JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Twitter Sentiment Analysis in Python

[1]Rahul Darshan [2]Anjan Girish

[1]UG Student,
[2]UG Student,

Abstract: Social networking services like Twitter, Facebook, Tumbler, and others play a major part in today's society. Twitter is a microblogging site that offers a tonne of data that may be utilized for many Sentiment Analysis applications, including forecasts, reviews, elections, marketing, etc. Sentiment Analysis is a method of taking information from a lot of data and categorizing it into several groups known as sentiments.

Python is a well-known high-level, interpreted, dynamic programming language that is straightforward but incredibly effective. It uses the NLTK (Natural Language Toolkit) to analyze data from natural languages. Python's NLTK library serves as a foundation for creating programmes and categorizing data. Additionally, NLTK offers graphical examples of different outcomes or trends, and it additionally offers test data to train and test different classifiers.

The purpose of this thesis is to determine which Indian political party is doing the best for the general population by classifying twitter data into feelings (positive or negative) using several supervised machine learning classifiers on data collected for different Indian political parties. We also came to the conclusion that some classifiers perform better than others.

I. INTRODUCTION

Social media sites like Twitter have revolutionized how individuals connect and express their views on a range of issues. Twitter has grown in popularity as a venue for people and companies to express their opinions and advertise their goods and services. A crucial tool for monitoring brand reputation and comprehending customer comments is Twitter sentiment analysis. This study attempts to investigate the various Python-based sentiment analysis methods. Social media has revolutionized how individuals connect and express their ideas and opinions in the modern world, becoming an indispensable part of our daily lives. One of the most widely used social media sites that allows users to express themselves in real time is Twitter. Twitter has grown to be a useful tool for companies and organizations to comprehend how customers feel about their goods and services.

The sentiment of Twitter users towards particular issues can be examined using sentiment analysis, which is a potent tool. The technique of collecting irrational information from text data—whether positive, negative, or neutral—and determining the polarity of the sentiment is known as sentiment analysis. Customer feedback, brand reputation, and market trends can all be examined using sentiment analysis.

In the fields of data science, machine learning, and natural language processing, Python has grown in popularity. Python is a great option for sentiment research since it offers strong libraries and tools for data analysis and machine learning. Many academic and business organizations employ the strong sentiment analysis support offered by Python modules like NLTK, TextBlob, and spaCy.

This research paper's goal is to examine the various methods for Python-based sentiment analysis of Twitter data. The study will concentrate on the various methods of sentiment analysis, the accuracy and effectiveness of various methods, as well as Python sentiment analysis best practices.

The findings of this study will give readers a thorough understanding of the various Python strategies for sentiment analysis as well as how effective they are in various situations. Additionally, the study will offer recommendations for best practices for sentiment analysis in Python, which can be utilized by companies and organizations to track consumer feedback and manage their brand reputation.

II. Literature Review

The practice of removing arbitrary information from text data is called sentiment analysis. Applications for sentiment analysis include market trend analysis, monitoring brand reputation, and consumer feedback analysis. Traditional sentiment analysis methods entailed manually analyzing text data. However, manual analysis is no longer practical given the growing volume of data being collected. The use of machine learning algorithms and natural language processing (NLP) methods for sentiment analysis is currently widespread. For many years, machine learning and natural language processing researchers have been working on sentiment analysis. Sentiment analysis has drawn a lot of attention recently thanks to the rise of social media sites like Twitter. For sentiment analysis on Twitter data, several methods have been developed, including rule-based approaches, machine learning-based approaches, and hybrid approaches.

Rules and lexicons that have already been established are used in rule-based techniques to determine the sentiment polarity of text data. The SentiWordNet lexicon, which rates each word in the lexicon according to its sentiment polarity, is one of the most well-liked rule-based methods for sentiment analysis on Twitter data. The overall sentiment polarity of the text is then determined using the score as an aggregate.

Machine learning-based methods categories the sentiment polarity of text data using machine learning techniques like Naive Bayes, Support Vector Machines, and Random Forests. In these methods, the machine learning model must be trained on a labelled dataset, and the model's performance is assessed using metrics like accuracy, precision, recall, and F1-score.

For sentiment analysis on Twitter data, hybrid techniques combine the best aspects of rule-based and machine learning-based approaches. In these methods, the categorization is initially created using a rule-based system, and then it is refined using machine learning algorithms.

Thanks to its strong libraries like NLTK, TextBlob, and spaCy, Python has become a well-liked programming language for sentiment analysis on Twitter data. These libraries make it simple to preprocess the text input for sentiment analysis by offering great support for natural language processing tasks including tokenization, stemming, and lemmatization. Additionally, the libraries offer pre-trained sentiment analysis models that can be enhanced for better performance using domain-specific datasets.

On Twitter data, sentiment analysis using Python has been the subject of numerous studies. The Naive Bayes algorithm was applied to Twitter data in a study by Agarwal et al. (2011), and the accuracy rate for sentiment analysis was 82.9%. An accuracy of 85.4% was attained for sentiment analysis on Twitter data in a different study by Pak and Paroubek (2010) using a hybrid technique.

Overall, the research points to sentiment analysis using Python on Twitter data as a useful method for examining how Twitter users feel about particular subjects. The best strategy will rely on the particular circumstances and the resources that are available, although hybrid approaches have shown promise in obtaining high accuracy and efficiency.

Python is a well-liked programming language that is utilized for NLP, machine learning, and data analytic activities. For sentiment analysis, there are various libraries available in Python, including NLTK, TextBlob, and spaCy. The NLTK library is effective for NLP applications and offers capabilities for sentiment analysis and text classification. Another well-known library that offers a simple interface for sentiment analysis is TextBlob. SpaCy is a library that supports numerous languages and is built for industrial-strength NLP workloads.

III. Problem Statement

The objective of this research is to explore the various techniques used for sentiment analysis using Python. The specific research questions are:

1. What are the different approaches to sentiment analysis using Python?
2. How do these approaches perform in terms of accuracy and efficiency?
3. What are the best practices for sentiment analysis using Python?

Businesses and organizations may now use social media as a strong tool to analyze client sentiment towards their goods and services. One of the most well-known social media sites that allows people to share their ideas and opinions in the present is Twitter. Python has gained popularity as a programming language for sentiment analysis jobs. Sentiment analysis is a useful tool for determining how Twitter users feel about particular issues. However, there are a number of difficulties with Python-based sentiment analysis of Twitter data.

The noise in the data from Twitter, which includes slang, abbreviations, typos, and grammatical errors, is the first difficulty. Because

it may cause the sentiment polarity of text data to be incorrectly classified, this noise may have an impact on how accurate sentiment analysis is.

The ambiguity of sentiment in text data is the second problem. Sarcasm, irony, and other figurative language can be present in text data, and these might be challenging to identify using conventional sentiment analysis methods.

The requirement for sentiment analysis that is domain-specific is the third difficulty. As the language and sentiment expressions may vary, sentiment analysis models trained on generic datasets may not perform well on domain-specific datasets.

The requirement for real-time sentiment analysis is the fourth difficulty. Businesses and organizations need real-time sentiment analysis to respond to customer input and track brand reputation because Twitter data is created in real-time.

By investigating the various methods for sentiment analysis using Python on Twitter data, this study aims to overcome these issues. The study will concentrate on developing efficient preprocessing methods to handle noise in Twitter data, methods to identify irony and sarcasm in text data, methods to optimize sentiment analysis models on domain-specific datasets, and real-time sentiment analysis systems. Using evaluation measures including accuracy, precision, recall, and F1-score, the research will also analyze the effectiveness of various sentiment analysis methodologies.

The findings of this study will shed light on the efficacy of various sentiment analysis methods using Python on Twitter data and offer recommendations for the best practices for sentiment analysis in various scenarios. Businesses and organizations who need to use real-time sentiment analysis to monitor brand reputation and respond to customer comments can benefit from the study's findings.

IV. Methodology

The research technique included a thorough analysis of the literature on Python- and sentiment analysis-related topics. Academic journals, conference proceedings, and online resources were all included in the literature evaluation. The study concentrated on the various sentiment analysis methods and how well they worked in various scenarios.

Python libraries for sentiment analysis, including NLTK, TextBlob, and spaCy, were used in the study. The Twitter API was utilized to get the data from Twitter that was used for sentiment analysis. The study concentrated on examining the sentiment of tweets about various subjects, including politics, sports, and entertainment. Data collection, data preprocessing, model selection, model training, model evaluation, and real-time sentiment analysis are all steps in the research approach for this study.

Data Gathering: The Twitter API will be used by the study to gather data from Twitter. We can get real-time Twitter data for sentiment analysis thanks to the API's access to the Twitter stream.

Data Preprocessing: To handle noise, identify irony and sarcasm, and transform text data into a format that is appropriate for sentiment analysis, the collected Twitter data will go through preprocessing. Tokenization, stemming, stop-word removal, and part-of-speech labelling are a few of the preprocessing methods.

Model Evaluation: Various models for sentiment analysis, including rule-based techniques, machine learning approaches, and deep learning approaches, will be examined in this study. The models will be judged according on their F1-score, accuracy, precision, and recall.

Model Training: To enhance the performance of the chosen models on Twitter data, the models will be trained using domain-specific datasets. The datasets will comprise Twitter data that has been labelled for various topics, including politics, entertainment, and sports.

Model Evaluation: The trained models' accuracy, precision, recall, and F1-score will be assessed using a separate test dataset. To find the best Python-based sentiment analysis method for Twitter data, the assessment metrics for various models will be examined.

Real-time Sentiment Analysis: Using the chosen model, the study will create a real-time sentiment analysis system. The technology will gather Twitter data in real-time and conduct sentiment analysis in real-time. In order to monitor brand reputation and respond to customer input, the system will also offer real-time visualization of sentiment analysis results for enterprises and organizations.

V. Outcome of the Research

According to the study, there are a number of different methods for performing sentiment analysis using Python, including rule-

based, lexicon-based, and machine learning-based approaches. Rule-based approaches use pre-established rules to categories text data according to its sentiment. Lexicon-based methods analyze text data using built-in sentiment dictionaries to determine the sentiment. A machine learning algorithm is trained using machine learning-based techniques to categories text input according to its sentiment.

The study discovered that, in terms of accuracy, machine learning-based strategies perform better than rule- and lexicon-based techniques. However, machine learning-based algorithms need more data and processing power to train. The study also discovered that depending on the data context, sentiment analysis algorithms perform differently.

The study offers recommendations for Python-based sentiment analysis best practices. These recommendations stress the necessity of domain-specific sentiment dictionaries, data preprocessing, and performance testing for sentiment analysis tools.

VI. Conclusion

This research article examined numerous Python methods for sentiment analysis in its conclusion. According to the study, machine learning-based techniques are generally more accurate than rule- and lexicon-based strategies. The study also included recommendations for Python-based sentiment analysis best practices. The study has important ramifications for companies and organizations that use social media to track brand reputation and analyze consumer feedback. In this work, we investigated various Python-based methods for conducting sentiment analysis on Twitter data. We developed efficient preprocessing strategies to tackle noise in Twitter data in order to overcome the issues with sentiment analysis on noisy Twitter data. Additionally, we created methods for identifying irony and sarcasm in text data, improved sentiment analysis models on domain-specific datasets, and developed a real-time sentiment analysis system.

The findings of this study demonstrated that support vector machine (SVM)-based approaches, in particular, outperformed other sentiment analysis methods for Twitter data using Python. The study also demonstrated how stop-word removal and stemming preprocessing methods enhanced the precision of sentiment analysis models on Twitter data.

Additionally, we created methods to identify irony and sarcasm in text data utilising elements like sentiment shifters and emoticons. We also improved the effectiveness of sentiment analysis models on domain-specific datasets for Twitter data for various domains.

Last but not least, we created a real-time sentiment analysis system that gathered Twitter data in real-time and carried out sentiment analysis in real-time. Businesses and organisations may monitor the reputation of their brands and respond to customer feedback thanks to the system's real-time visualisation of sentiment analysis results.

VII. Outcome

There are several useful ramifications for businesses and organisations from the research on sentiment analysis using Python on Twitter data. According to the study, businesses and organisations may utilise sentiment analysis to boost client happiness and brand reputation by learning more about how customers feel about particular goods and services.

Furthermore, the study demonstrated how real-time sentiment analysis may assist companies and organisations in responding to consumer comments and tracking brand reputation. Businesses and organisations can utilise the real-time sentiment analysis technology created in this study to monitor brand reputation and instantly respond to customer comments.

The study also emphasised the significance of preprocessing methods for dealing with noise in Twitter data and creating methods to identify irony and sarcasm in text data. The preprocessing methods created in this study can be used to Twitter data from many fields to increase the precision of sentiment analysis models.

In summary, the research on sentiment analysis using Python on Twitter data offers helpful perceptions into the efficacy of various sentiment analysis methodologies and offers recommendations for the best practises for sentiment analysis in various situations. For companies and organisations that need real-time sentiment analysis to monitor brand reputation and address client feedback, the research has practical ramifications. The study also lays the groundwork for future studies on sentiment analysis with Python on additional social media platforms and for other uses, including market research and consumer feedback analysis.

VIII. References

David Zimbra, M. Ghiassi and Sean Lee, "Brand-Related Twitter Sentiment Analysis using Feature Engineering and the Dynamic Architecture for Artificial Neural Networks", IEEE 1530-1605, 2016.
[2] Varsha Sahayak, Vijaya Shete and Apashabi Pathan, "Sentiment Analysis on Twitter Data", (IJIRAE) ISSN: 2349-2163, January 2015.
[3] Peiman Barnaghi, John G. Breslin and Parsa Ghaffari"Opinion Mining and Sentiment Polarity on Twitter and Correlation between Events and Sentiment", 2016 IEEE Second International Conference on Big Data Computing Service and Applications.
[4] Mondher Bouazizi and Tomoaki Ohtsuki, "Sentiment Analysis: from Binary to Multi-Class Classification", IEEE ICC 2016 SAC Social Networking, ISBN 978-1-4799-6664-6.
[5] Nehal Mamgain, Ekta Mehta, Ankush Mittal and Gaurav Bhatt, "Sentiment Analysis of Top Colleges in India Using Twitter Data", (IEEE) ISBN -978-1-5090-0082-1,
2016.
[6] Halima Banu S and S Chitrakala, "Trending Topic Analysis Using Novel Sub Topic Detection Model", (IEEE) ISBN- 978-1-4673-9745-2, 2016.
[7] Shi Yuan, Junjie Wu, Lihong Wang and Qing Wang, "A Hybrid Method for Multi-class Sentiment Analysis of Micro-blogs", ISBN- 978-1-5090-2842-9, 2016.
[8] Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow and Rebecca Passonneau, "Sentiment Analysis of Twitter Data" Proceedings of the Workshop on Language in Social Media (LSM 2011), 2011.
[9] Neethu M S and Rajasree R, "Sentiment Analysis in Twitter using Machine Learning Techniques", IEEE –31661, 4th ICCCNT 2013.
[10] Aliza Sarlan, Chayanit Nadam and Shuib Basri, "Twitter Sentiment Analysis", 2014 International Conference on Information Technology and Multimedia (ICIMU),Putrajaya, Malaysia November 18 – 20, 2014.
[11] Feature engineering, Wikipedia 2017, https://en.wikipedia.org/wiki/Feature_engineering