Online Community Monitoring
Tracking Changes in Subreddit Sentiments

Ashish Telukunta

Individual Final Report

Introduction

Our project harnesses the power of real-time Reddit data scraping to capture posts and comments either based on user input or from a preloaded dataset. This data is then processed using advanced transformer models like BERT, RoBERTa, or Electra, specifically fine-tuned to identify and predict sentiments across six emotions: Sadness, Joy, Love, Anger, Fear, and Surprise. By aggregating these sentiment scores, we generate a comprehensive sentiment trend for each emotion over a specified timeframe. This analysis is presented in an informative plot that visualizes the trend of these emotional sentiments over time, offering valuable insights into subreddit community trends.

In this project, my contributions were important in four key areas:

1. I developed a real-time scraper for Reddit posts and comments, incorporating multi-threading optimizations and rate-limiting management (allowing me to scrape more data) for enhanced efficiency.
2. Preprocessed training data using NLP concepts, Implemented and fine-tuned RoBERTa and ELECTRA models, adapting them for our specific use case.
3. Wrote an optimized code to expedite inference times on GPU (predict_emotions method), transitioning from sequential to batch processing in the EmotionClassifier Class.
4. Lastly, I was responsible for computing averages of sentiment scores and visualizing these insights using Matplotlib, resulting in clear and informative data representations. Also, made the plot for the histogram to show (the number of samples vs the Interval Number) to understand the distribution of Reddit data.

Individual Work Description

Overall, I was responsible for writing code to scrape Reddit data when required. To implement and fine-tune two pre-trained transformer models (RoBERTa and ELECTRA), write code to aggregate six sentiment scores for each interval and visualize the trend of average sentiment scores over each interval using matplotlib.

Detailed description of my work

Reddit Scraper Methods (RedditScraper Class in analysis_page.py)

1. **fetch_posts:**
    - Fetches top posts from a specified subreddit based on the time filter (daily, weekly, monthly) and the number of posts.
    - Sort the posts based on their creation time.
    - Divide posts into intervals (daily, weekly, monthly) and assign an interval number to each post.
    - Visualizes the distribution of posts across intervals using a bar chart.
    - Returns a data frame of posts and comment information with their interval numbers.

2. **fetch_comments:**

    Fetches top comments from a given submission up to a specified limit and returns details of each comment, including its text and interval number.

3. **create:**
    - Calls fetch_posts to get post data and a list of posts.
    - Uses a ThreadPoolExecutor for concurrent execution (multithreading) to fetch comments for each post. This is where multithreading is implemented, allowing simultaneous retrieval of comments from multiple posts, thus optimizing the scraping process.
    - Aggregates post and comment data.
    - Converts the aggregated data into a pandas DataFrame and saves it as a CSV file.

ELECTRA and RoBERTa - Implementation and Fine-Tuning on Twitter and Reddit Dataset (Electra_model.py, Roberta_model.py, predict_emotions function in EmotionClassifier class)

Electra Hyperparameters and Optimizer for the Best Model

{ batch_size: 128
initial_learning_rate: 5e-5
optimizer: Adam Optimizer
epochs for model training: 15 }

Roberta Hyperparameters and Optimizer for the Best Model

{ batch_size: 128
initial_learning_rate: 2e-5
optimizer: Adam Optimizer
epochs for model training: 15 }

Code Structure and Execution Flow

The files (Electra_model.py and Roberta_model.py) provide a comprehensive pipeline for fine-tuning and sentiment analysis: from preparing and processing text data to training and evaluating a transformer-based model using the pre-trained transformer models for predicting sentiments of new texts.

The methods and code structure is common to both Electra_model.py and Roberta_model.py. Code is duplicated to make the pipelines self-contained.

The **load_data_from_jsonl** method loads text data from a .jsonl file, extracting texts and their corresponding labels.

It then processes this text data to be compatible with the model using tokenization (converting text to a numerical format that the model can understand).

The **preprocess_text** method cleans and normalizes the text data by removing unnecessary characters and stopwords (common words that don't contribute much to the sentiment).

The **create_model** method sets up the model with specific configurations for sentiment analysis.

The **train_model** method trains this model on the prepared and processed data. It uses callback functions like Early Stopping to optimize the training process ModelCheckPoint to save the best model after each epoch.

After training, the **evaluate_model** method assesses the model's performance on a separate set of validation data, providing insights into how well the model predicts sentiment.

The **load_model** function allows the loading of the fine-tuned model. The **infer** method uses this model to predict the sentiment of new and unseen texts.

Compute Sentiment Score Averages and Plotting the Sentiment Trend
compute six sentiment scores for each scraped Reddit post or comment. compute the average for each of the scores for each interval using the code

```
combined_averages = df.groupby('Interval Number')[emotion_columns].mean()
```

This code returns data for concise analysis and plotting of emotional and sentiment trends across different time intervals in the Reddit posts and comments.

Utilized Matplotlib to plot the sentiment trend of six emotions over a period of time given by the interval number (day, week, or month). The plot (line graph) displays the interval number on the x-axis and the average score of each emotion for each interval on the y-axis.

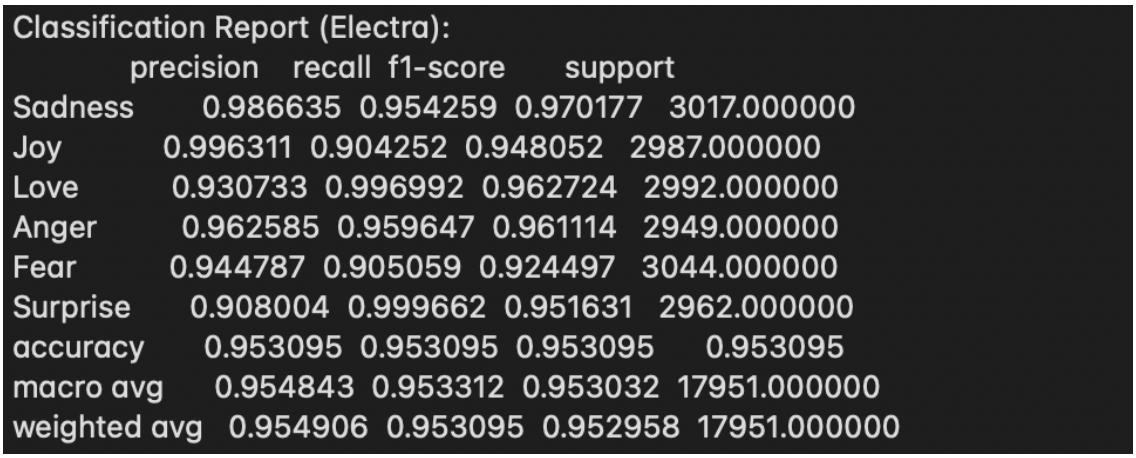
Results

Classification Report (RoBERTa):

	precision	recall	f1-score	support
Sadness	0.996178	0.950282	0.972689	3017.000000
Joy	0.994497	0.907600	0.949064	2987.000000
Love	0.930218	0.997995	0.962915	2992.000000
Anger	0.922177	0.988471	0.954173	2949.000000
Fear	0.963549	0.859724	0.908681	3044.000000
Surprise	0.908282	0.999662	0.951784	2962.000000
accuracy	0.950198	0.950198	0.950198	0.950198
macro avg	0.952484	0.950622	0.949884	17951.000000
weighted avg	0.952711	0.950198	0.949783	17951.000000

Fig 1. Classification Report of RoBERTa

The above figure represents different performance metrics to gauge how the fine-tuned RoBERTa model has performed across various emotion categories.



Classification Report (Electra):

	precision	recall	f1-score	support
Sadness	0.986635	0.954259	0.970177	3017.000000
Joy	0.996311	0.904252	0.948052	2987.000000
Love	0.930733	0.996992	0.962724	2992.000000
Anger	0.962585	0.959647	0.961114	2949.000000
Fear	0.944787	0.905059	0.924497	3044.000000
Surprise	0.908004	0.999662	0.951631	2962.000000
accuracy	0.953095	0.953095	0.953095	0.953095
macro avg	0.954843	0.953312	0.953032	17951.000000
weighted avg	0.954906	0.953095	0.952958	17951.000000

Fig 2. Classification Report of Electra

The above figure represents different performance metrics to gauge how the fine-tuned Electra model has performed across various emotion categories.

F-1 Scores are consistent across all the emotions in RoBERTa and Electra and are acceptable for a good model. However, ‘Fear’ has the lowest F-1 Score of 0.92 which is still high enough to be acceptable.

Model	Validation Accuracy	F1 Score
Bert (w/o Reddit data)	0.9520	0.9582
Bert (w/Reddit data)	0.91??	0.9140
Electra	0.9531	0.9530
Roberta	0.9502	0.9502

Fig 3. Validation Accuracy and F1_scores

The validation accuracy of Electra is 0.9531 with a 0.9530 overall F-1 Score and the validation accuracy of RoBERTa is 0.9502 and F-1 Score is 0.9502. There is no significant difference between the performance metrics between the two models.

	Post/Comment	ID	Text	Creation Date	Interval Number
0	Post	18eu311	Hermès heir plans to adopt 51-year-old gardener to pass on $11 billion fortune	2023-12-10	0
1	Post	18eordw	Don't forget guys it is our fault	2023-12-09	0
2	Post	18eiq1v	Crypto bro's growing bolder by the day	2023-12-09	0
3	Post	18ecxrr	Apple's head of iphone and Apple watch design leaving the company $AAPLPuts or calls for monday?	2023-12-09	0
4	Post	18e7i4k	TIL Baby Boomers hold $21 trillion worth of STOCKS. What will happen when they die? Will their survi	2023-12-09	0

Fig 4. Scraped subreddit data of r/wallstreetbets

The above figure is the result of running the Reddit scrape code with the following user input; num_comments = 5, num_posts = 100, subreddit_name = ‘wallstreetbets’, Time_filter = ‘month’, Interval = ‘daily’.

Interval Number	Sadness	Joy	Love	Anger	Fear	Surprise	Positive	Negative
0	0.2374	0.3779	0.0041	0.2747	0.0641	0.0418	0.2573	0.7427
1	0.2572	0.1457	0.0023	0.3023	0.2291	0.0633	0.0051	0.9949
2	0.2489	0.2825	0.0017	0.3091	0.0943	0.0634	0.0726	0.9274
3	0.2655	0.201	0.0348	0.3152	0.1177	0.0658	0.1639	0.8361
4	0.1575	0.3694	0.0021	0.2045	0.1537	0.1128	0.2766	0.7234
5	0.2165	0.241	0.0024	0.3056	0.1622	0.0722	0.17	0.83
6	0.1721	0.2979	0.0033	0.3021	0.1399	0.0846	0.1673	0.8327
7	0.3229	0.2815	0.0617	0.2267	0.0527	0.0545	0.2036	0.7964
8	0.2961	0.1818	0.0028	0.2805	0.1651	0.0737	0.2361	0.7639
9	0.2138	0.3987	0.0245	0.1863	0.0841	0.0926	0.294	0.706
10	0.2674	0.349	0.0034	0.2389	0.0835	0.0579	0.1274	0.8726
11	0.2032	0.2804	0.0026	0.3169	0.134	0.0628	0.1901	0.8099
12	0.1455	0.3591	0.0025	0.3217	0.1203	0.0508	0.3281	0.6719
13	0.2408	0.2597	0.0019	0.3107	0.0729	0.114	0.2111	0.7889

Fig 5. Dataframe after adding emotion scores

The above figure is the result after predicting the scores for each emotion of each sample in the live Reddit dataset for r/wallstreetbets.

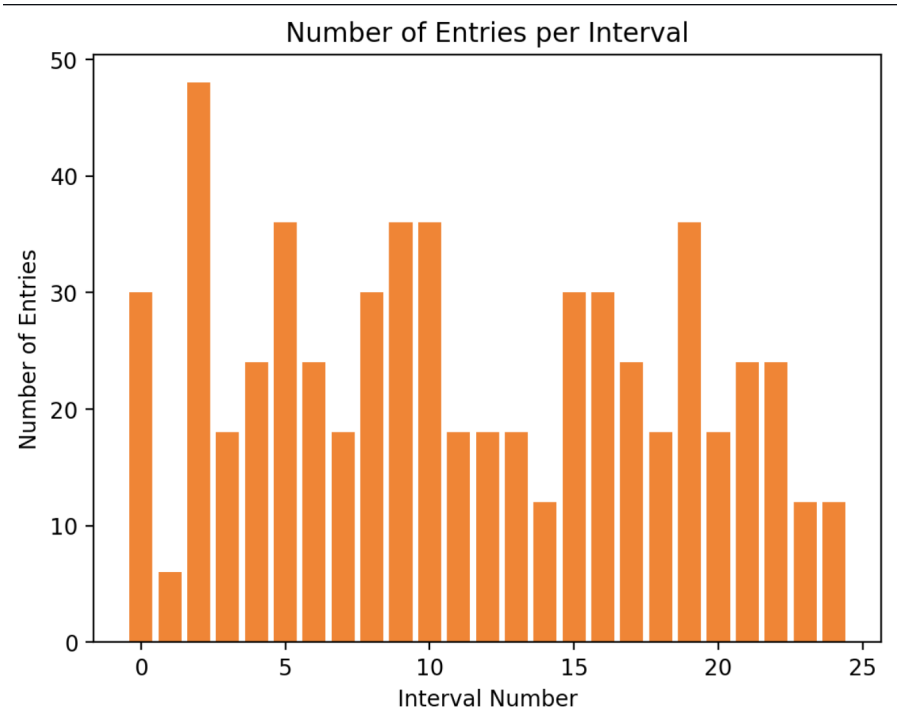


Fig 6. Number of Posts and Comments Histogram

The figure above illustrates the distribution of retrieved posts and comments from the subreddit r/wallstreetbets over various daily intervals. This histogram suggests that the input dataset contains a sufficient number of entries for each interval to yield accurate sentiment analysis.

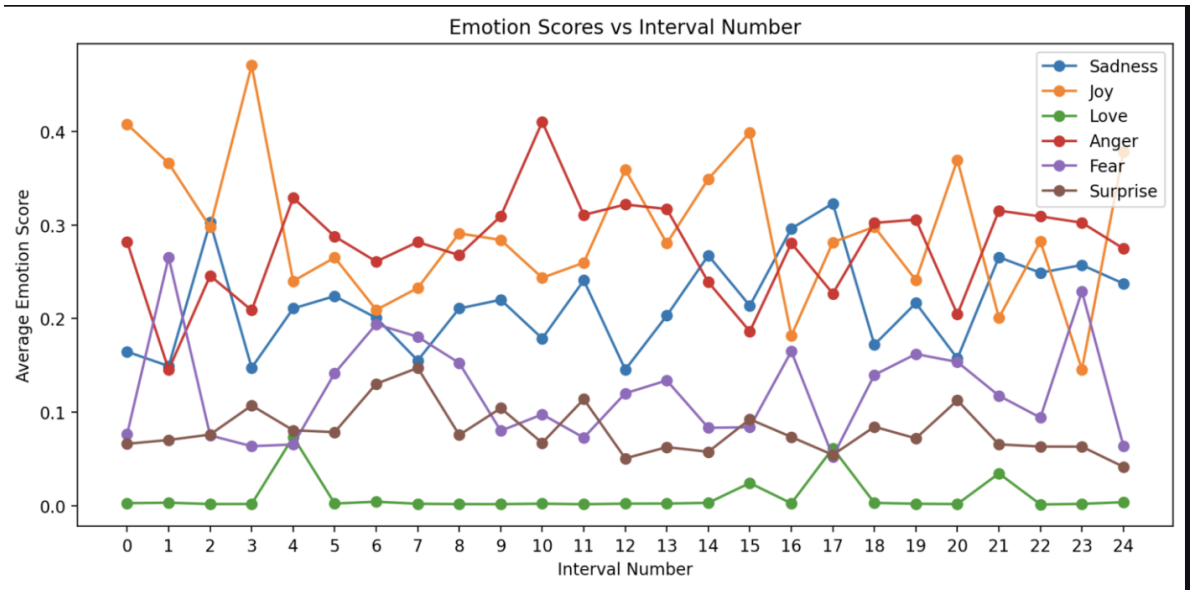


Fig 7. Average Sentiment Score Plot of r/wallstreetbets

r/wallstreetbets is primarily a community focused on stock market trading, often characterized by rapid shifts in investment positions, risk-taking, and the volatile nature of gains and losses. The primary emotions involved in this context are typically related to the outcomes of financial decisions—Joy in response to profits, Anger or Sadness due to losses, Fear from market uncertainty, and Surprise from unexpected market movements.

The figure above showcases a line graph depicting the average sentiment scores for six emotions across different intervals. A higher interval number corresponds to more recent data, while lower interval numbers indicate older data. From the graph, we can observe that members of r/wallstreetbets predominantly exhibit emotions of Anger, potentially due to financial losses in stocks, or Joy, likely resulting from profits. These predominant emotions are followed by Sadness, Fear, and Surprise, which are common in the context of r/wallstreetbets. The emotion of Love is observed the least as expected. As we can see the sentiment trend is characteristic of the r/wallstreetbets community.

Summary and Conclusions

The final output of my code includes generating a .csv file containing the text of subreddit posts and comments, along with their details such as 'ID', 'Creation Date', and 'Interval Number', obtained through subreddit scraping using PRAW. Additionally, it adds six more attributes for the emotion scores of each post and comment after prediction. These scores are then used to calculate the average sentiment score for each interval, resulting in another .csv file that includes the columns: ‘Sentiment’, ‘Average Sentiment Score’, and ‘Interval Number’. Finally, line plots are generated using these average scores and intervals for the six emotions.

For future enhancements to the project, I would consider the following:

User-Defined Date Ranges in PRAW: Update the Reddit scraping code to allow users to specify custom date ranges, providing more tailored data retrieval.

Real-Time Sentiment Trend Updates: Implement a system for automatically appending, analyzing, and predicting new data in real time to keep the sentiment trend analysis current and dynamic.

Extended Token Training: Increase the max_len attribute in the pre-trained Hugging Face models to utilize more tokens, potentially capturing more nuanced information in longer texts for improved model performance.

Percentage of the original code

From a coding perspective, we wrote our scripts on our own, improving and building upon each other, and sought assistance from online tools such as ChatGPT, Claude.ai, and Copilot when and where necessary. Measuring

specific lines of code across all of our scripts when each team member frequently made edits across all of them is difficult.

References

[1] https://praw.readthedocs.io/en/stable/, PRAW (Python Reddit API Wrapper) Documentation.
[2] https://huggingface.co/docs/transformers/model_doc/electra, ELECTRA Documentation.
[3] https://huggingface.co/docs/transformers/model_doc/roberta, RoBERTa Documentation.
[4] https://matplotlib.org/stable/users/explain/quick_start.html, Matplotlib Documentation.
[5] https://towardsdatascience.com/understanding-electra-and-training-an-electra-language-model-3d33e3a9660d, Understanding ELECTRA and Training an ELECTRA Language Model.
[6] https://docs.streamlit.io/, Streamlit Documentation.