SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V11I3P111

Volume 11 Issue 3, 106-114, March 2024

Original Article

An Improved Vectorization-Based Emotion Detection Using Tuned Inverse Document Frequency Approach

R. Vanitha[1], K. Sudharson[2], N. Sathish Kumar[2], S. Gunasundari[3]

[1]*Department of Cyber Security, Jerusalem College of Engineering, Tamil Nadu, India.*
[2]*Department of Artificial Intelligence and Machine Learning, R.M.D. Engineering College, Tamil Nadu, India.*
[3]*Department of Computer Science and Engineering, Velammal Engineering College, Tamil Nadu, India.*

[1]*Corresponding Author : vanithasenthilemail@gmail.com*

Received: 08 January 2024 Revised: 14 February 2024 Accepted: 12 March 2024 Published: 31 March 2024

Abstract - *Emotion analysis of social media content has garnered significant attention due to its potential to reveal valuable insights into people's feelings and opinions. This study is motivated by the need to understand better the emotions individuals express when posting their views on social media. The objective is to explore and compare the effectiveness of two machine learning methods, a Twin Support Vector Machine (TWSVM) and a novel approach called Tuned Inverse Document Frequency (TUNED-IDF) vectorizer, in accurately detecting emotions from textual data. To achieve this objective, the research process involves first applying the TWSVM algorithm, which examines the factors influencing emotions and their connection to the dependent variable. Next, our innovative TUNED-IDF vectorizer converts the textual content into numerical representations, leveraging its properties to improve accuracy in emotion analysis. The findings of this study showcase the remarkable performance of the TUNED-IDF approach, achieving an impressive accuracy level of 94.4%, surpassing existing methods in emotion detection. By employing this method, the research successfully predicts people's emotions with higher precision and efficacy than traditional machine learning models. The significance of this research lies in its contribution to the field of emotion analysis, particularly in the context of social media. Understanding the emotions conveyed in online communication is crucial for various applications, such as sentiment analysis, market research, and public opinion monitoring. The insights gained from this study offer valuable opportunities for better comprehension of individuals' sentiments in the digital age and lay the groundwork for enhanced emotion analysis techniques.*

Keywords - *Vectorization, Tuned-IDF, Data mining, Classification, Supervised learning.*

1. Introduction

Data analytics is a powerful technique that involves using algorithms combining deep learning, statistical analysis, and distributed databases to discover patterns within large datasets [2]. Meanwhile, data mining, an emerging field in computer engineering, focuses on using intelligent algorithms to extract knowledge from gathered information and present it in a comprehensible structure for further use.

This process, known as "Knowledge Discovery in Database" (KDD), includes various stages such as data administration, preparation, modelling, prediction techniques, relevance measurements, complexity examination, post-processing, visualization, and live updating. Contrary to its name, data analysis is not solely about gathering information directly; rather, it entails deriving patterns and insights from vast amounts of data, encompassing activities like data collection, mining, storage, modelling, statistical analysis, computerized recommendation systems, intelligent systems (e.g., machine learning), and business analytics [19].

Emotion analysis is crucial in determining the underlying intent behind people's expressions, particularly in reviews or community responses. For example, sentiment analysis on product or brand feedback can help improve quality marketing strategies, target specific customer groups, and ultimately boost revenue [1].

This process categorizes input sentiments as positive, neutral, or negative, offering valuable insights. This study focuses on calculating accuracy and F1 scores, using regression analysis to predict outcomes based on numerical input variables and coefficients. Non-target parameters are treated as independent variables, while the objective parameter is considered dependent. Graphs are utilized to represent the relationship between non-target and target variables. Additionally, the study explores the random forest technique, which involves multiple decision trees merging to form a forest. Each tree contributes predictions; the final result is determined by averaging the predictions [20].

Customer reviews are critical in establishing reputation, standards, and evaluation in a digital commerce mall. The study examines data from millions of customer datasets related to a specific brand, mainly focusing on Amazon Alexa's viewpoints. Overall, this research delves into data analytics, data mining, and emotion analysis, offering valuable insights and applications for various domains, including digital commerce and customer feedback assessment.

2. Related Works

Tan et al. [3] looked into and developed ways for evoking consumer emotions. When it comes to product reviews and extracting customer emotions via them, features and opinions on a product are essential. He uses a dataset of camera reviews in his study. The first step consists of information extraction utilizing natural language processing techniques such as NER and Dependency parser, then applying the learned knowledge to assess or grasp the new data. Many problems in the dataset must be corrected, including dependence faults, thoughts and sentiments, and other features. The frame detection approach starts with the separation of raw material, a lexicon for the system, and then moves on to representation, which entails evaluating a set of sources. The results are pretty decent with typical F1 levels.

When reviewing vast amounts of data, Bashir et al. [4] developed a content processing strategy, which is crucial in today's world. A comprehensive understanding of large amounts of data emerges when separating concepts into vocabulary, which produces high-quality metadata for content analysis. Word embeddings extract semantic commonalities from a set of words in a vector representation. Lock the semantically related word in the term to include it in a vector space. Subject modeling, which uses LDA to enhance the critical text search in a given document, is used here. This post's thought matrix technique collects the more diverse ideas and forms the expressions through them.

Dey et al. [5] demonstrated a method for interpreting emotional and perspective content in Twitter tweets, using the results to predict the occurrence of events. They look at various characteristics and methodologies, such as pos tagger, and perform polls to see which model is best for emotional interpretation using Naive-Bayes and SVM classifiers to construct and test the systems. The qualities that become highly significant to text or in a Twitter database to anticipate the user's thoughts are initially collected using the TF-IDF approach, which exposes the characteristics crucial to a page or in a collection of Twitter posts. It establishes the significance of a word in a phrase. For increased reliability, background subtraction expands to a bigram or trigram framework and NLP methods like POS tagger and bag of words model.

Atmaja et al. [6] reviewed current advances in question classification and the strategies employed the queries into two categories depending on the user's behavior and the query logs. Train and asses using ML techniques and compare the results, whether supervised or unsupervised. The amount of data generated every day and query logs are growing and coping with it. Big data techniques are offered and incorporated into query logs to analyze and display the data.

Alslaity et al. [7] investigated many text mining and text analysis algorithms. NLP applies a range of methodologies to assess the direction and relevance of passages and words. The initial strategy for identifying purposes in a phrase or blog is to divide the phrases and give n-gram loads to each word. The purposes are matched to the scales and predefined vocabulary to categorize the paragraph's or blog's purpose. The following approach investigated is "Part of Speech" (POS) tagging, a strategy for identifying additional phrases in a document by combining intriguing traits with lexicons. The POS tags provide all the necessary English language to separate the statements. Fit them in with other sentences to see how they connect. The TWSVM is a supervised learning approach that builds and tests models using the information from the previous two techniques. The TWSVM is a hyper-plane technique that separates two items and describes them based on their distance.

Kumar et al. [9] used NLP approaches to extract emotional information from data in an AI system. Emotional intelligence is still in its early stages, having passed through the most well-known developmental stage. In this case, AI can recognize pictures, sounds, and other objects in addition to text. The pre-computed dataset is divided into happy and unpleasant tweets on various emotions that we encounter regularly. Emotional classification employs a large amount of data, subsequently used for attribute analysis at a more acceptable level. The sentiments are physically recognized and then trained using supervised classifiers. The predicted results suggest that, while constrained to a defined set of parameters, these prediction models are not as successful as human labeling.

By merging the word2vec methodology with the TF-IDF, Sun et al. [10] presented a unique way to identify and eliminate features from phrases. The earlier technique employs a more distinct lexical subset to help categorize material more correctly. Builds Word2vec using a bundle of terms and skip-gram models. The skip-gram system learns and expects unexpected phrases based on the present expression, whereas word embedding guesses words based on obtained features. In this technique, the sum of the word2vec character limit in a document classification merges with the TDIDF weighted score. This technique uses vectors to define the phrases and compare the results of TDIDF without lexical elements. They employ Word2vec for training and assessing with the TWSVM classifier model, which results in enhanced correctness.

Shelke et al. [11] have outlined the most common text classification approaches. Fine-tune the text classifiers to fit into any model, classify text more correctly, and categorize it more generically since they are domain-oriented. The goal is to organize the classifiers hierarchically to improve their performance; however, this method has not gained much support due to unsatisfactory results. Because the proposed approach is hierarchical, the first stage requires manually categorizing and labeling data with a high accuracy score using expert systems. Parts trained with hierarchical levels prefer to choose just the best classification from the complete set of classifiers. One such structure is the neural network, training several input parameters to create a single actual data output (image, text, audio).

Acheampong et al. [12] have developed a strategy for automatic text categorization related to dynamically categorizing information based on their unique semantics and syntax, which is given or established by a machine learning algorithm based on supervised learning. Feature extraction is the primary method for training and assessing a system using a classifier. Multivariate regression frameworks, in which the collecting and processing of information or an array contains the frequency of each word and its regularity, are recommended because automatic identification has many difficulties, including unstructured data, working with numerous characteristics, and determining the best classifier for the dataset. The multiword model can assist in the correction of some pre-processing errors. Applying LSI and multiword models to reduce synonymous phrases and their recurrence. Use classifiers such as TWSVM to choose data features, albeit with limitations, such as the inability to conduct multilayer categorization.

Darokar et al. [13] submitted a study to discover plagiarised material in the articles to be analyzed using text similarity. This technique uses the n-gram and vector space models to tokenize articles. It uses the pos tagger and other LTK tools to classify plagiarised works and establish the source of the duplicated information. Instead, tokens lemmas are used to build the dictionary of words, convert each document into a vector space, and translate it into a unigram or bigram model. Then, the closeness information or coefficient is calculated by comparing the n-gram count obtained by applying a merge method between the source and the plagiarised material using vector n-gram. It includes the sorted lemmas, with the n-gram count obtained by merging the original and the plagiarised material. The final product is rather good, with just minor content detection improvements needed.

Goswami et al. [14] looked at query classification algorithms and developed them. You may improve your query results by correctly interpreting a user's search phrase. However, companies must first grasp the consumer's aim to enhance user pleasure. This article focuses on the campaign and search tool information to determine query context, an online business purpose when a person accesses a website to employ a commercial activity. A business inquiry is issued, and a list of all products that will be bought during the next few decades is made. Search Engine Result Pages (SERPs), which result from a user's choice or search word, are used to determine query meaning. The click-through rate is affected by various factors, including the type of material requested by the consumer. Divide the customer's search phrase into three categories: user-defined, navigational, and instructive, monitoring the customer's activity.

Jagadishwari et al. [15] have looked at how information gathered from a database might help users manage data by answering questions-using compression and clustering methods to the data obtained for the material connected to the user's query. Word compression is one of the tactics employed in this research. By determining the distance between phrases or the vector space of the articles, aggregate and group records to produce the similarity measure. The subjects are then examined topically for query forwarding phrases to the articles. They developed a specialized technique and modeling for hierarchical clustering to minimize the number of words in a document. The results were from a two-technique model that categorized many social media feeds and bulletins using subject assessment and word reduction methods. Finally, investigate the impact of content grouping and the consequences of compression methods.

Ho et al. [16] have shown how attributes might impact or decide how words are classified-converting the content of each article to a vector array to vector space for classification. The word selection strategy is critical in this scenario since it decides if a text belongs to a given category. Construct the confusion matrix once the dataset has been developed and evaluated using the feature-selected language. The single-labeled data is collected, and specific groupings, as well as multi-labeled groups, are deleted. The typeface's size, shape, and characters significantly impacted the messages.

Paula et al. [17] proposed a query generation mechanism based on the user's interactions as recorded in client logs. The primary goal of studying user logs is to uncover relationships between search phrases and objects logged. The vast majority of internet search engine queries are insignificant. They do not give enough information to judge the importance of a report. However, including the URL in the data may extract client sessions from which they browsed the inquiry. The searches have now been linked or connected with the content keywords spoken in the talks. As a result, the matching and presenting appropriate brief queries for lengthy questions.

3. Materials and Methods

Individuals complete items and purchases worldwide by looking at item reviews and the ubiquity of the item on the lookout. Audits are invariably emotional and are dependent on the item type examined. Audits must be tweaked and their exact significance removed to have a good outline of the survey and its undeniable relevance, which is a massive difficulty for any nostalgic examination initiatives [18].

Audits benefit many people, from clients to the companies who sell them; thus, they give a lot of weight. One of the most critical tasks for an eCommerce shop is maintaining its online reputation. Usually, it takes a lot of effort to earn a good reputation, but it takes minimal effort to lose it. Item reviews are the most effective approach to keep their winning streak going. Since the internet has evolved into a family thing, reviews and information sources have modified the online market's orientation.

The suggested system uses data from customer audits on Amazon Alexa, which includes ratings and feedback and data from verified questionnaires on the item [8]. Using TWSVM and TUNED-IDF vectorizer computations, evaluate their accuracy. TUNED-IDF calculation for highlight extraction with the help of a check vectorizer creates a jargon of words that can be encoded into another audit dataset to retrieve the vocabulary because the company's primary focus is data recovery and text mining [19].

Following highlight extraction, the arbitrary woods classifier is used for expectation and calibration, resulting in F1 scores for the audit that are much higher for positive names. The sensation of the survey predicts excellent and unfavorable marks and a plot of a Chi-square chart with the words with the most notable events or the element eliminated terms.

3.1. System Model

Figure 1 depicts the system architecture and the method for carrying out the process. The data is organized mainly into categories depending on the scoring objects. The scores range from 1 to 5, with 1 indicating a score higher than 3 and 0 indicating a score of 3 or less than 3. The count vectorizer fetches a word's lexicon and counts the number of words found. The extracted terms can be encoded or transformed into a new language for usage in another dataset or text.

The TWSVM classifier is fitted to the data to calculate computing efficiency (Janjua et al., 2021). The logistic technique uses the count vectorizer's recovered lexicon as input, with built-in packages for computing correlations between variables and ensuring accuracy. Use the TUNED-IDF vectorizer to extract the features. Frequency Counts and compute Inverse Document Frequency weights separately, using the product to determine the word's significance [20]. The random forest classifier can predict the F1 Score using the bag-of-words or n-gram framework. Use A Chi-square test plot to plot the relationship between the label and its occurrence.

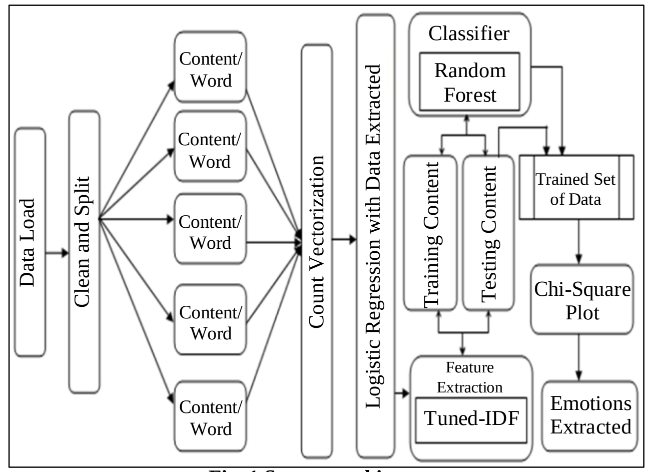

Fig. 1 System architecture

The techniques used in the study include linear regression, TUNED-IDF vectorizer, and random forest classifier. The count vectorizer examines the character count and produces the content's lexicon before processing the comments. First, the count vectorizer method analyses each phrase in the article and tells the user its significance. Next, we use the lexicon generated from the dataset to generate lexicons for additional reviews. After that, the vocabulary developed by the count vectorizer is exposed to TWSVM. Finally, the correlation between the independent and dependent variables is investigated, with the regression being multivariate or straightforward depending on the dataset's independent variables. The dependent variable or binary element is the favorably connected column, comprising 1 and 0 for favorable and unfavorable tags [21].

In contrast, the independent variable is the verified comments section, which is forecasting as a consequence. The TUNED-IDF vectorizer approach is a machine-learning technique that extracts features from a data collection's most frequently occurring phrases. Each phrase is included in the thesaurus, and each symbol will be translated into a grid format in the characteristic-derived index [22]. The TUNED-IDF weights provide floating numbers, with the highest number matching the collection's most common phrase-the algorithm used by the TUNED-IDF vectorizer and how the number of terms used to extract deciding features.

A random forest algorithm is a decision tree that categorizes a target variable using a tree structure. We use this method to test the TUNED-IDF vectorizer's efficacy by forecasting and fine-tuning it. It generates a report that includes the percentages of labels that are negative and positive, as well as their F1 ratings. The N-gram model separates the trigrams from the other retrieved features [23]. Next, we use the chi-square test model to calculate the chi coefficient by displaying graphs. Finally, we show the most often occurring trigram tokens and their occurrence throughout the review to determine which word appears the most frequently [24]. After customizing the random forest classifier, the proper testing collection tag quantity is given for the training set to project the review tags, and the conclusion will be either 1 or 0. (Pos or Neg).

3.2. Twin Support Vector Machine

Twin Support Vector Machine (TWSVM) is a machine learning algorithm used in text classification, particularly emotion classification, in this research. It is known for its high accuracy and efficiency in handling data in a lower dimension. The main objective of TWSVM is to evaluate the relationship between dependent and independent components by comparing and contrasting them. In sentiment analysis, the dependent part refers to the sentiment labels (positive or negative), while the independent parts are the features extracted from the text (words or phrases) [25]. The algorithm uses a kernel to determine the relationship between different data points. It establishes connections between pairs of data points as if they were in a higher-dimensional space [26]. This approach allows TWSVM to find a hyperplane that effectively separates the different classes, such as positive and negative reviews.

Algorithm

```
Begin
TWSVM = TWSVM()
TWSVM.fit([T1, T2])
a= TWSVM.transform([T1]).toarray()
b= TWSVM.transform([T2]).toarray()
col = TWSVM.get_feature_names()
df1 = pd.DF(x, col= col, id= ["T1"])
     df2 = pd.DF (y, col= col, id= ["T2"])
df = pd.concat([df1,df2])
df["tag"] = ["T1", "T2"]
df
End
```

The algorithm's role in the research is to classify the sentiment of the item reviews. Here is how TWSVM is applied in the study:

1. Initialization: The TWSVM algorithm is initialized.
2. Data Fitting: TWSVM is trained using the training data (T1 and T2). The algorithm learns the relationship between the features extracted from the reviews and their corresponding sentiment labels.
3. Transformation: After training, TWSVM transforms the training data (T1 and T2) into numerical vectors using the learned hyperplane. The transformed vectors represent the features of the reviews.
4. Concatenation: The transformed vectors are then concatenated into a data frame (df1 and df2) for further analysis.
5. Label Assignment: The data frame is labelled with "T1" and "T2" to distinguish between the two sets of reviews.
6. Result: The final data frame (df) contains the transformed features along with their assigned sentiment labels ("T1" or "T2").

The TWSVM algorithm plays a vital role in accurately classifying the sentiment of the item reviews. By transforming the textual data into numerical vectors and finding an effective hyperplane, TWSVM efficiently separates positive and negative emotions [27]. It enables the research to gain valuable insights into the opinions expressed by customers, aiding decision-making in the eCommerce domain. Additionally, TWSVM's ability to work efficiently in lower dimensions saves computational time, making it a suitable choice for large-scale sentiment analysis tasks [28].

3.3. Vectorizer

Vectorizer is a crucial component in the research as it converts textual data into numerical vectors that machine learning algorithms can process. In this study, the researchers utilize the TUNED-IDF vectorizer to extract features from the item reviews and transform them into a numerical representation. The TUNED-IDF vectorizer combines the concepts of Term Frequency (TF) and Inverse Document Frequency (IDF) to determine the importance of words in a document collection. The vectorizer calculates a score for each term based on its frequency in a document and occurrence across the entire dataset [29].

3.4. Tuned IDF

A textual vectorizer called Tuned-Inverse Document Frequency turns material into a vector used. Tuning and document frequency are two notions that are incorporated. The tuning of a phrase relates to how many times it appears in a publication. The tuning approach aids in determining the importance of a word in a sentence. Every word in the dataset is represented as a vector, with the row indicating the number of documents and the column showing the number of distinct phrases found throughout all texts. Document frequency refers to the amount of data that contains a specific term. The number of times a word appears in a text indicates how common it is. The inverse document frequency, which minimizes the value of a phrase if its occurrences are scatterers throughout all texts, determines the power of a word.

We calculate the Tuned-IDF with the following Equation 1:

$$Tidf_a = log\left(\frac{vi}{df_a}\right) \tag{1}$$

The Tuned-IDF score for the term ‘a’ is Tidfa, dfa for the volume of information including the term ‘a,’ and vi for the whole volume of information. The bigger the DF of a phrase, the lower the IDF for a word. Because log(1) is 0, the Tuned-IDF is 0 when the count of dfa = vi, indicating that the word appears in all texts. If in doubt, preserve this sentence in the final consonant group to avoid giving too much information.

As the name indicates, the Tuned-IDF rating is just a multiplication of the tuning frequency vector by its IDF. Therefore, we calculate using the following Equation 2:

$$R_{a,b} = Tv_{a,b} \times idf_a \tag{2}$$

Tva,b is the tuning frequency for a term ‘a’ in document b, and idfa is the IDF score for a phrase ‘a’, where Ra,b is the score for a word ‘a’ in document b, Tva,b is tuning frequency for term an in document b, and idfa is IDF score for term a.

The component extraction in the TUNED-IDF is calculated by applying loads to the words with the most notable or most minor event, and the loads with the highest worth have the smallest atonements in the report.

Algorithm

Begin

Def Tidf(term, inallDocs):
vi =0
for
terminallDocs:
if term.lower() inallDocs[doc].lower().split():
vi =vi +1
if vi> 0:
return1.0+log(float(len(vi)/inallDocs)
else
return1.0

End

The provided algorithm calculates the TUNED-IDF score for a given term in a collection of documents. The TUNED-IDF score represents the importance of a term in the dataset, taking into account both its frequency within individual documents and its occurrence across all documents.

Here is a step-by-step explanation of the algorithm:

1. Start: The algorithm begins.
2. DefTidf(term, inallDocs): The algorithm defines a function called DefTidf, which takes two parameters – “term” (the word for which the TUNED-IDF score is calculated) and “inallDocs” (the collection of all documents).
3. vi = 0: The variable “vi” is initialized to zero. It will be used to count the number of documents in which the term appears.
4. for the term in allDocs: The algorithm loops through each document in the collection.
5. if term.lower() in allDocs[doc].lower().split(): The algorithm checks if the given term (converted to lowercase for case insensitivity) is present in the current document (also converted to lowercase for matching).
6. vi = vi + 1: If the term is found in the document, the "vi" counter is incremented by one, indicating that the term is present in one more document.
7. if vi > 0: The algorithm checks if the term appears in at least one document.
8. return 1.0 + log(float(len(vi) / inallDocs): If the term appears in at least one document, the algorithm calculates the TUNED-IDF score using the formula: 1.0 + log(len(vi) / inallDocs), where “len(vi” is the total count of documents containing the term, and “inallDocs” is the total number of documents in the collection. The log function is used to scale the score.
9. else: If the term does not appear in any document, the algorithm returns a default value of 1.0, indicating that the term’s significance is neutral in the dataset.
10. End: The algorithm ends.

In summary, the provided algorithm calculates the TUNED-IDF score for a given term in a collection of documents. The score reflects the importance of the term in distinguishing the dataset’s content and is based on both its frequency in individual documents and its overall occurrence across all documents [30].

4. Results and Discussion

The findings achieved utilizing the suggested approach for emotion forecasting using TWSVM and TUNED-IDF are discussed in detail in this section. First, we produce an accuracy score to assess the proposed model’s performance, defining the ratio of correctly predicted reviews to the total number of studies in the dataset. Table 1 presents a summary of the findings.

Table 1. Accuracy in percentage

Sample Sizes		**TWSVM**	**TUNED-IDF**
Before Feature Extraction		70	61
After Feature Extraction	**1000000**	79	92
	2000000	79	91
	3000000	79	91

The accuracy scores achieved by comparing the two approaches are 70 percent and 61 percent for TWSVM and TUNED-IDF vectorizer, respectively, as shown in Figure 2. We extract and represent the number of features on the X-axis and represent the accuracy score in percent on the Y-axis. The features recovered after fine-tuning the TUNED-IDF approach using a random forest classifier are 10000, 20000, and 30000 in number, with 92 percent, 91 percent, and 91 percent accuracy, respectively. The TWSVM classifier's value remains constant during the prediction phase since it cannot extract features. The feature extraction approach can only utilize the TUNED-IDF algorithm; thus, features are recovered faster and more precisely than with the TWSVM, which we do after fine-tuning with the TWSVM classifier using Tuned-IDF [31].

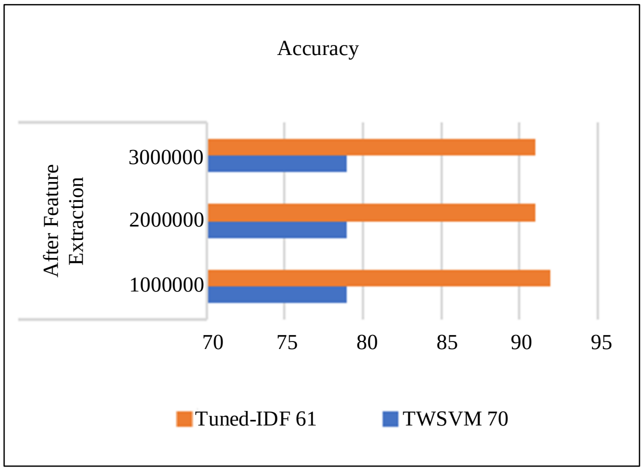

Fig. 2 Accuracy

We summarize the performance of the test subset to evaluate the trained sentiment classifiers in Table 2 (Appendix). According to the comprehensive data for each class, positive and negative kinds had higher Precision but poorer Recall. On the other hand, Precision and Recall are significantly higher in the neutral category (i.e., around 90 percent). To directly compare the two models, we also give the F1-scores for each class and the micro averaged F1-score for all categories (Ho et al., 2020).

TUNED-IDF improves the performance of TWSVM, which outperforms the sentiment classification task with three types, according to the micro-averaged F1 score in the last column of the table. Further, test the 2% and 2.5 percent improvements to see if they are meaningful. Nevertheless, they strongly indicate that the TUNED-IDF language model we developed can perform better in the emotion prediction test when using social media content. The number of nodes used in a different area of the data transmission from source to destination (de Paula and Alexandre, 2022).

Figure 4 depicts the F1 scores of the reviews. We display the Tuned-IDF classifier's predicted scores on the X-axis and F1 measurements on the Y-axis. In addition, we provide the F1 scores of the negative and positive labels in the categorization report. The positive labels have a high F1 score of 94 percent, and the negative labels have a score of 45 percent (Krommyda et al., 2021). Finally, we generate the F1 scores given below by the TWSVM and the TUNED-IDF vectorizer. Because positive labels have a far higher F1 than negative ones, TUNED-IDF vectorizer algorithms predict favorable reviews with greater accuracy than nasty comments.

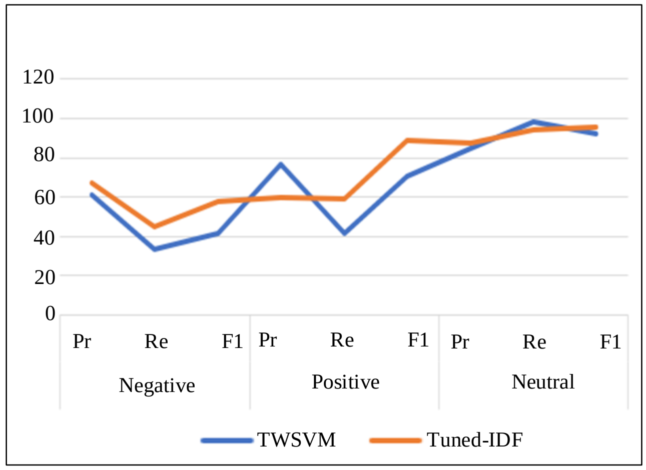

Fig. 3 Classification performance

Table 3. Performance of TUNED-IDF technique

Tuned-IDF	Precision	Recall	F1-Score
F1-Value Positive	0.95	0.98	0.94
F1-Value Negative	0.5	0.4	0.45

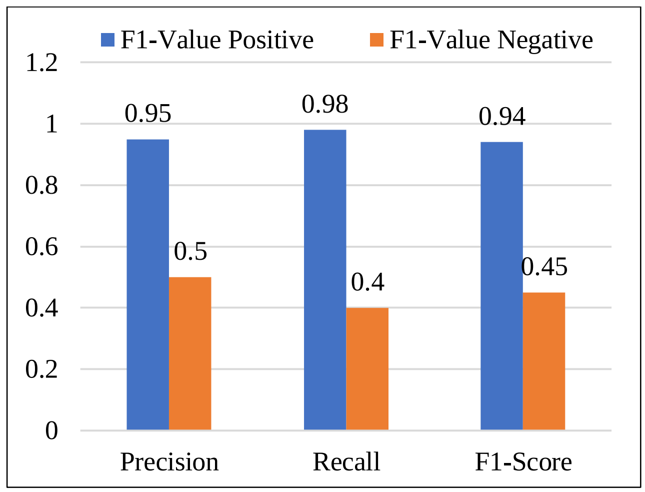

Fig. 4 Performance of TUNED-IDF

5. Conclusion and Future Work

The research purpose of achieving the user's emotions is assessed and has improved accuracy. By extracting word characteristics, machine learning and NLP technologies have helped increase the algorithm's efficiency in terms of F1 scores. The chi-square plot indicated the coefficients or feature extracted labels when fine-tuning the TUNED-IDF approach using the random forest classifier.

The TUNED-IDF method achieved good results. Because the dataset utilized here is small, with just thousands of evaluations, there is still interest in technology. A more extensive dataset may be employed in the future to compare and monitor the consequences. We must improve pessimistic review prediction because negative reviews have lower ratings than good reviews. To further deliver more precise findings, intent analysis, such as emotional labeling, which uses advanced deep learning algorithms, can be utilized to establish the user's intent. Because linear regression and TUNED-IDF are not as excellent at predicting negative reviews as they are at predicting positive reviews, we must solve this problem by employing a large dataset with a much more extensive range of perspectives.

References

[1] Pansy Nandwani, and Rupali Verma, "A Review on Sentiment Analysis and Emotion Detection from Text," *Social Network Analysis and Mining*, vol. 11, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[2] Santosh Kumar Bharti et al., "Text-Based Emotion Recognition Using Deep Learning Approach," *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1-8, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[3] Kian Long Tan, Chin Poo Lee, and Kian Ming Lim, "A Survey of Sentiment Analysis: Approaches, Datasets, and Future Research," *Applied Sciences*, vol. 13, no. 7, pp. 1-21, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[4] Muhammad Farrukh Bashir et al., "Context-Aware Emotion Detection from Low-Resource Urdu Language Using Deep Neural Network," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 22, no. 5, pp. 1-30, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[5] Ranit Kumar Dey, and Asit Kumar Das, "Modified Term Frequency-Inverse Document Frequency Based Deep Hybrid Framework for Sentiment Analysis," *Multimedia Tools and Applications*, vol. 82, pp. 32967-32990, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[6] Bagus Tris Atmaja, and Akira Sasou, "Sentiment Analysis and Emotion Recognition from Speech Using Universal Speech Representations," *Sensors*, vol. 22, no. 17, pp. 1-11, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[7] Alaa Alslaity, and Rita Orji, "Machine Learning Techniques for Emotion Detection and Sentiment Analysis: Current State, Challenges, and Future Directions," *Behaviour & Information Technology*, vol. 43, no. 1, pp. 139-164, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[8] Chris Jonathan D., and Sujitha Juliet, "Text-Based Emotion Recognition Using Sentiment Analysis," *2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*, Salem, India, pp. 844-848, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[9] Pedamuthevi Kiran Kumar, and Ishan Kumar, "Emotion Detection and Sentiment Analysis of Text," *Proceedings of the International Conference on Innovative Computing & Communication (ICICC)*, pp. 1-4, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[10] Licai Sun et al., "Multimodal Emotion Recognition and Sentiment Analysis via Attention Enhanced Recurrent Model," *Proceedings of the 2nd on Multimodal Sentiment Analysis Challenge*, pp. 15-20, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[11] Nilesh Shelke et al., "An Efficient Way of Text-Based Emotion Analysis from Social Media Using LRA-DNN," *Neuroscience Informatics*, vol. 2, no. 3, pp. 1-10, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[12] Francisca Adoma Acheampong, Chen Wenyu, and Henry Nunoo-Mensah, "Text-Based Emotion Detection: Advances, Challenges, and Opportunities," *Engineering Reports*, vol. 2, no. 7, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[13] Madhavi S. Darokar, Atul D. Raut, and Vilas M. Thakre, "Methodological Review of Emotion Recognition for Social Media: A Sentiment Analysis Approach," *2021 International Conference on Computing, Communication and Green Engineering (CCGE)*, Pune, India, pp. 1-5, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[14] Anjali Goswami et al., "Sentiment Analysis of Statements on Social Media and Electronic Media Using Machine and Deep Learning Classifiers," *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1-19, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[15] V. Jagadishwari et al., "Sentiment Analysis of Social Media Text-Emoticon Post with Machine Learning Models Contribution Title," *Journal of Physics: Conference Series*, vol. 2070, pp. 1-8, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[16] Vong Anh Ho et al., "Emotion Recognition for Vietnamese Social Media Text," *International Conference of the Pacific Association for Computational Linguistics*, pp. 319-333, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[17] Diandre de Paula, and Luís A. Alexandre, "Facial Emotion Recognition for Sentiment Analysis of Social Media Data," *Iberian Conference on Pattern Recognition and Image Analysis*, pp. 207-217, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[18] Maria Krommyda et al., "An Experimental Analysis of Data Annotation Methodologies for Emotion Detection in Short Text Posted on Social Media," *Informatics*, vol. 8, no. 1, pp. 1-15, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[19] Amira Samy Talaat, "Sentiment Analysis Classification System Using Hybrid BERT Models," *Journal of Big Data*, vol. 10, no. 1, pp. 1-18, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[20] Yafeng Ren et al., "Context-Sensitive Twitter Sentiment Classification Using Neural Network," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, pp. 215-221, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[21] Barakat AlBadani, Ronghua Shi, and Jian Dong, "A Novel Machine Learning Approach for Sentiment Analysis on Twitter Incorporating the Universal Language Model Fine-Tuning and SVM," *Applied System Innovation*, vol. 5, no. 1, pp. 1-16, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[22] Praphula Kumar Jain, Vijayalakshmi Saravanan, and Rajendra Pamula, "A Hybrid CNN-LSTM: A Deep Learning Approach for Consumer Sentiment Analysis Using Qualitative User-Generated Contents," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 20, no. 5, pp. 1-15, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[23] Sadaf Hussain Janjua et al., "Multi-Level Aspect Based Sentiment Classification of Twitter Data: Using Hybrid Approach in Deep Learning," *PeerJ Computer Science*, vol. 7, pp. 1-25, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[24] Ushmani Babar et al., "Data Mining Techniques for Sentiment Analysis Social Media Networking Using Different Information Resources," *Global Sci-Tech*, vol. 13, no. 3&4, pp. 138-147, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[25] Deepak Kumar Jain et al., "An Intelligent Cognitive-Inspired Computing with Big Data Analytics Framework for Sentiment Analysis and Classification," *Information Processing & Management*, vol. 59, no. 1, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[26] Monika Kabir et al., "An Empirical Research on Sentiment Analysis Using Machine Learning Approaches," *International Journal of Computers and Applications*, vol. 43, no. 10, pp. 1011-1019, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[27] Hetu Patel, and Anilkumar Suthar, "Sentiment Analysis for Rating Cooking Recipes Based on User's Feedback Using Data Mining Techniques (Classification)," *Gedrag & Organisatie Review*, vol. 34, no. 1, pp. 75-80, 2021. [Google Scholar] [Publisher Link]

[28] Ukhti Ikhsani Larasati et al., "Improve the Accuracy of Support Vector Machine Using Chi Square Statistic and Term Frequency Inverse Document Frequency on Movie Review Sentiment Analysis," *Scientific Journal of Informatics*, vol. 6, no. 1, pp. 138-149, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[29] Nur Syafiqah Mohd Nafis, and Suryanti Awang, "An Enhanced Hybrid Feature Selection Technique Using Term Frequency-Inverse Document Frequency and Support Vector Machine-Recursive Feature Elimination for Sentiment Classification," *IEEE Access*, vol. 9, pp. 52177-52192, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[30] C.S. Anita, and R. Sasikumar, "Learning Automata and Lexical Composition Method for Optimal and Load Balanced RPL Routing in IoT," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 40, no. 4, pp. 288-300, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[31] B. Sai Mani Teja et al., "A CNN Based Facial Expression Recognizer," *Materials Today: Proceedings*, vol. 37, Part 2, pp. 2578-2581, 2021. [CrossRef] [Google Scholar] [Publisher Linka]

Appendix

Table 2. Classification performance using the different models (Pr denotes precision, Re denotes recall, and F1-S denotes F1 − score) in the three classes (negative, positive and neutral)

Models	Negative			Positive			Neutral		
	Pr	**Re**	**F1-S**	**Pr**	**Re**	**F1-S**	**Pr**	**Re**	**F1-S**
TWSVM	61.3	33.1	41.7	77.0	41.3	70.4	84.6	98.2	92.1
TUNED-IDF	67.2	45.2	57.7	60.1	59.1	88.7	87.3	94.1	95.8