Political Sentiment and Community Analysis

State of The Art

Coordinator: Constantinescu Gabriel

Team Members: Lazurcă Samuel-Ionuț, Rotaru Florin Eugen, Nastasiu Ștefan, Nestor Maria, Hriscu Alexandru-Gabriel

GitHub: https://github.com/TAIP-Project-2024

Introduction

Our application is a real-time dashboard that fetches live social media data (e.g., from Twitter, Reddit, etc.) and analyzes sentiment in posts and comments. The results will be displayed on an interactive graph where nodes represent users and edges represent interactions (likes, replies, mentions, etc.).

The interest in sentiment analysis and community analysis has increased recently due to upcoming political events, an example being this year's elections in the USA. Due to the influence offered by the media networks, political campaigns use the resources offered by the online environment such as posts and debates on different platforms (e.g. 𝕏 formerly Twitter).

Therefore, a very good indicator of the effectiveness of a political campaign is the way in which a political party or a candidate is perceived by the users of social media networks. To determine political sentiment, the following can be taken into account: the way people respond to a post, observing the number of likes and comments, what kind of comments are at the top of a post and how often a post is shared. Also, the social media post of a candidate can decisively change the political sentiment, for example when he makes a mistake or the way he responds to certain controversies (for

example illegal immigration). However, the sentiment and community analysis should not be limited to politics or the elections of a single country, but to include several topics, for example Apple vs Samsung.

Related works in the field

The field of sentiment analysis has evolved significantly, marked by foundational contributions that have shaped its current methodologies and applications. An example of an application that is useful in the analysis of political sentiment at the level of social media is Crimson Hexagon (Brandwatch) which, although it is also used for other purposes such as following trends in a market space, can be used for politics, therefore being useful for the analysis of political sentiment during the Brexit period (more info). Another popular application is Talkwalker, which offers social listening, media monitoring and social benchmarking services. In addition, Facebook has developed a tool called CrowdTangle, which isn't used anymore, to see what kind of posts go viral. It was based on the way the posts are distributed, and its purpose was to identify influencers who commented on a certain topic, while also analyzing the polarization of communities. In addition, it was used to analyze the spread of fake news.

Recent academic projects have demonstrated the effective use of machine learning models to analyze political sentiment and polarization on social media. Research from universities has focused on the application of natural language processing (NLP), sentiment analysis, and community detection in order to map out polarized political communities.

A study by Zhang et al. [1] explored various approaches to sentiment analysis on Reddit comments related to the 2019 Indian General Elections, leveraging advanced deep learning architectures including Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks. Their objective was to gain comprehensive insights into public opinion dynamics and sentiment trends surrounding political events discussed on social media platforms like Reddit. To achieve this, they conducted a series of experiments employing different model architectures, namely CNN, LSTM, LSTM+CNN, and CNN+LSTM. Their experimental results revealed that the hybrid CNN+LSTM model achieved superior performance in sentiment classification, consistently outperforming the individual CNN and LSTM models, obtaining a Macro F1-Score of 0.84.

A study conducted by Marozzo et al. [2] on analyzing political sentiment using social media data describes how posts are categorized based on specific keywords that signal political alignment. These keywords are pre-processed and classified into factions, allowing the system to detect user polarization and predict trends during election campaigns. This research emphasizes using keyword-based approaches in combination with advanced algorithms like multilayer perceptrons (MLPs) and convolutional neural networks (CNNs) to improve sentiment classification and topic modeling. These methodologies help map out the communities of users with distinct political preferences by examining their engagement and interaction patterns on platforms like Twitter or Facebook.

In another project by Belcastro et al. [3] from an European university, researchers applied community detection algorithms to group users by their interaction networks and identified polarized clusters based on their political sentiment. These clusters were then visualized using color-coded graphs that show the strength of interaction between users aligned with different political opinions. These visual tools can be helpful for political scientists to better understand the evolving polarization during important political events, such as elections.

From a security perspective, academic discussions around data protection in social media analytics stress the need for encrypting both data at rest and in transit. Ensuring that passwords are hashed and never stored in plain text is critical to prevent unauthorized access to sensitive political data. These practices, combined with principles like Open-Closed Design (allowing system extensions without altering existing functionality), are essential to protect the system from vulnerabilities and external threats. Additionally, techniques such as role-based access control (RBAC) and database encryption ensure that data, especially user-generated content from social networks, remains secure against potential breaches.

These combined efforts in sentiment analysis, visualization, and security form the backbone of modern political analysis tools that monitor public opinion and polarization trends across digital platforms.

Data Processing and Storage

The data used by our application will come from the 𝕏 platform and possibly from Reddit or Facebook, using the APIs provided by these platforms along with existing datasets. To achieve optimal results, we need to work with a well-structured and analysis-ready dataset. This can be accomplished by applying appropriate preprocessing methods to the input data.

The most used preprocessing methods are:

- *lowercasing* - converting all text to lowercase helps ensure that the same tokens are not treated differently if they happen to have different capitalization;
- *tokenization* - breaking the text into elementary units, usually words, in order to evaluate the vocabulary and prepare for further analysis;
- *removing punctuation* - periods, commas, question marks and others are not useful for most of the text mining tasks. In some cases, we may want to keep the ”-” symbol, as it helps saving frequent word combinations such as ”real-time”, ”follow-up”, etc. from being tokenized;
- *removing irrelevant parts* - for example removing email addresses, document headers, dates, numbers, URLs, special characters and the possible non-decodable characters;
- *dealing with frequent words* - common words like ”a”, ”the”, ”is” etc. don’t carry significant meaning and we could remove or subsample them;
- *stemming* - is used to identify the essential part of the word. Shortening the word saves computation resources, for example: ”Connect”, ”Connected”, ”Connection”, etc. can all be mapped to ”Connect”.

Once the dataset has been preprocessed, the next step is to represent its relevant features. We can do this using feature selection. A good way to do that is to use word embeddings, but there are also other ways like bag-of-words.

Regarding data storage, a good option is to use a NoSql database such as MongoDb due to the data structure. But for storing confidential data such as user accounts, it is more appropriate to use a relational database such as PostgreSql.

For the development of the services offered by our application, we will use Django as it is a well-documented framework and it is also made for the Python programming language, which is suitable for applications that work with large amounts of data. There are also libraries like TensorFlow or PyTorch, that ease the development of machine learning algorithms.

And for the frontend of our application we will use ReactJs.

Sentiment Analysis Methods

Sentiment analysis involves analyzing human opinions or comments about a product, service, or event shared through platforms like social media, websites, or emails. The complexity arises because a single piece of text can have multiple interpretations, making sentiment analysis challenging. It is conducted using text mining techniques combined with machine learning and natural language processing tools. This analysis helps uncover emotions expressed in online text, providing insights into an individual's psyche. The two main branches of sentiment analysis are opinion mining and emotion mining. Opinion mining focuses on detecting subjectivity and classifying polarity, while emotion mining involves emotion detection, polarity classification, and emotion categorization. In this study, we will focus on both opinion and emotion mining.

There are a lot of methods or approaches in sentiment analysis. Some of them include:

1. Lexicon based approach

Lexicons are collections of tokens where each token is assigned a predefined score indicating the positive, neutral, or negative nature of the text. In the lexicon-based approach, the text is tokenized into individual words. These words are mapped with sentiment scores, and by summing the positive, negative, and neutral scores separately, we determine the overall sentiment of the text. There are two main approaches:

- **Corpus based approach.** This approach uses large sets of corpora to understand the sentiment of a word from the context. This allows words to have different sentiments depending on the domain or context.
- **Dictionary based method.** Each word has a predefined sentiment in a static lexicon. The primary assumption behind this approach is that synonyms have the same polarity as the word, while antonyms have

opposite polarity. Lexicons such as WordNet, which links words to their synonyms, antonyms, and various meanings, and SentiWordNet, which assigns numerical sentiment scores to WordNet synsets, can be used in this approach.

2. Machine learning approach

Machine Learning Algorithms are used to categorize sentiments. The task of analyzing the sentiments can be accomplished using both supervised and unsupervised learning methodologies. The most commonly used algorithms include:

- **Naive Bayes (NB).** This method is utilized for both training and categorization. It computes the probability of a text belonging to a particular sentiment class. On smaller datasets, this approach performs well, but may struggle with negative sentiment classification.
- **Support vector machine (SVM).** This approach is commonly used for sentiment polarity classification and is known for its high accuracy**.**
- **Logistic regression (LR).** This method is effective for binary classification tasks, such as two opposite sentiments (positive and negative).
- **Decision trees (DT)**
- **K-nearest Neighbors (KNN)**

3. Hybrid approach

The hybrid approach combines machine learning and lexicon-based approaches. The lexicon handles cases where domain-specific terms appear, while machine learning techniques capture contextual nuances.

4. Deep learning techniques

This approach uses neural networks in sentiment analysis to automatically learn patterns and relationships in data. A comprehensive overview of deep learning approaches suited for sentiment detection, analysis, and classification are shown in Fig 1.

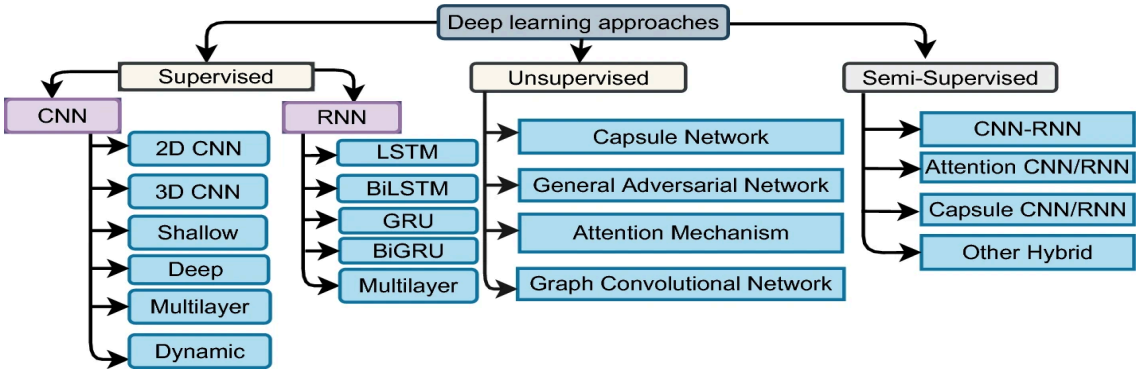


Fig 1: Taxonomy of sentiment analysis method [4]

Hybrid model CNN-LSTM

In the sentiment analysis universe, the **Convolutional Neural Network** (**CNN**) plays a key role in identifying and extracting local features and patterns from the text. The process begins with preprocessing the data (obtained from 𝕏), transforming the text into word embeddings, which represent each word as a continuous vector in a high-dimensional space. These word embeddings are then fed into the CNN. As the input passes through the CNN layers, the model applies several convolutional filters of different sizes to the text. These filters move across the input, capturing local features and patterns at various levels of abstraction. Each filter generates a feature map, which highlights specific patterns within the text. The final output of the CNN consists of these feature maps, each capturing unique aspects of the input. These feature maps, also known as CNN embeddings, encapsulate the local features and patterns from the 𝕏's (tweets), creating a detailed representation of the text. These embeddings emphasize the important textual features necessary for accurate sentiment analysis.

After the Convolutional Neural Network (CNN) in the sentiment analysis architecture for 𝕏 data, a **Long Short-Term Memory** (**LSTM**) network is used to capture contextual dependencies and sequential information within the text. The LSTM processes the embeddings generated by the CNN, leveraging its capacity to retain long-term information and understand the sequence of the text. These CNN-generated embeddings, which capture local features and patterns from the preprocessed 𝕏's, serve as the input to the LSTM. The LSTM then uses these embeddings to interpret the contextual relationships and dependencies between words in the sequence. As the embeddings pass through the LSTM layers, the network produces output embeddings that encapsulate both contextual information and sequential dependencies in the

text. These output embeddings represent a rich, context-aware understanding of the input, capturing the intricate relationships between words and phrases in the 𝕏 posts and comments. The LSTM's ability to model these relationships is essential for effective sentiment analysis, as it allows the model to interpret the text within its broader context, leading to a more accurate and comprehensive sentiment assessment of the data.

Visualization

Data can be represented in multiple ways:

1. **As networks** - representing data through “node-link” diagrams: vertices connected by curved segments. Both the vertices and the curved segments have geometric and graphical attributes, such as color, style, route, etc.

   The goal of a network representation is to be faithful - encode all the facts in the data and only those facts, and effectively - convey information in an accessible way. There is no universally good layout, since it depends on the audience, task, data, etc.

   A popular way to represent human activity on social media and political trends can be a network graph, (for example, nodes can represent users and edges interactions like tweets/replies). Some standard techniques are:
   - **Force-Directed, Energy-Based Layouts [5]**: modeling the graph as a system that has overall energy, which has to be minimized for a good drawing. An example is: unconnected vertices repel each other and vertices linked by edges attract each other, similarly to electrically charged steel rings and springs. Algorithms like Spring Embedder and Fruchtermann Rheingold [6] are used.

   - **Powergraph analysis**: A lossless compression of a graph that relies on finding groups of nodes that have the same external connections, (these can be common pages followed, etc). Group together nodes using / optimizing a certain measure (nodes „closer“ to each other than to nodes outside cluster, intra-cluster density vs inter-cluster sparsity)

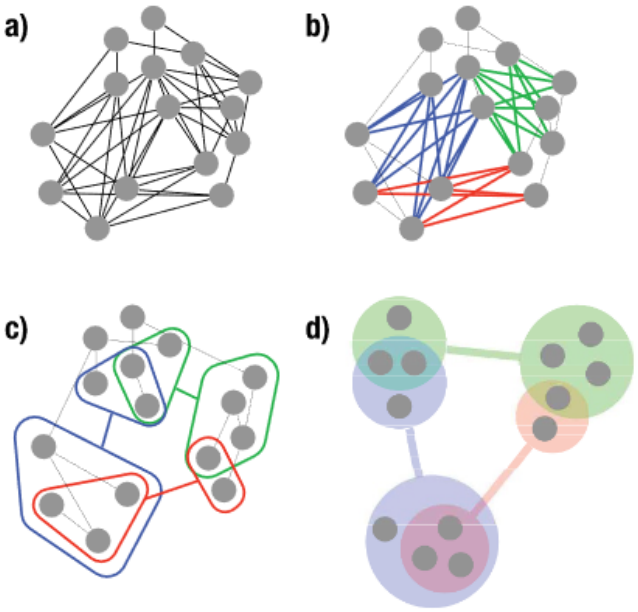
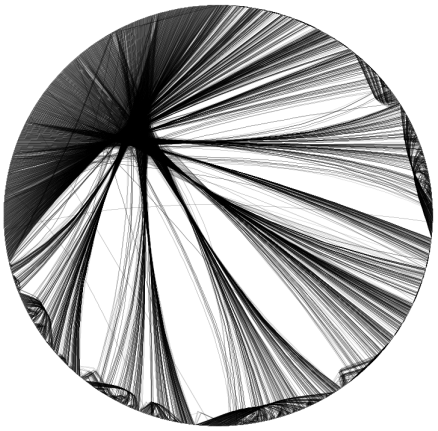


Illustration of the compression of a network into a power graph with overlapping power nodes, Ahnert 2014

- **Circular Layouts** Nodes are arranged evenly around the circumference of a circle, with edges drawn between them. Advantages are: reduced clutter and edge crossings, providing a clear view of relationships. Highlights symmetrical relationships and community structures. Good for smaller networks.



Circular layout example with edge bundling

Other techniques for network visualization:

- **Various aggregation**, such as edge bundling - cluster compatible edges together. Also can be done with a force-directed approach.

- **Dealing with large graphs**
  i. Multi-level methods - construct a sequence of increasingly smaller graph representations (“coarsening levels”) that approximately conserve the global structure of the input graph G, and then compute a sequence of approximate solutions, starting with the smallest representation. Intermediate results can be used on the subsequent level to speed up the computation and to achieve a certain quality. After computing a layout for the coarsest level from scratch, for each of the intermediate levels a force-directed layout method is applied
  ii. Graph Sampling - reducing size but keeping the unsampled graph structure, keeping cluster quality, density, etc.
  E.g. snowball sampling - begins with a small group of nodes (participants), which recruit additional nodes through their edges (relationships). This iterative process continues until the desired sample size is achieved or no new nodes can be identified. It is particularly useful for exploring hard-to-reach populations in social network analysis and other graph-related studies.

2. **Non Networks:**
   - **Word Clouds:** By displaying frequently used words or phrases associated with political topics, word clouds can illustrate the prevailing themes in public discourse. Interactive word clouds can highlight different sentiment aspects when hovered over or clicked.
   - **Interactive Charts (VR/AR, GUI) -** allowing user exploration. https://transparency.tube/.
   - **Sentiment Heatmaps**: These maps display geographical sentiment variations, allowing users to see areas of strong support or opposition. They provide immediate visual cues about where sentiments are concentrated.
   - Barplots, Pie-charts, Line graphs for time series.

Resources and relevant links:

1. Zhang, Ning & Xiong, Jize & Zhao, Zhiming & Feng, Mingyang & Wang, Xiaosong & Qiao, Yuxin & Jiang, Chufeng. (2024). Dose My Opinion Count? A CNN-LSTM Approach for Sentiment Analysis of Indian General Elections. Journal of Theory and Practice of Engineering Science. 4. 40-50. 10.53469/jtpes.2024.04(05).06.
2. Marozzo, Fabrizio & Bessi, Alessandro. (2017). Analyzing Polarization of Social Media Users and News Sites during Political Campaigns. Social Network Analysis and Mining. 8. 10.1007/s13278-017-0479-5.
3. Belcastro, L. & Cantini, Riccardo & Marozzo, Fabrizio & Talia, Domenico & Trunfio, Paolo. (2019). Discovering Political Polarization on Social Media: A Case Study. 10.1109/SKG49510.2019.00038.
4. Islam, M.S., Kabir, M.N., Ghani, N.A. et al. "Challenges and future in deep learning for sentiment analysis: a comprehensive review and a proposed novel hybrid approach". Artif Intell Rev 57, 62 (2024). https://doi.org/10.1007/s10462-023-10651-9
5. Peter Eades. A heuristics for graph drawing. Congressus numerantium, 42:146–160, 1984.
6. Fruchterman, Thomas MJ, and Edward M. Reingold. "Graph drawing by force-directed placement." Software: Practice and experience 21.11 (1991): 1129-1164.
7. Neppare, Christoffer. "A Force Directed Graph for Visualization of Voters Preferences Relative to Political Parties." (2018).
8. Wankhade, M., Rao, A. C. S., & Kulkarni, C. (2022). A survey on sentiment analysis methods, applications, and challenges. Artificial Intelligence Review, 55(7), 5731–5780. https://doi.org/10.1007/s10462-022-10144-1

Tools:

- https://react.dev/reference/react - ReactJs
- https://www.djangoproject.com/ - Django
- https://www.mongodb.com/ - MongoDb
- https://www.postgresql.org/ - PostgreSQL
- https://www.tensorflow.org/ - TensorFlow
- https://developer.x.com/en/docs/x-api/getting-started/about-x-api - 𝕏
- https://ogdf.uos.de/ - OGDF
- https://cytoscape.org/ - Cytoscape
- https://gephi.org/ - Gephi
- https://www.yworks.com/products/yed - yEd