Unveiling Polarization: Community Detection and Sentiment Analysis on Reddit Discussions

Exploring the Israeli-Palestinian Conflict Through Social Media Data

Matilde De Luigi
matilde.deluigi@unifr.ch

Matej Kutirov
matej.kutirov@students.unibe.ch

Flaminia Trinca
flaminia.trinca@students.unibe.ch

Repository
https://github.com/matejkutirov/community-detection-israel-palestine-reddit

Abstract

This study investigates the dynamics of community detection within a social graph constructed from over one million Reddit posts about the Israeli-Palestinian conflict. By employing sentiment analysis and various community detection algorithms, we analyzed the interplay between social support and emotional intensity in online discussions. Our customized Louvain algorithm demonstrated strong performance, validated through modularity and Normalized Mutual Information (NMI) metrics. Additionally, centrality measures provided insights into network structure and connectivity. Future work includes detailed sentiment analysis within major communities, temporal sentiment analysis, and the application of advanced machine learning models to enhance the accuracy and depth of insights. These efforts aim to deepen our understanding of community dynamics and sentiment expression in conflict-related social networks.

1 Introduction

Today, social media platforms play a crucial role in expression of public opinion, offering an open virtual space where voices from all kinds of different backgrounds converge. In conflict-ridden contexts, these platforms often become polarized, straying from neutral discussions.

This project explores these dynamics using a dataset from Kaggle, containing over one million Reddit posts about the Israeli-Palestinian conflict[1]. Our goal is not to form opinions but to analyze public sentiment regarding the ongoing war.

By analyzing posts and comments, we aim to understand the relationship between social support, expressed through "likes," and the intensity of language used. We constructed a graph where nodes represent posts and edges represent relationships based on sentiment. This graphical representation enables community detection. Applying sentiment analysis and community detection algorithms, we aim to uncover how emotional intensity and social support interact in online discussions during conflict.

2 Data Loading and Preprocessing

The dataset contains over one million data points and 24 variables, totaling approximately 20 GB. For our graph construction, we focused on the score (likes and dislikes for each post), comments, comment IDs, post IDs, and subreddits. Seven variables had significant missing values and were excluded from the analysis.

To understand the nature of the score variable, representing the sum of likes and dislikes, we analyzed its correlation and distribution. The score exhibited a positively skewed distribution, with many values near zero, few negative values, and some exceptionally high values exceeding seven thousand likes. This distribution is typical of social media platforms, where most users have low visibility, and a few outliers have significant social impact.

The correlation heatmap showed weak associations between variables, prompting us to create a new variable to calculate the intensity expressed in each comment using the SentimentIntensityAnalyzer. Sentiments were categorized as negative (below -0.05), positive (above 0.05), and neutral (between -0.05 and 0.05). We visualized the average sentiment trend from September 2023 to March 2024, noting a peak in negative sentiments following the attack on October 7th, which worsened until December. Emotional positivity increased in December during Egyptian ceasefire negotiations and UN decisions.

For our analysis, we focused on the following variables: score, sentiment score, comment ID, and post ID. After selecting these columns, we removed duplicates and missing values to ensure data integrity. Using the SentimentIntensityAnalyzer with parallel processing, we reduced computational time from eight minutes to two and a half minutes, optimizing the analysis for quicker and more reliable outcomes.

[1]https://www.kaggle.com/datasets/asaniczka/reddit-on-israel-palestine-daily-updated

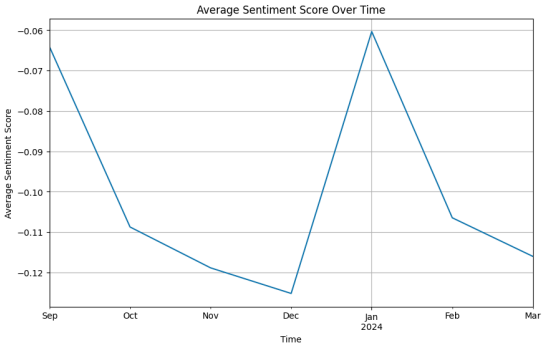

Figure 1: Average emotional intensity in post comments since the start of the conflict

3 Graph Construction

We opted to store the data as a weighted edge list instead of a matrix for efficiency. Storing the matrix would require about 2.8 GB of memory, whereas the edge list requires only about 12 MB.

To develop the weighted edge list for our network analysis, we first grouped the dataset by post_id, then computed the mean sentiment_score and the mean score for each grouped post. We sorted the posts by sentiment_score from -1 to 1, facilitating the application of a sliding window technique to generate the weighted edge list effectively. This method ensured a more structured and focused approach to network construction.

In the sliding window technique, we set a threshold to determine the range within which nodes can be connected based on their sentiment scores. For example, if a node has a sentiment score of 0.5000 and the threshold is 0.0005, this node connects to all other nodes with sentiment scores between 0.4995 and 0.5005. This method ensures that nodes with closely aligned sentiment values are linked, reflecting nuanced sentiment relationships in the network structure.

We constructed a graph with approximately 17,000 nodes and 490,000 edges. We removed nodes with a degree of 5 or less, representing less than 10% of the total nodes. The node with the highest degree had 712 connections, while the lowest had 1. This pruning strategy focused on maintaining more interconnected nodes, enhancing the robustness and relevance of the network analysis.

By eliminating nodes with low degrees, we aimed to concentrate the analysis on the more interconnected and potentially influential nodes within the network topology. This strategy improved the overall quality and significance of subsequent network exploration and community detection processes. Retaining highly interconnected nodes helped us achieve a more robust and cohesive structure, facilitating the identification of meaningful communities and patterns.

4 Network Exploration

Before analyzing our network, we explored its properties through various visualizations. First, we visualized a subgraph to check edge density. Second, we computed the degree centrality of the nodes to understand their distribution. Lastly, we implemented k-means clustering to visualize the distribution of the score and sentiment score.

4.1 Subgraph and Degree Centrality

To begin, we created a smaller subgraph consisting of 1,000 nodes to visualize the graph's density. We then computed the degree centrality to analyze the degree distribution. The degree distribution revealed that most nodes have a small degree, while a few nodes have a degree approximately four times higher. This skewed pattern indicates that many nodes are likely to be part of small, disconnected communities, while a few highly connected nodes may serve as hubs, facilitating larger, more cohesive communities.

The mode of the degree distribution lies in the lower range, showing that most nodes have limited connections. However, the distribution's long tail includes a small number of nodes with exceptionally high degrees, deviating significantly from the average.

This skewed degree distribution has significant implications for community detection. The numerous low-degree nodes suggest a tendency for smaller, potentially disconnected communities. In contrast, the few highly connected nodes could act as hubs, supporting the formation of larger, more integrated communities. In our graph, nodes represent created posts, and edges are formed based on sentiment scores, weighted by the number of likes. The degree distribution reflects the inherent nature of sentiment expression in online discourse.

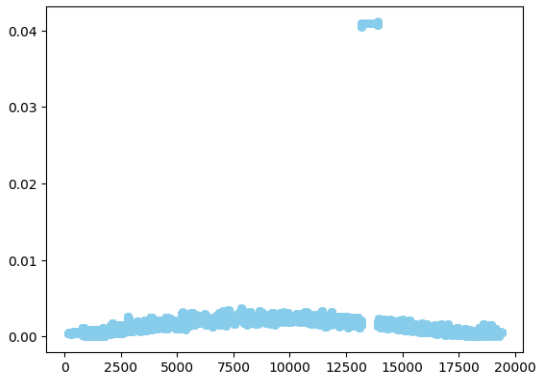

Figure 2: Degree Centrality for each of the nodes (posts) in our graph

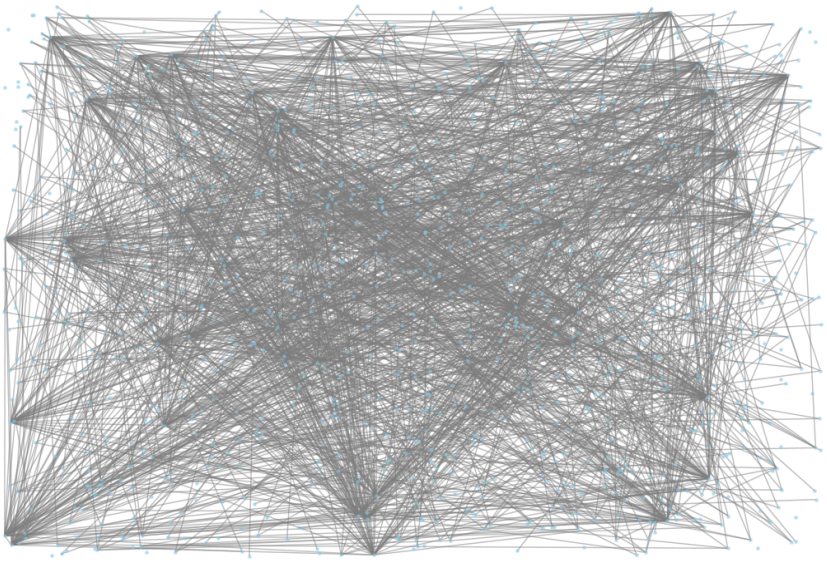

Figure 3: Visualization of randomly sampled subgraph made from our graph

4.2 k-Means

We used the k-Means clustering algorithm to categorize and visualize posts according to their sentiment and popularity. The x-axis represents the score of each post, indicating its popularity, while the y-axis depicts the sentiment score. This approach delineates three distinct clusters: popular posts with neutral sentiment, posts with positive sentiment but lower popularity, and posts with negative sentiment and lower popularity.

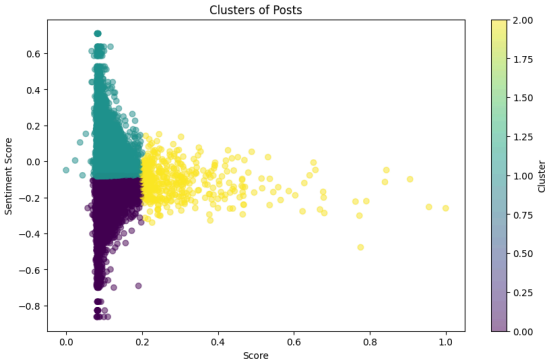

Figure 4: KMeans clustering of the posts based on score (popularity) and sentiment score

5 Network Analytics

For the analytics part of the project, we implemented our own version of the Louvain community detection algorithm and analyzed its runtime and community quality. We then compared it to other community detection algorithms using selected metrics. Finally, we computed centrality measures on the hypergraph of communities and visualized the overall sentiment using word clouds.

5.1 Louvain Implementation

We chose the Louvain algorithm for community detection, basing our implementation on the "GVE-Louvain" article by Ghosh et al. (2023)[2], which is one of the fastest and most optimized implementations available. The first_phase function optimizes modularity by moving nodes between communities, and the second_phase function aggregates the communities into a new, smaller graph. This process repeats until no further modularity gain is possible or the number of iterations reaches a predefined max_iterations to avoid infinite loops.

We included a max_iterations parameter to balance between correctness and time efficiency. The default value is set to 1 iteration for simplification. We also implemented helper functions to visualize the graph with its communities.

5.1.1 Execution Time

To test our functions, we generated random weighted graphs and applied our Louvain implementation, varying the number of iterations. We compared the execution times with the networkx implementation of Louvain. The results showed that execution time increases proportionally to the number of iterations. Our implementation is as fast, or faster, than the networkx implementation when performing only one iteration.

5.1.2 Community Quality

We analyzed community quality using three random graphs of different sizes. For each graph, we:

[2]GVE-Louvain: Fast Louvain Algorithm for Community Detection in Shared Memory Setting: https://arxiv.org/html/2312.04876v4

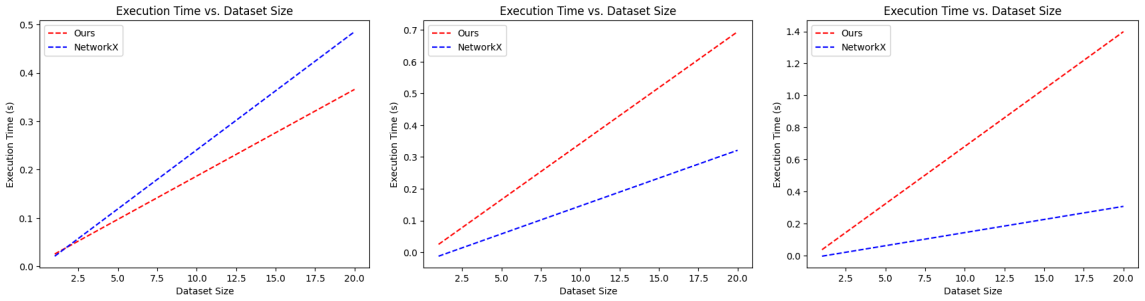

Figure 5: Comparison between the execution time of our algorithm and the algorithm from networkx. As we can, even though our algorithm performs better for 1 iteration (left image) it struggles as the number of iterations increases.

1. Detected communities using our Louvain implementation with 1, 5, 10, and 20 iterations.
2. Detected communities using the networkx Louvain implementation.
3. Compared the number of communities, the largest and smallest community sizes.
4. Calculated NMI (Normalized Mutual Information) and Max Modularity.
5. Visualized the communities and hypergraphs.

Our analysis showed that as iterations increase, the number of communities decreases, and the size of the largest community and modularity increase. The networkx implementation generally produced fewer but larger communities with higher modularity, indicating better clustering. Interestingly, the NMI was highest with only one iteration, suggesting that our implementation with one iteration is most similar to the networkx implementation.

5.2 Community Detection

5.2.1 Community Detection on Original Graph

We applied our algorithm to the original graph with 17,309 nodes and 493,728 edges, varying the maximum number of iterations (1, 5, and 10). We compared the community partitions to networkx communities using NMI. The NMI remained high (~0.77) regardless of the number of iterations, indicating that our implementation is accurate and fast, running in less than 4 minutes.

5.2.2 Other Community Detection Algorithms

We further analyzed the social graph using additional community detection algorithms:

- Label Propagation Algorithm (LPA)
- Greedy Modularity Communities

We compared these communities using metrics like Max Modularity, Silhouette Score, Coverage, and Purity. The NMI was also calculated between our Louvain communities and those from other algorithms.

Our findings showed that modularity is high for all methods, with networkx Louvain having slightly higher values. All methods exhibited negative silhouette scores, indicating possible overlaps or misplacements. Coverage was good across all algorithms, implying strong internal connectivity. The Greedy Modularity algorithm had the highest NMI, suggesting it produces communities most similar to our implementation.

5.2.2 Comparing the Communities

We analyzed the quality of the selected communities using various metrics. Our algorithm achieved a purity of 0.15, while the NetworkX Louvain algorithm achieved a higher purity of 0.25, indicating more homogeneous communities. We also evaluated modularity, silhouette score, coverage, and Normalized Mutual Information (NMI) using our Louvain algorithm, the NetworkX Louvain algorithm, the Label Propagation Algorithm, and the Greedy Modularity Communities algorithm. Our Louvain algorithm demonstrated good modularity, with the NetworkX implementation slightly better. Both the Label Propagation and Greedy Modularity algorithms had lower modularity scores. Negative silhouette scores across all methods suggested potential overlaps in community assignments. Despite this, all algorithms showed good coverage, indicating strong internal connectivity within communities. The Greedy Modularity algorithm had the highest NMI score, indicating that its communities closely aligned with those detected by our Louvain implementation. This suggests that Greedy Modularity effectively identifies similar community structures.

Figure 6: Visualization of the communities of our graph (left) and the hypergraph of communities (right)

	Implementation	Number of Communities	Largest Community	Smallest Community
0	Our Louvain	334	4460	1
1	NetworkX Louvain	203	709	1
2	Label Propagation Algorithm	675	712	1
3	Greedy Modularity Communities	163	8333	1

	Partition	Modularity	Silhouette Score	Coverage	Purity
0	Louvain	0.730484	-0.992374	0.993198	0.145762
1	nx-Louvain	0.778637	-0.999422	0.994732	0.232827
2	Label Propagation Algorithm	0.702259	-0.999422	0.920393	0.855913
3	Greedy Modularity Communities	0.656246	-0.999422	0.991033	0.131203

	Partition	NMI
0	nx-Louvain	0.773438
1	Label Propagation Algorithm	0.624259
2	Greedy Modularity Communities	0.823647

Table 1: Results of the analytics part

Overall, while the NetworkX Louvain algorithm performed slightly better in purity and modularity, our implementation was efficient and comparable. The Greedy Modularity algorithm excelled in NMI, showing strong alignment with our community structures.

5.3 Network Exploration on Community-Hypergraph

We explored the hypergraph of communities from our Louvain implementation by measuring degree centrality, betweenness centrality, and closeness centrality.

Degree Centrality: This measure showed a few highly central communities with many connections, indicating hub-like structures, while most communities had relatively few connections.

Betweenness Centrality: A few communities had high betweenness centrality, acting as key connectors within the network. Most communities had low betweenness centrality, indicating they were not significant bridges.

Closeness Centrality: Some communities had high closeness centrality, meaning they could quickly reach other nodes, while most had low closeness centrality, indicating longer average paths to other nodes. This suggests a network with a few well-connected communities facilitating efficient communication, while most communities were more peripheral.

These centrality measures revealed the presence of crucial hub communities and the overall connectivity structure within the hypergraph.

5.4 Sentiment Analysis

Finally, we performed sentiment analysis by categorizing sentiment scores into three categories:

- Negative (sentiment score between -1 and -0.5)
- Neutral (sentiment score between -0.5 and 0.5)
- Positive (sentiment score between 0.5 and 1)

We visualized word clouds and plotted the distribution of comments per category. The general sentiment in Reddit posts was predominantly negative. Word clouds revealed that "Israel" appeared more in positive and neutral contexts, while "Hamas" was used more neutrally.

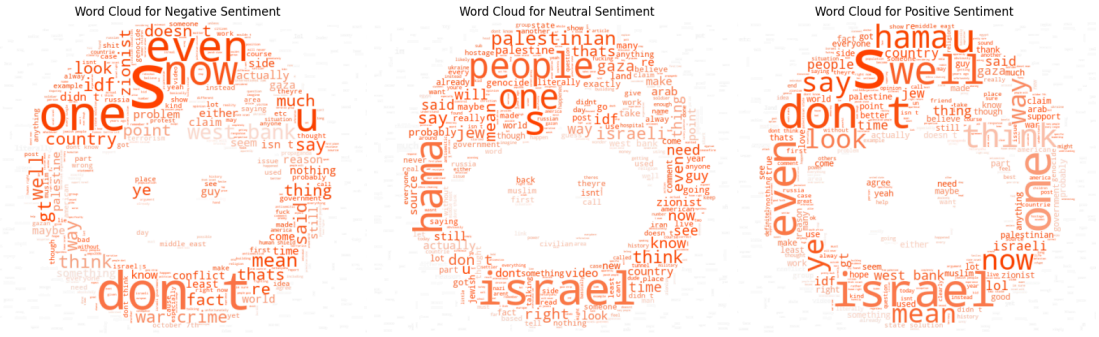

Figure 7: Word Clouds of the different sentiments

6 Graph Persistence

As we concluded our analyses, we explored professional graph database systems to deepen our understanding of the constructed graph and explore future research avenues. The Neo4j graph database was particularly valuable, using Cypher, a simple yet powerful query language.

We analyzed relationships between subreddits (dark blue), posts (pink), post creation time (celestial hue), comments (green), and comment scores (yellow). This intricate structure highlighted the multi-hierarchical interactions between these variables, supporting our strategy of using sentiment scores averaged per post instead of per comment.

Applying the same method to post creation times could reveal how the sentiment of words changes over months, offering deeper insights into temporal sentiment dynamics.

7 Conclusion and Further Steps

In this project, we explored community detection on a social graph constructed from Reddit posts about the Israeli-Palestinian conflict. By applying sentiment analysis and various community detection algorithms, we identified key patterns in the data, revealing the relationship between social support and emotional intensity in online discussions.

Our Louvain algorithm implementation, compared to other methods, showed strong performance in terms of modularity and NMI, validating its effectiveness. Additionally, the exploration of centrality measures on the community-hypergraph provided insights into the network's structure and connectivity.

For further work, we suggest performing sentiment analysis on the largest detected communities to determine if their sentiment distribution matches the general sentiment score distribution, and assessing the purity of sentiment within each community to understand the homogeneity of sentiments. Additionally, investigating temporal changes in sentiment could reveal how discussions evolve over time. Extending the analysis to other conflict-related datasets would help generalize findings and compare community structures across different contexts. Implementing more advanced machine learning models for sentiment analysis could improve accuracy and depth of insights. Finally, exploring the impact of external events on community formation and sentiment dynamics could enhance our understanding of real-world influences on online discourse. These steps could refine our current findings and expand our understanding of community dynamics and sentiment expression in online social networks.

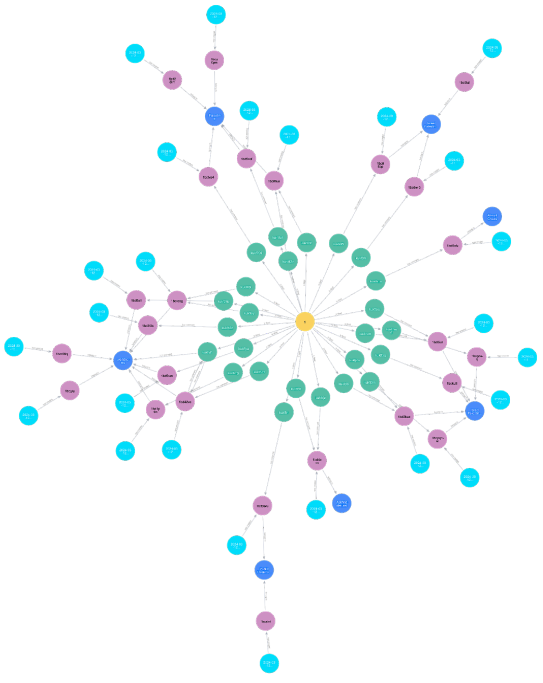

Figure 8: Visualization of Graph Persistence

Work Contribution: *Throughout the whole process, ideas were continuously discussed and exchanged. So was the code: it was modified, corrected and optimized by all parties involved. While Matej took the lead in creating the project pipeline, doing the graph construction and the network analysis, Matilde contributed to the network analysis and led the preprocessing of the data and the graph persistence. Flaminia took the lead in writing the Louvain implementation, doing the network analysis and finally, putting everything into format for the final report.*