Sentiment Analysis of Tweets on Social Issues using Machine Learning Approach

Salma GUEMMOUT

(*) : FSJES de Tanger - Université Abdelmalek Essaâdi

Abstract— Since the emergence of Web 2.0, there has been an increasing interest in capturing the spontaneous and real-time opinions of Internet users. This vast amount of opinion data can be accessed using web mining tools, which continually gather information. Several websites have specialized in collecting opinions in specific domains (e.g., movie reviews), and Internet users have become accustomed to consulting these opinions and ratings when making purchasing decisions for technical products or hotel reservations. Opinions have become valuable to Internet users, leading to the development of various applications and services, creating a virtuous cycle where individuals are encouraged to express their opinions and gain recognition for providing relevant insights that others follow. However, this data also holds significance for brands and research firms, as they aim to understand the aggregated sentiment of the crowd. Brands, often worried about the potential harm of negative blog comments on their reputation, are concerned about their online identity and seek to comprehend the expectations and criticisms of Internet users. Consequently, techniques to capture these evaluations from Internet users have seen significant advancements, ranging from simple positive or negative comment counts to more in-depth analysis of comment content. This document aims to outline the detailed steps involved in sentiment analysis using machine learning on a given dataset.

Keywords: Sentiment Analysis, Machine Learning, Deep Learning, Opinion Mining, Natural Language Processing (NLP), Polarity, Subjectivity.

1 Introduction

The Internet, a global system of interconnection, serves as a bridge between billions of devices and people worldwide. The rapid growth of social networks has led to an exponential increase in users and digital content [1]. This expansion provides opportunities for individuals with diverse skills and knowledge to share their experiences and wisdom with one another. Numerous websites, such as Yelp, Wikipedia, and Flickr, leverage the Internet's power to assist users in making informed decisions.

The term "Sentiment Analysis" was initially coined by Nasukawa and Yi in 2003 [1]. It refers to the process of determining the subjectivity polarity (positive or negative) and polarity strength (strongly positive, mildly positive, weakly positive, etc.) of a given review text, essentially capturing the writer's opinion. Turney's groundbreaking work on Sentiment Analysis [2] utilized an unsupervised approach to classify review data into positive and negative classes. However, this paper focuses on analyzing sentiments in a different type of dataset: financial news, while establishing a correlation between the sentiments expressed in financial news and stock market variations.

Moreover, there are websites that facilitate user consultations with professionals, with investment being a particularly popular topic. Established companies like Goldman Sachs and Lehman Brothers have been offering investment advice for over 150 years. In the age of the Internet, independent analysts and retail investors worldwide can collaborate through online platforms. Seeking Alpha and Stock-Twits are two examples of financial social media platforms centered around the stock market, enabling users to connect with information, each other, and grow their investments [2]. Financial social media platforms bring together individuals, companies, and organizations to generate ideas and share information with one another. These platforms provide an abundance of unstructured data (Big Data) that can be integrated into the decision-making process. Such Big Data serves as a valuable source of real-time estimation due to its high frequency of creation and low-cost acquisition. Sentiment analysis, a widely used method, helps evaluate the sentiments of social media users towards a particular subject. Data mining is the most popular approach employed to conduct sentiment analysis.

The objective of sentiment classification is to determine the polarity of sentences extracted from review texts. For example, document [2] showcases the effectiveness of applying machine learning techniques to classify whether a film is successful or not, based on people's opinions and criticisms [5]. This analysis enables relevant organizations to gauge public sentiment towards films, whether positive or negative. In the realm of behavioral finance, investor sentiment refers to the excessive optimism or pessimism displayed by investors, independent of objective economic factors. Numerous studies have been conducted to measure investor sentiment, employing various types of measures: direct declarative measures (questionnaires), direct measures from the Internet, indirect measures, and exogenous measures.

Doms and Morin (2004) provide an explanation for the "sentimental" nature of the media, attributing it to three channels. Firstly, the media disseminate information about the economic world and expert opinions in this domain. Secondly, the content of media outlets serves as signals regarding the overall state of the economy. Lastly, the media potentially influences the formation of consumer expectations. Through an empirical study conducted in the United States between March 1978 and June 2003, these authors validate the influence of the media on the University of Michigan's confidence index, a widely recognized direct measure of investor sentiment. They demonstrate that people's moods are more reactive to the tone and volume of economic news than its actual content, even after controlling for economic fundamentals. Similarly, Tims et al. (1989) analyze a random sample of articles published between 1977 and 1988 in major American newspapers, focusing on economic topics. They conclude that analyzing the content and tone (favorable or unfavorable) of these articles can predict changes in the University of Michigan confidence index. The authors note that the media not only report economic news but also shape its meaning, effectively disseminating public sentiment about the economy. Lastly, Johnson and Tversky (1983), in the field of cognitive psychology, demonstrate through experimental research that reading a newspaper excerpt with very negative or positive content influences individuals' perception of risk frequencies, regardless of any direct link between the article's content and the risk being evaluated. Their findings support the hypothesis that individuals tend to make judgments consistent with their current mood, even when the subject of judgment is unrelated to the cause of their mood. These results elucidate why positive and negative media content can influence investor sentiment.

Sentiment analysis involves the process of determining the underlying opinion, judgment, and emotion behind natural language expressions. When individuals leave online reviews, comment on brands, or respond to market research, their ratings inherently carry sentiment, be it positive, negative, or neutral. When consumers share their opinions by typing them into a text box, this textual data can be transformed into categorical data, such as "positive," "negative," or "neutral." By categorizing all the responses, companies can compile survey reports to gain an overview of the sentiments and opinions of the respondents.

Sentiment analysis is a part of text mining, which involves extracting meaning from various textual sources, including survey responses, online reviews, or social media comments. A sentiment score is assigned based on the expressed sentiment, such as (-1) for negative and (+1) for positive, using natural language processing (NLP).

Sentiment analysis proves highly valuable when dealing with large volumes of textual data, enabling the extraction of information and generalization. While humans can comprehend the emotions conveyed in individual statements, processing vast amounts of comments from multiple sources in a reasonable time frame is beyond our capabilities. Sentiment analysis tools provide accurate and unbiased reporting, delivering a comprehensive and cohesive score and verdict in a matter of clicks. Through the obtained scores, companies can understand the emotions their brand evokes within a given population, whether it be happiness, sadness, anger, or a more neutral sentiment. Armed with this information, organizations can develop an action plan to reinforce or positively shift these sentiments.

2 Methodology

In this paper, we use the Natural language processing and Machine learning classification algorithm to analyze the sentiments as positive, negative, and neutral of people's opinions/emotions. This workflow diagram explains the complete process of the proposed study.

1. Data Preprocessing: Preprocess your dataset to clean and prepare the text data for analysis. This typically involves steps like removing special characters, lowercasing the text, removing stop words, and tokenizing the text into individual words or n-grams.

2. Feature Extraction: Convert the preprocessed text data into numerical features that can be used by the GMM classifier. There are several feature extraction techniques you can use, such as bag-of-words, TF-IDF (Term Frequency-Inverse Document Frequency), word embeddings (e.g., Word2Vec or GloVe), or document embeddings (e.g., Doc2Vec or BERT embeddings). Choose an appropriate feature extraction method based on your dataset and requirements.

3. Split the Dataset: Split your dataset into training and testing sets. The training set will be used to train the GMM classifier, while the testing set will be used to evaluate its performance.

4. GMM Training : Train the GMM classifier using the training set and the extracted features. The GMM model is a generative probabilistic model that represents the data as a mixture of Gaussian distributions. You can use libraries such as scikit-learn in Python to create and train the GMM classifier.

5. Model Evaluation: Once the GMM classifier is trained, evaluate its performance on the testing set. Calculate metrics such as accuracy, precision, recall, and F1-score to assess how well the classifier is predicting the sentiment of the test data.

6. Prediction: After evaluating the model, you can use the trained GMM classifier to predict the sentiment of new, unseen text data. Preprocess the new data using the same steps as in the preprocessing stage, extract the features, and pass them through the trained GMM classifier to obtain sentiment predictions.

3 SENTIMENT ANALYSIS PROCEDURE: MATERIALS AND METHODS

In preparation for conducting sentiment analysis, it is crucial to format the data appropriately and extract the relevant features of the sentiments [17]. To accomplish this task, the following steps should be followed:

1. Remove special characters: Remove any special characters, symbols, or punctuation marks from the text data. These characters may not contribute significantly to sentiment analysis and can be safely removed. You can use regular expressions or string manipulation functions to achieve this.

2. Lowercase the text: Convert the text to lowercase to ensure uniformity. This step helps in treating uppercase and lowercase versions of the same word as identical. Most sentiment analysis algorithms consider the lowercase representation of words.

3. Tokenization: Split the text into individual words or tokens. Tokenization breaks down the text into meaningful units, which can be further processed. You can use libraries like NLTK (Natural Language Toolkit) or spaCy in Python for tokenization.

4. Remove stop words: Remove common words that do not carry much sentiment or meaning, such as articles, prepositions, and conjunctions. These words, known as stop words, can be filtered out to reduce noise in the data. Libraries like NLTK provide predefined stop word lists that you can use.

5. Stemming or Lemmatization: Reduce words to their base or root form. Stemming and lemmatization are techniques used to normalize words and reduce inflectional forms to a common base. For example, "running" and "runs" can both be reduced to "run" using stemming or lemmatization. This step helps in reducing the dimensionality of the data and treating similar words as the same. NLTK and spaCy provide functions for stemming and lemmatization.

6. Handle contractions: If your dataset contains contractions like "can't" or "won't," it's important to expand them to their full forms. For example, "can't" can be expanded to "cannot" and "won't" to "will not". This step ensures that sentiment analysis algorithms can interpret the words correctly. You can create your own mapping or use existing libraries or functions to handle contractions.

7. Additional cleaning: Depending on your dataset, you may need to perform additional cleaning steps specific to your data. For example, removing URLs, handling emoticons or emojis, or replacing specific patterns or expressions. These steps can be customized based on the requirements of your sentiment analysis task.

These techniques collectively contribute to cleaning and enhancing the textual data for more effective analysis and machine learning model performance.

C. Feature Extraction Methods

Following the pre-processing phase, the data was partitioned into a "training subset" and a "testing subset" in a 3:1 ratio for training and testing purposes, respectively. Feature extraction methods were then applied to the training subset. These same feature extraction techniques were also applied to the testing data during the classification phase. The extraction of relevant features is crucial for sentiment analysis. Some of these features include:

1. Bag-of-Words (BoW): One common approach is to use the bag-of-words model. Convert each text document into a numerical vector representation, where each dimension represents a unique word in the corpus vocabulary. The value in each dimension can be the frequency of the word in the document (term frequency) or a binary indicator of whether the word is present or not.

2. TF-IDF (Term Frequency-Inverse Document Frequency): TF-IDF is another popular technique that takes into account not only the frequency of words in a document but also their importance in the entire corpus. It assigns higher weights to words that appear frequently in a document but less frequently in the corpus overall. This technique helps to identify important and distinguishing terms.

3. Word Embeddings: Word embeddings represent words as dense, low-dimensional vectors, capturing semantic and contextual information. Popular word embedding models like Word2Vec, GloVe, or fastText can be used to obtain word embeddings. Each word in the text is replaced with its corresponding embedding vector, resulting in a fixed-length representation for the entire document.

4. Document Embeddings: Document embeddings represent the entire text document as a dense, fixed-length vector. Techniques like Doc2Vec or BERT embeddings can be used to obtain document embeddings. These models consider the context and semantics of the entire document, capturing a higher level of information than word-level representations.

5. N-grams: Instead of considering single words, n-grams are sequences of adjacent words. By including n-grams (bi-grams, tri-grams, etc.) in addition to individual words, you can capture the context and local dependencies within a text. N-grams can be used along with the bag-of-words or TF-IDF approaches.

6. Feature Selection: Depending on the dimensionality and sparsity of the feature space, you may consider applying feature selection techniques to select a subset of the most informative features. Methods like chi-square test, mutual information, or feature importance from machine learning models can help identify the most relevant features.

7. Normalization: Normalize the extracted features to ensure they have similar scales. Common normalization techniques include z-score normalization (subtracting the mean and dividing by the standard deviation) or min-max scaling (scaling values between a specified range).

4 MACHINE LEARNING ALGORITHMS FOR SENTIMENT CLASSIFICATION

This section provides the necessary details regarding the machine learning classifiers utilized in this study for tweet classification. There are several machine learning algorithms commonly employed in sentiment analysis, including:

•**GMM :** stands for Gaussian Mixture Model. It is a probabilistic model that represents a dataset as a mixture of multiple Gaussian distributions. In the context of sentiment analysis, GMM can be used as a classification algorithm to predict the sentiment of text data.

•**Naive Bayes**: Naive Bayes is a simple yet effective probabilistic classifier. It assumes that features are conditionally independent given the class label. Naive Bayes classifiers work well for text classification tasks, including sentiment analysis.

•**Neural Networks**: Deep learning techniques, particularly recurrent neural networks (RNNs) and their variants like LSTM and GRU, have shown promising results in sentiment analysis. These models can capture the sequential nature of text and learn complex representations that capture sentiment information effectively.

•**Support Vector Machines (SVM):** SVM is a powerful algorithm for binary and multi-class classification. SVMs aim to find the optimal hyperplane that separates the data into different classes while maximizing the margin. SVMs can handle high-dimensional feature spaces and perform well for sentiment analysis tasks.

•**Random Forest:** Random Forest is an ensemble learning method that combines multiple decision trees. It operates by constructing a multitude of decision trees and outputs the class that is the mode of the classes predicted by individual trees. Random Forests can handle non-linear relationships and capture complex interactions between features.

•Logistic Regression: Logistic regression is a supervised learning algorithm used for binary classification tasks. In the given code, logistic regression is used to train a model on the keyword features extracted from the tweets. Logistic regression models the relationship between the features and the binary sentiment labels.

•Gradient Boosting Algorithms: Gradient Boosting algorithms, such as XGBoost and LightGBM, build an ensemble of weak learners (usually decision trees) in a sequential manner. They optimize a loss function by iteratively adding models that correct the mistakes of the previous models. Gradient Boosting algorithms are known for their high predictive power and are often used in sentiment analysis tasks.

These algorithms are commonly employed in sentiment analysis tasks and provide various approaches for classifying sentiment in textual data.These algorithms are commonly employed in sentiment analysis tasks and provide various approaches for classifying sentiment in textual data.

5 PERFORMANCE EVALUATIONS PARAMETERS

When evaluating the performance of a sentiment analysis model, you can consider various evaluation parameters to assess its effectiveness. Here are some commonly used performance evaluation parameters:

1. Accuracy: Accuracy measures the overall correctness of the model's predictions by calculating the ratio of correctly predicted instances to the total number of instances

2. Precision: Precision evaluates the proportion of correctly predicted positive instances out of all instances predicted as positive. It measures the model's ability to avoid false positives.

3. Recall: Recall (also known as sensitivity or true positive rate) calculates the proportion of actual positive instances that are correctly identified by the model. It measures the model's ability to avoid false negatives.

4. F1-score: The F1-score is the harmonic mean of precision and recall. It provides a balanced measure that considers both precision and recall, especially in imbalanced datasets

5. Confusion Matrix: The confusion matrix provides a tabular summary of the model's predictions against the actual labels. It includes counts of true positives, true negatives, false positives, and false negatives, which can be used to calculate various performance metrics.

6. ROC Curve and AUC: Receiver Operating Characteristic (ROC) curve illustrates the trade-off between true positive rate and false positive rate at various classification thresholds. The Area Under the ROC Curve (AUC) summarizes the overall performance of the model.

7. Precision-Recall Curve: The Precision-Recall curve shows the trade-off between precision and recall at different classification thresholds. It is particularly useful when dealing with imbalanced datasets.

8. Cross-Validation: Cross-validation techniques, such as k-fold cross-validation, can provide more robust estimates of model performance by evaluating the model on multiple train-test splits.

9. Macro/Micro/Average Metrics: In multi-class classification scenarios, macro-average, micro-average, or weighted average precision, recall, and F1-score can be calculated to account for class imbalances.

10. Specificity: Specificity measures the proportion of actual negative instances correctly identified by the model. It is the complement of the false positive rate.

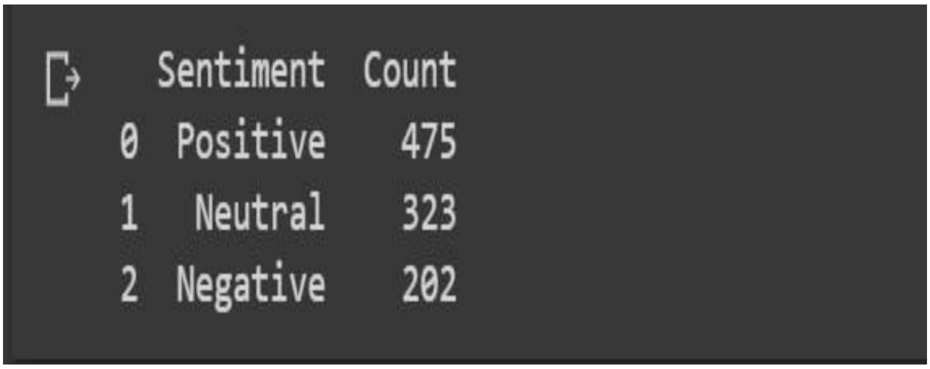
These evaluation parameters help assess different aspects of the sentiment analysis model's performance. Depending on the specific requirements of your analysis and the characteristics of your dataset, you can select the most appropriate metrics to evaluate the model's accuracy, precision, recall, F1-score, and overall effectiveness.

6 EXPERIMENTAL RESULTS AND ANALYSIS

Opinion Mining, also known as sentiment analysis, enables the automatic analysis of textual data to identify and emphasize various opinions expressed about a particular subject, such as a brand, a press article, or a product. This process involves extracting sentiments or attitudes from the text to gain insights into how people perceive and evaluate the subject matter.

Opinion Mining, also referred to as sentiment analysis, enables the automated analysis of textual data to identify and highlight diverse opinions expressed about a specific subject, such as a brand, a press article, or a product. In this project, the proposed approach follows the steps outlined in Section 4 for analyzing sentiments on Twitter data. Twitter, being the most popular microblogging platform, was chosen as the social network for data collection. Given an idea about Chatgbt and education, and what is people’s opinion about the bot, Twitter provides a wealth of real-time events, news, and opinions regarding the apparition of Chatgbt. To gather relevant data, tweets were collected using the Python programming language and specific keywords (#Education, #Chatgbt).

To initiate Twitter analysis, the primary requirement is to develop a Twitter application. This application facilitates analytical operations by establishing a connection between your console and Twitter via the Twitter API. Our project's objective revolves around analyzing the response of individuals to Chatgbt and its impact on education . This analysis will be founded on the principles of sentiment analysis, artificial intelligence, and deep learning.



```
   Sentiment  Count
0   Positive    475
1    Neutral    323
2   Negative    202
```

Figure 1 : NUMBER OBTAINED OF TWEETS OF EACH TYPE OF SENTIMENT

And that’s the representation of number of tweets of each type of sentiment:

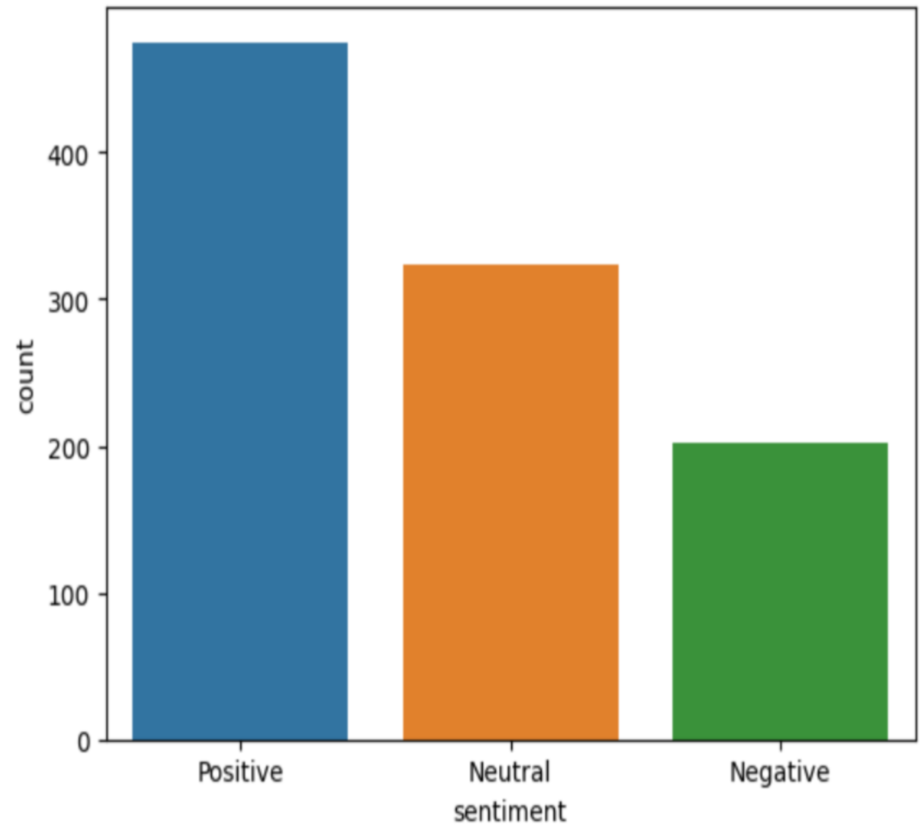
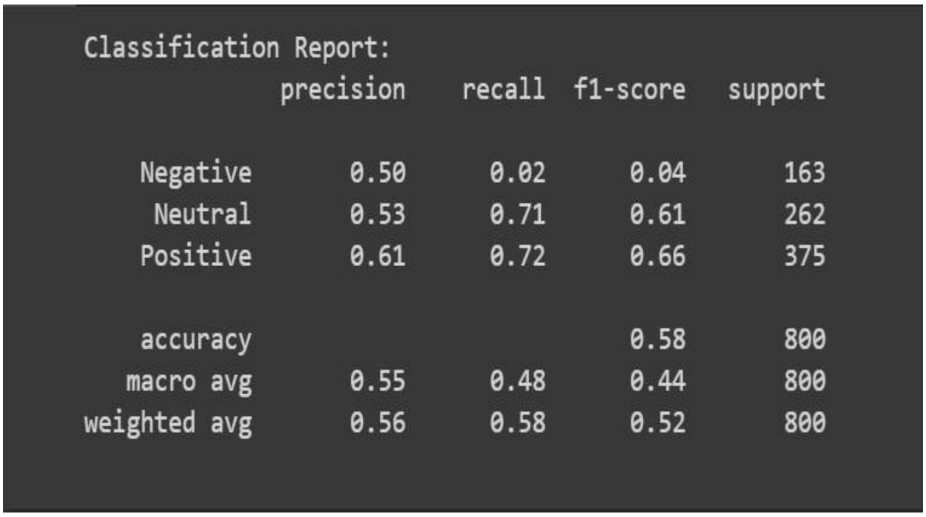


Figure 2 GRAPH OF NUMBER OBTAINED OF TWEETS OF EACH TYPE OF SENTIMENT

•Training machine-learning model:

The model is trained using the Logistic Regression algorithm.. In the code, it is utilized to train a sentiment analysis model on the keyword features extracted from the tweets. The trained logistic regression model can then be used to make predictions on new data or evaluate its performance using evaluation metrics such as accuracy, precision, recall, or F1-score.



```
Classification Report:
              precision    recall  f1-score   support

    Negative       0.50      0.02      0.04       163
     Neutral       0.53      0.71      0.61       262
    Positive       0.61      0.72      0.66       375

    accuracy                           0.58       800
   macro avg       0.55      0.48      0.44       800
weighted avg       0.56      0.58      0.52       800
```

Figure 3 : RESULTS OBTAINED USING THE BAG OF WORDS MODEL

7 Evaluation

Based on the classification, here are the interpretation of the performance evaluation metrics:

1. Precision: Precision measures the accuracy of the positive predictions made by the model. In this case, the precision for the positive class is 0.61, indicating that out of all the tweets predicted as positive, 61% of them are correctly classified.

2. Recall: Recall calculates the proportion of actual positive instances correctly identified by the model. The recall for the positive class is 0.72, which suggests that the model correctly identifies 72% of the actual positive tweets.

3. F1-score: The F1-score is the harmonic mean of precision and recall, providing a balanced measure that considers both precision and recall. The F1-score for the positive class is 0.66, indicating a reasonable balance between precision and recall.

4. Support: The support refers to the number of instances in each class in the test data. In this case, there are 375 instances of positive tweets

5. Accuracy: Accuracy represents the overall correctness of the model's predictions. The accuracy of 0.58 suggests that the model correctly predicts the sentiment for 58% of the tweets in the test data.

8 DISCUSSION AND FUTURE WORK

Additional steps for improving the model's performance could involve collecting more training data, exploring different feature representations, or trying alternative algorithms. Indeed, here are some additional steps we can take to improve the model's performance:

1. Increase Training Data: Gathering more labeled training data can help the model learn more diverse patterns and improve its performance. An expanded and balanced dataset may help address class imbalances and provide more representative samples for each sentiment class.

2. Feature Engineering: Experiment with different feature representations or engineering techniques to capture more relevant information. For text data, you can try techniques such as using n-grams, word embeddings (e.g., Word2Vec or GloVe), or contextual embeddings (e.g., BERT or GPT) to enhance the representation of the text and capture contextual information.

3. Hyperparameter Tuning: Optimize the hyperparameters of your model to find the best configuration for your specific task. This can include parameters related to the algorithm itself (e.g., number of components in GMM) or parameters related to feature extraction or regularization techniques. Use techniques like grid search or random search to explore different combinations efficiently.

4. Ensemble Methods: Consider using ensemble methods to combine predictions from multiple models. Ensemble methods, such as bagging or boosting, can help reduce bias and variance in the model's predictions and improve overall performance.

5. Model Selection: Experiment with different algorithms or models for sentiment classification, such as random forests, gradient boosting algorithms (XGBoost, LightGBM), or deep learning architectures (LSTM, CNN, transformer models). Each algorithm has its own strengths and may perform differently depending on the dataset.

6. Address Class Imbalance: If the dataset has imbalanced classes, employ techniques like oversampling, undersampling, or generating synthetic samples to balance the classes. This can help prevent the model from being biased towards the majority class and improve its ability to capture sentiment patterns in the minority classes.

7. Regularization Techniques: Apply regularization techniques such as L1 or L2 regularization, dropout, or early stopping to prevent overfitting and improve generalization performance. Regularization can help control model complexity and improve its ability to generalize to unseen data.

8. Cross-Validation: Perform cross-validation to assess the model's performance more robustly and to ensure that the results are not specific to a particular train-test split. Cross-validation provides a more reliable estimate of the model's performance on unseen data.

9 CONCLUSION

Sentiment Analysis can be approached using three main methods: lexical-based, machine learning-based, or a hybrid approach that combines both. However, the lexicon-based approach has a limitation where the effectiveness of sentiment classification relies heavily on the size of the lexicon or dictionary. As the lexicon grows larger, this approach becomes more prone to errors and time-consuming. In this study, we developed a sentiment analysis system that analyzes the sentiment expressed in text data. We followed a systematic approach, starting with the acquisition of a labeled dataset containing text samples and sentiment labels. The data was then preprocessed using various NLP techniques, including tokenization, stemming, removal of stopwords, and handling special characters.

To extract relevant features from the preprocessed data, we employed techniques such as Bag of Words, Word2Vec, and FastText. We optimized the parameters of our machine learning models using techniques like GridSearchCV to ensure optimal performance. Several machine learning classifiers, including Naïve Bayes, Support Vector Machines, Logistic Regression, Decision Trees, Random Forest, and Gradient Boosting, were trained on the labeled dataset. We evaluated the performance of these models using precision, accuracy, recall, and F-score metrics. After comparing the results from different models and representations, we identified the best-performing model. This model demonstrated superior performance in sentiment analysis tasks, providing valuable insights into the expressed sentiment in the text data. Overall, our study highlights the effectiveness of utilizing labeled datasets,preprocessing techniques, feature extraction methods, model optimization, and evaluation metrics in developing a robust sentiment analysis system. The insights generated from our system can aid in understanding and analyzing sentiment across various domains and applications.

REFERENCES

[1] G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. (references)

[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[3] I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271– 350.

[4] K. Elissa, “Title of paper if known,” unpublished. [5] R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.

[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[7] M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.

[8] Jaspreet Singh, Gurvinder Singh and Rajinder Singh.: Optimization of sentiment analysis using machine learning classifiers. Department of Computer Science, GuruNanak Dev University, Amritsar, India Singh et al. Hum, (2017).

[9] David Pereira Coutinho and Mario A. T. Figueredo.: An Information Theoretic Approach to Text Sentiment Analysis. Instituto Superior de Engenharia de Lisboa, (2015).

[10] Wael Korani and Malek Mouhoub.: Sentiment Analysis of Serious Suicide References in Twitter Social Network. Department of Computer Science, University of Regina. (2016).

[11] Natarajan R., Barua G., Patra M.R.: Distributed Computing and Internet Technology. Lecture Notes in Computer Science (2015).

[12] Gonzalez-Marron D., Mejia-Guzman D., Enciso-Gonzalez A.: Exploiting Data of the Twitter Social Network Using Sentiment Analysis. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, (2017).

[13] Jaspreet Singh, Gurvinder Singh and Rajinder Singh: Optimization of sentiment analysis using machine learning classifiers, Department of Computer Science, Guru Nanak Dev University, Amritsar, India

[14] Natarajan R., Barua G., Patra M.R: Distributed Computing and Internet Technology, ICDCIT, Lecture Notes in Computer Science, (2015).

[15] Gonzalez-Marron D., Mejia-Guzman D., Enciso-Gonzalez A: Exploiting Data of the Twitter Social Network Using Sentiment Analysis, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, (2017).

[16] Furqan Rustam, Imran Ashraf, Arif Mehmood, Saleem Ullah and Gyu Sang Choi: Tweets Classification on the Base of Sentiments for US Airline Companies, Licensee MDPI, Basel, Switzerland, (2019).