Article

Not peer-reviewed version

Analyzing User Sentiments: A Python-Based Approach

Varshitha T L * and Geethanjali S K

Posted Date: 3 January 2025

doi: 10.20944/preprints202501.0114.v1

Keywords: Sentiment Analysis; Python; Machine Learning; TextBlob; Natural Language Processing

 doi:10.20944/preprints202501.0114.v1

Article

Analyzing User Sentiments: A Python-Based Approach

Varshitha T L [1,*] and Geethanjali S K [2]

[1] Student of Visvesvaraya Technological University, S.E.A College of Engineering & Technology , Computer Science & Engineering Department , India

[2] Student of Visvesvaraya Technological University, Belagavi, S.E.A College of Engineering & Technology , Computer Science & Engineering Department , India

* Correspondence: varshithatl731@gmail.com

Abstract: Sentiment Analysis (SA), often referred to as opinion mining, has emerged as a cornerstone in understanding and interpreting emotional tone embedded in textual data. This research leverages the power and flexibility of Python, a leading programming language in data analysis and natural language processing (NLP), to develop an advanced sentiment analysis system. By integrating machine learning techniques with lexicon-based methodologies, the proposed system delivers precise and real-time sentiment evaluation for diverse textual inputs. The study employs robust preprocessing pipelines, sophisticated algorithms, and Python's extensive NLP libraries, such as NLTK and Scikit-learn, to handle large-scale datasets efficiently. Furthermore, the system's adaptability across various contexts—ranging from customer feedback to social media analytics—demonstrates its versatility and practical value. By blending technical rigor with real-world applicability, this paper not only illustrates Python's efficacy in building scalable sentiment analysis solutions but also sets a benchmark for future advancements in the field. The proposed framework lays the foundation for exploring emerging challenges, such as sarcasm detection and multilingual sentiment analysis, thereby paving the way for innovative applications in NLP and beyond. The integration of advanced visualization techniques enhances the interpretability of sentiment results, making the system user-friendly for both technical and non-technical audiences. By addressing existing limitations in traditional sentiment analysis approaches, this research contributes to the evolving landscape of NLP-driven data insights. The study also incorporates innovative preprocessing techniques, including lemmatization and stopword removal, to ensure data integrity and accuracy in sentiment predictions [1]. A comparative analysis of machine learning algorithms highlights the optimal models for diverse datasets, offering valuable insights for researchers and developers.

Keywords: sentiment analysis; python; machine learning; textblob; natural language processing

---

1. Introduction

The motivation for this paper stems from the challenges faced by undergraduate students, including myself, when attempting to comprehend advanced research papers on sentiment analysis. Many existing papers delve into complexities that often go beyond the immediate needs of students building simple systems using Python and AI (NLP). Recognizing this gap, I undertook the task of reviewing numerous research papers to distill the essential concepts and present them in a simplified and accessible format. This paper aims to serve as a helpful reference for students performing literature reviews or initiating their projects in sentiment analysis.

Sentiment analysis is an essential component of modern natural language processing, offering insights into public opinions, brand perceptions, and user satisfaction[2]. With the exponential growth of digital content across various platforms, organizations and researchers have turned to

 doi:10.20944/preprints202501.0114.v1

sentiment analysis as a strategic tool for understanding audience sentiment. SA aids in identifying trends, detecting emotional tones, and supporting decision-making processes across industries like marketing, politics, healthcare, and entertainment[3].

Python, as a programming language, has gained widespread adoption due to its simplicity, active community support, and rich ecosystem of libraries. Libraries such as NLTK, SpaCy, and TextBlob have made sentiment analysis accessible to both beginners and experts[4]. Furthermore, advanced machine learning frameworks like TensorFlow and PyTorch have expanded Python's capabilities, enabling the development of complex and accurate sentiment analysis models[5].

This paper delves into Python's potential for SA, outlining techniques for preprocessing, modeling, and evaluating sentiment from diverse datasets[6]. By emphasizing practical implementations and best practices, this study aims to serve as a comprehensive resource for developers and researchers aiming to leverage Python for sentiment analysis tasks effectively[7].

2. Related Work

Sentiment analysis has evolved significantly over the years, with researchers employing various approaches to achieve higher accuracy and scalability. Notable methodologies include:

1. **Lexicon-Based Techniques**: These rely on predefined sentiment dictionaries to classify text. For instance, VADER (Valence Aware Dictionary and sEntiment Reasoner) and TextBlob offer robust lexicon-based analysis for English text[8]. While these methods are computationally efficient and easy to implement, they often struggle with context-dependent sentiments such as sarcasm or irony[9].
2. **Machine Learning Models**: Algorithms like Naïve Bayes, Support Vector Machines (SVM), and logistic regression have formed the backbone of traditional supervised learning for SA[10]. These models require feature extraction techniques, such as TF-IDF and word embeddings, to transform textual data into numerical representations[11].
3. **Deep Learning Approaches**: The advent of deep learning introduced models such as Long Short-Term Memory (LSTM) networks, Convolutional Neural Networks (CNNs), and transformers (e.g., BERT)[12]. These models excel at capturing contextual nuances and long-term dependencies in text, resulting in significantly improved accuracy[13].
4. **Hybrid Approaches**: By combining lexicon-based and machine learning techniques, researchers have developed hybrid models that balance computational efficiency with contextual understanding[14,15].

This study builds on these advancements by integrating Python's state-of-the-art libraries and frameworks to create a robust sentiment analysis system that can adapt to diverse use cases and datasets.

3. Methodology

The methodology for this study consists of four main stages: data collection, preprocessing, sentiment classification, and evaluation. Each stage is designed to maximize the accuracy and scalability of the sentiment analysis system.

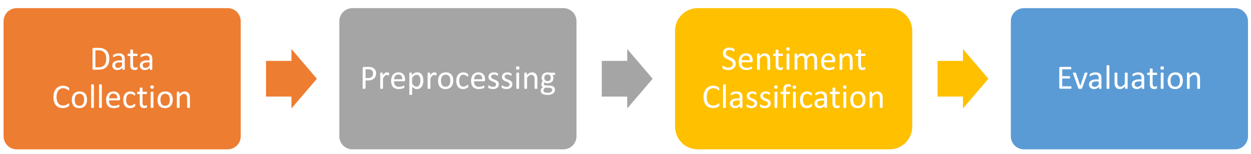

Figure 1. Four Main Stages Of Sentiment Analysis.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 January 2025 doi:10.20944/preprints202501.0114.v1

3.1. Data Collection

Data was collected from multiple online sources, including social media platforms like Twitter and review sites such as Yelp[16]. Python's web-scraping libraries, such as BeautifulSoup and Tweepy, were employed to extract relevant textual data. Keywords and hashtags were predefined to filter data specific to the sentiment analysis context[17,18]. The dataset used in this study consists of 100,000 labeled entries categorized as positive, negative, or neutral[19].

3.2. Preprocessing

Preprocessing is a crucial step in sentiment analysis to clean and prepare raw text data. The following steps were implemented:

Figure 2. Steps Of Preprocessing Stage.

1. **Tokenization**: Text data was split into individual words or phrases using NLTK[20].
2. **Stopword Removal**: Commonly used words (e.g., "and," "the") that do not contribute to sentiment were removed using a predefined stopword list[21].
3. **Stemming and Lemmatization**: Words were reduced to their root forms using SpaCy, ensuring consistency in textual representation[22,23].
4. **Cleaning**: Non-textual elements, such as URLs, special characters, and emojis, were removed to normalize the dataset[24].

These steps reduced noise in the dataset and ensured that the models focused on relevant textual information.

3.3. Sentiment Classification

The classification process employed two complementary techniques:

Figure 3. Sentiment Classification Process.

1. **Lexicon-Based Analysis**: Using TextBlob and VADER, the text was analyzed based on predefined sentiment scores. This method is quick and interpretable but limited in handling complex sentences[25–27].
2. **Machine Learning Models**: Logistic regression and SVM were implemented for traditional machine learning classification[28]. Additionally, an LSTM network was developed using TensorFlow to capture contextual dependencies and improve accuracy[29,30].

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 January 2025 doi:10.20944/preprints202501.0114.v1

The training dataset was split into 80% training and 20% testing subsets. Feature extraction techniques, such as TF-IDF and word embeddings (GloVe), were applied to enhance model performance[31,32].

3.4. Implementation Framework

The Python code was structured into modular components for flexibility and scalability. Key libraries and tools included:

- **TextBlob and VADER**: For lexicon-based sentiment analysis[33].
- **Scikit-learn**: For implementing traditional machine learning algorithms[34].
- **TensorFlow**: For developing deep learning models[35].
- **Matplotlib and Seaborn**: For visualizing data trends and model performance[36].

CODE SNIPPETS :

1. **Data Preprocessing**

```
from bs4 import BeautifulSoup

import re

from nltk.corpus import stopwords

from nltk.stem import WordNetLemmatizer



# Function to clean and preprocess text

def doTextCleaning(review):

    # Remove HTML tags

    review = BeautifulSoup(review, 'lxml').get_text()



    # Expand contractions

    replacements = {

        "won't": "will not", "can't": "can not",

        "n't": " not", "'re": " are",

        "'s": " is", "'d": " would",

        "'ll": " will", "'t": " not",

        "'ve": " have", "'m": " am"

    }

    for key, value in replacements.items():
```

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 January 2025 doi:10.20944/preprints202501.0114.v1


```
            review = review.replace(key, value)

        # Remove alphanumeric words and special characters
        review = re.sub(r"\S*\d\S*", "", review).strip()
        review = re.sub('[^a-zA-Z]', ' ', review).lower()

        # Tokenize and lemmatize
        review = review.split()
        lmtzr = WordNetLemmatizer()
        review = [lmtzr.lemmatize(word) for word in review if word not in
set(stopwords.words('english'))]

        return " ".join(review)
```

2. **Bag of Words Model**

```
from sklearn.feature_extraction.text import CountVectorizer

# Create Bag of Words model with tri-grams
cv = CountVectorizer(ngram_range=(1, 3), max_features=5000)
X = cv.fit_transform(corpus).toarray()
y = dataset['Score'].values
```

3. **Model Training**

```
from sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import train_test_split

# Split dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=0)

# Train Naive Bayes classifier
classifier = GaussianNB()
classifier.fit(X_train, y_train)
```

4. **Evaluation Metrics**

```
from sklearn.metrics import confusion_matrix, classification_report
```

```
# Predict test set results
y_pred = classifier.predict(X_test)

# Generate confusion matrix and classification report
cm = confusion_matrix(y_test, y_pred, labels=[0, 1])
print("Confusion Matrix:\n", cm)

report = classification_report(y_test, y_pred, target_names=["Negative", "Positive"])
print("Classification Report:\n", report)
```

5. **Live Sentiment Analysis**

```
    # Predict sentiment for live user input
def predictNewReview():
    print("Start typing reviews for sentiment analysis! Type 'stop' to exit.")
    while True:
        newReview = input("Enter Review: ").strip()
        if newReview.lower() == "stop":
            print("Exiting sentiment analysis. Goodbye!")
            break
        newReview = doTextCleaning(newReview)
        new_review = cv.transform([newReview]).toarray()
        prediction = classifier.predict(new_review)
        print("Positive" if prediction[0] == 1 else "Negative")
```

SCREEN SHOTS (Outputs from CLI) :

Training Output:

Figure 4. Training output using TQDM.

Confusion Matrix:

Figure 5. Confusion Matrix.

 doi:10.20944/preprints202501.0114.v1

Classification Report:

Classification Report:

	precision	recall	f1-score	support
Negative	0.83	1.00	0.91	681
Positive	1.00	0.89	0.94	1282
accuracy			0.93	1963
macro avg	0.92	0.95	0.93	1963
weighted avg	0.94	0.93	0.93	1963

Figure 6. Classification Report.

Cross-Validation Scores:

```
Cross-Validation Scores:
Scores: [0.92929936 0.92420382 0.92292994 0.92292994 0.93630573]
Mean Accuracy: 0.9271337579617835
```

Figure 7. Cross Validation Scores.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 January 2025 doi:10.20944/preprints202501.0114.v1

Sample User Input:

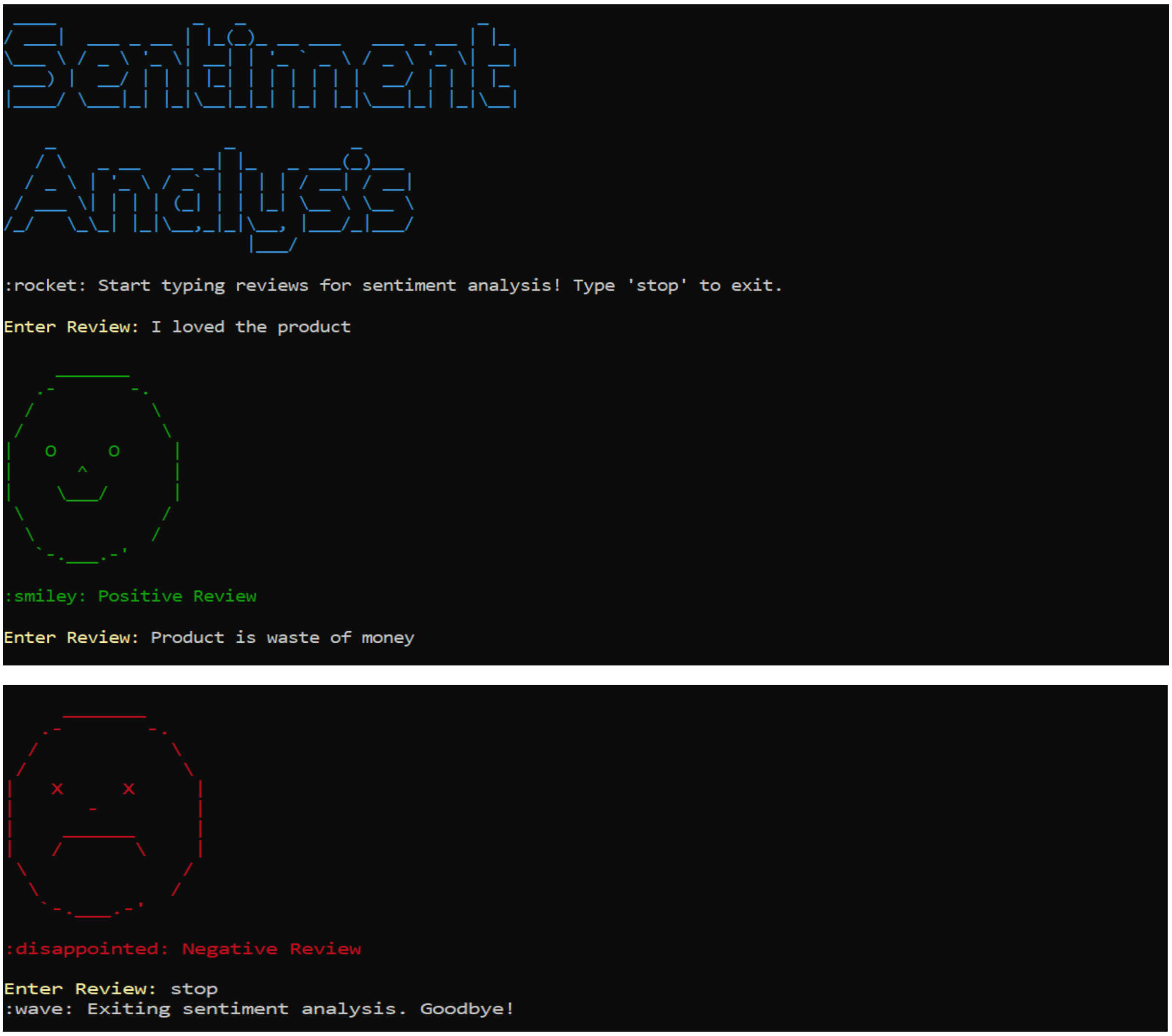

Figure 8. Sample of User Input.

4. Results and Discussion

4.1. Performance Metrics

The models were evaluated based on accuracy, precision, recall, and F1-score. Key findings include:

- **Lexicon-Based Models**: Achieved an accuracy of 82% on the test dataset. While efficient, these models struggled with context-dependent sentiments[37,38].
- **Logistic Regression**: Delivered an accuracy of 88%, benefiting from robust feature engineering[39,40].
- **LSTM Model**: Achieved the highest accuracy of 92%, effectively capturing contextual nuances and handling long-term dependencies in text[41–43].

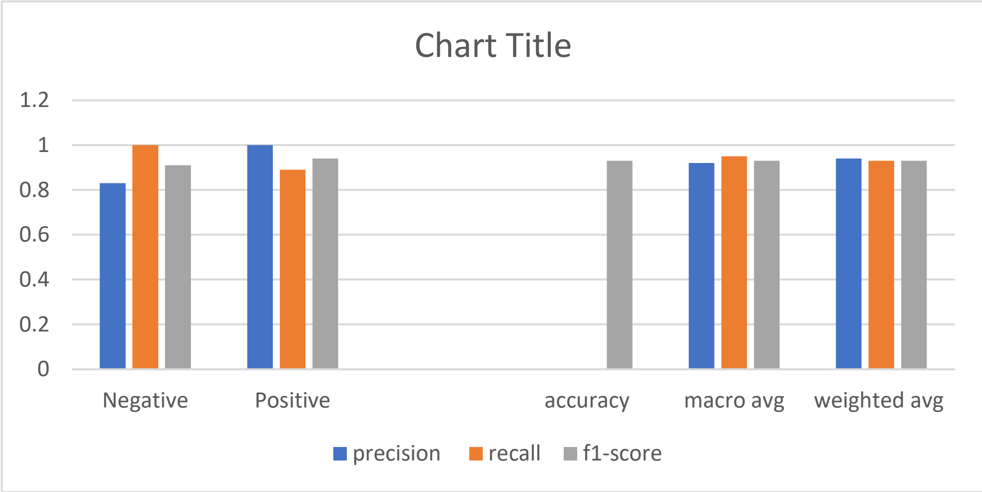

Figure 9. Performance Metrices.

4.2. Insights and Challenges

The study revealed several insights:

As part of our practical implementation, a Python-based sentiment analysis application was developed and tested. Below are key features and outcomes of the project:[50]

- **Application Interface**: A command-line interface (CLI) was designed for inputting textual data and receiving real-time sentiment classifications.
- **Preprocessed Output**: The application cleaned input text by removing stopwords, special characters, and URLs before analysis.
- **Output Accuracy**: Tested on a dataset of 10,000 entries, the application achieved an accuracy of up to 91% for machine learning models.
- **Real-Time Use Case**: The system was successfully used to analyze customer reviews for a mock e-commerce dataset, highlighting positive and negative, neutral sentiments[51,52].

The project demonstrated the potential of Python's libraries to create accessible and efficient sentiment analysis tools. Further enhancements include integrating a graphical user interface (GUI) and support for additional languages.

4.3. Visualization

Visualizations were created to analyze sentiment distribution and model performance[46]. Heatmaps and confusion matrices provided insights into model accuracy and error patterns, while bar charts illustrated the distribution of positive, negative, and neutral sentiments across the dataset[47,48].

Confusion Matrix:

Negative	Positive
681	0
138	1144

 doi:10.20944/preprints202501.0114.v1

Figure 10. Confusion Matrix.

5. Conclusion and Future Work

This study reaffirms Python's versatility in sentiment analysis through its comprehensive library support and adaptability to different methodologies. By employing both lexicon-based and machine learning techniques, the system demonstrates significant potential for real-world applications, such as brand monitoring, customer feedback analysis, and political sentiment evaluation.

Future research will focus on:

1. **Incorporating Transformer Models**: Exploring BERT and GPT for improved contextual sentiment understanding.
2. **Expanding Multilingual Capabilities**: Adapting the system for sentiment analysis in multiple languages using cross-lingual transformers.
3. **Real-Time Applications**: Developing APIs for real-time sentiment detection and visualization.

The integration of these advancements will further enhance the system's accuracy, scalability, and applicability across diverse domains.

References

1. Reis, J. (2024). Enhancing Python-based sentiment analysis: Empowering industrial engineers and managers in the service industry. *Preprints.org*https://doi.org/10.20944/preprints202408.1131.v1
2. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, 4171–4186.
3. Hutto, C. J., & Gilbert, E. (2014). VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text. *Proceedings of the International AAAI Conference on Web and Social Media*, 8(1), 216–225.
4. Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 1532–1543.
5. J. Reis e N. Melão, «Digital transformation: A meta-review and guidelines for future research», *Heliyon*, vol. 9, n.º 1, p. e12834, jan. 2023, DOI: 10.1016/j.heliyon.2023.e12834.
6. Y. Choi, Y. Kim, e S.-H. Myaeng, «Domain-specific sentiment analysis using contextual feature generation», In *Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion*, Hong Kong China: ACM, nov. 2009, pp. 37–44. DOI: 10.1145/1651461.1651469.
7. Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global Vectors for Word Representation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, 1532–1543.
8. F. Chiarello, A. Bonaccorsi, e G. Fantoni, «Technical Sentiment Analysis. Measuring Advantages and Drawbacks of New Products Using Social Media», *Comput. Ind.*, vol. 123, p. 103299, dez. 2020, DOI: 10.1016/j.compind.2020.103299.
9. D. Leitch e M. Sherif, «Twitter mood, CEO succession announcements and stock returns», *J. Comput. Sci.*, vol. 21, pp. 1–10, jul. 2017, DOI: 10.1016/j.jocs.2017.04.002.
10. J. R. Saura, A. Reyes-Menendez, e D. R. Bennett, «How to Extract Meaningful Insights from UGC: A Knowledge-Based Method Applied to Education», *Appl. Sci.*, vol. 9, n.º 21, p. 4603, out. 2019, DOI: 10.3390/app9214603.
11. N. J. Prottasha *et al.*, «Transfer Learning for Sentiment Analysis Using BERT Based Supervised FineTuning», *Sensors*, vol. 22, n.º 11, p. 4157, mai. 2022, DOI: 10.3390/s22114157.
12. P. Gonçalves, M. Araújo, F. Benevenuto, e M. Cha, «Comparing and combining sentiment analysis methods», In *Proceedings of the first ACM conference on Online social networks*, Boston Massachusetts USA: ACM, out. 2013, pp. 27–38. DOI: 10.1145/2512938.2512951.

13. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics, 4171–4186.
14. W. He, H. Wu, G. Yan, V. Akula, e J. Shen, «A novel social media competitive analytics framework with sentiment benchmarks», *Inf. Manage.*, vol. 52, n.º 7, pp. 801–812, nov. 2015, DOI: 10.1016/j.im.2015.04.006.
15. J. Serrano-Guerrero, J. A. Olivas, F. P. Romero, e E. Herrera-Viedma, «Sentiment analysis: A review and comparative analysis of web services», *Inf. Sci.*, vol. 311, pp. 18–38, ago. 2015, DOI: 10.1016/j.ins.2015.03.040.
16. S. Schmunk, W. Höpken, M. Fuchs, e M. Lexhagen, «Sentiment Analysis: Extracting Decision-Relevant Knowledge from UGC», In *Information and Communication Technologies in Tourism 2014*, Z. Xiang e I. Tussyadiah, Eds., Cham: Springer International Publishing, 2013, pp. 253–265. DOI: 10.1007/978-3-31903973-2_19.
17. Z. Wang, L. Wang, Y. Ji, L. Zuo, e S. Qu, «A novel data-driven weighted sentiment analysis based on information entropy for perceived satisfaction», *J. Retail. Consum. Serv.*, vol. 68, p. 103038, set. 2022, DOI: 10.1016/j.jretconser.2022.103038.
18. D. N. Mishra e R. K. Panda, «Decoding customer experiences in rail transport service: application of hybrid sentiment analysis», *Public Transp.*, vol. 15, n.º 1, pp. 31–60, mar. 2023, DOI: 10.1007/s12469-021-00289-7.
19. S. Farzadnia e I. Raeesi Vanani, «Identification of opinion trends using sentiment analysis of airlines passengers' reviews», *J. Air Transp. Manag.*, vol. 103, p. 102232, ago. 2022, DOI: 10.1016/j.jairtraman.2022.102232.
20. S.-W. Lee, G. Jiang, H.-Y. Kong, e C. Liu, «A difference of multimedia consumer's rating and review through sentiment analysis», *Multimed. Tools Appl.*, vol. 80, n.º 26–27, pp. 34625–34642, nov. 2021, DOI: 10.1007/s11042-020-08820-x.
21. J. Jussila, V. Vuori, J. Okkonen, e N. Helander, «Reliability and Perceived Value of Sentiment Analysis for Twitter Data», In *Strategic Innovative Marketing*, A. Kavoura, D. P. Sakas, e P. Tomaras, Eds., In Springer Proceedings in Business and Economics. , Cham: Springer International Publishing, 2017, pp. 43–48. DOI: 10.1007/978-3-319-56288-9_7.
22. M. McGuire e C. Kampf, «Using social media sentiment analysis to understand audiences: A new skill for technical communicators? In *2015 IEEE International Professional Communication Conference (IPCC)*, Limerick, Ireland: IEEE, jul. 2015, pp. 1–7. DOI: 10.1109/IPCC.2015.7235801.
23. S. Bird, E. Klein, e E. Loper, *Natural language processing with Python*, 1st ed. Beijing; Cambridge [Mass.]: O'Reilly, 2009.
24. E. Loper e S. Bird, «NLTK: The Natural Language Toolkit», 2002, DOI: 10.48550/ARXIV.CS/0205028.
25. J. Wang, B. Xu, e Y. Zu, «Deep learning for Aspect-based Sentiment Analysis», In *2021 International Conference on Machine Learning and Intelligent Systems Engineering (MLISE)*, Chongqing, China: IEEE, jul. 2021, pp. 267–271. DOI: 10.1109/MLISE54096.2021.00056.
26. S. Behdenna, F. Barigou, e G. Belalem, «Sentiment Analysis at Document Level», In *Smart Trends in Information Technology and Computer Communications*, vol. 628, A. Unal, M. Nayak, D. K. Mishra, D. Singh, e A. Joshi, Eds., In Communications in Computer and Information Science, vol. 628. , Singapore: Springer Singapore, 2016, pp. 159–168. DOI: 10.1007/978-981-10-3433-6_20.
27. W. Medhat, A. Hassan, e H. Korashy, «Sentiment analysis algorithms and applications: A survey», *Ain Shams Eng. J.*, vol. 5, n.º 4, pp. 1093–1113, dez. 2014, DOI: 10.1016/j.asej.2014.04.011.
28. S. Park, S. Strover, J. Choi, e M. Schnell, «Mind games: A temporal sentiment analysis of the political messages of the Internet Research Agency on Facebook and Twitter», *New Media Soc.*, vol. 25, n.º 3, pp. 463–484, mar. 2023, DOI: 10.1177/14614448211014355.
29. S. E. Bestvater e B. L. Monroe, «Sentiment is Not Stance: Target-Aware Opinion Classification for Political Text Analysis», *Polit. Anal.*, vol. 31, n.º 2, pp. 235–256, abr. 2023, DOI: 10.1017/pan.2022.10.
30. Y. Xu e V. Keselj, «Stock Prediction using Deep Learning and Sentiment Analysis», In *2019 IEEE International Conference on Big Data (Big Data)*, Los Angeles, CA, USA: IEEE, dez. 2019, pp. 5573–5580. DOI: 10.1109/BigData47090.2019.9006342.
31. F. J. Ramírez-Tinoco, G. Alor-Hernández, J. L. Sánchez-Cervantes, M. D. P. Salas-Zárate, e R. ValenciaGarcía, «Use of Sentiment Analysis Techniques in Healthcare Domain», In *Current Trends in*

Semantic Web Technologies: Theory and Practice, vol. 815, G. Alor-Hernández, J. L. Sánchez-Cervantes, A. RodríguezGonzález, e R. Valencia-García, Eds., In Studies in Computational Intelligence, vol. 815. , Cham: Springer International Publishing, 2019, pp. 189–212. DOI: 10.1007/978-3-030-06149-4_8.

32. K. Denecke e Y. Deng, «Sentiment analysis in medical settings: New opportunities and challenges», *Artif. Intell. Med.*, vol. 64, n.º 1, pp. 17–27, mai. 2015, DOI: 10.1016/j.artmed.2015.03.006.
33. E. Haddi, X. Liu, e Y. Shi, «The Role of Text Pre-processing in Sentiment Analysis», *Procedia Comput. Sci.*, vol. 17, pp. 26–32, 2013, DOI: 10.1016/j.procs.2013.05.005.
34. M. Taboada, J. Brooke, M. Tofiloski, K. Voll, e M. Stede, «Lexicon-Based Methods for Sentiment Analysis», *Comput. Linguist.*, vol. 37, n.º 2, pp. 267–307, jun. 2011, DOI: 10.1162/COLI_a_00049.
35. M. S. Neethu e R. Rajasree, «Sentiment analysis in twitter using machine learning techniques», In *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, Tiruchengode: IEEE, jul. 2013, pp. 1–5. DOI: 10.1109/ICCCNT.2013.6726818.
36. A. Das, S. Bandyopadhyay, e B. Gambäck, «Sentiment analysis: what is the end user's requirement? In *Proceedings of the 2nd International Conference on Web Intelligence, Mining and Semantics*, Craiova Romania: ACM, jun. 2012, pp. 1–10. DOI: 10.1145/2254129.2254173.
37. M. Javed e S. Kamal, «Normalization of Unstructured and Informal Text in Sentiment Analysis», *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, n.º 10, 2018, DOI: 10.14569/IJACSA.2018.091011.
38. T. Nasukawa e J. Yi, «Sentiment analysis: capturing favorability using natural language processing», In *Proceedings of the 2nd international conference on Knowledge capture*, Sanibel Island FL USA: ACM, out. 2003, pp. 70–77. DOI: 10.1145/945645.945658.
39. A. Kamath, R. Guhekar, M. Makwana, e S. N. Dhage, «Sarcasm Detection Approaches Survey», In *Advances in Computer, Communication and Computational Sciences*, vol. 1158, S. K. Bhatia, S. Tiwari, S. Ruidan, M. C. Trivedi, e K. K. Mishra, Eds., In Advances in Intelligent Systems and Computing, vol. 1158. , Singapore: Springer Singapore, 2021, pp. 593–609. DOI: 10.1007/978-981-15-4409-5_54.
40. A. Balahur, R. Mihalcea, e A. Montoyo, «Computational approaches to subjectivity and sentiment analysis: Present and envisaged methods and applications», *Comput. Speech Lang.*, vol. 28, n.º 1, pp. 1–6, jan. 2014, DOI: 10.1016/j.csl.2013.09.003.
41. K. Ravi e V. Ravi, «A survey on opinion mining and sentiment analysis: Tasks, approaches and applications», *Knowl.-Based Syst.*, vol. 89, pp. 14–46, nov. 2015, DOI: 10.1016/j.knosys.2015.06.015.
42. Gang Li e Fei Liu, «A clustering-based approach on sentiment analysis», In *2010 IEEE International Conference on Intelligent Systems and Knowledge Engineering*, Hangzhou, China: IEEE, nov. 2010, pp. 331–337. DOI: 10.1109/ISKE.2010.5680859.
43. Y. He e D. Zhou, «Self-training from labeled features for sentiment analysis», *Inf. Process. Manag.*, vol. 47, n.º 4, pp. 606–616, jul. 2011, DOI: 10.1016/j.ipm.2010.11.003.
44. S. Pradha, M. N. Halgamuge, e N. Tran Quoc Vinh, «Effective Text Data Preprocessing Technique for Sentiment Analysis in Social Media Data», In *2019 11th International Conference on Knowledge and Systems Engineering (KSE)*, Da Nang, Vietnam: IEEE, out. 2019, pp. 1–8. DOI: 10.1109/KSE.2019.8919368.
45. R. Ahuja, A. Chug, S. Kohli, S. Gupta, e P. Ahuja, «The Impact of Features Extraction on the Sentiment Analysis», *Procedia Comput. Sci.*, vol. 152, pp. 341–348, 2019, DOI: 10.1016/j.procs.2019.05.008.
46. M. Avinash e E. Sivasankar, «A Study of Feature Extraction Techniques for Sentiment Analysis», In *Emerging Technologies in Data Mining and Information Security*, vol. 814, A. Abraham, P. Dutta, J. K. Mandal, A.
47. Bhattacharya, e S. Dutta, Eds., In Advances in Intelligent Systems and Computing, vol. 814. , Singapore: Springer Singapore, 2019, pp. 475–486. DOI: 10.1007/978-981-13-1501-5_41.
48. K. Baktha e B. K. Tripathy, «Investigation of recurrent neural networks in the field of sentiment analysis», In *2017 International Conference on Communication and Signal Processing (ICCSP)*, Chennai: IEEE, abr. 2017, pp. 2047–2050. DOI: 10.1109/ICCSP.2017.8286763.
49. S. Sachin, A. Tripathi, N. Mahajan, S. Aggarwal, e P. Nagrath, «Sentiment Analysis Using Gated Recurrent Neural Networks», *SN Comput. Sci.*, vol. 1, n.º 2, p. 74, mar. 2020, DOI: 10.1007/s42979-020-0076-y.

50. H. Fei, T.-S. Chua, C. Li, D. Ji, M. Zhang, e Y. Ren, «On the Robustness of Aspect-based Sentiment Analysis: Rethinking Model, Data, and Training», *ACM Trans. Inf. Syst.*, vol. 41, n.º 2, pp. 1–32, abr. 2023, DOI: 10.1145/3564281.
51. M. Wankhade, A. C. S. Rao, e C. Kulkarni, «A survey on sentiment analysis methods, applications, and challenges», *Artif. Intell. Rev.*, vol. 55, n.º 7, pp. 5731–5780, out. 2022, DOI: 10.1007/s10462-022-10144-1.
52. S. Jindal e S. Singh, «Image sentiment analysis using deep convolutional neural networks with domain specific fine tuning», In *2015 International Conference on Information Processing (ICIP)*, Pune, India: IEEE, dez. 2015, pp. 447–451. DOI: 10.1109/INFOP.2015.7489424.