Reddit Sentiment Analysis

Project Report

Ethan Boyar

Bhavpreet Singh

Sameer Bhatti

Ayman Haque

Danielle Ongsohu

CSCI-426/626 W01 – Information Retrieval

Professor Houwei Cao

May 1st, 2024

Table of Contents

Abstract

Analyzing and understanding the public opinion of any given person, topic, or idea is key to making it popular, and more importantly, keeping it popular. Sentiment analysis is perfect for this task, allowing us to determine the sentiment polarity associated with a person, topic, or idea, and therefore draw conclusions on public opinion regarding it.

Understanding public opinion is crucial for the popularity and sustainability of any person, topic, or idea. Sentiment analysis emerges as a powerful tool for this purpose, enabling the assessment of sentiment polarity associated with a specific entity. By analyzing sentiment, we are able to draw accurate conclusions in regards to the public and their opinion. Sentiment Analysis involves the application of natural language processing (NLP) techniques which allows is to analyze the sentiment polarity associated with a specific entity. By analyzing sentiment, the provided text data can be found and classified as positive, negative, or neutral, this provides insights into public perception.

Description

Background

The computational study of people's feelings and attitudes as they are expressed in text is known as sentiment analysis. It entails classifying textual material as positive, negative, or neutral using Natural Language Processing (NLP) approaches. So in essence we are quantifying feelings, and this is so important because with quantitative data, the possibilities of interpreting such data is seemingly endless. Moreover, comprehending the attitude present in online chats is becoming more and more important for political campaigns and businesses alike. Just pause for a moment to consider how effective it may be to quickly get the consensus of a large number of people. Social media conversation analysis, such as that done on Reddit, can be used to identify patterns in public opinion, evaluate brand perception, and gain an understanding of people's attitudes about different subjects. Reddit provides a rich supply of text data for sentiment analysis because of its numerous communities (subreddits) devoted to a wide range of topics. The platform's conversational structure allows researchers to explore how sentiment evolves and spreads within specific communities.

The process of sentiment analysis begins with data collection, in which text data from various sources such as social media, new articles, and customer reviews is gathered. Preprocessing techniques are then applied to clean and structure the data removing noise and irrelevant information. Afterwards, sentiment analysis is employed to analyze the acquired text data and determine the sentiment polarity associated with the entity of interest. These algorithms may include machine learning models trained on labeled data, lexicon-based approaches, or deep learning techniques. The results of sentiment analysis can be visualized through

graphical representation which includes sentiment histograms, word clouds, or sentiment timelines. These visualizations provide a clear and concise overview of public sentiment trends over time.

Overall, in our project, sentiment analysis data acts as a valuable tool for analyzing public opinion. By combining NLP techniques and sentiment analysis algorithms, we are able to gain insights on organizations and individuals into how we are able to support them in making informed decisions and strategies to maintain or enhance popularity through their brand, product, or message as they are perceived by the public.

Motivation

This project analyzes the sentiment shared in Reddit comments to determine the public's perceptions on particular subjects, figures, or concepts. We will be able to detect changes in general sentiment over time and identify trends in public opinion thanks to the new insights. Decisions made by companies, policymakers, and social researchers can be improved by having a better knowledge of public mood. The analysis's conclusions may have an impact on campaign message, marketing tactics, or even changes to policy recommendations. Our goal is to test several sentiment analysis tools (such as TextBlob and VADER) and assess how well they work. This comparative method will demonstrate the benefits and possible drawbacks of several social media sentiment analysis tools.

Application

Businesses and even individuals can assess how the public views their brand, goods, or services by using Reddit sentiment research. Reddit does not appeal to

just companies or famous people. But instead serves as an environment to anyone to have an open discourse about anything. Leveraging sentiment analysis in this sense can enable anyone to pinpoint areas in need of development and gauge consumer response in addition to modifying marketing strategies in response to input. Political parties and candidates can keep an eye on Reddit sentiment to see what the masses think of their public persona and policies for example. Reddit's signature subreddit system, allows for a multitude of conversations to be had about the same person, so aggregating the overall sentiment of these many subreddits proves to be a fairly effective method in generalizing opinions. With this data, platforms may be changed, communications can be improved, and possible conflicts can be delicately handled. Reddit sentiment analysis is a useful tool for all ranging from researchers to journalists, and even at a more official capacity with government organizations to follow the evolution of public opinion on current affairs as well and social/cultural movements. With such data available, users can facilitate educated discussion and aid in comprehending the perspectives of other populations.

Dataset(s)

Pullpush.io

pullpush.io is a free online indexing service, which indexes and catalogs user-submitted content to Reddit. It exists primarily to support high quality search-and-retrieval functions (it's actually what most of Reddit search is built on!), and academic research, like our project.

All of our data was retrieved exclusively using the pullpush API, and was retrieved within the default lifespan of *verbosely indexed* posts (11 cycles, so 5 ½ years old at

most). Posts older than that are reduced to just the post title and some metadata(simple indexed) so we cannot really use them.

Pulling data

Pullpush provides two general-purpose *endpoints* that accept and respond to generic HTTP requests. Our preprocessing script dynamically builds requests depending on criteria that we set (such as net-limits, specific subreddits, specific sorting criteria), and then feeds the output into a processing pipeline.

A single post (and associated comments) are stored in memory in a standard Map/K:V pair, where a single post (key) can have many comments (values). The pair is run through various preprocessing stages, and then exported into a JSON file with the same K:V format.

The memory is cleared, and then the process is repeated until every post is handled.

Pullpush imposes a hard limitation of 100 comments per-post, and 100 posts per-request. 10,000 comments is sufficient data to make conclusions (at least if the topic is focused).

To keep datasets reasonably sized, and keep pre-processing times under an hour, each dataset we built originated from one request, and consisted of 10,000 raw comments (and 100 associated posts). The final dataset that we performed most tests on originated entirely from the subreddit **r/politics**.

Preprocessing

[removed] and [deleted]

Reddit contains a ton of [deleted] or [removed] comments, which were useless for our analysis. They were removed right away from the processing pipeline.

To lowercase

Capital letters generally create extra noise for sentiment analysis(at least in English, not in every language). They may appear as a more extreme polarity, or incorrectly indicate the start of a new sentence. To avoid potential inaccuracies, we lowercase the entire dataset.

Useless characters

We needed to remove the irrelevant characters and simplify the comment. To do this, we removed characters that aren't alphanumeric or whitespace. We checked if the comment is alphanumeric (*ch.isalnum*()) or a whitespace character (*ch.isspace*()). If both conditions are satisfied, the comment is retained. If they aren't, the comment is excluded from the resulting string. We then join the filtered characters into a string, further simplifying the comment.

User mentions

Since Reddit is a conversational platform, we had a lot of data with users mentioning other users, whether that was replying to them or simply mentioning them. We needed to remove this, as it was redundant and we don't need user mentions to be filtered into our dataset. By doing this, it allows us to keep the context of the comment itself to analyze, and not the individuals mentioned.

Emojis

Emojis are a huge part of our current digital world. We use them to convey certain emotions, whether it's a sad face or a laughing face. As useful as they are, we didn't need to include it in our filtered data, so we cleaned the comments further to remove them. We implemented a method to normalize the emoji; replacing the emoji with a tag like "sad" or "happy".

Tokenization

The first expensive preprocessing stage - tokenization is absolutely required to correctly prepare the data for the final two stages, which (by design) will only work on vectors consisting of discrete elements. We used a pretty standard tokenizer, and structured the raw comment into a token-filled vector.

Stopwords and short words

Stopwords and short words have no statistical significance on net polarity (i.e. the, but, is, to). They simply create extra noise and increase real-processing time, which

for medium datasets like ours, can already get quite long. Perhaps the single largest efficiency optimization throughout the preprocessing stages, removing these tokens can save up to 30% processing time on that vector.

Lemmatization

Sentiment analysis lexicons are built from existing *dictionaries*, and assume no inflection in a word when assigning sentimental polarity to it. Naturally, this means, every word being compared to the lexicon needs to be in a standardized dictionary-form, aka Lemma. Lemmatization is more-or-less required for sentiment analysis to actually work, and is by far the most expensive preprocessing stage, and thus occurs last, once the dataset has already been reduced to its near-final form.

Export

The resulting, fully preprocessed vector is exported as the value to the key (post) inside a JSON file.

Processing

Processing pipeline

The data is read from a JSON file named ‘data.json’. Various processing steps are performed on the read-out K:V pairs.

Sentiment Analysis

For each post in the loaded data, sentiment analysis is performed on its related comments using the 'analyze_sentiment' function. This function employs TextBlob and Vader to analyze the sentiment Polarity of each comment and categorize them as 'Very Positive', 'Positive', 'Neutral', 'Negative', or 'Very Negative'.

TextBlob

TextBlob is a Python library used for processing textual data. It helps with common NLP tasks like part-of speech tagging, noun phrase extraction, sentiment analysis, and more. The sentiment analysis feature relies on a pattern-based approach and is a pre-trained model. The 'analyze_sentiment' function utilizes TextBlob to analyze the sentiment polarity of each comment. The TextBlob object is created for the comment, and then the sentiment attribute is accessed to get the polarity score, which ranges from -1 (very negative) to 1 (very positive).

VADER

Valence Aware Dictionary and Sentiment Reasoner is a rule-based sentiment analysis tool specifically designed for social media text. It is lexicon and rule-based, meaning it uses a predefined list of words with associated sentiment scores and rules to analyze sentiment in text. The 'analyze_sentiment' function in the VADER implementation initializes a SentimentIntensityAnalyzer() object. This analyzer computes the sentiment polarity scores for the given comment using rules and heuristics. The compound score, which represents the overall sentiment intensity, is extracted from the result. Based on this compound score, the function categorizes the sentiment into one of five categories.

Topics Labeling

Each post's title is passed through a pre-trained zero-shot classification model. This model, part of the Hugging Face Transformers library, is designed to predict relevant topics for text inputs without being explicitly trained on topic-label data. It leverages a large language model, likely based on transformer architecture, fine-tuned for multi-class classification tasks, the model outputs a list of candidate topics along with their confidence scores. The 'label_topics' function then selects the most probable topic from the list. For example, a post's title is "Finally! A Republican Shows Some Spine, Says She's Voting for Biden", the function may output "Joe Biden" as the label.

Compilation and Storage

The sentiment analysis results, along with the associated topic label and comment sentiment count for each post, are compiled and stored in a CSV file. This process involves gathering and organizing the data for each post, calculating the sentiment distribution, determining the topic labels, and updating the counters. Finally, the compiled results are written to the CSV file.

Visualization

The results from the respective CSV file are read and prepared for plotting, The relevant columns containing sentiment counts for post are selected. Then, a stacked bar plot is generated using Matplotlib. Each bar represents a post, and the height of each segment within the bar corresponds to the count of sentiments.

This visualization allows for a clear comparison of sentiment distributions across different posts, aiding in the analysis of sentiment trends in the dataset.

Results

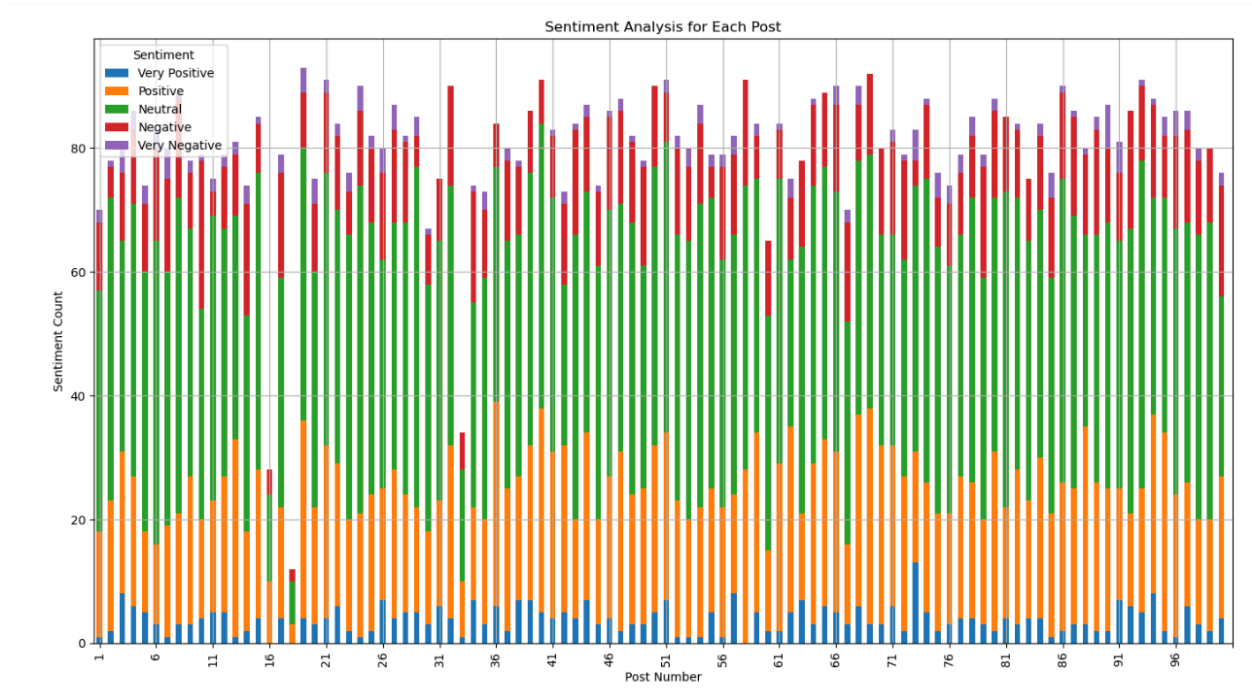
Frequency

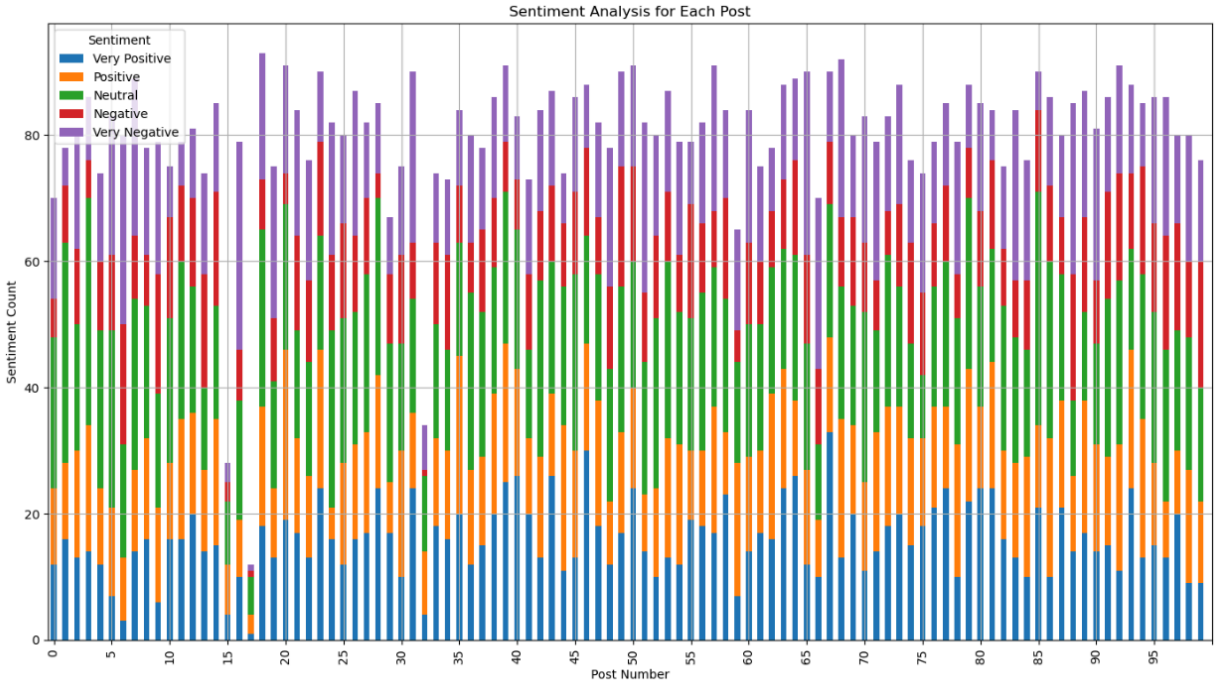
- **Donald Trump** and **Joe Biden** were the topics with the most posts by a substantial margin, with Trump ranking at #1 overall.
- **Election News**, **FBI** and **Ukraine** were 3rd, 4th and 5th, respectively.
- Approximately 20% of the dataset were one-off topics (fairly common on Reddit due to megathreads).

Polarization

- **Donald Trump** and **Joe Biden** were the *most polarizing* topics among the posts. They had the highest distribution at the extremes (very positive or very negative), with overall sentiment being roughly split between the two.
- **Taxes** was the *least polarizing* topic, with almost the entire distribution being negative-neutral. Meaning most people agree and feel the same way about it.

Visualizations





Conclusion

In conclusion, our project on Reddit sentiment analysis has provided valuable insights into public opinion in regards to the various topics, persons and ideas collected through our program. Through the application of sentiment analysis techniques, we were able to quantify and analyze the emotional tone expressed in comments on Reddit. By leveraging the natural language processing tools such as TextBlob and Vader, we are able to categorize sentiments into positive, negative, or neutral, which allows us to draw more concise and accurate conclusions about public perceptions.