Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 11, No.1, November : 2024

A SENTIMENT ANALYSIS FOR TWEETS TO DETECT THE GENUINENESS OF TWEETS.

Tapaswi Mestry, Department of Information Technology Vidyalankar Institute of Technology Mumbai, India Tapaswi.mestry@vit.edu.in
Dr. Vipul Dalal Professor, Department of Information Technology Vidyalankar Institute of Technology Mumbai, India Vipul.dalal@vit.edu.in

ABSTRACT
Sentiment analysis involves identifying and classifying the emotions or opinions expressed in text. Social media platforms, especially Twitter, generate an enormous volume of rich data in the form of tweets, status updates, blog posts, and more. Analysing sentiment from this user-generated content is extremely valuable for understanding public opinion. However, sentiment analysis on Twitter is more challenging than general text analysis due to the use of slang, abbreviations, and frequent misspellings. Two primary methods for sentiment analysis are the knowledge-based approach and the machine learning approach. Social media platforms like Twitter allow individuals to freely express their views on a wide range of topics, making it a key source for understanding public and private opinions. For businesses and organizations, Twitter provides a fast and effective way to gauge customer sentiment, which is crucial for success in the competitive market. Developing a sentiment analysis tool helps automate the process of measuring public perceptions. This project aims to explore various patterns in tweets, utilizing multiple strategies to identify the sentiment behind each tweet and assess whether a tweet is authentic or not.

Keywords:
NLP (Natural Language Processing), Sentiment analysis, machine learning, influence of tweets, POS (Part of Speech).

INTRODUCTION

Twitter, with over 100 million daily active users and more than 500 million tweets posted every day, has become a significant platform for people to express their opinions on various issues, brands, companies, and other topics of interest. This large user base makes Twitter an essential source of information for organizations, institutions, and companies seeking to gauge public sentiment. On Twitter, users can share their thoughts in the form of short tweets, limited to 140 characters. As a result, people often use slang, abbreviations, emoticons, and shorthand, which can make the language highly condensed and informal. Furthermore, sarcasm and polysemy (multiple meanings of a word) are common, making the language even more complex. Due to these characteristics, Twitter language can be considered unstructured and difficult to analyse directly. To extract meaningful sentiment from tweets, **sentiment analysis** is employed. This process can be applied in various areas, such as monitoring sentiment shifts around a particular event, understanding public opinion on a specific brand or product release, or analysing attitudes toward government policies. Extensive research has been conducted on Twitter data to classify tweets and analyse sentiment patterns. In this project, our goal is to predict emotions from tweets by determining their polarity—whether they are positive, negative, or irrelevant. Sentiment analysis is essentially the process of interpreting the sentiment behind a statement or sentence. It is a classification technique that extracts opinions from tweets and organizes them into categories. The sentiment of a tweet is subjective and depends on the context or topic being discussed. The model's effectiveness is often determined by how well it can classify the sentiment based on the features it analyses. In the context of this project, **sentiment** refers to the specific categories or emotions that the sentiment analysis model seeks to identify in the tweets. The dimension of the sentiment categories is crucial to the performance of the model. For instance, we could have a two-class sentiment classification (positive vs. negative), or a three-class classification (positive, negative,

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 11, No.1, November : 2024
and irrelevant). Sentiment analysis approaches can be broadly classified into two categories: **lexicon-based** and **machine learning-based**. The lexicon-based approach is unsupervised, relying on predefined sentiment lexicons and scoring methods to analyse opinions. In contrast, the machine learning-based approach involves extracting features from the tweets and training a model using labelled datasets to classify sentiment. In this project, we aim to classify tweets into multiple sentiment categories beyond just polarity. We propose using classes such as "sadness," "happiness," and "love" to categorize tweets based on the emotions they express. Unlike many traditional sentiment analysis methods that use static datasets, our system will fetch live tweets from users that the system follows, ensuring a dynamic and real-time analysis of sentiment on Twitter.

LITERATURE SURVEY

Beakcheol Jang and Jungwon Yoon [2] conducted a comprehensive study comparing the characteristics of data from news sources and social media platforms (SNS). They identified several key differences between the two: **News** focuses on official events and typically maintains consistent topics over time, using a single keyword to identify an event. **SNS**, on the other hand, responds more to personal interests, with topics shifting rapidly from one to another. SNS discussions tend to be more dynamic, with new topics emerging daily, and multiple keywords are often needed to gather relevant data.

Pros: This study provides valuable insights into the differences between news and SNS data and offers good predictive capabilities.

Cons: It requires a robust keyword database for effective analysis.

Fang and Zhang [3] proposed a method for analysing Chinese sentiment phrases by calculating their polarities and strengths using a fuzzy logic approach. This method differs from conventional sentiment analysis by employing probability values to measure sentiment strength.

Pros: The introduction of fuzzy logic allows for more nuanced sentiment analysis.

Cons: This method is language-specific (focused on Chinese) and may not be applicable to other languages.

Mondher and Tomoaki [4] introduced a sentiment analysis approach that classifies tweets into seven distinct classes. The results showed a promising accuracy of 60.2% for multi-class sentiment analysis. However, they suggested that with more optimized training datasets, the performance could be improved.

Pros: The approach supports multi-class sentiment classification.

Cons: The accuracy achieved with the current dataset was not high, and more refined datasets would be needed for better results.

Aldo Hernández and Victor Sanchez [5] proposed a **linear regression model** for sentiment analysis of tweets, specifically to assess negative sentiment in the context of hacking policy discussions. The model uses natural language processing (NLP) to analyse a corpus of tweets and predict the responses of specific groups when sentiments are negative.

Pros: This method is effective for analysing sentiment on specific issues and predicting responses.

Cons: It requires a dataset specific to each issue, limiting its general applicability.

Manju Venugopalan and Deepa Gupta [6] developed a **hybrid tweet sentiment classification model** that incorporates **domain-specific lexicons, unigrams**, and tweet-specific features. By applying machine learning techniques, the model showed a modest improvement in classification accuracy across different domains.

Pros: The use of lexicons and unigrams, combined with machine learning, enhances sentiment classification.

Cons: The model focuses more on the techniques than on the overall results, with modest improvements.

Rincy Jose and Varghese S Chooralil [7] implemented a **real-time, domain-independent sentiment analysis system** using sentiment lexicons like **Sent WordNet** and **WordNet**. The system compared

political sentiment between two politicians using real-time data from Twitter. They also improved accuracy by applying **Word Sense Disambiguation (WSD)** and **negation handling**, which resulted in small but notable improvements in classification accuracy (1% for negation and 2.6% for WSD).
Pros: The use of online dictionaries like WordNet enhances the dataset, and the system is domain-independent. **Cons**: The focus was primarily on election prediction, limiting its broader application.
Cons: Focused on particular issue of election prediction.

PROPOSED METHODOLOGY

Tweet Collection and Sentiment Analysis Process

1. Tweet Collection: The process begins with collecting tweets related to a specific event or topic, such as "#ParisAttacks" or "#Brussels". Since the Twitter API allows the collection of tweets only within a ten-day window, it's crucial to start collecting tweets as soon as an event occurs or a fake tweet begins circulating.

- Tweets can be fetched using the Twitter API (via tools like Twit4j API), which allows for searching specific keywords or hashtags.
- After the tweets are retrieved, they undergo preprocessing (e.g., tokenization, removal of symbols, URLs, and stopwords) to clean the data before further analysis.

2. Feature Extraction: The next step involves extracting features from the tweets. These features are primarily textual and can be used to detect sentiment class. Key features are based on keywords relevant to the sentiment analysis of the tweets.

3. Sentiment Analysis: To determine the polarity of each tweet, sentiment analysis is performed using well-known NLP algorithms, such as the Stanford NLP or Apache OpenNLP algorithms. These algorithms evaluate the sentiment of a tweet and assign a sentiment score (usually a value of -1, 0, or 1), which we refer to as SentiScore.

- -1 indicates a negative sentiment
- 0 indicates a neutral sentiment
- 1 indicates a positive sentiment

After the NLP sentiment scoring, pattern matching techniques (as shown in Table 1) are applied to match specific sentiment patterns in the tweet.

4. Identifying Fake Tweets: Once sentiment analysis is completed, a strategy is applied to detect potentially fake tweets. If two or more analysis methods (such as sentiment analysis and pattern matching) return conflicting or suspicious results, the tweet is flagged as potentially non-genuine or fake.

5. Feature Calculation and Sentiment Classification: The classification of the sentiment is further refined by manually maintained sentiment data dictionaries. The following calculations help to determine the sentiment class:

- Number of positive words in the tweet.
- Number of negative words in the tweet.
- Number of highly emotional positive words: These are words that have a sentiment score greater than or equal to 1.
- Number of highly emotional negative words: These are words that have a sentiment score less than or equal to -1.
- Ratio of emotional words: This is calculated as p(t)p(t)p(t), the proportion of words in the tweet that carry a strong emotional sentiment (either positive or negative).

Based on these metrics, the overall sentiment of the tweet is determined, and the result is compared to the predefined sentiment dictionary to classify the tweet as positive, negative, neutral, or irrelevant.

$$\rho(t) = \frac{PW(t) - NW(t)}{PW(t) + NW(t)}$$

Fake Tweet Detection and Analysis Process

1. **Tweet Preprocessing and Sentiment Scoring:**

○ **PW (Positive Words) and NW (Negative Words)**: These are the total sentiment scores for positive and negative words in a tweet, as returned by the **Sent Score** algorithm. In cases where the tweet does not contain any emotional (positive or negative) words, the **emotion ratio** p(t)p(t)p(t) is set to **0**.

2. **Step-by-Step Fake Tweet Detection:**

(a) Manual Analysis and Filtering Irrelevant Tweets:

- In this step, tweets are manually reviewed to filter out irrelevant ones. For example, if tweets related to “Brussels” are collected due to the Brussels attacks, tweets about vacations or unrelated events in Brussels are removed.
- The goal is to focus only on tweets that are relevant to the specific topic or event under investigation (e.g., the attacks in Brussels) and exclude any off-topic content.

(b) Collect Additional Tweets (Optional Step):

- After manually filtering, additional tweets can be collected using missed keywords or new keywords that are identified during the analysis. For instance, if during the filtering process you identify tweets discussing fake news about the Brussels attacks, you might add keywords related to the fake tweet (e.g., “#FakeBrusselsNews”) to the search query.
- This helps refine the data set by capturing tweets that may have been missed in the initial keyword search.

(c) Categorize Tweets as Fake:

- After filtering and refining the collection, tweets are categorized into fake tweets. Tweets that refer to or propagate a particular fake news story are grouped together. This helps in analysing patterns across tweets and detecting common indicators of fake content.

(d) Identify Users Involved in Fake Tweets:

- Identify all unique users who are involved in sharing or propagating fake tweets. These users become the focus of further analysis in the next steps.
- Collect the most recent 400 tweets (from before the fake tweet) of each of these users. The purpose is to analyse their past behaviour and sentiment before the fake tweet was posted, in order to detect if their writing style or sentiment changes when posting the fake tweet.
- This aspect is novel in the academic literature for fake tweet detection, as it focuses on user behaviour and sentiment shifts over time.

Pattern	Description
atti_cnt	No of users who have favored in weibo
cmt_cnt	No of users who have commented in weibo
repo_cnt	No of users who have reposted in weibo
sent_score	Sentiment score of weibo
pic_cnt	No of pictures posted in weibo
tag_cnt	No of #tag in weibo
mention_cnt	No of @mentioned in weibo
smiley-cnt	No of smileys in weibo
qm_cnt	No of question marks in weibo
fp_cnt	No of first person in the weibo
Length	Length of the Weibo
is_rt	Whether the weibo is repost
Hour	Hour the weibo was posted
Source	How the Weibo was posted

Table 1 Patterns for tweets analysis

The system operates on tweets and tweets are fetched by using Twitt4j API using Java as programming platform. The steps involved in working are as follows,

1. Authenticate
2. Fetch Tweets
3. Preprocess (Tokenization, removal of special symbols and URLs)
4. Read Tweet features such as mention counts, tag counts, smiley, URL, etc.
5. Perform NLP to detect polarity of preprocessed tweet.
6. Match patterns from read features of tweets as shown in table 1.
7. Apply strategies such as time of tweet, username, location and compare this info in the know database.

If polarity is negative and any of the pattern matches or polarity is positive but patterns are matching and strategies returns true then tweet is not genuine.

Mathematical Model

Let S be the closed system defined as,

S = {Ip, Op, A, Ss, Su, Fi}

Where, Ip=Set of Input, Op=Set of Output, Su= Success State, Fi= Failure State and A= Set of actions, Ss= Set of user's states.

- Set of input=Ip={username, password}
- Set of actions =A={F1,F2,F3,F4,F5,F6} Where,
  - F1= Authentication of user
  - F2 =Fetching Tweets from twitter
  - F3 = Compare all Tweets
  - F4 = Influence of negativity spread by tweets
  - F5= Classification on tweets
  - F6= Show Result
- Set of user's states=Ss={registration state, login state, selection of tweets, feature selection, classification, logout}
- Set of output=Op={Show twits analysis results}
- Su=Success state={Registration Success, Login Success, Fetch/Search tweets success}
- Fi=Failure State={Registration failed, Login failed, API failure}

Set of Exceptions= Ex ={Null Pointer Exception while various states, Record Not Found (Invalid Password) state , Null Values Exception while fetching tweets, Limit exhaust while fetching tweets}

SYSTEM ARCHITECTURE

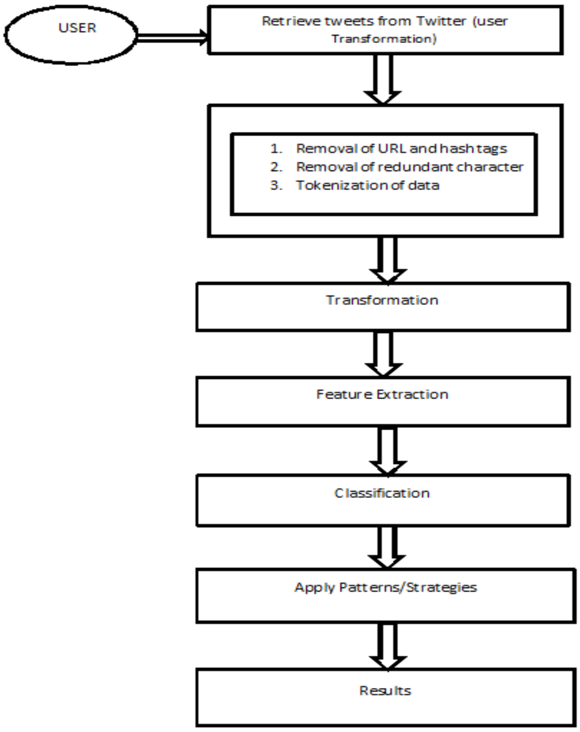

FIG.SYSTEM ARCHITECTURE

ALGORITHM

Project leverages several foundational algorithms commonly used in natural language processing (NLP) and text analysis. Here's a breakdown of how each algorithm functions in this context:

1. NLP Algorithm for Sentiment Analysis:

- NLP, or Natural Language Processing, is a broad field that includes techniques for processing and analyzing human language.
- For sentiment analysis, your algorithm likely applies techniques to determine the "polarity" of tweets, identifying them as positive, negative, or neutral. This could involve various methods like rule-based approaches, machine learning models, or pre-trained language models designed to detect sentiment in text.

2. TF-IDF (Term Frequency-Inverse Document Frequency):

- TF-IDF is a statistical measure used to evaluate the importance of a word in a document relative to a collection of documents.
- It's commonly used in text search and pattern matching, helping to determine if specific terms appear in documents and to rank documents based on their relevance to a query. In your case, TF-IDF could be helping to highlight patterns in tweet content and identify key terms for further analysis.

3. Naïve Bayes Classifier:

- Naïve Bayes is a probabilistic classifier based on Bayes' theorem, often used in text classification because of its simplicity and efficiency.
- Here, it classifies user sentiments into different categories, potentially more nuanced than just positive, negative, and neutral. This could include classes like "very positive" or "slightly negative" based on the likelihood of certain words or phrases appearing in each sentiment category.

Each algorithm plays a unique role, with TF-IDF supporting term relevance and matching, Naïve Bayes facilitating classification, and NLP techniques enabling sentiment analysis.

Own Contribution

As discussed earlier in the paper many of the authors focused on only sentiment analysis of the tweets, some used online tools for that as Senta[1], and some used machine learning approaches as SVM classification[4]. Most of the approaches use offline dataset and it cannot be verified if those tweets are legitimate or has been removed by twitter. Hence we use online twitter API to fetch current tweets and process them. We not only find polarity of tweets but add sentiment classes to it, then by applying patterns and strategies we check if the tweets was genuine, if the tweet was rumor and the user who posted the tweet is legitimate user or not. Many times it may happen that some screen names of genuine users will match in our database but if their tweets is not matched in disrespectful sentiment class or not of negative polarity and pattern matching score is low then they will be ignored

PREDICTED RESULT AND DISCUSSIONS

In the proposed system, based on the query, we will fetch tweets from twitter account using twitter API. The fetched tweets will be subjected for preprocessing. We will then apply the various patterns and strategic algorithms as well as few machine learning algorithms for NLP for supervise the data. The algorithms result i.e. the sentiment and influence will be represented in graphical manner (pie charts/bar charts). The proposed system is more practical than the existing one. This is as a result of we'll be ready to shrewdness the statistics determined from the illustration of the result will have a sway in a very specific field furthermore influence of negativity spread by fake tweets. Many of the existing systems as discussed in literature focuses on only NLP algorithms and resulting in sentiment or polarity of the tweet. Our system extends these results with applying various pattern matching as shown in table 1 along with strategies to detect if the tweets were genuine or not so that other users will know how much to trust such tweets. The system also can be used in predicting anonymous or fake profiles on twitter which can be helpful further to mitigate such false tweets.
Comparative results of existing and proposed system is as following Table 2,

Parameters	**Existing System**	**Proposed System**
Sentiment Analysis	Yes	Yes
Polarity Detection	Somewhat	Yes
Classification	Somewhat	Yes
Pattern Matching	No	Yes
Fake Tweets	No	Yes
Graphical Analysis	No	Yes
User Alerts	No	Yes

Table 2: Comparative Results

With reference to table 2, it is clear that we overcome various problems in existing system and our approach works efficiently.

CONCLUSIONS

The proposed system set out to solve a specific problem of sentiment analysis and genuinely check of Twitter posts. This system proposed a method using knowledge base patterns, strategies and machine learning approaches. These methods are proposed to increase the accuracy of sentiment check for tweets. Patterns can be used to evaluate if the tweets were an influenced rumour or a genuine post by any user. By using API of twitter, it is possible to work on live tweets than to work on offline data. Querying and fetching of particular tweets from twitter are possible by using its API. Finding influence or negativity spread by users can be useful in various analytical tasks.

REFERENCES

1. Mondher Bouazizi and Tomoaki Ohtsuki, "A Pattern-Based Approach for Multi-Class Sentiment Analysis in Twitter," Journal of Computational Intelligence and Neuroscience, vol. 2017, pp. 2169-3536, August 2017.

2. Beakcheol Jang and Jungwon Yoon, "Characteristics Analysis of Data from News and Social Network Services," International Journal of Information and Education Technology, vol. 10, no. 3, pp. 2169-3536, March 2018.
3. Hai Tan and Jun Zhang, "Multi-Strategy Sentiment Analysis of Consumer Reviews Based on Semantic Fuzziness," International Journal of Computer Science and Network Security, vol. 18, pp. 2169-3536, April 2018.
4. Mondher Bouazizi and Tomoaki Ohtsuki, "A Pattern-Based Approach for Multi-Class Sentiment Analysis in Twitter," Journal of Computational Intelligence and Neuroscience, vol. 2017, pp. 2169-3536, August 2017. (Repeated entry)
5. Aldo Hernández and Victor Sanchez, "Security Attack Prediction Based on User Sentiment Analysis of Twitter Data," International Journal of Computer Applications, vol. 131, pp. 22-27, May 2016.
6. Manju Venugopalan and Deepa Gupta, "Exploring Sentiment Analysis on Twitter Data," International Journal of Data Mining and Knowledge Management Process, vol. 5, no. 8, pp. 45-60, August 2015.
7. Rincy Jose and Varghese S. Chooralil, "Prediction of Election Result by Enhanced Sentiment Analysis on Twitter Data using Word Sense Disambiguation," International Journal of Advanced Computer Science and Applications, vol. 6, no. 9, pp. 1-8, November 2015.
8. Anurag P. Jain and Vijay D. Katkar, "Sentiments Analysis of Twitter Data Using Data Mining," Proceedings of the 2015 International Conference on Computing, Communication and Networking Technologies, pp. 978-1-4673-7758-4, December 2015.
9. Gaurav D. Rajurkar and Rajeshwari M. Goudar, "A Speedy Data Uploading Approach for Twitter Trend and Sentiment Analysis using HADOOP," Proceedings of the 2015 IEEE International Conference on Big Data (Big Data), pp. 978-1-4799-6892-3, February 2015.
10. Ahmed Talal Suliman, Khaled Al Kaabi, "Event Identification and Assertion from Social Media Using Auto-Extendable Knowledge Base," International Journal of Advanced Computer Science and Applications, vol. 7, pp. 2161-4407, July 2016.
11. Mohd Fazil and Muhammad Abulaish, "A Hybrid Approach for Detecting Automated Spammers in Twitter," Proceedings of the 2018 International Conference on Communication, Control, Computing and Electronics Engineering (C3E), November 2018.
12. Yan Zhang and Weiling Chen, "Detecting Rumors on Online Social Networks Using Multi-layer Autoencoder," Proceedings of the 2017 International Conference on Information and Communication Technology, pp. 2161-4407, June 2017.