت زمان ساخته شدن توییت، دریافت متن توییت، مشخص کردن اینکه آیا توییت مربوطه retweet است یا خیر و بسیاری متد کاربردی دیگر دارد. در این پروژه، از بین تمام ویژگی‌های موجود، ویژگی‌های زمان ساخته شدن توییت، آیدی کاربر، متن توییت، retweet بودن یا نبودن توییت مربوطه، quote بودن یا نبودن توییت مربوطه و reply بودن یا نبودن توییت مربوطه را جدا کردیم و همچنین ویژگی دیگری با نام tags که مجموعه‌ای از برچسب‌ها است را به آن اضافه کردیم و در فایل‌های جداگانه ذخیره کردیم. ویژگی tags مشخص می‌کند که هر توییت برای چه محصول یا برند خاص بوده است تا بتوانیم با توجه به آن دسته‌بندی و رتبه‌بندی‌های آخر را انجام دهیم.

^۱ <https://github.com/bianjiang/tweeta>

همچنین دلیل اینکه این ویژگی به صورت مجموعه است، این است که برخی از توییت‌ها تنها مربوط به یک محصول یا برند نیستند و کاربر مربوطه درباره مجموعه‌ای از آن‌ها نظر داده است. به عنوان مثال برای سه اپراتور ایرانسل، همراه اول و رایتل توییت‌هایی مانند توییت "همراه اول مثل ایرانسل وفا نداره" داریم که همان‌طور که مشخص است در این توییت هم اپراتور ایرانسل و هم همراه اول ذکر شده‌اند. این نوع از توییت‌ها، کار را برای مدل‌های آموزش دیده و تصمیم‌گیری بسیار سخت می‌کنند اما در خروجی نهایی می‌توان این‌گونه توییت‌ها را نادیده گرفت اما در این پروژه ما این توییت‌ها را نگه می‌داریم و تصمیم بر نادیده گرفتن یا نگرفتن آن‌ها را به شخصی که از خروجی استفاده می‌کند واگذار کردیم.

ویژگی‌های موردنظر:

features

```
# created_time
# user_id = tweet.user_id()
# text
# is_retweet = tweet.is_retweet()
# is_quote = tweet.is_quote()
# is_reply = tweet.get('in_reply_to_user_id') != None
# tags = 'hamrah_aval', 'irancell', 'rightel'
```

در نهایت پس از تکمیل مراحل پس پردازش، داده‌های پردازش شده را در قالب فایل‌های CSV ذخیره می‌کنیم تا بعداً با استفاده از مدل‌های آموزش دیده، برچسب‌های تحلیل احساسات و نظر هرز بودن یا نبودن به آن‌ها اضافه شود.

نمونه ابتدای فایل CSV پس از پس پردازش برای اپراتور همراه اول:

	created_time	user_id	text	is_retweet	is_quote	is_reply	tags
0	2022-07-19 19:18:27	1355911468902981639	هیچ کس تنها نیست ... همراه اول	False	False	True	{hamrah_aval}
1	2022-07-19 19:13:32	614986877	بلی، ایرانسل از همراه اول بهتره حتی	False	False	True	{hamrah_aval, irancell}
2	2022-07-19 19:09:10	941220112845164544	... بسته سه ماهه همراه اول رو دو دوره یجی شش ماه	False	False	True	{hamrah_aval}
3	2022-07-19 19:05:41	822277959176306688	همراه اول هم همینه	False	False	True	{hamrah_aval}
4	2022-07-19 19:00:44	1111363641545236480	... بسته‌ی سه گیگ ماهانه‌ی اینترنت همراه اول می‌خر	False	False	False	{hamrah_aval}

فصل ۳

تحلیل احساسات

۱-۳ مقدمه

با رشد روز افزون استفاده از اینترنت و گسترش خدمات اینترنتی، هر روزه داده‌های بسیار زیادی در هر حوزه تولید می‌شوند. این داده‌ها که امروزه از آن‌ها با عنوان گنج نو یاد می‌شود. می‌تواند شامل اطلاعات بسیار زیادی باشد که برای کاربردهای گوناگون می‌توان از آن‌ها استفاده کرد. قسمت عمده‌ای از داده‌های موجود در اینترنت را داده‌های متنی تشکیل می‌دهند.

با استفاده از روش‌های مختلف پردازش زبان طبیعی می‌توان این داده‌ها را پردازش کرد و اطلاعات مفیدی را از آن‌ها استخراج کرد. کسب و کارهای مختلف از داده‌ها برای به دست آوردن اطلاعات به منظور بهبود کسب و کار خود استفاده می‌کنند. یکی از این اطلاعات مفید که کاربردهای زیادی دارد احساس یک متن است. تحلیل احساسات یک روش هوشمند پردازش زبان طبیعی است که به وسیله‌ی آن می‌توان میزان مثبت یا منفی بودن یک متن را تشخیص داد. یکی از کاربردهای مهم تحلیل احساسات، تحلیل احساسات بر روی نظرات کاربران است. یعنی نظراتی که برای یک محصول، یک فیلم، یک رستوران و ... نوشته می‌شود. با به دست آوردن احساس یک نظر می‌توان میزان رضایت و یا عدم رضایت افراد از یک محصول را به دست آورد و با انجام این کار به بهبود محصول و کسب و کار پرداخت. این تنها یکی از کاربردهای تحلیل احساسات است در حالی که احساس یک متن می‌تواند در موارد مختلف زیادی کاربرد داشته باشد. در زبان فارسی علی‌رغم اینکه داده‌های بسیار زیاد و کاربران زیادی که از سرویس‌های فارسی استفاده می‌کنند، به دلایل مختلف مدل‌های خوبی برای تحلیل احساسات در زبان فارسی وجود ندارد، البته در این اواخر تحقیق‌ها و پروژه‌ها در این رابطه بیشتر شده است و نتایج نسبتاً قابل قبول هم اکتساب کرده‌اند. با توجه به کاربردهای مختلف تحلیل احساسات، نیاز است مدل مناسبی برای زبان فارسی توسعه داده شود تا بتوان از کاربردهای مختلف این مدل بهره‌مند شد.

۲_۳ کارهای مرتبط

تحقیقات مختلف زیادی در سال‌های مختلف برای مسئله تحلیل احساسات در زبان‌های مختلف انجام شده است اما برای زبان فارسی علی‌رغم نیاز فراوان به یک مدل مناسب برای تحلیل احساسات، پژوهش‌های زیادی برای تحلیل احساسات متن انجام نشده است و برخی تحقیقات انجام شده نیز به دلیل اینکه از روش‌های کلاسیک یادگیری ماشین استفاده می‌کنند به دقت‌های مناسبی نمی‌رسند.

اما در سال‌های اخیر استفاده از یادگیری عمیق برای مسئله تحلیل احساسات در زبان فارسی شدت گرفته است. در تحقیقی که در سال ۲۰۱۷ انجام شد [۹] چند مجموعه داده‌ی جدید برای تحلیل احساسات در زبان فارسی معرفی شد که همگی آن‌ها از وب‌سایت دیجی‌کالا جمع‌آوری شده‌اند. این مجموعه داده‌ها شامل دو برچسب هستند. یکی برچسب قطبیت نظر و یکی مربوط به امتیاز داده شده برای محصول است. در نهایت با استفاده از روش بیزی ساده به آموزش و ارزیابی مدل خود بر روی این مجموعه داده‌ها پرداختند.

تحقیق دیگری نیز در سال ۲۰۱۷ [۱۰] به منظور بررسی روش‌های مبتنی بر واژه‌نامه برای تحلیل احساسات در زبان فارسی صورت گرفته است. موضوعی که در این مقاله بررسی شده است مسئله ترجمه در تحلیل احساسات برای زبان فارسی است. در این مقاله بیان شده است که ترجمه به زبان دیگر عملکرد ضعیف‌تری نسبت به انجام پیش پردازش و سپس استفاده از روش مبتنی بر واژه‌نامه دارد.

در تحقیق دیگری که در سال ۲۰۱۸ انجام شد [۴] نیز مجموعه داده‌ی دیگری به نام SentiPers معرفی شد که در تحقیق پیش‌رو نیز از این مجموعه داده استفاده شده است. این مجموعه داده نیز از وب‌سایت دیجی‌کالا جمع‌آوری شده است. این مجموعه داده بیش از ۲۶۰۰۰ جمله دارد و توسط دانشجویان دانشگاه گیلان و به صورت دستی برچسب گذاری شده است. در سال ۲۰۱۹ نیز تحقیقی انجام شد [۱۱] که مسئله تحلیل احساسات در زبان فارسی را با استفاده از دو رویکرد بررسی کرده است. رویکرد اول استفاده از ترکیب مدل‌های یادگیری ماشین و رویکرد دوم استفاده از روش‌های یادگیری عمیق با کمک جاسازی کلمات می‌باشد. در این تحقیق با آموزش و ارزیابی مدل بر روی مجموعه داده‌ای شامل نظرات مربوط به هتل‌ها که از وب‌سایت booking.ir جمع‌آوری شده بود نتیجه‌ای که به دست آمد این بود که روش‌های یادگیری عمیق به مراتب عملکرد بهتری نسبت به روش‌های یادگیری ماشین کلاسیک دارند. همچنین در میان مدل‌های یادگیری عمیق نتایج به دست آمده در این تحقیق برای شبکه‌های عصبی کانولوشنال^۱ بهتر از دیگر مدل‌های یادگیری عمیق می‌باشد. در این تحقیق یادگیری در دو سطح انجام شده است. یادگیری در سطح سند و یادگیری در سطح جمله که نتایج به دست آمده برای هر سطح نشان دهنده تفاوت عملکرد مدل در این سطوح است و هر کدام از آن‌ها در بخشی نسبت به دیگری برتری دارند. تحقیق دیگری نیز در سال ۲۰۲۰

^۱ Convolutional Neural Network (CNN)

انجام شد [۱۲] که به مسئله تحلیل احساسات در زبان فارسی برای داده‌های موجود در شبکه‌های اجتماعی پرداخته است. در این تحقیق از شبکه‌های عصبی کانولوشنال برای تحلیل احساسات استفاده شده است و در پایان به این نتیجه رسیده‌اند که این مدل عملکرد بهتری نسبت به شبکه‌های حافظه طولانی کوتاه مدت^۱، شبکه‌های عصبی بازگشتی^۲ و روش‌های یادگیری ماشین کلاسیک دارد.

همان‌طور که گفته شد تحقیقات در این زمینه رو به رشد است و در سال‌های اخیر تعداد تحقیقات بیشتر شده است. در سال ۲۰۲۰ تحقیقی [۱۳] انجام شد که سعی داشت با دخیل کردن قواعد گرامری در یادگیری به موضوع تحلیل احساسات در زبان فارسی پردازد همچنین مدل‌های یادگیری عمیق در این تحقیق استفاده شده است. در سال ۲۰۲۱ تحقیقات مختلفی برای تحلیل احساسات در زبان فارسی انجام شده است که از جمله آن‌ها می‌توان به [۱۴] اشاره کرد که با استفاده از داده‌های مختلف صوتی، تصویری و متنی به تحلیل احساسات پرداخته است. در این تحقیق از مدل‌های مختلف یادگیری عمیق برای تحلیل احساسات استفاده شده است. بهترین نتیجه به دست آمده برای داده‌های متنی مربوط به مدلی است که از شبکه‌های حافظه طولانی کوتاه مدت استفاده کرده است. یکی دیگر از تحقیقات سال ۲۰۲۱ تحقیق [۱۵] است که با هدف تحلیل احساسات نظرات مربوط به فیلم‌ها انجام شده است. در این تحقیق انواع مدل‌های یادگیری عمیق عملکردشان با یکدیگر مقایسه شده است. در نهایت با نتایج به دست آمده در این تحقیق این نتیجه به دست آمده که عملکرد شبکه‌های حافظه طولانی کوتاه مدت دو طرفه بهتر از سایر روش‌های یادگیری عمیق است. همچنین به طور کلی عملکرد روش‌های یادگیری عمیق بهتر از روش‌های یادگیری ماشین کلاسیک است.

با توجه به تحقیقات اخیر و روش‌های استفاده شده در موضوع تحلیل احساسات در زبان فارسی، به نظر می‌رسد استفاده از یادگیری عمیق نتایج بهتری به دست می‌دهد.

۳-۳ مدل پیشنهادی

توسعه مدلی برای تحلیل احساسات در زبان فارسی با توجه به کاربردهای زیاد آن در موضوعات مختلف اهمیت ویژه‌ای دارد. در این پروژه مدلی برای تحلیل احساسات در زبان فارسی پیاده‌سازی شده است. مدل ارائه شده، مدل یادگیری عمیق Transformers و ParsBERT است که با دقت بالایی احساس متن را تشخیص می‌دهد.

^۱ Long Short-Term Memory (LSTM)

^۲ Recurrent Neural Network (RNN)

Transformers ۱_۳_۳

«ترانسفورمر» یک مدل یادگیری عمیق است که مکانیزم توجه به خود، به طور متفاوتی اهمیت هر بخش از داده‌های ورودی را وزن می‌کند. مانند شبکه‌های عصبی بازگشتی، ترانسفورمرها برای مدیریت داده‌های ورودی متوالی، مانند زبان طبیعی، برای کارهایی مانند ترجمه و خلاصه متن طراحی شده‌اند. با این حال، برخلاف شبکه‌های عصبی بازگشتی، ترانسفورمرها لزوماً داده‌ها را به ترتیب پردازش نمی‌کنند. در عوض، مکانیزم توجه زمینه را برای هر موقعیتی در دنباله ورودی فراهم می‌کند. به عنوان مثال، اگر داده ورودی یک جمله زبان طبیعی باشد، ترانسفورمر نیازی به پردازش ابتدای جمله قبل از پایان ندارد. در عوض، زمینه‌ای را مشخص می‌کند که به هر کلمه در جمله معنا می‌بخشد. این ویژگی اجازه می‌دهد تا موازی‌سازی بیشتر از شبکه‌های عصبی بازگشتی باشد و بنابراین زمان آموزش را کاهش می‌دهد. ترانسفورمرها در سال ۲۰۱۷ توسط تیمی در Google Brain [۱] معرفی شدند و به طور فزاینده‌ای مدل انتخابی برای مشکلات NLP هستند. این مدل جایگزین مدل‌های شبکه عصبی بازگشتی مانند حافظه طولانی کوتاه مدت می‌باشد. موازی‌سازی آموزش، امکان آموزش بر روی مجموعه داده‌های بزرگتر را می‌دهد. این منجر به توسعه مدل‌های از پیش آموزش دیده مانند BERT (مدل زبانی) و GPT [۱۶] که با مجموعه داده‌های زبانی بزرگ، مانند ویکی‌پدیا و Common Crawl آموزش دیده‌اند، و می‌توان آن‌ها را برای کارهای خاص به خوبی تنظیم کرد.

۱_۱_۳_۳ معماری رمزگذار-رمزگشا^۱

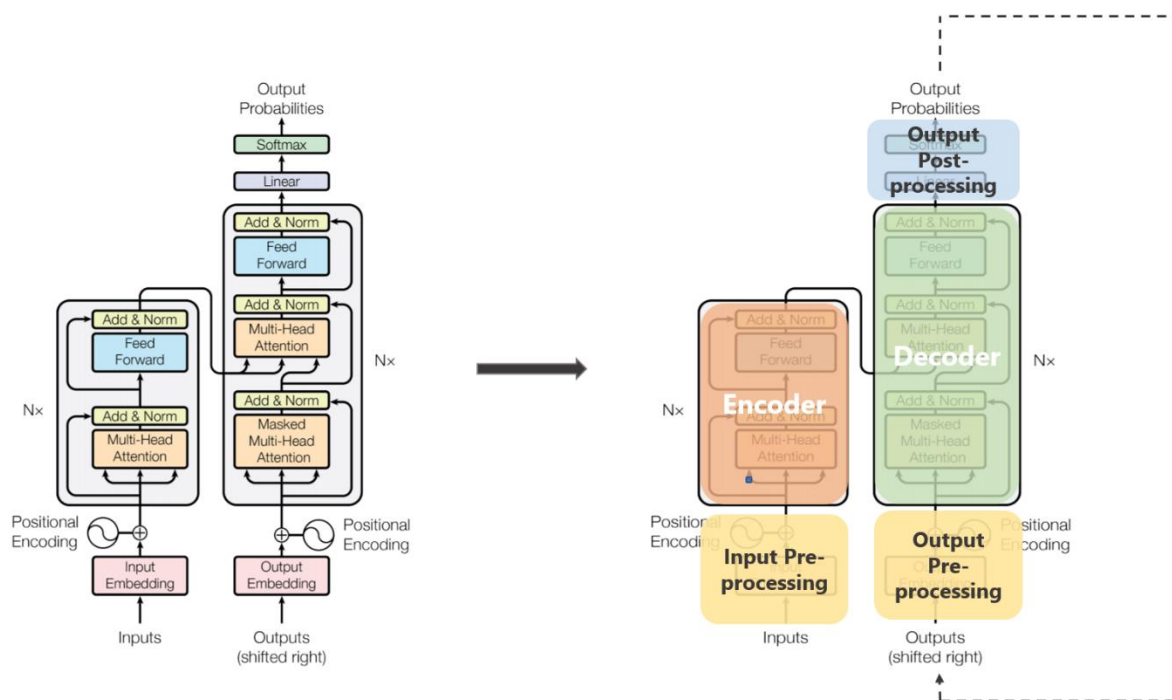
مانند مدل‌های قبلی [۱۷] seq2seq، مدل اولیه ترانسفورمر از معماری «رمزگذار-رمزگشا» استفاده می‌کرد. رمزگذار از لایه‌های رمزگذاری تشکیل شده است که ورودی را به طور مکرر یکی پس از دیگری پردازش می‌کند، در حالی که رمزگشا شامل لایه‌های رمزگشایی است که همان کار را با خروجی رمزگذار انجام می‌دهد. وظیفه هر لایه رمزگذار تولید رمزگذاری‌هایی است که حاوی اطلاعاتی در مورد اینکه کدام بخش از ورودی‌ها با یکدیگر مرتبط هستند. رمزگذاری‌های خود را به عنوان ورودی به لایه رمزگذار بعدی ارسال می‌کند. هر لایه رمزگشا برعکس عمل می‌کند، تمام رمزگذاری‌ها را می‌گیرد و از اطلاعات متنی آن‌ها برای تولید یک دنباله خروجی استفاده می‌کند. برای هر ورودی، توجه، ارتباط هر ورودی دیگر را می‌سنجد و از آن‌ها برای تولید خروجی استفاده می‌کند. هر لایه رمزگشا دارای مکانیزم توجه اضافی است که اطلاعات را از خروجی‌های رمزگشاهای قبلی، قبل از اینکه لایه رمزگشا اطلاعات را از رمزگذاری‌ها ترسیم کند، می‌گیرد. هر دو لایه رمزگذار و رمزگشا دارای یک شبکه عصبی پیش‌خور^۲ برای پردازش اضافی خروجی‌ها هستند و شامل اتصالات باقی‌مانده و مراحل عادی‌سازی لایه هستند.

^۱ Encoder-Decoder

^۲ Feedforward Neural Network

۳_۳_۱_۲ توجه چند سر^۱

توجه چند سر یک ماژول برای مکانیزم های توجه است که چندین بار به طور موازی از یک مکانیزم توجه عبور می کند. هر لایه در یک مدل ترانسفورمر دارای سرهای توجه متعددی است. در حالی که هر سر توجه به نشانه های مربوط به هر نشانه توجه می کند، با سرهای توجه متعدد، مدل می تواند این کار را برای تعاریف مختلف «ارتباط» انجام دهد. علاوه بر این، میدان تأثیر نشان دهنده ارتباط می تواند به تدریج در لایه های متوالی گشاد شود. بسیاری از سرهای توجه ترانسفورمر روابط مرتبطی را رمزگذاری می کنند که برای انسان معنادار است. برای مثال، سرهای توجه می توانند بیشتر به کلمه بعدی توجه کنند، در حالی که دیگران عمدتاً از افعال به مفعول مستقیم خود توجه می کنند. محاسبات برای هر سر توجه را می توان به صورت موازی انجام داد، که امکان پردازش سریع را فراهم می کند. خروجی های لایه توجه به هم متصل می شوند تا به لایه های شبکه عصبی پیش خور منتقل شوند.

¹ Multi-Head Attention

ParsBERT ۲_۳_۳

ParsBERT یک مدل زبانی تک‌زبانه بر اساس معماری BERT گوگل است. این مدل بر روی مجموعه‌های بزرگ فارسی با سبک‌های نوشتاری مختلف از موضوعات متعدد (مانند علمی، رمان، اخبار) با بیش از ۳/۹ میلیون سند، ۷۳ میلیون جمله و ۱/۳ میلیارد کلمه از قبل آموزش داده شده است.

BERT در هسته خود یک مدل زبانی ترانسفورمر با تعداد متغیر لایه‌های رمزگذار و سرهای خود توجه^۱ است. معماری آن با ترانسفورمر اصلی تقریباً یکسان است.

BERT روی دو کار از قبل آموزش دیده بود: مدل‌سازی زبان (۱۵٪ توکن‌ها Mask شده بودند و BERT برای پیش‌بینی آن‌ها از متن مربوطه، آموزش داده شد) و پیش‌بینی جمله بعدی (BERT آموزش داده شد تا پیش‌بینی کند که آیا جمله بعدی انتخابی محتمل است یا جمله اول داده نشده است). در نتیجه فرآیند آموزشی، BERT جاسازی‌های متنی کلمات را یاد می‌گیرد. پس از پیش آموزش، که از نظر محاسباتی گران است، BERT را می‌توان با منابع کمتری برای مجموعه‌های داده کوچک‌تر تنظیم کرد تا عملکرد آن در کارهای خاص بهینه شود.

ورودی مدل BERT ترکیبی از سه جاسازی است:

۱. جاسازی توکن: واژگان توکن WordPiece (WordPiece) یکی دیگر از الگوریتم‌های تقسیم‌بندی کلمه مشابه BPE است).

۲. جاسازی‌های بخش: برای جملات زوج [A-B] که به عنوان EA یا EB علامت گذاری شده‌اند به این معنی است که به جمله اول یا دوم تعلق دارد.

۳. جاسازی موقعیت: موقعیت کلمات را در یک جمله مشخص می‌کند.

همچنین مدل BERT تنها از قسمت رمزگذار Transformers استفاده می‌کند. معماری‌های مربوط به جاسازی ورودی‌ها و مدل BERT که ParsBERT هم از همین معماری‌ها استفاده می‌کند، در بخش معماری مدل آمده‌اند.

¹ Self-Attention Heads

۳-۳-۳ پیش پردازش^۱

در این تحقیق بر روی متن ورودی مراحل مختلف پیش پردازش شامل نرمال سازی، توکن بندی و ... با استفاده از کتابخانه هضم^۲ انجام شده است. همچنین به دلیل نامتوازن بودن مجموعه داده از روش متوازن سازی نمونه برداری بیشینه^۳، به این صورت که داده های چندین مجموعه داده SentiPers با هم ترکیب شده اند و همچنین استفاده از داده های ترجمه معکوس^۴ و موارد دیگری برای بهبود مدل، نیز استفاده شده است.

۳-۳-۴ جاسازی کلمات^۵

برای اینکه بتوانیم کلمات را با یکدیگر مقایسه کنیم و میزان نزدیکی و ارتباط آن ها با یکدیگر را توسط ماشین مقایسه کنیم نیاز داریم هر کدام از این کلمات را به یک بردار نسبت دهیم یعنی یک بردار از اعداد نمایانگر یک کلمه است و مقایسه این بردارها با یکدیگر می تواند ارتباط این کلمات با یکدیگر را به ما نشان دهد. به تولید این بردارها جاسازی کلمات گفته می شود [۱۸]. برای تولید این بردارها نیز می توان آن ها را توسط یک مدل آموزش داد و یا از بردارهای تولید شده توسط مدل های پیش آموزش دیده استفاده کرد. در این پروژه، از مدلی که درون خود مدل پیش آموزش دیده ParsBERT وجود دارد (مدل های پیش آموزش دیده BERT، جاسازی کلمات را درون خود دارند)، استفاده کرده ایم.

۳-۳-۵ معماری مدل

مدل ارائه شده در این پروژه همان مدل ParsBERT و Transformers است. در این پروژه از روش یادگیری نظارت شده استفاده شده است یعنی برای آموزش مدل به متن و برچسب آن (قطبیت نظر)، نیاز است. تحلیل احساسات در این پروژه به صورت پنج دسته [-2, -1, 0, 1, 2] انجام می شود که 2- نشان دهنده حالت Furious، 1- نشان دهنده حالت Angry، 0 نشان دهنده حالت Neutral، 1 نشان دهنده حالت Happy و 2 نشان دهنده حالت Delighted است و در واقع دارای پنج کلاس هستیم. همانطور که در معماری ها مشخص است، پس از انجام پیش پردازش بر روی متن ورودی و توکن بندی آن (با

¹ preprocessing

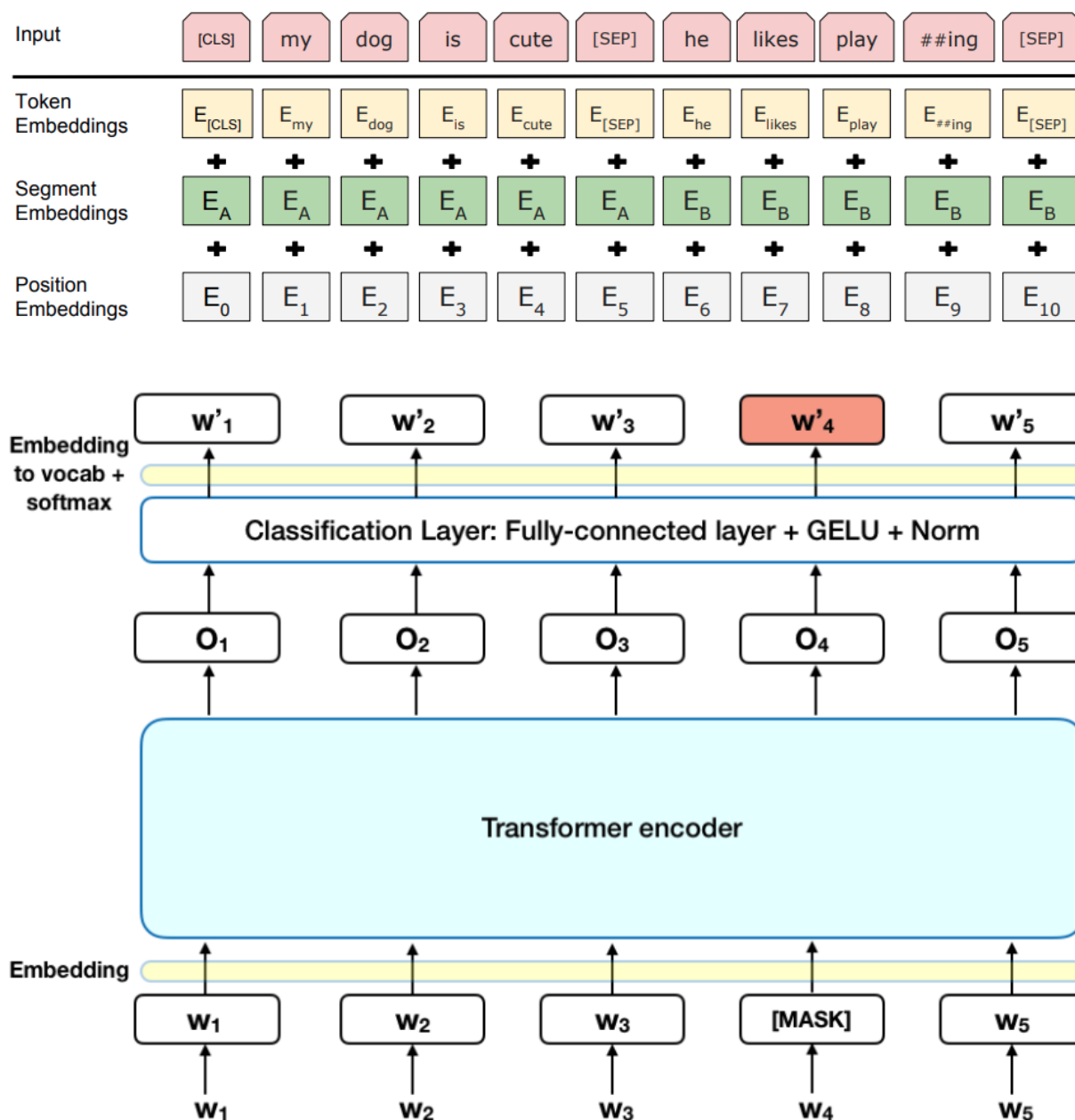
² Hazm

³ Oversampling

⁴ Back translation

⁵ Word Embedding

استفاده از کتابخانه هضم) با استفاده از سه حالت جاسازی کلمات در مدل‌های BERT، بردارهای متناسب با کلمات در هر سه حالت استخراج شده و به مدل ParsBERT داده می‌شود.



۴-۳ نتایج تجربی

در این بخش نتایج به دست آمده برای آموزش و ارزیابی مدل پیشنهادی بر روی مجموعه داده SentiPers بررسی می‌شود.

۱_۴_۳ مجموعه داده^۱

مجموعه داده‌ای که برای آموزش و ارزیابی استفاده شده، مجموعه داده SentiPers است. این مجموعه داده در سال ۲۰۱۸ و در تحقیق [۴] معرفی گردید. این مجموعه داده شامل نظراتی است که از وبسایت دیجی‌کالا جمع‌آوری گردیده و توسط دانشجویان دانشگاه گیلان برچسب گذاری شده است. این مجموعه داده شامل ۲۶۰۰۰ نظر است و دارای ۵ سطح احساس است که از نظرات بسیار منفی با برچسب ۲- تا نظرات بسیار مثبت با برچسب ۲+ در ۵ سطح طبقه‌بندی شده‌اند.

۲_۴_۳ تنظیمات مدل^۲

تنظیمات مدل اعم از برچسب‌ها، بیشینه طول^۳، batch_size، تعداد Epoch و ... طبق موارد زیر است:

```
[ ] # general config
MAX_LEN = 128
TRAIN_BATCH_SIZE = 16
VALID_BATCH_SIZE = 16
TEST_BATCH_SIZE = 16

EPOCHS = 5
EEVERY_EPOCH = 1000
LEARNING_RATE = 2e-5
CLIP = 0.0

MODEL_NAME_OR_PATH = 'HooshvareLab/bert-fa-base-uncased'
OUTPUT_PATH = '/content/bert-fa-base-uncased-sentiment/pytorch_model.bin'

os.makedirs(os.path.dirname(OUTPUT_PATH), exist_ok=True)
```

^۱ Dataset

^۲ Configuration

^۳ MAX_LEN

```
[ ] # create a key finder based on label 2 id and id to label

label2id = {label: i for i, label in enumerate(labels)}
id2label = {v: k for k, v in label2id.items()}

print(f'label2id: {label2id}')
print(f'id2label: {id2label}')

label2id: {'angry': 0, 'delighted': 1, 'furious': 2, 'happy': 3, 'neutral': 4}
id2label: {0: 'angry', 1: 'delighted', 2: 'furious', 3: 'happy', 4: 'neutral'}
```

```
[ ] # setup the tokenizer and configuration

tokenizer = BertTokenizer.from_pretrained(MODEL_NAME_OR_PATH)
config = BertConfig.from_pretrained(
    MODEL_NAME_OR_PATH, **{
        'label2id': label2id,
        'id2label': id2label,
    })
```

۳_۴_۳ نتایج

مدل پیشنهادی پس از ارزیابی نتایج زیر را به دست آورد:

F1: 0.9491487299204161				
	precision	recall	f1-score	support
furios	0.77	0.83	0.80	12
angry	0.94	0.96	0.95	183
neutral	0.97	0.97	0.97	712
happy	0.94	0.92	0.93	545
delighted	0.93	0.95	0.94	337
accuracy			0.95	1789
macro avg	0.91	0.93	0.92	1789
weighted avg	0.95	0.95	0.95	1789

۵_۳ جمع بندی

در این پروژه مدلی بر اسلِس Transformers و ParsBERT برای تحلیل احساسات در زبان فارسی ارائه شد و با آموزش و ارزیابی آن بر روی مجموعه داده SentiPers که از وبسایت دیجی کالا جمع آوری شده عملکرد قابل قبولی به دست آمده است. با مقایسه مدل پیشنهادی با کارهای مشابه مشخص شد که مدل پیشنهادی با رسیدن به F1 برابر با ۹۴/۹ درصد عملکرد بهتری نسبت به سایر مدل ها داشته است.

فصل ۴

تشخیص نظرات هرز

۴_۱ مقدمه

نزدیک به ۹۵ درصد افراد قبل از خرید آنلاین یک محصول، ابتدا نظرات نوشته شده درباره آن را مطالعه کرده سپس تصمیم به خرید می گیرند [۱۹]. تاثیر این نظرات فقط بر روی کاربران نیست و کسب و کارها نیز از این نظرات برای بهبود کیفیت خدمات و کالاهای خود استفاده می کنند. با توجه به اهمیت این نظرات و تاثیر آن بر فروش محصولات، تولید نظرات هرز نیز گسترش یافته است. نظر هرز به نظرانی گفته می شود که حاصل تجربه واقعی فرد نیست و با هدف تبلیغ و یا تخریب یک برند نوشته می شود که می تواند کاربران دیگر را دچار اشتباه در تصمیم گیری کند.

در سال های اخیر تحقیقات زیادی برای تشخیص نظرات هرز انجام شده است و کسب و کارهای مختلف نیز به دنبال راهکاری برای مقابله با نظرات هرز هستند. تعداد تحقیقات انجام شده در این حوزه به صورت نمایی رو به افزایش است [۲۰] اما برای زبان فارسی علی رغم گسترش سرویس های مبتنی بر اینترنت، تحقیقات زیادی برای تشخیص نظرات هرز انجام نشده است و بیشتر تحقیقات موجود نیز از روش های یادگیری ماشین کلاسیک استفاده کرده اند.

در این پروژه مدلی با استفاده از روش های یادگیری عمیق، Transformers و ParsBERT پیاده سازی شده است که تنها روی متن نظر آموزش می بیند.

پس از آموزش و ارزیابی مدل پیشنهادی بر روی مجموعه داده های Digikala و fake-review-dataset که پرکاربردترین مجموعه داده زبان فارسی برای این حوزه است، نتایج به دست آمده نشان دهنده عملکرد بهتر مدل پیشنهادی نسبت به روش های موجود است.

۲_۴ کارهای مرتبط

تحقیق بر روی مسئله تشخیص هرزنامه‌ها (تشخیص ایمیل هرز، تشخیص فکس هرز و ...) سابقه‌ای طولانی دارد اما تشخیص نظرات هرز به صورت کلی موضوعی نسبتاً تازه است. تحقیقات برای شناسایی نظرات هرز اولین بار توسط Jindal و همکارانش شروع شد [۲۱]. آن‌ها نظرات هرز را به سه دسته نظرات غیر واقعی، نظرات روی برند (نظراتی که فقط در مورد برند صحبت کرده‌اند و از کالا هیچ حرفی نزده‌اند) و نظرات غیر مرتبط تقسیم کردند. آن‌ها بیان کردند که دسته دوم و سوم نظرات مشکلی ایجاد نمی‌کنند و توسط کاربران قابل تشخیص هستند. اما نظرات دسته اول به راحتی قابل تشخیص نیستند و باید برای تشخیص آن‌ها مدلی ایجاد شود [۱۹]. در تحقیق [۲۳] از اندازه‌گیری شباهت نظرات با یکدیگر برای تشخیص نظرات هرز استفاده شده است. آن‌ها در تحقیقشان بیان کرده‌اند که هرزنویسان از چند الگوی ثابت برای نوشتن نظرات هرز استفاده می‌کنند و بر همین اساس نظرات هرز شباهت زیادی با یکدیگر دارند. هر چند این استدلال به نظر ناقص می‌آید و مدل خیلی خوبی نیست اما می‌توانست برای شروع روش خوبی باشد.

بعد از تحقیقات بیشتری در این زمینه انجام شد و روش‌های بهتری ارائه گردید. در تحقیق Wael و همکارانش [۲۴] موضوع تاثیر مراحل مختلف پیش پردازش بر روی داده‌ها، در کارایی مدل تشخیص نظرات هرز مورد بررسی قرار گرفت. چند روش پیش پردازش بر روی متن از جمله حذف کلمات توقف، حذف نشان‌های تاکید، ریشه‌یابی و ... در این تحقیق مورد بررسی قرار گرفت و تاثیر هر کدام از این روش‌ها با آموزش چند مدل مختلف یادگیری ماشین مثل شبکه بیزی ساده، ماشین بردار پشتیبان، جنگل تصادفی و ... اندازه‌گیری شد.

با گسترش روش‌های یادگیری عمیق و عملکرد خوب این مدل‌ها در مسائل مختلف، در تحقیقات تشخیص نظرات هرز نیز از این روش‌ها استفاده شد. Lie و همکارانش از شبکه‌های CNN برای تشخیص نظرات هرز استفاده کرده‌اند [۲۶]. بردارهای کلمات به عنوان ویژگی‌های ورودی به شبکه داده می‌شود و به طور مستقیم به شناسایی نظرات هرز با استفاده از CNN پرداخته می‌شود.

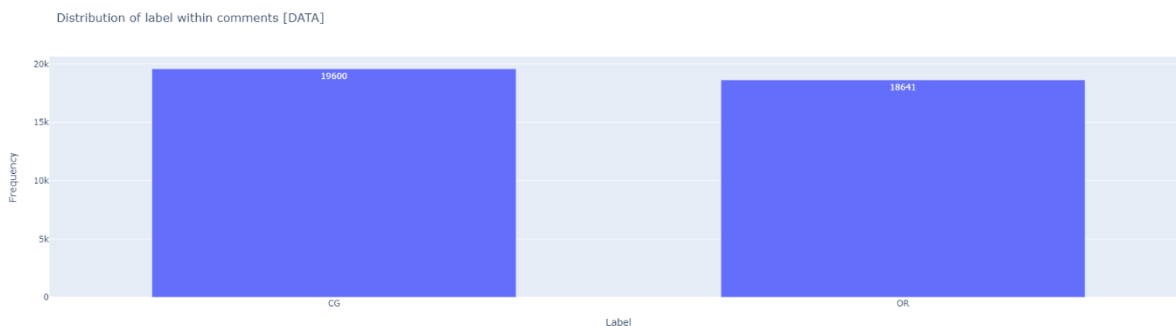
برای تشخیص نظرات هرز در زبان فارسی تحقیقات کمی انجام شده است. بیشتر تحقیقات موجود نیز از روش‌های یادگیری ماشین کلاسیک برای این موضوع استفاده کرده‌اند به همین دلیل نتایج آن‌ها خیلی خوب نیست. Safarian و همکارانش [۲۵] سعی کرده‌اند که به بررسی ویژگی‌های مختلف که برای آموزش مدل در مسئله تشخیص نظرات هرز استفاده می‌شود، بپردازند. آن‌ها مدل‌های مختلفی مثل روش‌های بیزی ساده، درخت تصمیم، ماشین بردار پشتیبان و ... را استفاده کرده‌اند و هر کدام از این مدل‌ها را با ویژگی‌های مختلف مثل امتیاز کلی کالا، احساسات نظرات، تگ POS و ... آموزش دادند. در این تحقیق داده‌های آموزشی از نظرات کاربران وبسایت دیجی‌کالا (بزرگترین سایت خرده فروشی در ایران) استفاده شده است.

Basiri و همکارانش [۲۷] نیز سعی کردند با استفاده از روش‌های مختلف یادگیری ماشین مثل بیزی ساده، درخت تصمیم، ماشین بردار پشتیبان و استفاده از ویژگی‌های مختلفی که از متن نظر استخراج می‌شود و همچنین استفاده از فراداده‌های دیگر که در مجموعه داده موجود است به شناسایی نظرات هرز در زبان فارسی بپردازند. این تحقیق بر روی داده‌های متوازن و نامتوازن انجام شده است و با توجه به نتایج به دست آمده، ماشین بردار پشتیبان برای داده‌های نامتوازن و درخت تصمیم برای داده‌های متوازن بهترین عملکرد را دارند.

۳_۴ مدل پیشنهادی

این بخش مشابه با بخش مدل پیشنهادی فصل سوم (تحلیل احساسات) است و تنها مجموعه داده ورودی و تعداد کلاس‌ها (که در اینجا ۲ کلاس، ۰ برای نظرات هرز و ۱ برای نظرات واقعی، است) متفاوت است، همچنین در پیش پردازش داده‌ها برای متوازن سازی به جای OverSampling از UnderSampling استفاده می‌شود به این صورت که از کلاسی که داده‌های بیشتری دارد به صورت تصادفی تعداد کمتری را نگه می‌داریم.

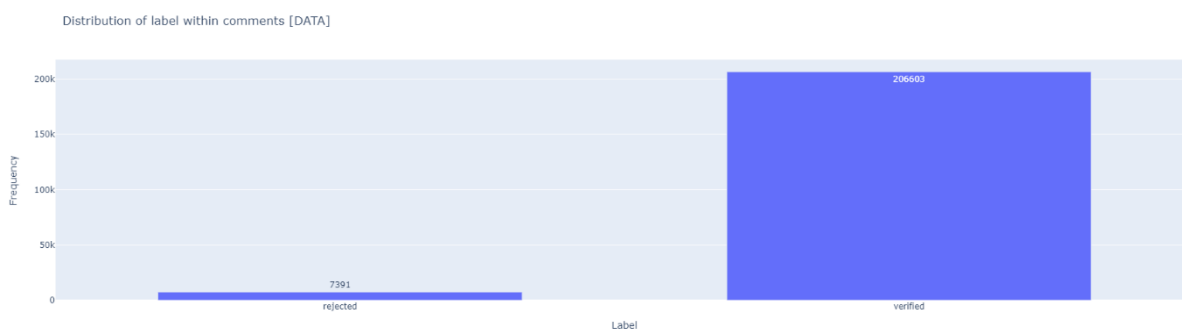
نمودار فراوانی داده‌ها قبل از UnderSampling در مجموعه داده fake-reviews-dataset:



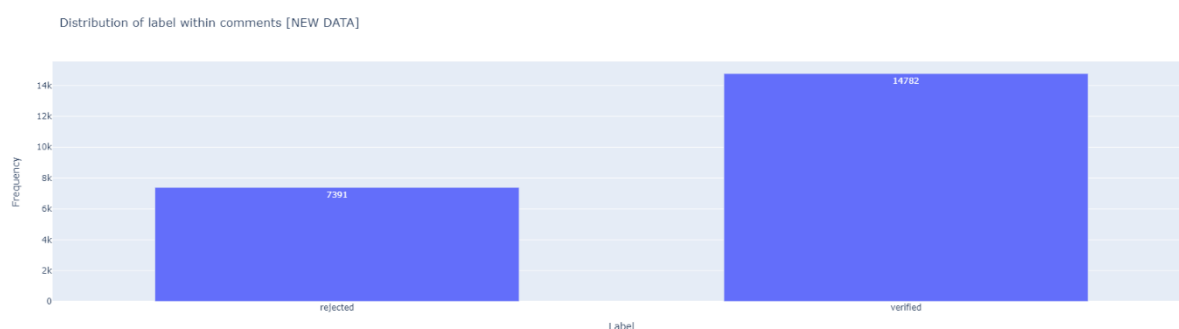
نمودار فراوانی داده‌ها بعد از UnderSampling در مجموعه داده fake-reviews-dataset:



نمودار فراوانی داده‌ها قبل از UnderSampling در مجموعه داده Digikala:



نمودار فراوانی داده‌ها بعد از UnderSampling در مجموعه داده Digikala:



۴_۴ نتایج تجربی

در این بخش نتایج به دست آمده از آموزش و ارزیابی مدل بر روی مجموعه داده‌های Digikala و fake-reviews-dataset آورده شده است.

۱_۴_۴ مجموعه داده‌ها

مجموعه داده‌های استفاده شده در این پروژه:

مجموعه داده fake-reviews-dataset که توسط Joni Salminen و همکارانش [۷] با استفاده از یک مدل زبانی که روی نظرات هرز آموزش دیده شده تا بتواند با این روش نظرات هرز تولید کند و تعداد نظرات هرز را در این مجموعه داده افزایش دهد، منتشر شده است. همچنین این مجموعه داده به زبان انگلیسی است و برای آموزش و ارزیابی مدل پیشنهادی که برای زبان فارسی است روی این مجموعه داده از ترجمه ماشینی Google Translate استفاده کردیم و تمام نظرات را به زبان فارسی ترجمه کردیم. این مجموعه داده شامل ۴۰۰۰۰ نظر است که از این تعداد ۲۰۰۰۰ نظر واقعی (با برچسب OR) و ۲۰۰۰۰ نظر نیز غیر واقعی (با برچسب CG) هستند که پس از پاک کردن مغایرت‌ها، ۱۸۶۴۱ نظر واقعی (با برچسب OR) و ۱۹۶۰۰ نظر غیر واقعی (با برچسب CG)، باقی می‌ماند و در انتها پس از UnderSampling تعداد ۱۸۶۴۱ نظر واقعی (با برچسب OR) و ۱۸۶۴۱ نظر غیر واقعی (با برچسب CG)، باقی می‌ماند. برای هر نظر ۴ ویژگی (متن نظر، برچسب نظر، امتیاز و دسته‌بندی) موجود است که از بین آن‌ها برای آموزش تنها به متن نظر و برچسب آن نیاز است.

مجموعه داده Digikala که توسط شرکت دیجی‌کالا که بزرگ‌ترین شرکت خرده‌فروشی در ایران است، منتشر شده است. این مجموعه داده شامل ۲۶۴۳۹۹ نظر است که از این تعداد ۲۵۴۴۸۱ نظر تایید شده (با برچسب ۱) و ۹۹۱۸ نظر نیز تایید نشده (با برچسب ۰) هستند که پس از پاک کردن مغایرت‌ها، ۲۰۶۶۰۳ نظر تایید شده (با برچسب ۱) و ۷۳۹۱ نظر تایید نشده (با برچسب ۰)، باقی می‌ماند و در انتها پس از UnderSampling تعداد ۱۴۷۸۲ نظر تایید شده (با برچسب ۱) و ۷۳۹۱ نظر تایید نشده (با برچسب ۰)، باقی می‌ماند. در این پروژه، نظرات تایید نشده به عنوان نظر هرز و نظرات تایید شده به عنوان نظرات واقعی در نظر گرفته شده است. برای هر نظر ۱۰ ویژگی موجود است (متن نظر، برچسب نظر، آیدی محصول، عنوان محصول و ...) که از بین آن‌ها برای آموزش تنها به متن نظر و برچسب آن نیاز است.

۲_۴_۴ تنظیمات مدل

تنظیمات مدل اعم از برچسب‌ها، بیشینه طول^۱، batch_size، تعداد Epoch و ... طبق موارد زیر است:

تنظیمات مدل برای مجموعه داده fake-reviews-dataset:

^۱ MAX_LEN

```
[ ] # general config
MAX_LEN = 128
TRAIN_BATCH_SIZE = 16
VALID_BATCH_SIZE = 16
TEST_BATCH_SIZE = 16

EPOCHS = 10
EEVERY_EPOCH = 1000
LEARNING_RATE = 2e-5
CLIP = 0.0

MODEL_NAME_OR_PATH = 'HooshvareLab/bert-fa-base-uncased'
OUTPUT_PATH = '/content/bert-fa-base-uncased-fake-review-detection/pytorch_model.bin'

os.makedirs(os.path.dirname(OUTPUT_PATH), exist_ok=True)
```

```
[ ] # create a key finder based on label 2 id and id to label

label2id = {label: i for i, label in enumerate(labels)}
id2label = {v: k for k, v in label2id.items()}

print(f'label2id: {label2id}')
print(f'id2label: {id2label}')

label2id: {'CG': 0, 'OR': 1}
id2label: {0: 'CG', 1: 'OR'}
```

```
[ ] # setup the tokenizer and configuration

tokenizer = BertTokenizer.from_pretrained(MODEL_NAME_OR_PATH)
config = BertConfig.from_pretrained(
    MODEL_NAME_OR_PATH, **{
        'label2id': label2id,
        'id2label': id2label,
    })
```

تنظیمات مدل برای مجموعه داده Digikala:

```
[ ] 1 # general config
2 MAX_LEN = 512
3 TRAIN_BATCH_SIZE = 16
4 VALID_BATCH_SIZE = 16
5 TEST_BATCH_SIZE = 16
6
7 EPOCHS = 3
8 EVERY_EPOCH = 500
9 LEARNING_RATE = 2e-5
10 CLIP = 0.0
11
12 MODEL_NAME_OR_PATH = 'HooshvareLab/bert-fa-zwnj-base'
13 OUTPUT_PATH = '/content/bert-fa-zwnj-base-fake-review-detection/pytorch_model.bin'
14
15 os.makedirs(os.path.dirname(OUTPUT_PATH), exist_ok=True)
```

```
[ ] 1 # create a key finder based on label 2 id and id to label
2
3 label2id = {label: i for i, label in enumerate(labels)}
4 id2label = {v: k for k, v in label2id.items()}
5
6 print(f'label2id: {label2id}')
7 print(f'id2label: {id2label}')
```

```
label2id: {'rejected': 0, 'verified': 1}
id2label: {0: 'rejected', 1: 'verified'}
```

```
[ ] 1 # setup the tokenizer and configuration
2
3 tokenizer = BertTokenizer.from_pretrained(MODEL_NAME_OR_PATH)
4 config = BertConfig.from_pretrained(
5     MODEL_NAME_OR_PATH, **{
6         'label2id': label2id,
7         'id2label': id2label,
8     })
```

۳_۴_۴ نتایج

مدل پیشنهادی پس از ارزیابی نتایج زیر را به دست آورد:

:fake-rewies-dataset

F1: 0.95738929191844				
	precision	recall	f1-score	support
CG	0.96	0.95	0.96	1678
OR	0.95	0.96	0.96	1678
accuracy			0.96	3356
macro avg	0.96	0.96	0.96	3356
weighted avg	0.96	0.96	0.96	3356

Digikala:

F1: 0.9059858651362502				
	precision	recall	f1-score	support
rejected	0.84	0.89	0.86	665
verified	0.94	0.91	0.93	1331
accuracy			0.91	1996
macro avg	0.89	0.90	0.90	1996
weighted avg	0.91	0.91	0.91	1996

۴_۵ جمع بندی

در این پروژه مدلی بر اسلس Transformers و ParsBERT برای تشخیص نظرات هرز در زبان فارسی ارائه شد و با آموزش و ارزیابی آن بر روی مجموعه داده‌های fake-rewies-dataset و Digikala عملکرد قابل قبولی به دست آمده است. با مقایسه مدل پیشنهادی با کارهای مشابه مشخص شد که مدل پیشنهادی با رسیدن به F1 برابر با ۹۵/۷ درصد در مجموعه داده fake-rewies-dataset و F1 برابر با ۹۰/۵ درصد در مجموعه داده Digikala عملکرد بهتری نسبت به سایر مدل‌ها داشته است.

فصل ۵

استفاده از مدل‌های آموزش داده شده روی مجموعه داده

۵_۱ نحوه برچسب گذاری

برای برچسب گذاری تحلیل احساسات و تشخیص نظرات هرز روی هر مجموعه داده، به موارد زیر نیاز است:

- مجموعه داده آماده یا جمع‌آوری شده (که در این پروژه از مجموعه داده‌ای که با استفاده از خزنده tweetf0rm از تویتر جمع‌آوری کردیم استفاده شده است)
- مدل آموزش داده شده تشخیص نظرات هرز
- مدل آموزش داده شده تحلیل احساسات

مجموعه داده جمع‌آوری شده از تاریخ 07/09/2022 تا تاریخ 07/19/2022 در بستر تویتر برای سه اپراتور ایرانسل، همراه اول و رایتل:

	created_time	user_id	text	is_retweet	is_quote	is_reply	tags
0	2022-07-19 19:18:27	1355911468902981639	هیچ کس تنها نیست همراه اول	False	False	True	{'hamrah_aval'}
1	2022-07-19 19:13:32	614986877	بلی، ایراسل از همراه اول بدتره حتی	False	False	True	{'hamrah_aval', 'irancell'}
2	2022-07-19 19:09:10	941220112845164544	... یکنه سه ماهه همراه اول رو دو دوره یخی شش ماه	False	False	True	{'hamrah_aval'}
3	2022-07-19 19:05:41	822277959176306688	همراه اول هم همینه	False	False	True	{'hamrah_aval'}
4	2022-07-19 19:00:44	1111363641545236480	...بیست‌ی سه گیگ ماهانه‌ی اینترنت همراه اول می‌خر	False	False	False	{'hamrah_aval'}
...
2261	2022-07-10 07:35:43	1474119738490040324	...این چه سمی بود! من خیلی یچه بودم زنها تازه تاز	False	False	True	{'rightel'}
2262	2022-07-09 20:08:21	1355434208022421504	...کل روز اینترنت قطع بود حتی ملت ۹۱۱ هم نمی‌تولس	True	False	False	{'rightel'}
2263	2022-07-09 19:29:11	967671943750184960	آره باید تو خود رایتل هم شارژش کنی تا وصل شه	False	False	True	{'rightel'}
2264	2022-07-09 17:41:06	1280293767581499394	رایتل	False	False	True	{'rightel'}
2265	2022-07-09 17:30:08	989783064988483584	محمدرضا رایتل	False	False	True	{'rightel'}

2266 rows x 7 columns

پس از آماده‌سازی موارد ذکر شده، می‌توانیم با استفاده از بارگذاری مدل‌های آموزش داده شده طبق تنظیمات اولیه‌ای (بیشینه طول^۱، batch_size، مشخص کردن برچسب‌های هر مدل و ...) که برای آموزش دادن آن‌ها مشخص کرده بودیم و

¹ Max_len

استفاده از تابع Predict ای (این تابع، مدل، توکنایزر، بیشینه طول، batch_size و نظرات متنی را ورودی می‌گیرد و به ازای هر نظر، طبق برچسب‌های موجود برای مدل، آن نظر را برچسب گذاری می‌کند) که برای آن‌ها نوشته‌ایم، مجموعه داده موردنظر را برچسب گذاری کنیم.

به عنوان مثال در این پروژه مدل آموزش داده شده تحلیل احساسات دارای ۵ برچسب 2, 1, 0, -1, -2 است، بنابراین پس از استفاده تابع Predict بر روی نظرات موجود در مجموعه داده، یک ستون در مجموعه داده اضافه می‌شود که برای هر سطر (نظر) دارای یکی از مقادیر ۲- تا ۲ است. بنابراین پس از اجرای مدل آموزش داده شده تحلیل احساسات، هر نظر در مجموعه داده دارای برچسبی با مقدار ۲- تا ۲ است که نشان دهنده حس آن نظر است. ۲- نشان دهنده حالت Furious، ۱- نشان دهنده حالت Angry، ۰ نشان دهنده حالت Neutral، ۱ نشان دهنده حالت Happy و ۲ نشان دهنده حالت Delighted است. همچنین مدل آموزش داده شده تشخیص نظرات هرز نیز مانند همان مدل آموزش داده شده تحلیل احساسات است با این تفاوت که برچسب‌ها ۰ یا ۱ خواهد بود. ۰ نشان دهنده هرز بودن نظر و ۱ نشان دهنده واقعی و غیر هرز بودن نظر است.

در انتها پس از اجرای مدل‌های آموزش داده شده تحلیل احساسات و تشخیص نظرات یک ستون برای نشان دادن امتیاز حس^۱ و ستون دیگری برای نشان دادن واقعی بودن یا نبودن^۲ به آن اضافه می‌شود.

نمونه اجرای مدل‌های آموزش داده شده تحلیل احساسات و تشخیص نظرات روی مجموعه داده جمع آوری شده از

توییتر:

	created_time	user_id	text	is_retweet	is_quote	is_reply	tags	sentiment_scores	is_real
0	2022-07-19 19:18:27	1355911468902981639	هیچ کس تنها نیست ... همراه اول	False	False	True	{'hamrah_aval'}	1	1
1	2022-07-19 19:13:32	614986877	بابی، ایراسل ان همراه اول بدتره حتی	False	False	True	{'hamrah_aval', 'irancell'}	-1	1
2	2022-07-19 19:09:10	941220112845164544	... بسته سه ماهه همراه اول رو دو دوره یعنی شش ماه	False	False	True	{'hamrah_aval'}	0	1
3	2022-07-19 19:05:41	822277959176306688	همراه اول هم همینه	False	False	True	{'hamrah_aval'}	1	1
4	2022-07-19 19:00:44	1111363641545236480	...بسته سه گیگ ماهانه ایترت همراه اول می‌خر	False	False	False	{'hamrah_aval'}	-1	1
...
2261	2022-07-10 07:35:43	1474119738490040324	...اون چه سمی بودا من خیلی بچه بودم زنده تازه داز	False	False	True	{'rightel'}	-1	1
2262	2022-07-09 20:08:21	1355434208022421504	...کل روز اینترنت قطع بود حتی ملت ۹۱۱ هم نمی‌فونس	True	False	False	{'rightel'}	-1	1
2263	2022-07-09 19:29:11	967671943750184960	آره باید تو خود رایئل هم شارژش کنی تا وصل شه	False	False	True	{'rightel'}	0	1
2264	2022-07-09 17:41:06	1280293767581499394	رایئل	False	False	True	{'rightel'}	1	1
2265	2022-07-09 17:30:08	989783064988483584	محمدرضا رایئل	False	False	True	{'rightel'}	0	1

2266 rows x 9 columns

¹ Sentiment_score

² is_real

فصل ۶

داشبورد

۱_۶ رابط کاربری

برای طراحی داشبورد این پروژه از کتابخانه streamlit^۱ که برای بسیاری از پروژه‌های حوزه یادگیری ماشین بکار می‌رود استفاده شده است. این کتابخانه به ما این قابلیت را می‌دهد که رابط کاربری بسیار جذابی را برای پروژه خود طراحی کنیم. داشبوردهایی که به شدت تعاملی بوده و کاربر از کار با آن لذت می‌برد. معمولاً ارائه و گزارش نتایج آزمایش‌های علم داده / یادگیری ماشین می‌تواند کار سختی باشد. درحالی‌که درگذشته، یک ارائه پاورپوینت یا یک فایل LaTeX برای نمایش پروژه کافی بود ولی با محبوبیتی که امروزه وبسایت‌ها دارند، بهتر است که پروژه خود را در بستر یک سایت نمایش دهیم.

این همان جایی است که Streamlit وارد عمل می‌شود. Streamlit یک فریم‌ورک^۲ متن‌باز^۳ پایتون است که این امکان را فراهم می‌کند تا وبسایت‌های زیبا و تعاملی برای پروژه‌های یادگیری ماشین و علم داده را بدون نیاز به داشتن مهارت‌های طراحی وب، بسازید. با Streamlit به راحتی و تنها با اضافه کردن تعدادی تابع، یک وبسایت می‌توان ایجاد کرد و پروژه یادگیری ماشین خود را در قالب یک وب اپلیکیشن ارائه داد. از قابلیت‌های Streamlit پویایی آن است به صورتی که با تغییر داده‌ها نمودارها و گزارش‌ها به روزرسانی می‌شوند. در حال حاضر Streamlit قابلیت نمایش متون، داده‌ها و کد و همچنین بسیاری

^۱ <https://streamlit.io>

^۲ Framework

^۳ Open source

از ویجت^۱ (ابزارک) های تعاملی را فراهم می کند. همچنین به راحتی می توان از بسیاری از کتابخانه های رسم نمودار مثل Matplotlib^۲ و Plotly^۳ در وب اپلیکیشن ها استفاده کرد.

در ادامه درباره ی نصب و استفاده و توضیح بیشتر قابلیت های این کتابخانه محبوب در محدوده پروژه خود می پردازیم.

۲_۶ نحوه نصب و اجرای کد

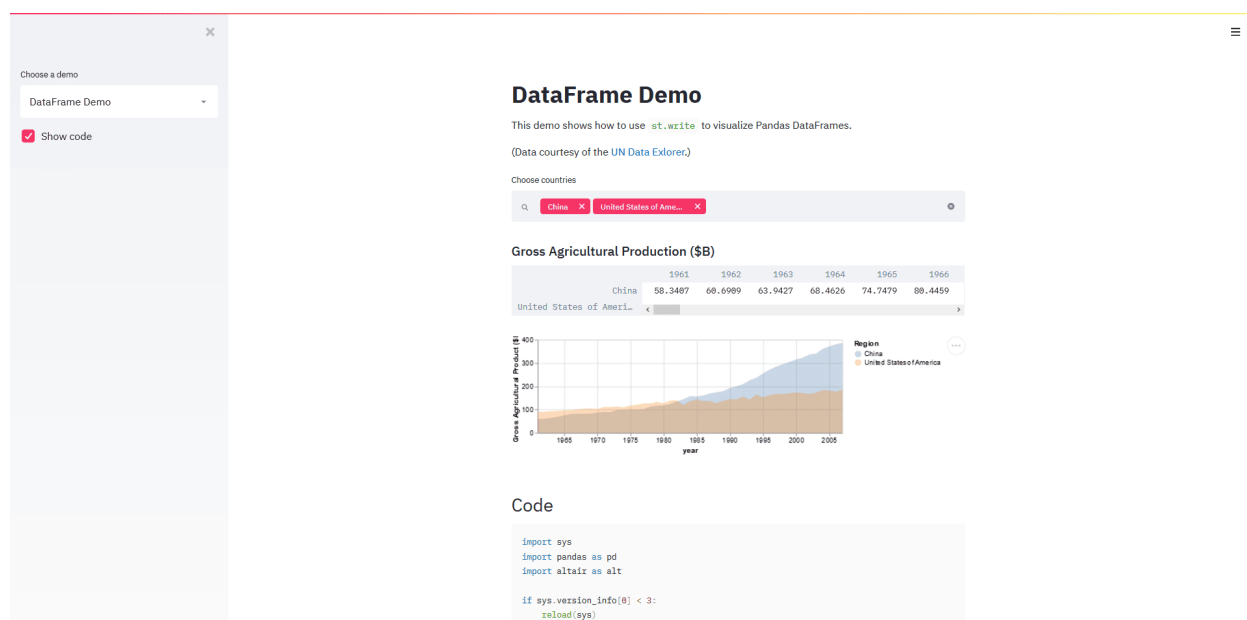
نخست در مورد نحوه نصب این ابزار را توضیح می دهیم. Streamlit را می توان با استفاده از PIP نصب کرد:

Pip install streamlit

پس از نصب Streamlit می توانید با اجرای برنامه hello world آزمایش کنید که همه چیز به درستی کار می کند:

streamlit hello

این کار یک وبسایت بر روی localhost:8501 برایتان باز می کند که در آن می توانید تعدادی دمو که با ابزار Streamlit ایجاد شده را ببینید.



¹ Widget

² <https://matplotlib.org>

³ <https://plotly.com/python>

می‌توانید فایل‌های مربوط به Streamlit را با نوشتن دستور زیر اجرا کنید:

```
streamlit run [your_app_name].py
```

این کار یک برگه جدید (tab) در مرورگر باز می‌کند و کد فعلی‌تان را به شما نمایش می‌دهد. این برگه در مرورگر به طور اتوماتیک پس از هر تغییر به‌روز خواهد شد.

۳_۶ نحوه نمایش متن

Streamlit برای اضافه‌کردن متن، چندین روش از جمله `st.write`، `st.markdown` و `st.title` دارد. برای مثال از دستور `write` می‌توان استفاده کرد و عبارت “Hello World” را روی صفحه نوشت.

```
st.write(“Hello World”)
```

تابع `write` در واقع به صورت خودکار تشخیص می‌دهد که چه نوع شی‌ای برای ارسال شده و آن را نمایش می‌دهد.

۴_۶ استفاده از ابزارک‌ها

با استفاده از ابزارک‌ها، streamlit این امکان را به شما می‌دهد تا داشبوردهای تعاملی زیبا بسازید. Streamlit دکمه‌ها، چک‌باکس، نوار لغزنده^۱، ورودی متنی و بسیاری امکان دیگر را مهیا کرده است. برای کسب اطلاعات بیشتر درباره تمام ابزارک‌های موجود، به API^۲ مراجعه کنید.

از `Selectbox` برای انتخاب یک گزینه از چند گزینه از یک لیست می‌توان استفاده کرد. این کار برای فیلتر داده بسیار مفید است. با استفاده از آن می‌توانیم روی ستون‌های `data-frame`، فیلتر اعمال کنیم.

همچنین می‌توانید با استفاده از `st.multiselect` به جای `st.selectbox`، چندین مورد را هم‌زمان انتخاب کنید.

^۱ Slider

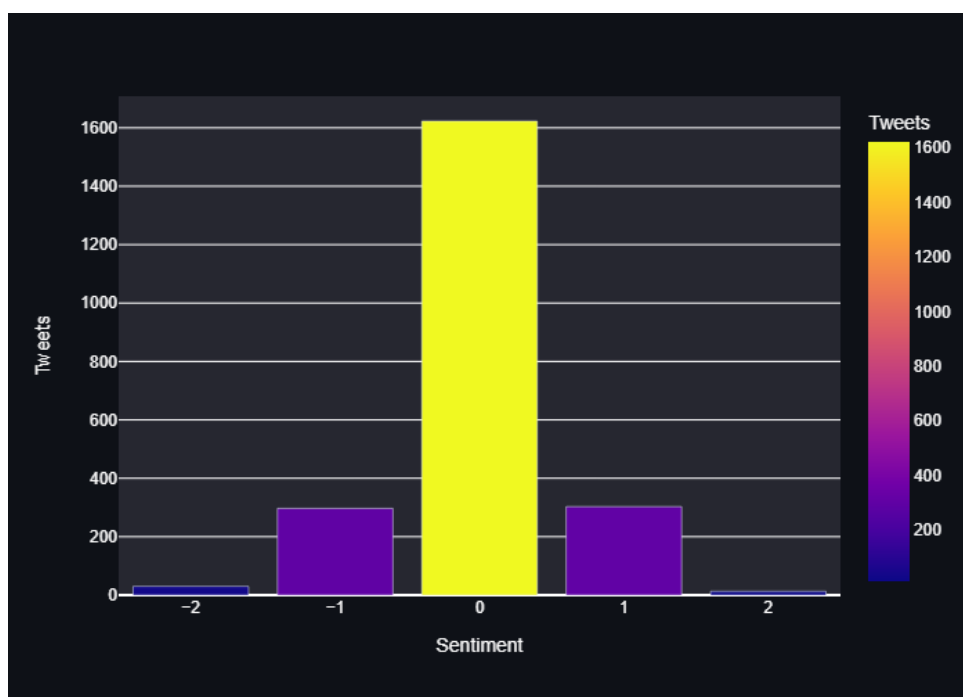
^۲ <https://docs.streamlit.io/library/api-reference/widgets>

۵_۶ بخش‌های برنامه

در این وب اپلیکیشن ما دو سربرگ داریم که یکی برای دید کلی^۱ آمار و ارقام که شامل نمودارهای میله‌ای و دایره‌ای و فیلتر کردن نمایش توییت‌ها می‌باشد و در برگه دیگر عملیات جستجو^۲ برای نمایش توییت‌ها بر اساس کلمه‌ای که کاربر وارد می‌کند انجام می‌شود.

۱_۵_۶ دید کلی

در این بخش در داخل نوار^۳ کنار صفحه کاربر می‌تواند تنظیمات دلخواه خود را انتخاب کند و نتایج و گزارشات در سمت راست نوار به کاربر نمایش داده می‌شود. این تنظیمات شامل انتخاب نوع نمودار است که کاربر می‌تواند یکی از دو نمودار میله‌ای^۴ یا دایره‌ای^۵ را انتخاب کند و بلافاصله نمودار به نمایش می‌آید. این نمودار نشانگر تعداد توییت‌ها برای هر یک از پنج دسته‌بندی داده‌ها می‌باشد.



¹ Overview tab

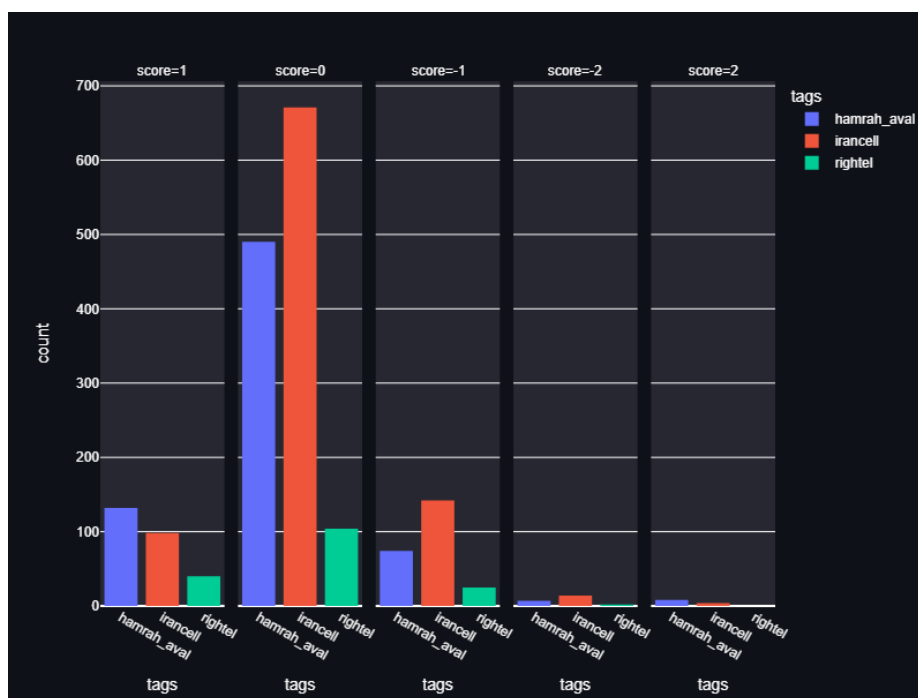
² Search tab

³ Sidebar

⁴ Histogram

⁵ Pie chart

آپشن بعدی انتخاب نوع اپراتور می‌باشد که کاربر می‌تواند به صورت همزمان چند اپراتور را با هم انتخاب کند تا نمودار میله‌ای مقایسه‌ای به او نمایش داده شود. در نمودار زیر مقایسه‌ای بین سه اپراتور معروف ایران را از نظر تعداد توییت‌ها در هر دسته‌بندی امتیازی می‌توان مشاهده و بررسی کرد.

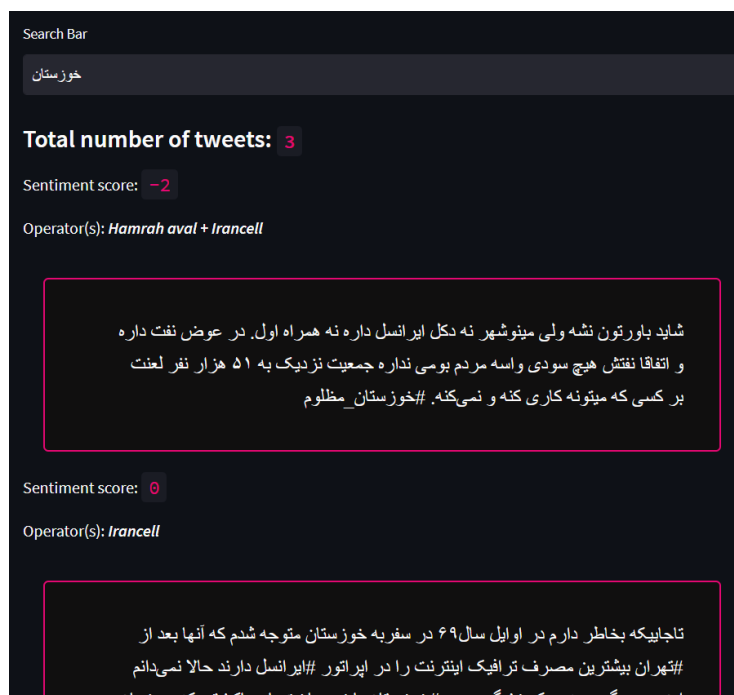


در سه آپشن بعدی به ترتیب می‌توانیم نام اپراتور موردنظر و نوع توییت‌ها از نظر هرز یا واقعی بودن یا هر دو به صورت کلی و در آخر امتیاز توییت از نظر تحلیل احساس را مشخص کنیم و بر این اساس توییت‌های فیلترشده را به کاربر نمایش دهیم.

نکته دیگری که باید به آن اشاره شود این است که به دلیل اینکه بعضی توییت‌ها شامل نام چند اپراتور می‌شدند، ما نیز توییت‌ها را با برچسب‌های جایگشت‌های نام این اپراتورها انتخاب کردیم برای مثال اگر در توییتی نام دو اپراتور همراه اول و ایرانسل همزمان قید شود، ویژگی tag آن توییت شامل نام هر دو اپراتور هست و آن توییت در دسته مخصوص خودش طبقه‌بندی می‌شود که برای این کار باید در قسمت انتخاب نام اپراتور هر دو اپراتور را انتخاب کنیم تا بتوانیم فقط توییتی‌هایی را ببینیم که نام هر دوی این اپراتورها در آن آمده باشد.

۶_۵_۲ جستجو

در سربرگ جستجو نیز کاربر با وارد کردن کلمه موردنظر می‌تواند توییتی‌هایی که شامل آن کلمه میشوند را به همراه امتیاز و دسته بندی اپراتور آن مشاهده کند. برای مثال، جستجوی کلمه خوزستان در تصویر زیر مشاهده می‌شود که تعداد ۳ توییت متناسب با این واژه پیدا شده است.



۶_۵_۳ کتابخانه‌های دیگر

برای رسم نمودارها از کتابخانه plotly استفاده شده است که یکی از بهترین کتابخانه‌ها برای رسم نمودارهای مختلف و باکیفیت است.

۶_۶ نتیجه‌گیری

کتابخانه streamlit یکی از بهترین کتابخانه‌ها برای نمایش گزارشات و نمودارها و توضیحات پروژه‌های مختلف علل‌الخصوص پروژه‌های یادگیری‌ماشین و علم‌داده می‌باشد که به کمک آن و دیگر کتابخانه‌ها نظیر plotly میتوان داشبوردهای فوق‌العاده‌ای را در کمترین زمان و با کیفیت مناسب تولید نمود. داشبورد به ما این امکان را

می‌دهد که داده‌ها را به صورت نمودارهای مختلف و یا متنی به آسانی مورد بررسی قرار بدهیم و با توجه به آن بتوانیم نتیجه‌گیری درباره داده‌ها و تحقیقات خود داشته باشیم.

فصل ۷

خلاصه، نتیجه‌گیری و کارهای آینده

۱_۷ خلاصه

یکی از مهم‌ترین بخش‌های هر سرویس اینترنتی و فروشگاه‌های بخش نظرات کاربران است. در این بخش کاربران به تبادل نظر با یکدیگر می‌پردازند و یکی از مطمئن‌ترین منابع برای شناختن بیشتر یک محصول و تصمیم به خرید و یا عدم خرید آن، همین بخش نظرات کاربران است. هر چند این بخش برای کاربران بسیار مفید است و می‌توانند از طریق آن اطلاعات بسیار خوبی در مورد یک محصول یا سرویس به دست آورند اما به دلیل وجود جنبه‌های مختلف در نظرات و ضد و نقیض بودن برخی از آن‌ها با یکدیگر یا با امتیازی (یا ممکن است امتیازی وجود نداشته باشد) که خود کاربر داده است و همچنین وجود افرادی که از این ظرفیت‌ها سوء استفاده می‌کنند و با تولید نظرات غیر واقعی اقدام به تبلیغ و یا تخریب یک محصول یا برند می‌کنند، نیاز داریم روش‌هایی ایجاد کنیم تا این جنبه‌های مختلف را تحلیل و بررسی کنند و آن‌ها را تبدیل به اطلاعات مفیدی برای تصمیم‌گیری کنند.

بنابراین باید مدلی داشته باشیم که این نظرات را تحلیل کند و احساسات فرد را از آن متن بفهمد و همچنین مدل دیگری که به وسیله آن نظرات هرز را از نظرات واقعی تشخیص دهیم. برای طراحی چنین مدل‌هایی ابتدا از روش‌های آماری و روش‌های کلاسیک یادگیری ماشین برای حل این مسئله استفاده می‌شد ولی در سال‌های اخیر با توجه به نتایج خوب استفاده از یادگیری عمیق در مسائل مختلف، استفاده از یادگیری عمیق برای تشخیص نظرات هرز نیز گسترش یافت. روش‌های مختلف نظارتی و غیر نظارتی و حتی نیمه نظارتی برای حل این مسئله مورد استفاده قرار گرفته‌اند که هر کدام از این روش‌ها نقاط ضعف و قوت خود را دارند.

مسئله تشخیص نظرات هرز به صورت کلی یکی از مسائل پیچیده است که حتی تشخیص این نظرات برای انسان نیز بسیار دشوار است. یعنی به صورت عادی یک انسان قادر به تشخیص نظرات هرز از نظرات واقعی نیست. تحقیقاتی وجود دارند که در مدل خود از انواع فراداده‌ها استفاده کرده‌اند و با ارتباط دادن ویژگی‌های نظر، ویژگی‌های یک کاربر، ویژگی‌های محصول و انواع داده‌ها به یکدیگر توانسته‌اند به دقت‌های خوبی در این حوزه برسند اما یکی از مشکلات اساسی این گونه مدل‌ها و به صورت کلی مسئله تشخیص نظرات هرز، تشخیص نظرات تکی است. یعنی افرادی که تنها یک نظر ثبت کرده‌اند، از آن جا که نمی‌توانند اطلاعات زیادی به مدل بدهند در تحقیقات مختلف از مجموعه داده‌ها حذف می‌شوند. همچنین مسئله تحلیل

احساسات نیز مشکل نظرات تکی را دارد و همچنین مجموعه داده‌های متفاوت در این زمینه، ویژگی‌های مختلفی دارند. بنابراین بهتر است مدل‌هایی که طراحی می‌شوند تنها به متن نظر وابسته باشند.

نظرات تکی در مدل‌هایی که تنها بر مبنای ویژگی‌های نظر و متن آن آموزش می‌بینند، در دسترس نیست. دلیل این مسئله هم این است که این مدل‌ها به ویژگی‌های کاربر و یا محصول وابسته نیستند. از طرف دیگر مشکلی که وجود دارد این است که هم مجموعه داده برچسب‌دار برای این مسئله خیلی کم است و هم مجموعه داده‌های موجود از نظر ویژگی‌های موجود در مجموعه داده با یکدیگر خیلی متفاوت هستند. یعنی اگر بخواهیم مدلی را بر اساس یک مجموعه داده و ویژگی‌های موجود در آن پیاده‌سازی کنیم ممکن است بر روی مجموعه داده دیگری آن ویژگی‌ها موجود نباشند و به همین دلیل نمی‌توانیم مدل را بر روی این مجموعه داده‌ها آموزش دهیم.

در این پروژه از Transformers که تا به این لحظه قدرت‌مندترین مدل برای پردازش زبان طبیعی است و مدل پیش آموزش دیده ParsBERT که همان BERT برای زبان فارسی است (مدل‌های مبتنی بر BERT در سال‌های اخیر موفقیت‌های زیادی در زمینه پردازش زبان طبیعی به دست آورده‌اند)، برای تحلیل احساسات و تشخیص نظرات هرز، استفاده شده است. همچنین برای آموزش این مدل‌ها تنها از متن نظرات و برچسب آن‌ها استفاده شده است که مشکل نظرات تکی را حل می‌کند و همچنین این قابلیت را به مدل می‌دهد که روی هر مجموعه داده‌ای قابل اجرا باشد و نتایج بسیار خوبی را به دست آورد.

در این پروژه از Transformers که تا به این لحظه قدرت‌مندترین مدل برای پردازش زبان طبیعی است و مدل پیش آموزش دیده ParsBERT که همان BERT برای زبان فارسی است (مدل‌های مبتنی بر BERT در سال‌های اخیر موفقیت‌های زیادی در زمینه پردازش زبان طبیعی به دست آورده‌اند)، برای تحلیل احساسات و تشخیص نظرات هرز، استفاده شده است. همچنین برای آموزش این مدل‌ها تنها از متن نظرات و برچسب آن‌ها استفاده شده است که مشکل نظرات تکی را حل می‌کند و همچنین این قابلیت را به مدل می‌دهد که روی هر مجموعه داده‌ای قابل اجرا باشد و نتایج بسیار خوبی را به دست آورد.

در این پروژه از مجموعه داده‌های مختلفی برای آموزش و ارزیابی مدل خود استفاده کرده‌ایم و دلیل این مسئله هم این است که بتوانیم کارایی مدل خود را با مدل‌های دیگر مقایسه کنیم. در این تحقیق از مجموعه داده SentiPers برای تحلیل احساسات و از مجموعه داده‌های fake-reviews-dataset و Digikala برای تشخیص نظرات هرز استفاده کرده‌ایم و همچنین از مدل‌های جاسازی کلمات پیش آموزش دیده که برای مدل‌های BERT تعریف شده‌اند، برای تولید ۳ جاسازی (جاسازی توکن، جاسازی‌های بخش و جاسازی موقعیت) استفاده شده است.

همچنین برای جمع‌آوری نظرات کاربران در بسترهای مختلفی مانند توییتر، به جهت استفاده از آن‌ها برای برچسب گذاری با مدل‌های آموزش دیده و استفاده از این نظرات و برچسب‌هایی که مدل‌های آموزش دیده ساخته‌اند برای ایجاد تصمیم‌گیری، نیاز به پیاده‌سازی خزنده‌ای است تا این نظرات را برای ما جمع‌آوری کند. در این راستا، در این پروژه از خزنده tweetform استفاده کرده‌ایم که به خوبی نظرات کاربران توییتر روی محصولات یا برندهای مورد نظر را که به عنوان Query

به آن می‌دهیم برای ما جمع‌آوری می‌کند و پس از آن نظرات جدیدی که اضافه می‌شوند را به صورت آنلاین و جریان داده برای ما جمع‌آوری می‌کند.

۲_۷ نتیجه‌گیری

در این پروژه سعی کرده‌ایم مدل‌هایی را برای تحلیل احساسات و تشخیص نظرات هرز با استفاده از یادگیری عمیق، Transformers و مدل ParsBERT که مبتنی بر مدل BERT است، ارائه کنیم که قابلیت آموزش بر روی مجموعه داده‌های مختلف را داشته باشد. ما این مدل‌ها را برای زبان فارسی آموزش داده و ارزیابی کرده‌ایم. همچنین با استفاده از مدل‌های دیگر BERT مانند XLM-RoBERTa به راحتی می‌توان این مدل‌ها را برای هر زبان دیگری نیز پیاده‌سازی کرد. در این مدل‌ها از روش‌های پیش پردازش مختلف و متوازن سازی‌های مختلف استفاده کرده‌ایم. این مدل‌ها با استفاده از روش نظارتی پیاده‌سازی شده‌اند و برای آموزش نیاز به داده‌های دارای برچسب دارند.

در این پروژه سعی کرده‌ایم از کمترین ویژگی‌ها برای آموزش مدل استفاده کنیم و به همین دلیل تنها به متن و برچسب نظر برای آموزش مدل نیاز داریم. با استفاده از روش‌های به کار رفته، به دقت مطلوبی بر روی مجموعه داده‌های مختلف رسیدیم همچنین در این پروژه سعی کرده‌ایم خزنده‌ای داشته باشیم که نسبت به محصولات یا برندهای خاصی که مورد نظر است بتواند نظرات کاربران درباره آن‌ها را برای ما جمع‌آوری کند و پس از آن نظرات جدیدی که اضافه می‌شوند را به صورت آنلاین و جریان داده برای ما جمع‌آوری کند.

نتایج به دست آمده در مدل‌های پیشنهادی نسبت به مدل‌های دیگر، نشان دهنده کارایی مدل است. از معیارهای مختلف ارزیابی F1، Precision و Recall برای ارزیابی مدل خود استفاده کرده‌ایم. در نتیجه با استفاده از مدل‌های پیشنهادی برای زبان فارسی و بر روی مجموعه داده SentiPers برای تحلیل احساسات به F1 برابر با ۹۴/۹ درصد، برای مجموعه داده fake-reviews-dataset برای تشخیص نظرات هرز به F1 برابر با ۹۵/۷ درصد و برای مجموعه داده Digikala برای تشخیص نظرات هرز به F1 برابر با ۹۰/۵ درصد رسیدیم که با توجه به اینکه تنها از متن نظر برای آموزش مدل‌های خود استفاده کرده‌ایم نتایج بسیار خوبی به دست آمده است.

۳_۷ کارهای آینده

هر چند در حوزه‌های تحلیل احساسات و تشخیص نظرات هرز کارهای زیادی انجام شده است و در تحقیقات مختلف دقت‌های خوبی نیز گزارش شده است اما آن چه در واقعیت وجود دارد این است که این مسائل پیچیده هستند و به خصوص مسئله تشخیص نظرات هرز یکی از مسائل پیچیده‌ای است که برای انسان نیز به راحتی قابل تشخیص نیست و حتی یک فرد خبره هم نمی‌تواند به راحتی نظر هرز را از نظر واقعی تشخیص دهد. به همین دلیل مشکلات و چالش‌های زیادی در این حوزه وجود دارد و تا رسیدن به مدلی که به خوبی بتواند این نظرات را در محیط‌های مختلف و سیستم‌های مختلف تشخیص دهد راه زیادی باقی مانده است. در این بخش قصد داریم تعدادی از کارهایی که در آینده می‌توان برای بهبود مدل‌ها انجام داد را بررسی کنیم.

• ترکیب مدل با روش‌های غیر نظارتی

از آن جا که تعداد زیادی داده بدون برچسب در این حوزه‌ها وجود دارد که می‌تواند در رسیدن به یک مدل مطلوب به ما کمک کند، می‌توان مدل پیشنهادی را با روش‌های غیرنظارتی ترکیب کرد تا بتوانیم از این داده‌ها نیز برای آموزش مدل استفاده کنیم.

• استفاده از روش‌ها و مجموعه داده‌های دیگر برای تشخیص نظرات هرز

از آن جایی که مسئله تشخیص نظرات هرز، همان‌طور که گفته شد از مسائل بسیار پیچیده و سخت می‌باشد، باید روش‌های دیگری مانند شباهت‌های برداری نظرات، یادگیری تقویتی و ... را برای این مسئله آزمایش کرد تا نتایج این روش‌ها را نیز با روش‌های موجود مقایسه کنیم و به بهترین مدل دست پیدا کنیم. همچنین مجموعه داده‌های بسیار کمی در این زمینه وجود دارد و آن‌هایی هم که وجود دارند کیفیت مناسب و کافی را ندارند به خصوص در زبان فارسی که حتی مجموعه داده Digikala که تنها مجموعه داده در این زمینه است اخیراً حذف شده است و همچنین نامتوازن بودن این مجموعه داده‌ها به علت پیچیده بودن مسئله تشخیص نظر هرز برای به وجود آوردن داده با برچسب هرز بودن، باعث ایجاد کمبود داده‌های آموزشی می‌شود و همچنین نمی‌توان از روش‌هایی مانند ترجمه معکوس^۱، به دلیل پیچیدگی نظرات با برچسب هرز و تشخیص سخت آن حتی برای انسان، برای افزایش این داده‌ها استفاده کرد. بنابراین در نظر است که در آینده مجموعه داده‌های دیگری را نیز برای مسئله تشخیص نظرات هرز به کار ببریم یا با استفاده از روش‌های مختلف مجموعه داده‌های مناسبی را ایجاد کنیم تا بتوانیم از آن‌ها استفاده کنیم.

¹ Back Translate

- استفاده از فرایادگیری

استفاده از روش‌های فرایادگیری^۱ می‌تواند در تمام مسائل یادگیری ماشین و به خصوص یادگیری عمیق به ما کمک کند. منظور از فرایادگیری این است که در حال حاضر تنظیم فرایارامترهای مدل توسط خود ما و به صورت آزمون و خطا و بر اساس قوانین اندکی که می‌شناسیم انجام می‌گیرد اما در صورتی که بتوانیم یادگیری یک مدل را تحت یک مدل دیگر انجام دهیم و مقدار بهینه برای این فرایارامترها توسط مدلی یادگرفته شود می‌توان به دقت بهتری دست پیدا کرد.

- استفاده در بسترهای دیگر

در نظر داریم کارهای انجام شده را در بسترهای دیگری غیر از تویتر نیز به کار ببریم تا نتایج حاصل را با نتایج به دست آمده روی بستر تویتر مقایسه کنیم و همچنین به جایی برسیم که در هر بستری بتوان از مدل‌های پیاده‌سازی شده استفاده کرد.

¹ Metalearning

1. Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need." Advances in neural information processing systems 30 (2017).
2. Farahani, M., Gharachorloo, M., Farahani, M. et al. ParsBERT: Transformer-based Model for Persian Language Understanding. Neural Process Lett 53, 3831-3847 (2021).
<https://arxiv.org/abs/2005.12515>
3. Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).
<https://arxiv.org/abs/1810.04805>
4. Hosseini, Pedram, Ali Ahmadian Ramaki, Hassan Maleki, Mansoureh Anvari, and Seyed Abolghasem Mirroshandel. "SentiPers: a sentiment analysis corpus for Persian." arXiv preprint arXiv:1801.07737 (2018).
5. Erick Kauffmann, Jesús Peral, David Gil, Antonio Ferrández, Ricardo Sellers, Higinio Mora, A framework for big data analytics in commercial social networks: A case study on sentiment analysis and fake review detection for marketing decision-making, Industrial Marketing Management, Volume 90, 2020
6. Peng, Qingxi & Zhong, Ming. (2014). Detecting Spam Review through Sentiment Analysis. Journal of Software. 9. 10.4304/jsw.9.8.2065-2072.
7. Joni Salminen, Chandrashekhar Kandpal, Ahmed Mohamed Kamel, Soon-gyo Jung, Bernard J. Jansen, Creating and detecting fake reviews of online products, Journal of Retailing and Consumer Services, Volume 64, 2022
8. Conneau, Alexis, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. "Unsupervised cross-lingual representation learning at scale." arXiv preprint arXiv:1911.02116 (2019). <https://arxiv.org/abs/1911.02116>

9. M. E. Basiri and A. Kabiri, "Sentence-level sentiment analysis in Persian," 2017 3rd International Conference on Pattern Recognition and Image Analysis (IPRIA), 2017
10. M. E. Basiri and A. Kabiri, "Translation is not enough: Comparing Lexicon-based methods for sentiment analysis in Persian," 2017 International Symposium on Computer Science and Software Engineering Conference (CSSE), 2017
11. Dehkharghani, R., & Emami, H. (2020). A novel approach to sentiment analysis in Persian using discourse and external semantic information, arXiv, 2020.
12. Rohanian, Morteza, Mostafa Salehi, Ali Darzi, and Vahid Ranjbar. "Convolutional Neural Networks for Sentiment Analysis in Persian Social Media." *arXiv preprint arXiv:2002.06233* (2020).
13. Dashtipour, Kia & Gogate, Mandar & Li, Jingpeng & Jiang, Fengling & Kong, Bin & Hussain, Amir. A Hybrid Persian Sentiment Analysis Framework: Integrating Dependency Grammar Based Rules and Deep Neural Networks. *Neurocomputing*. 380. 2019
14. Dashtipour, Kia & Gogate, Mandar & Cambria, Erik & Hussain, Amir. (2021). A Novel Context-Aware Multimodal Framework for Persian Sentiment Analysis.
15. Dashtipour, Kia, Mandar Gogate, Ahsan Adeel, Hadi Larijani, and Amir Hussain. "Sentiment Analysis of Persian Movie Reviews Using Deep Learning" *Entropy* 23, no. 5: 596. 2021
16. Brown, Tom, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan et al. "Language models are few-shot learners." *Advances in neural information processing systems* 33 (2020): 1877-1901.
17. Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. "Sequence to sequence learning with neural networks." *Advances in neural information processing systems* 27 (2014).
18. Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. "Efficient estimation of word representations in vector space." *arXiv preprint arXiv:1301.3781* (2013).
19. Zeng Z-Y, Lin J-J, Chen M-S, Chen M-H, Lan Y-Q, Liu J-L. A Review Structure Based Ensemble Model for Deceptive Review Spam. *Information*. 2019; 10(7):243.
<https://doi.org/10.3390/info10070243>
20. Alessandro Bondielli, Francesco Marcelloni, A survey on fake news and rumour detection techniques, *Information Sciences*, Volume 497, 2019, <https://doi.org/10.1016/j.ins.2019.05.035>.

21. N. Jindal and B. Liu, "Analyzing and Detecting Review Spam," Seventh IEEE International Conference on Data Mining (ICDM 2007), 2007, doi: 10.1109/ICDM.2007.68.
22. Nitin Jindal and Bing Liu, Review spam detection, In Proceedings of the 16th international conference on World Wide Web (WWW '07), Association for Computing Machinery, New York, NY, USA, 1189- 1190, 2007 <https://doi.org/10.1145/1242572.1242759>
23. Nitin Jindal and Bing Liu, Opinion spam and analysis, In Proceedings of the 2008 International Conference on Web Search and Data Mining (WSDM '08), Association for Computing Machinery, New York, NY, USA, 219-230, 2008, <https://doi.org/10.1145/1341531.1341560>
24. Wael Etaiwi, Ghazi Naymat, The Impact of applying Different Preprocessing Steps on Review Spam Detection, Procedia Computer Science, Volume 113, 2017, <https://doi.org/10.1016/j.procs.2017.08.368>
25. Safarian, N., Basiri, M., Khosravi, H. Feature ranking for Persian Review Spam detection. Journal of Soft Computing and Information Technology, 2019; 8(2): 1– 16.
26. Li, Luyang et al. "Learning Document Representation for Deceptive Opinion Spam Detection." CCL (2015).
27. M. E. Basiri, N. Safarian and H. K. Farsani, "A Supervised Framework for Review Spam Detection in the Persian Language," 2019 5th International Conference on Web Research (ICWR), 2019, doi: 10.1109/ICWR.2019.8765275.



**University of Tehran
College of Farabi
Faculty of Engineering
Department of Computer Engineering**

Sentiment Analysis and Fake Review Detection on Twitter

By:

**M. Dehghani
M. Rashidkhan**

Under Supervision of:
Dr. Kazim Fouladi-Ghaleh

**A Project Report as a Requirement for
the Degree of Bachelor of Science in
Computer Engineering**

September 2022

