



دانشکده مهندسی کامپیوتر

مبانی پردازش زبان طبیعی

---

تحلیل احساسات بازار stock و crypto بر اساس توییت های

توییتر

گزارش پروژه

فاز ۱ - جمع‌آوری داده

---

سید عمامد موسوی

## فهرست مطالب

## ۱ منبع دقیق داده

ابتدا قصد داشتم که توییت های مورد نیاز برای این تسک را از توییتر کراول کنم و یک crawler هم نوشتیم اما به علت فیلتر بودن توییتر و مشکلاتی از این قبیل نتواستم این کار را انجام دهم. به همین خاطر از دیتابست های Kaggle استفاده کرده ام. برای اینکه داده ها bias نداشته باشند از چندین دیتابست مختلف استفاده کرده ام. دیتابست های استفاده شده به شرح زیر است:

### stock-tweet-sentiment-dataset ۱.۱

این دیتابست شامل تقریبا ۲۸۰۰۰ توییت در مورد سهام های مختلف در بازار Stock آمریکا است و همچنین بعضی از رمز ارز ها را هم شامل می شود. ستون های این دیتابست عبارتند از:

timestamp •

text •

source •

symbols •

company-names •

sentiment •

هر توییت در این دیتابست با ۳ برچسب به صورت زیر مشخص شده است:

1 : (Positive) •

0 : (Neutral) •

-1 : (Negative) •

## stock-market-tweets ۲.۱

این دیتاست از `hugging face` می باشد و شامل تقریبا ۱.۷ میلیون توییت در مورد سهام های AMZN (Amazon), GOOG (Google), MSFT (Microsoft), TSLA(Tesla), AAPL (Apple) می باشد.

ستون های این دیتاست عبارتند از:

`tweet_id` •

`writer` •

`post_date` •

`body` •

`comment_num` •

`retweet_num` •

`like_num` •

`ticker_symbol` •

لازم به ذکر است که `sentiment` این دیتاست برچسب گذاری نشده است و در ادامه لازم است تا با ابزار هایی این دیتاست را برچسب گذاری کنیم.

## stocks-with-sentiment-and-emotion-analyst ۳.۱

این دیتاست شامل تقریبا ۱.۲ میلیون توییت در مورد سهام های AMZN (Amazon), GOOG (Google), MSFT (Microsoft), TSLA(Tesla), AAPL (Apple) می باشد.

ستون های این دیتاست عبارتند از:

`tweet_id` •

`writer` •



post\_date •

body •

body\_cleaned •

comment\_num •

retweet\_num •

like\_num •

probabilities •

برچسب گذاری این دیتاست در ستون probabilities انجام شده است و دارای ۳ مقدار زیر می باشد:

positive •

neutral •

negative •

## cryptocurrency-tweets-with-sentiment-analyst ۴.۱

این دیتاست شامل تقریبا ۸۲۵۰۰۰ توییت در مورد سهام های مختلف می باشد که برچسب گذاری هم شده اند. این دیتاست شامل ۴۳ ستون می باشد که برخی از مهم ترین آنها عبارتند از:

date •

username •

retweet\_counts •

tweet •

neg •

neu •

pos •

compound •

که مقدار نهایی برچسب در ستون compound می باشد که مقادیر آن به شرح زیر می باشد:

positive •

neutral •

negative •

## ۲ روش جمع‌آوری و ابزار مورد استفاده

همانطور که پیش تر ذکر شد به دلیل فیلتر بودن توییت امکان کراول کردن وجود نداشت . برای همین از چندین دیتاست مختلف موجود در Hugging Face و kaggle استفاده کردم که تنها بر روی یک دیتاست خاص bias نداشته باشم. تعداد توییت های موجود در این دیتاست ها به بیش از ۵ میلیون توییت می رسید. برای راحتی کار به صورت زیر از هر دیتاست به صورت رندوم نمونه گیری کردم:

• دیتاست اول: ۲۸۰۰۰ توییت

• دیتاست دوم: ۵۰۰۰۰ توییت

• دیتاست سوم: ۵۰۰۰ توییت

• دیتاست چهارم: ۵۰۰۰ توییت

که تعداد نهایی توییت ها جمعا به ۶۲۸۰۰۰ توییت رسید. از تمام دیتاست ها فقط ۳ ستون زیر را استخراج کردم:

post\_date •

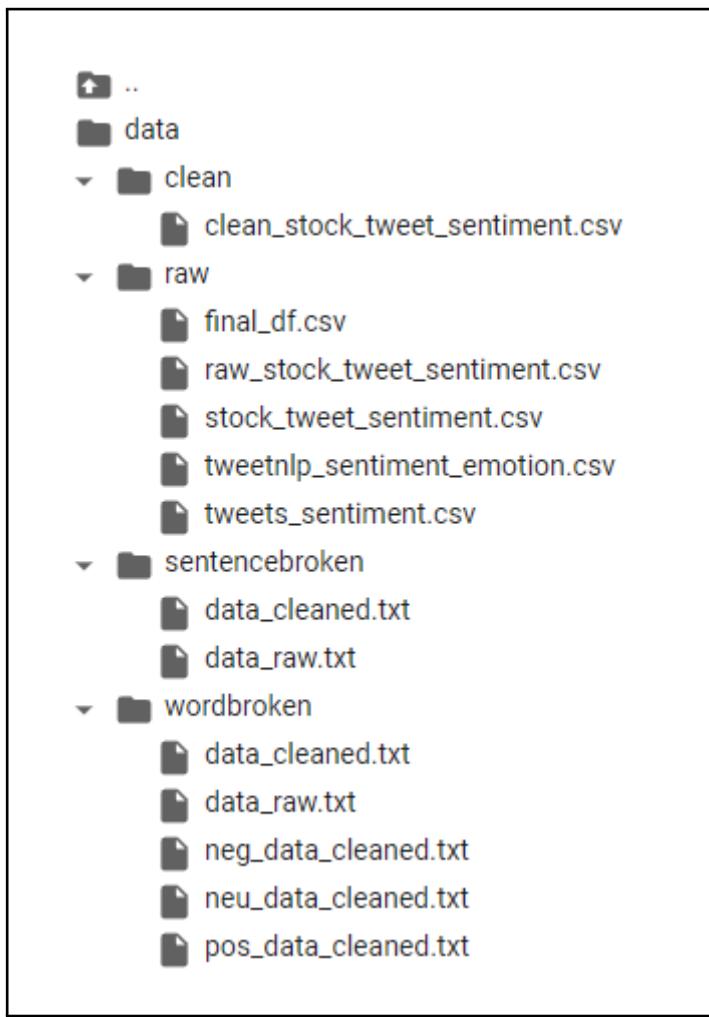
tweet •

sentiment •

بخش خوبی از دیتاست که ۵۰۰۰۰ توییت می شد هم برچسب نداشت که با استفاده از ابزار برچسب زنی آنها را مقدار دهی کردم که در ادامه توضیح خواهم داد  
در آخر هم برای اینکه توزیع داده ها به صورت استاندارد باشد از هر برچسب ۴۰۰۰ توییت به صورت رندوم انتخاب کردم که در نهایت ۱۲۰۰۰ توییت داشته باشیم.

### ۳ فرمت داده‌ها

در کل، داده‌ها در چهار پوشه‌ی clean، raw، sentencebroken و wordbroken قرار گرفته‌اند.



- در پوشه clean داده‌های تجمعی شده تمیز شده اند و روی آنها پیش پردازش صورت گرفته است و در ستون tweet\_cleaned ذخیره شده اند.

- در پوشه raw در صورتی که کد مربوط به load کردن داده‌ها را اجرا کنید فایل‌های CSV مربوط به هر دیتاست در آن قرار می‌گیرند. به صورت پیش فرض این پوشه حاوی یک فایل text است که در آن لینک دیتاست‌ها موجود است به همراه یک فایل اکسل raw\_stock\_tweet\_sentiment.csv

های ذکر شده است که شامل ۱۲۰۰۰۰ توییت می باشد

- در پوشه sentencebroken داده های تمیز و خام بر اساس جمله tokenize شده اند به صورتی که در هر خط این فایل ها یک جمله قرار گرفته است.
- در پوشه wordbroken داده های تمیز و خام بر اساس کلمه tokenize شده اند به صورتی که هر کلمه با یک space از همدیگر جدا شده است همچنین توییت های موجود بر اساس برچسب نیز بر حسب کلمه tokenize شده اند

## ۴ پیش‌پردازش‌های انجام‌شده

### ۱.۴ روش/ابزار تفکیک جملات

برای تفکیک جملات از ابزار nltk که یک ابزار معروف در حوزه NLP می باشد استفاده کردم. این ابزار یک تابع sent\_tokenize دارد که یک عبارت می گیرد و در خروجی یک لیست از جملاتی که در آن عبارت وجود دارد بر می گرداند.

### ۲.۴ روش/ابزار تفکیک توکن‌ها/کلمات

برای جدا کردن کلمات از BERT Tokenizer BERT یک مدل زبانی transformer-based می باشد که به صورت گسترده برای تولید embedding کلمات به کار می رود. من در این پژوهش از این tokenizer استفاده کردم و ابتدا هر جمله را encode کردم و به input\_ids\_to\_tokens با تابع convert\_ids\_to\_tokens آنها را تبدیل به توکن/کلمه کردم و ذخیره کردم لازم به ذکر است که چون تعداد توییت ها بالا است برای جلوگیری از زمان بر بودن این فرایند از استفاده کردم که پیاده سازی آن با زبان Rust BertTokenizerFast بوده و بسیار سریع تر عمل می کند.

## ۳.۴ روش/معیارهای تمیزکردن داده

### ۱.۳.۴ حذف کاراکتر های غیر لاتین

در بین توییت های موجود در دیتاست برخی به زبان فارسی یا چینی و زبان های مختلفی بودند که مجبور شدیم آنها را حذف کنیم. دلایل این کار عبارتند از:

- حجم عظیمی از دیتاست به زبان انگلیسی است و BERT tokenizer هم با زبان انگلیسی آموزش دیده است در صورتی که کلماتی غیر از کلماتی که در vocabulary خود وجود دارد ببینید به آنها توکن unknown نسبت می دهد که این کار باعث می شود تا حجم داده غیر مفید بالا برود و بازدهی مدل در هنگام یادکری پایین بیاید
- وجود کلمات چینی به دلیل اینکه بیشتر بصری هستند و از این لحاظ به هم نزدیکی دارند نیاز دارد تا یک لایه CNN برای در ک بهتر به tokenizer اضافه شود. ولی خب من ترجیح دادم تا تمام کاراکتر ها به یک زبان باشند تا مدل به راحتی یاد بگیرد و رابطه بین کلمات را بهتر متوجه بشود تا در نهایت یک embedding بهتر خروجی بدهد

### ۲.۳.۴ حذف کلمات پالایشی (stopwords)

منظور از کلمات پالایشی کلماتی مثل the ، in ، is ، and ، it ... است. این کلمات در بردارنده بار معنایی ارزشمندی برای تشخیص لحن نیستند و به همین دلیل با حذف آنها از متن ورودی می توانیم حجم نویز ورودی را کم کنیم و به مدل اجازه دهیم بتواند روی کلمات کلیدی تر متن ورودی تمرکز کند.

برای حذف این کلمات از پکیج nltk استفاده شده است.

### ۲.۳.۴ حذف کردن (mentions)

در توییت ها زیاد اتفاق می افتد تا در آخر توییت چندین نفر را به صورت "@username" منشن کنند. این بخش از توییت هیچ ارزش خبری ندارد و داده اضافی و غیر مفید است و باید حذف شود

### ۴.۳.۴ حذف (urls)

در توییت ها ممکن است لینک دسترسی به یک خبر یا چیز های مختلفی برای ارجاع به چیزی داده

شود که محتوی خود این url برای ما ارزشی ندارد و مدل نباید آن را در نظر بگیرد زیرا ممکن است گمراه شود و به اشتباه کاراکتر های موجود در url را مهم تلقی کند. این اطلاعات هم غیر مفید است و باید حذف شود

#### ۵.۳.۴ حذف اعداد و علائم نشانه گذاری (punctations)

در توییت ها برای خوانایی بیشتر از علائم نشانه گذاری استفاده می شود که عبارتند از:

(colon) : •

(semicolon) ; •

(apostrophe) ' •

(comma) , •

(square brackets) [] •

(parentheses) () •

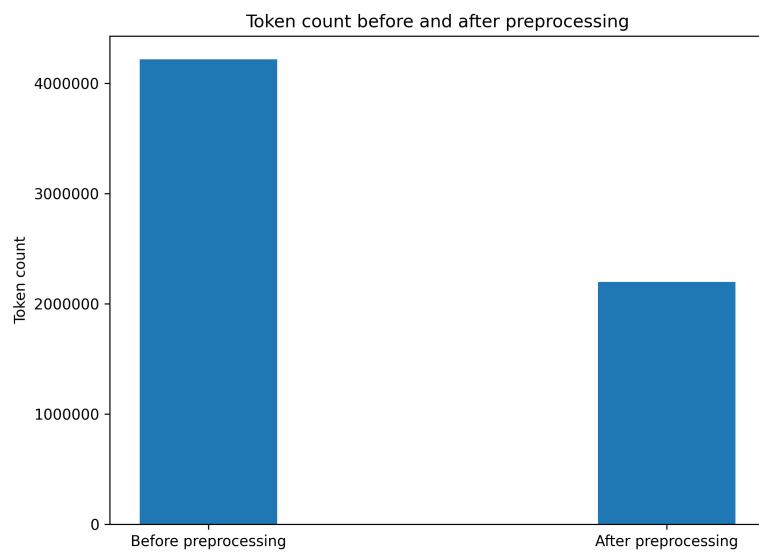
به نظر نمی رسد تاثیری در لحن یک جمله داشته باشد و بهرت است حذف شود همچنین ارقام نیر تاثیری در لحن یک توییت ندارند. مثلاً این طور نیست که عدد ۹ لحن مثبت داشته باشد و عدد ۰ لحن منفی داشته باشد و برای جلوگیری از این بدفهمی بهتر است حذف شوند و به آنها توجی نشود.

#### ۶.۲.۴ تبدیل کلمات به lowercase

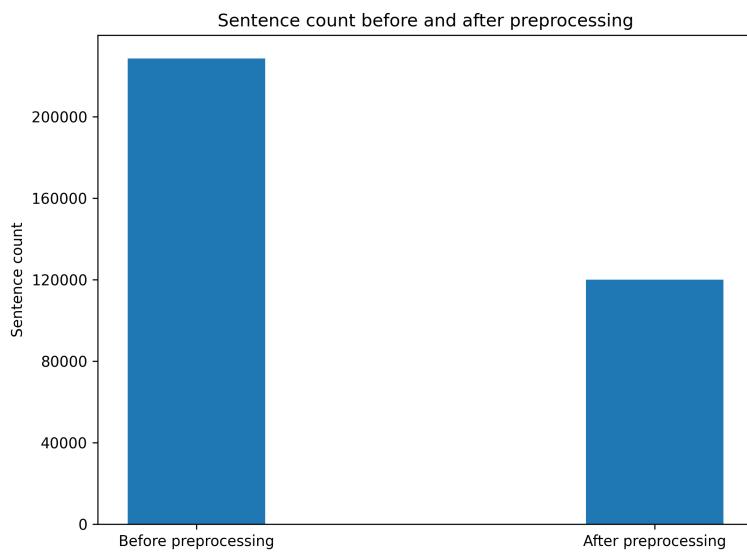
در آخر کار تمام کلمات را به lowercase تبدیل کردیم تا مدل راحت تر بتواند یاد بگیرد و به طور مثال فرقی بین A و a نگذارد زیرا در عمل هم فرقی ندارند.

#### ۴.۴ اندازه داده قبل/بعد تمیز کردن داده

برای کلمات:



برای جملات:



## ۵ واحد و روش برچسب‌گذاری

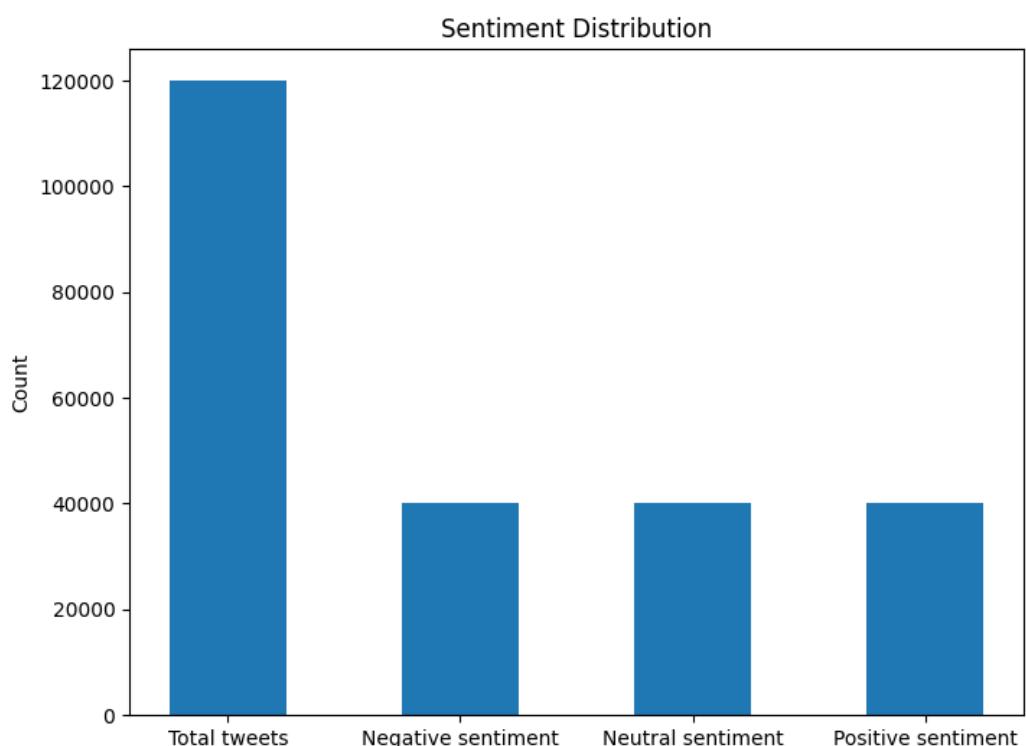
واحد داده، یک توییت است و برچسب‌گذاری به ازای هر توییت انجام شده است. هر توییت بر اساس لحنی که دارد می‌تواند یکی از برچسب‌های مثبت، منفی و خنثی را به خود نسبت دهد. ۳ تا از دیتاست‌های مورد استفاده برچسب‌گذاری شده بودند. بخش عظیمی از توییت‌ها که تقریباً ۵۰۰۰۰۰ توییت می‌شد برچسب نداشت.

برای برچسب زدن از nltk.sentiment.SentimentIntensityAnalyzer استفاده کردم که در واقع یک ابزار پیش‌آموزش‌دیده برای تحلیل احساس متن است که با استفاده از روش لغتنامه‌ای، امتیاز احساسی هر کلمه را براساس سه حالت مثبت، منفی یا بی‌طرف مشخص می‌کند. این ابزار شامل یک لغتنامه بزرگ است که بیش از ۷۰۰۰ کلمه و عبارت مختلف را در بر می‌گیرد و می‌تواند برای تحلیل احساس متون مختلف از جمله خبرها و شبکه‌های اجتماعی و اخبار مالی به کار رود با استفاده از این ابزار یک عبارت به عنوان ورودی می‌دهم و به عنوان خروجی میزان منفی بودن، مثبت بودن و خنثی بودن را به صورت ۳ عدد بین ۰ و ۱ می‌گیرم. یک نتیجه نهایی هم به عنوان compound می‌گیریم. اگر این عدد بزرگتر از ۰.۵ بود توییت را مثبت، اگر عدد کمتر از -۰.۵ بود منفی و اگر بین این دو عدد بود خنثی برچسب می‌زنیم نمونه خروجی روی یک توییت:

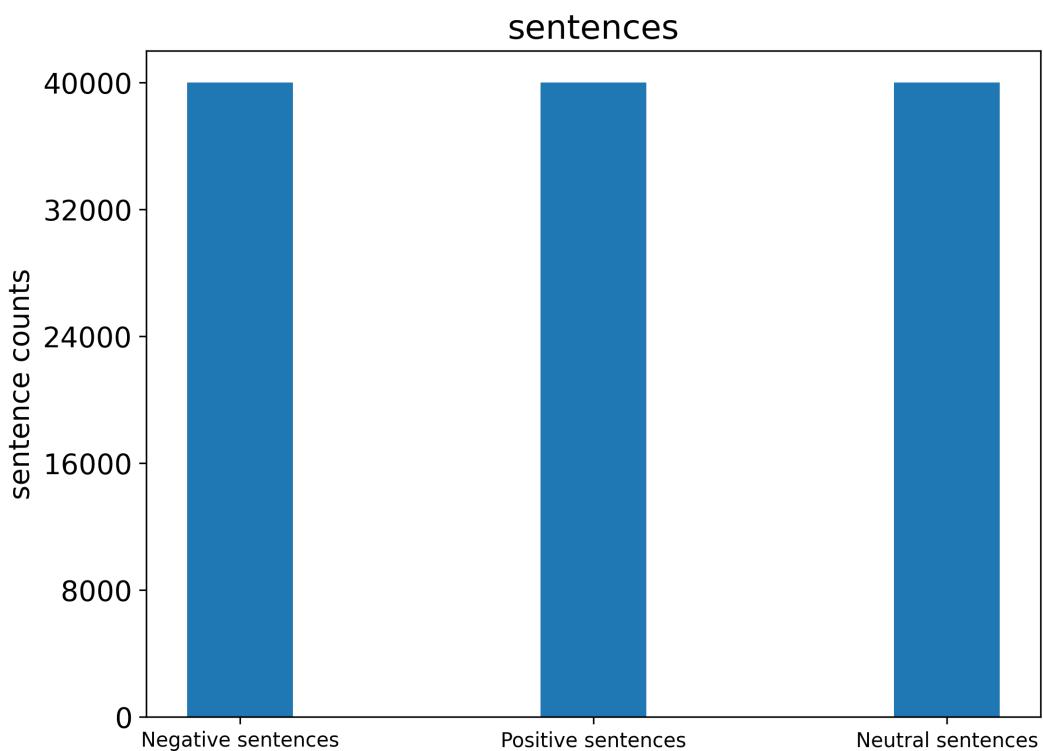
{neg:0.301, neu:0.547, pos:0.152, compound: - 0.4404}

## ۶ آمار داده به تفکیک برچسب

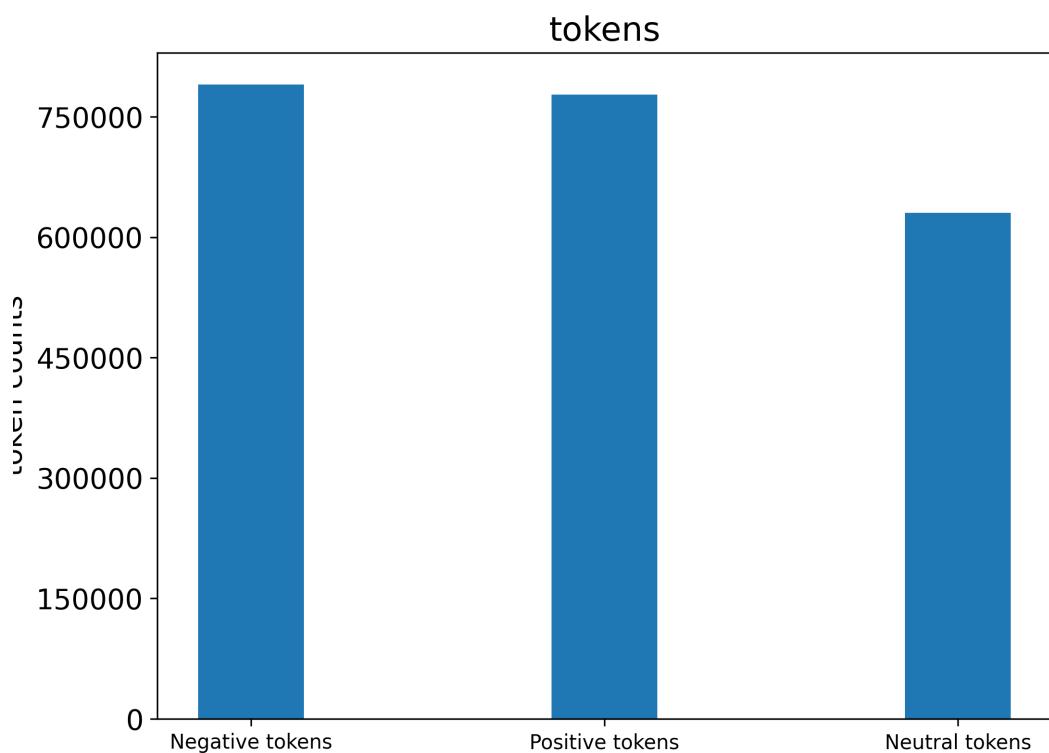
### ۱.۶ تعداد توييت ها



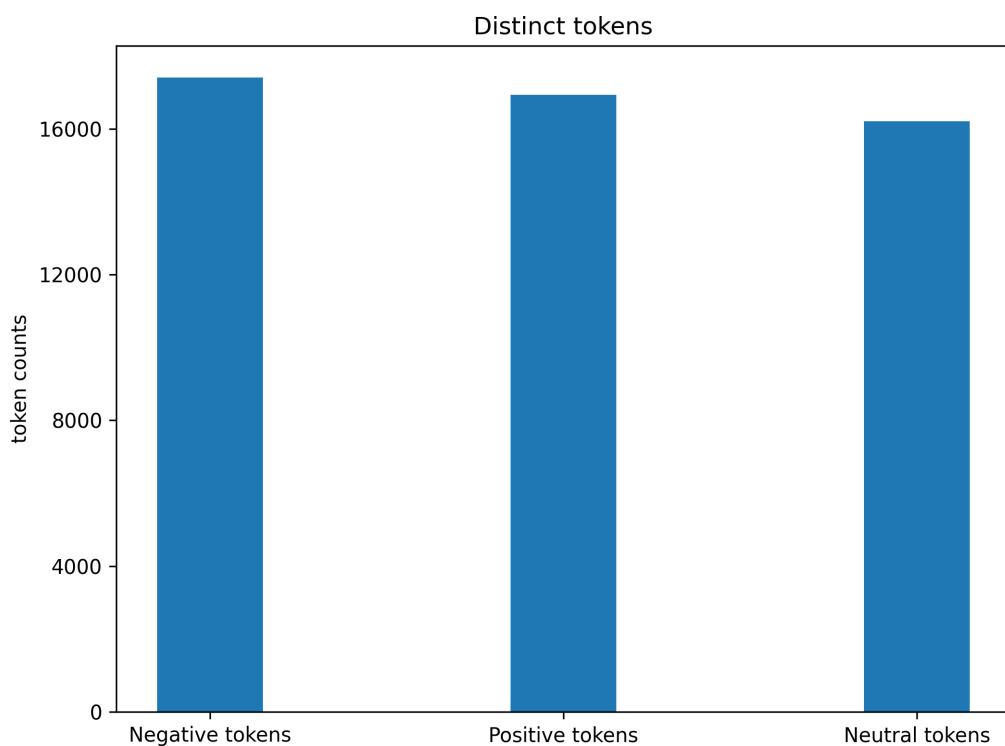
۲.۶ تعداد جملات



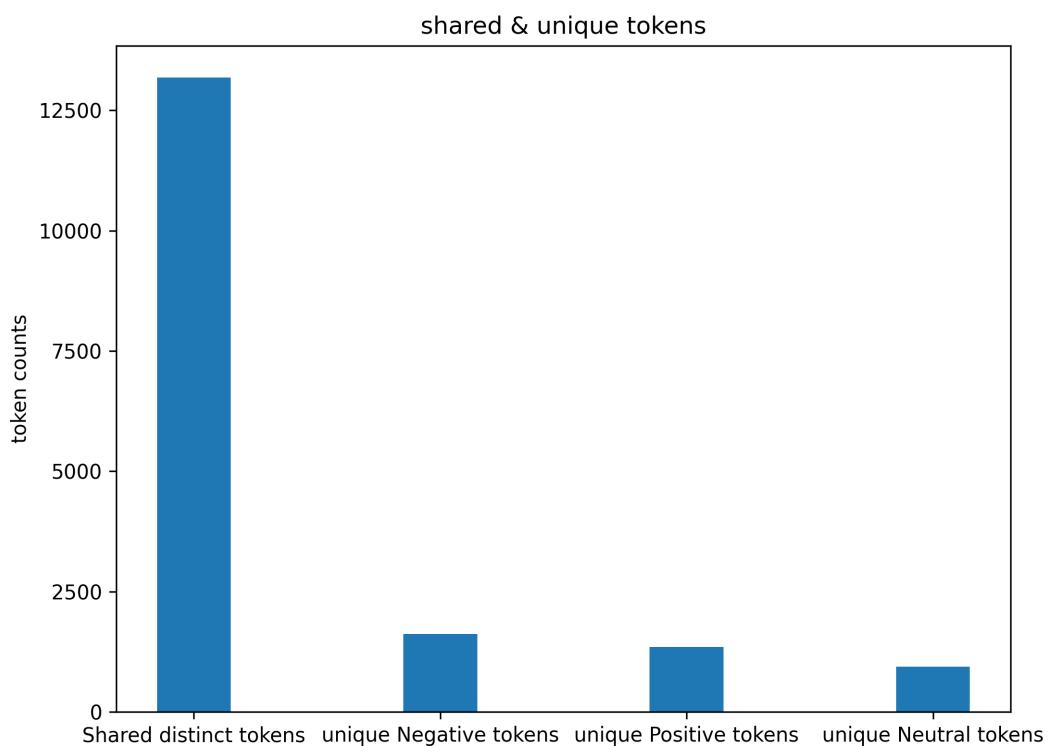
### ۳.۶ تعداد کلمات



#### ۴.۶ تعداد کلمات منحصر بفرد

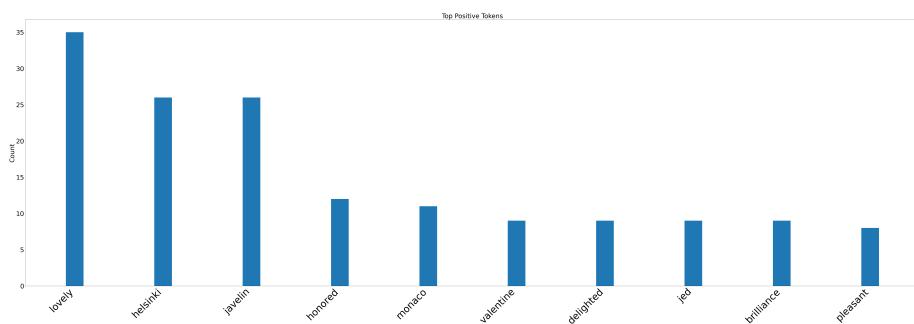


## ۵.۶ تعداد کلمات منحصرفرد مشترک و غیر مشترک بین برچسب‌ها

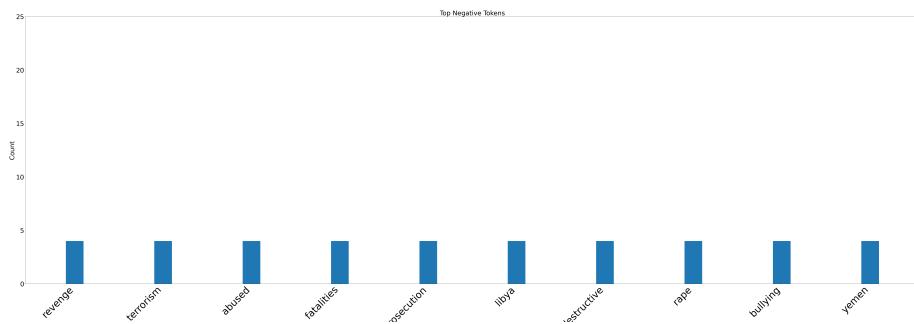


## ۶.۶ ۱۰ کلمه پر تکرار غیر مشترک هر برچسب

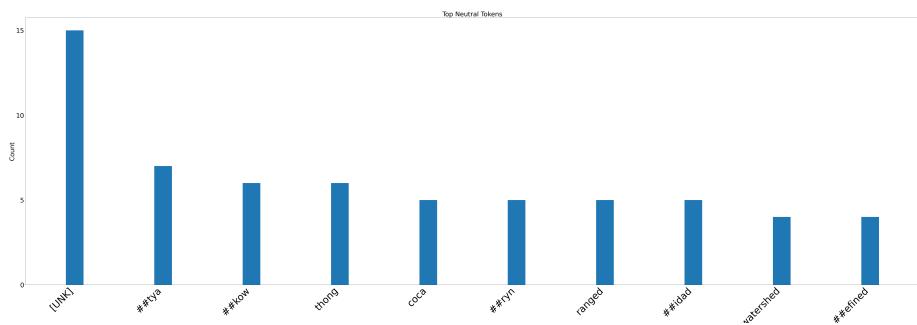
۱۶۶ مثبت



۲۶۶ منفی

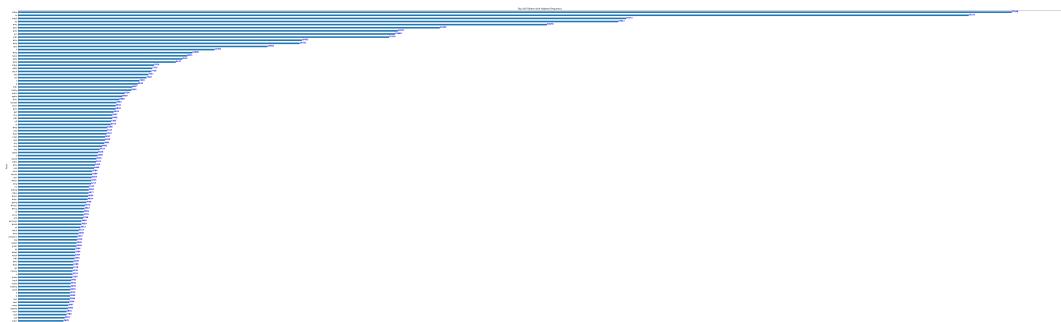


۳۶۶ خنثی

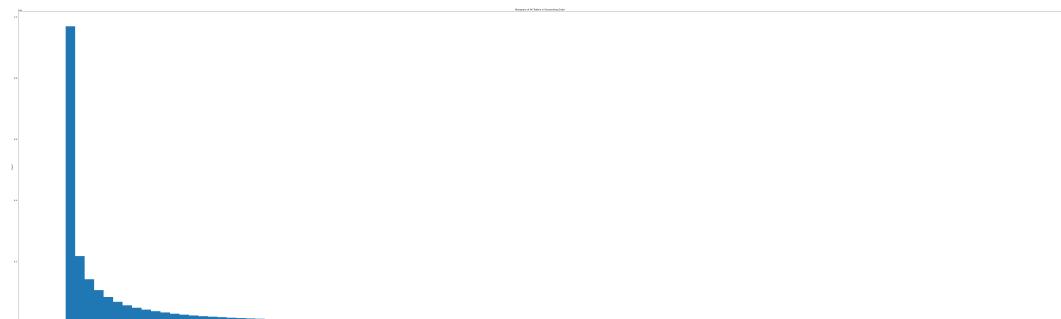


## ۷.۶ هیستوگرام تعداد تکرار هر کلمه منحصر بفرد

۱.۷۶ کلمه برتر ۱۰۰



۲.۷۶ تمام کلمات



## ۷ آدرس گیت هاب و Hugging Face

• لینک ریپازیتوری گیت هاب

• لینک دیتابست Hugging Face