I.J. Information Technology and Computer Science, **2024, 2, 16-31**
Published Online on April 8, 2024 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2024.02.02

Augmenting Sentiment Analysis Prediction in Binary Text Classification through Advanced Natural Language Processing Models and Classifiers

Zhengbing Hu
School of Computer Science, Hubei University of Technology, Wuhan, China
E-mail: drzbhu@gmail.com
ORCID iD: https://orcid.org/0000-0002-6140-3351

Ivan Dychka
Computer Systems Software Department, Igor Sikorsky Kyiv Polytechnic Institute, Kyiv, Ukraine
E-mail: dychka@pzks.fpm.kpi.ua
ORCID iD: https://orcid.org/0000-0002-3446-3076

Kateryna Potapova
Department of System Programming and Specialized Computer Systems, Igor Sikorsky Kyiv Polytechnic Institute, Kyiv, Ukraine
E-mail: avatarkina-avatarochka@ukr.net
ORCID iD: https://orcid.org/0000-0002-3347-6350

Vasyl Meliukh*
Department of System Programming and Specialized Computer Systems, Igor Sikorsky Kyiv Polytechnic Institute, Kyiv, Ukraine
E-mail: vasylmeliukh430@gmail.com
ORCID iD: https://orcid.org/0009-0009-3783-9954
*Corresponding Author

Received: 03 January 2024; Revised: 15 February 2024; Accepted: 10 March 2024; Published: 08 April 2024

Abstract: Sentiment analysis is a critical component in natural language processing applications, particularly for text classification. By employing state-of-the-art techniques such as ensemble methods, transfer learning and deep learning architectures, our methodology significantly enhances the robustness and precision of sentiment predictions. We systematically investigate the impact of various NLP models, including recurrent neural networks and transformer-based architectures, on sentiment classification tasks. Furthermore, we introduce a novel ensemble method that combines the strengths of multiple classifiers to improve the predictive ability of the system. The results demonstrate the potential of integrating state-of-the-art Natural Language Processing (NLP) models with ensemble classifiers to advance sentiment analysis. This lays the foundation for a more advanced comprehension of textual sentiments in diverse applications.

Index Terms: Binary Text Classification, Natural Language Processing, Deep Learning, Ensemble Methods, Neural Networks.

1. Introduction

In the era of digital communications, the large amount of written content generated daily across various platforms has gained popularity in the field of natural language processing (NLP) for a variety of applications, ranging from monitoring social media interactions to scrutinizing customer feedback. The difficulty of sentiment analysis lies in the precise determination of the emotional tone conveyed in a particular textual composition, which is usually classified as positive, negative, or neutral sentiment. In addition, the study of opinions and public opinion is becoming increasingly important, emphasizing the nuances of extracting subjective information from various textual sources [1].

In the domain of sentiment analysis, a core task is to examine and extrapolate sentiments and opinions from written information using machine learning and natural language processing approaches, thereby classifying processed content into different categories. The accuracy and reliability of sentiment predictions are crucial for decision-making processes in various areas, especially in the area of binary text classification. Binary text classification, a nuanced subset within the broader field of sentiment analysis, is particularly concerned with carefully distinguishing between positive and negative sentiments encoded in written information [2, 3].

For convoluted and unclear phrases that contain an excess of contextual ambiguity, it may be difficult for traditional sentiment analysis techniques to fully represent the complexity of natural language. However, new opportunities to improve the accuracy and performance of sentiment analysis predictions have emerged due to recent advances in natural language processing (NLP) based on deep learning architectures, transfer learning strategies, and ensemble methods. Sentiment analysis, viewed as a whole, plays a crucial role in understanding and researching the content of human language. Improving the prediction of sentiment analysis in binary text classification is becoming more and more achievable as NLP models and classifiers develop [4].

The study's goal is to investigate how sophisticated models of natural language processing (NLP), such as recurrent neural networks and transformer architecture, affect tasks involving sentiment classification. With the help of these models, emotive texts' subtleties can be more accurately captured, and they are made to comprehend intricate linguistic patterns and situations. Additionally, we describe a novel strategy that uses ensemble methods to harness the combined intelligence of many machine learning classifiers. We aim to create a strong and resilient sentiment analysis system that excels at handling a wide range of text data sources and linguistic subtleties by merging the advantages of several classifiers.

While a large number of optimization algorithms have been presented in the literature, little is known about their effectiveness in improving sentiment analysis prediction using sophisticated natural language processing models and classifiers, particularly in the context of binary text classification. Through thorough comparative analysis, we hope to provide insightful information about the applicability of various language processing models and classifiers for producing binary classification prediction results, while presenting their advantages and disadvantages. The main goal of this study is to address this gap, focusing on the challenge of integrating numerous models into ensemble classifiers, including begging, voting, and batch classifiers, to maximize prediction efficiency and accuracy.

The performance of the proposed methodology is evaluated through extensive experiments and benchmarking on a baseline Disaster Tweets dataset. The results show that our method is superior to traditional sentiment analysis techniques and show how sophisticated NLP models and complex classifiers can completely transform the sentiment analysis industry. Our work opens the door to more precise, in-depth, and context-sensitive sentiment analysis techniques at a time when organizations, policymakers, and researchers need to understand public sentiment.

Nevertheless, the intrinsic challenges of understanding complex deep learning architectures, especially when it comes to comprehending ensemble methods' processes for making decisions, is a major downside. Even if our method's precision and accuracy have been substantially enhanced, the complex advanced models may find it difficult to grasp the rationale behind certain of the predictions. Since the application of our results to various text datasets and domains may show variability, more investigation and validation within a wider framework are required. Regardless of these limitations, our research provides significant new insights into the way challenging natural language processing models might be used with ensemble classifiers to improve sentiment analysis in binary text classification.

A number of important objectives are the authors' ambitions. Our main objective is to increase comprehension of sentiment analysis within the framework of applications involving Natural Language Processing (NLP), with an emphasis on binary text classification specifically. We endeavor to dramatically increase the resilience and accuracy of sentiment predictions by utilizing state-of-the-art methodologies, including ensemble techniques, transfer learning, and complex deep learning architectures like recurrent neural networks and transformer-based models. We also hope to give a thorough overview of the efficacy of these models through our systematic investigation of how various NLP models affect sentiment categorization tasks. The system's predictive capacity is meant to be enhanced by the addition of a novel ensemble technique that integrates the advantages of several classifiers. Our research aims to show how cutting-edge NLP models may be integrated with ensemble classifiers to improve text sentiment understanding in a variety of applications and pave the stage for future advancements in sentiment analysis.

The intended results of this research are expected to considerably extend the corpus of existing knowledge about the enhancement of sentiment analysis prediction in binary text classification through the utilization of advanced Natural Language Processing models and classifiers. With the use of cutting-edge techniques including ensemble approaches, transfer learning, and deep learning architectures like transformer-based models and recurrent neural networks, our method seeks to improve the accuracy and dependability of sentiment predictions. It is anticipated that methodologically analyzing how various NLP models perform on sentiment classification tasks would yield informative data that will help us better understand how successful these models are. Using a novel ensemble strategy that makes use of many classifiers is also bound to boost the overall prediction capability.

2. Background

In recent years, there has been a growing interest in enhancing sentiment analysis prediction in binary text classification using advanced natural language processing (NLP) models and classifiers. One such model is BERT (Bidirectional Encoder Representations from Transformers), which has shown remarkable performance in various natural language processing (NLP) tasks, finding that BERTLARGE significantly outperforms BERTBASE across all tasks, especially those with very little training data. BERT is a pre-training model that utilizes deep bidirectional transformers for language understanding. It has been successfully applied to tasks such as natural language inference and paraphrasing [4].

Another popular approach to sentiment analysis is the use of convolutional neural networks (CNNs). Convolutional Neural Networks (CNNs) are widely used in natural language processing (NLP) for sentence classification. These models learn word vector representations through neural language models and perform composition over the learned word vectors for classification. The Kim [5] paper shows that simple CNN models built on top of pre-trained word vectors can achieve excellent results for sentence classification, and fine-tuning the word vectors provides additional gains.

Furthermore, sentiment analysis has practical applications in various domains, including politics, news, and online reviews [1]. In the field of online homestay reviews, a multimedia sentiment model that combines text and image sentiment analysis was proposed. This model utilizes Word2vec for topic clustering and Bayesian classifiers for sentiment analysis of text data, while Convolutional Neural Networks (CNNs) are used for sentiment analysis of image data. The research found that fusing text and image sentiment analysis models through decision-level fusion achieved higher classification accuracy (88.6%) than using text or image models alone [6].

The authors of the study [7] conclude evidence that augmenting sentiment analysis with advanced NLP can yield more accurate predictions, especially for tasks involving subtle or context-dependent sentiment expression. Combining linear algebra and probabilistic topic model for Latent Semantic Relations (LSR) revealing allowed to eliminate their limitations. Such an approach allowed to bring the average value of the topic recognition recall rate indicator close to 90–95% and increase the precision indicator from 62 to 70–75% [8].

Traditionally, sentiment analysis approaches relied on rule-based methods and basic machine learning algorithms such as Naive Bayes and Support Vector Machines [9]. Although these methods were reasonably effective, they often struggled to capture the intricacies of natural language and context, resulting in limited accuracy and generalization. With the advent of advanced natural language processing techniques and machine learning models, researchers have explored innovative methods to improve sentiment analysis predictions in binary text classification.

A notable advancement in this area is the application of deep learning models, particularly neural networks, which have shown remarkable performance in various natural language processing (NLP) tasks. Recurrent Neural Networks (RNNs) and Long Short-Term Memory Networks (LSTMs) are the types of neural networks that are widely used to analyze sequential data, such as sentiment analysis. These models are able to capture sequential dependencies in text data, enabling more accurate sentiment predictions, incorporating additional information like user/product information, syntactic/semantic information, cross-domain information to improve performance [10].

Furthermore, the introduction of transformer-based architectures, exemplified by models such as BERT (Bidirectional Encoder Representations of Transformers) [11] and GPT (Generative Pre-trained Transformer) [12], transforming the field of NLP. These transformer models are pre-trained on large amounts of text data and can be further optimized for specific tasks, such as binary text classification. Leveraging contextual embeddings from transformer models enables the capture of nuanced semantic meanings, improving the accuracy and depth of sentiment analysis.

Additionally, researchers have investigated ensemble methods, which combine predictions from multiple models to improve overall performance. Ensemble techniques such as bagging and boosting have been used to improve the robustness and reliability of sentiment analysis systems. These methods exploit the diversity of different models to collectively make more accurate predictions than individual models. Using a multilayer perceptron network and combining the results with both deep learning and traditional feature-based models, Akhtar et al. proposed a stacked ensemble method for predicting mood intensity [1].

In the context of binary text classification, researchers have also explored the inclusion of advanced classifiers such as random forests, gradient boosting machines, and deep neural networks. Empirical results from Leo Breiman's research show that Random Forests are more competitive than boosting approaches. Furthermore, the resilience of random forests to noise is discussed, which is important for evaluating classifier performance in sentiment analysis [13]. These classifiers often outperform traditional machine learning algorithms by capturing complex patterns and relationships within the data, resulting in better sentiment prediction.

This paper builds upon these advancements by proposing an innovative approach to enhance sentiment analysis prediction in binary text classification. By utilizing advanced natural language processing models, such as transformer architectures, and integrating them with cutting-edge classifiers and ensemble techniques, the proposed methodology seeks to greatly improve the precision and dependability of sentiment analysis systems. The following sections detail the methodology, experiments, and results, demonstrating the effectiveness of the proposed approach in enhancing sentiment analysis prediction in binary text classification tasks.

3. Methodology

The thorough selection and preprocessing of data are crucial for improving the prediction of sentiment analysis in binary text classification. We delve into the intricacies of data preprocessing, which includes tokenization, text cleaning, handling unbalanced data, and using embedding techniques to convert textual data into numerical representations. These basic procedures lay the foundation for the later utilization of advanced natural language processing models and classifiers [14].

3.1. Data Preprocessing

A. Text cleaning and Tokenization

The quality of the input data has a significant impact on the effectiveness and accuracy of text classification models and sentiment analysis in the field of natural language processing (NLP). Unstructured, erratic, and noisy text data can be obtained from various sources [15,16]. A number of rigorous data pretreatment procedures were used to enhance the accuracy and consistency of sentiment analysis prediction in binary text categorization.

a. Lowercasing

The first step in data preprocessing involves converting all text to lowercase. It offers several benefits, including text normalization, vocabulary reduction, compatibility with word embedding models, and ensuring consistency in text processing pipelines. This standardizes the text and ensures that the model does not treat words with different cases as distinct entities [16]. By converting the entire corpus to lowercase, potential discrepancies in capitalization are eliminated, thus improving the consistency and comparability of the dataset. However, it is important to consider the specific requirements of the task, as lowercasing can lead to the loss of information related to proper nouns and can affect the interpretation of acronyms and abbreviations.

b. Named Entity Recognition

Named entities, encompassing phrases specific to individuals, places, organizations, and other entities, may introduce noise into the data, resulting in a low sentiment value. To address this, named entity recognition techniques were applied to identify and replace these entities with generic tags. This process effectively removed entity-specific information while preserving the text's syntactic structure. Streamlining the text through this step significantly improved its suitability for sentiment analysis [16, 17]. In a study by Lample et al., the authors examined various sequence labeling models and evaluated their performance with and without the use of external labeled data, such as gazetteers and knowledge bases. The results showed that the LSTM-CRF model showed significant performance improvement over all previous methods, even outperforming models containing externally labeled data [18].

c. URL Link Removal

Text frequently contains URLs, which are irrelevant for sentiment analysis and can skew the results. As the dataset is formed from tweets, it might also contain a number of mentions and hashtags (such as #NYfire and @KyleUps0). Various approaches, such as regular expressions, library functions, and preprocessing tools, were employed to detect and remove URL links from the dataset.

d. Punctuation Removal

Punctuation marks, including periods, commas, and exclamation points, are essential for language structure but do not significantly contribute to sentiment analysis. Removing these symbols minimizes noise and reduces the dimensionality of the dataset. By eliminating punctuation, the model can focus on the semantic meaning of the words and phrases, thereby enhancing the accuracy of sentiment prediction.

e. Spelling Correction

Misspelled words are common in unedited text data and can result in misinterpretation during analysis. To address this issue, a spelling correction mechanism has been implemented. Leveraging advanced algorithms, misspelled words are identified and replaced with their correct counterparts. SymSpell uses a prefix tree data structure to efficiently store and search for words.

The core idea lies in the Symmetric Delete spelling correction method. It starts by generating a dictionary that contains correctly spelled words. For each word in the dictionary, it generates a set of candidate words by deleting one or more characters from the original word (1).

$$C(w) = \{w_i \mid w_i = w[0:i] + w[i+1], i \in [o, len(w)]\} \tag{1}$$

The method utilizes an edit distance algorithm, specifically the Damerau-Levenshtein (DL) distance or Levenshtein distance, to quantify the similarity between two words, showcasing a greater improvement over previous algorithms across time, space, and energy metrics [19,20]. This distance metric calculates the number of single-character edits (insertions, deletions, substitutions, or transpositions) required to change one word into another. After generating a list of candidate words for a misspelled word, the system ranks these candidates based on their frequency in the corpus.

f. Lemmatization

Lemmatization, a technique for morphological analysis, involves reducing words to their base or root form. By lemmatizing the text data, variations of words (such as 'running' to 'run' or 'better' to 'good') are standardized. This process reduces the dimensionality of the dataset, simplifies the feature space, and ensures that the sentiment analysis model focuses on the core meaning of words, thereby enhancing prediction accuracy. The main difference between lemmatization and stemming is that lemmatization takes into consideration the context and converts the word to its base form, whereas stemming only removes the last few characters of the word.

g. Stop Words Removal

Stop words are common words such as 'and', 'the', and 'is' that occur frequently in a language. These words do not carry significant sentiment value and can be safely removed from the text. Removing stop words reduces noise in the dataset, making it more focused and meaningful for sentiment analysis [14]. The exclusion of stop words ensures that the model gives priority to words that express sentiment, resulting in more precise predictions in binary text classification.

B. Handling Imbalanced Data with Entity Recognition

Imbalanced data poses a challenge in binary text classification tasks, where one class significantly outnumbers the other. One common challenge faced during the preprocessing stage is handling missing data. In the field of sentiment analysis and binary text classification, it is crucial to ensure data completeness and address the challenge of missing data. As investigated in the research by Jacob Eisenstein et al., the Stack LSTM model exhibits a notable characteristic as it integrates a transition-based algorithm that draws inspiration from shift-reduction parsers. This particular approach effectively enables the creation and attribution of segments that represent the stack within the input, thereby harnessing the power of stack LSTMs to enhance the performance of the model [18].

Upon initial data inspection, it was identified that the dataset contained missing entries, particularly in important fields such as keywords and locations. These missing data points can significantly impact the accuracy of sentiment analysis and text classification models. Entity recognition, a subtask of information extraction, involves identifying and classifying entities in text into predefined. In our research, entity recognition serves a dual purpose: not only does it help in populating missing location entries by identifying relevant geographical entities, but it also contributes to filling gaps in the keyword column by extracting essential keywords from the text. As a result, missing entries in the imbalanced keyword and location columns were intelligently populated, ensuring a more complete dataset and it enhanced the quality and relevance of the imputed data. This nuanced approach mitigated the impact of missing data on the subsequent analysis, ensuring a more accurate representation of the dataset.

C. Embedding Techniques for Transforming Textual Data into Numerical Representation

Transforming textual data into numerical representation is imperative for machine learning models to process the information effectively. Embedding techniques, especially word embeddings such as Word2Vec, GloVe, or fastText, are used to transform words or tokens into compact vectors of real numbers [14,16]. These embeddings capture semantic relationships between words while preserving contextual information. Additionally, techniques such as Doc2Vec extend this concept to entire documents, providing numerical representations for complete texts. These embeddings serve as crucial features for Natural Language Processing models, allowing them to understand the textual content and acquire significant patterns.

In the context of Twitter data related to disasters, these preprocessing steps are particularly crucial. Disaster-related tweets often contain noisy information, abbreviations, and misspellings, making text cleaning pivotal. Additionally, imbalanced data may occur, where tweets expressing sentiments about disasters might be significantly outnumbered by neutral or unrelated tweets. Through appropriate preprocessing techniques and robust embedding methods, the proposed Sentiment Analysis Prediction model can leverage the capabilities of advanced Natural Language Processing, leading to more precise and insightful binary text classification outcomes.

The disaster Twitter dataset, which has been robustly preprocessed and enriched through these techniques, will be used as input for the advanced Natural Language Processing models and classifiers discussed in the following sections of this paper.

a. Word Embeddings

The use of pre-trained word embeddings, such as Word2Vec and GloVe, is crucial in converting words from the disaster Twitter dataset into dense vectors of real numbers [16]. These embeddings capture semantic relationships between words, enabling the conversion of textual context into meaningful numerical representations. By utilizing these embeddings, the model acquires a comprehension of the fundamental meaning of words, facilitating a more sophisticated sentiment analysis.

b. BERT Embeddings

Transformer-based language models, particularly BERT (Bidirectional Encoder Representations from Transformers), are utilized to generate contextual word embeddings. BERT embeddings provide a deep understanding of word meanings by considering the context in which they occur within a sentence [11]. This contextual comprehension allows for a deeper analysis of tweets related to disasters, capturing subtle nuances and intricacies in language usage. BERT embeddings enhance the model's ability to capture the contextual variations in sentiment expressions, thereby improving the accuracy of sentiment analysis predictions.

c. Token Embeddings

Token embeddings are generated using tokenizers specifically designed for transformer models. These embeddings represent individual tokens within a sentence, providing a detailed analysis of textual data [14,16]. Token embeddings are particularly valuable for transformer-based architectures as they enable the model to focus on the detailed semantics of each token. By incorporating token embeddings, the model can discern sentiment-related nuances at the token level, effectively capturing variations in sentiment within tweets.

d. Term Frequency-Inverse Document Frequency

TF-IDF (Term Frequency-Inverse Document Frequency) is a numerical statistic used in natural language processing and information retrieval to quantify the significance of a word in a document compared to a collection of documents (corpus). TF-IDF is a popular technique for extracting text features and is commonly employed in tasks such as text mining, information retrieval, and text classification [21].

In TF-IDF based learning models, each document is represented as a vector where each dimension corresponds to a unique term in the corpus. The TF-IDF score of a term in the document is used as the value in the corresponding dimension. These numerical vectors can then be fed into machine learning models like logistic regression, Naive Bayes, SVM, or any other model capable of handling numerical data for tasks such as text classification, sentiment analysis, and information retrieval.

In TF-IDF-based learning models, textual data is transformed into numerical vectors using the TF-IDF representation. Here's what TF-IDF stands for:

- Term Frequency: It measures the frequency of occurrence of a term (word) in a document (d). The more frequently a term appears in a document, the higher its TF score (2).

$$TF(t,d) = \frac{Number\ of\ times\ t\ appears\ in\ d}{Total\ number\ of\ terms\ in\ d} \tag{2}$$

- Inverse Document Frequency: It measures how important a term is across a collection of documents (corpus). Words that occur frequently in many documents have a lower IDF score, while words that occur rarely have a higher IDF score (3).

$$IDF(t,D) = \log\left(\frac{Total\ number\ number\ of\ d\ in\ the\ corpus\ |D|}{Number\ of\ d\ containing\ the\ term\ \mathrm{t}+1}\right) \tag{3}$$

- TF-IDF Score: The TF-IDF score of a term in a document is calculated by multiplying its TF score and IDF score (4).

$$TF - IDF(t,d,D) = TF(t,d) * IDF(t,D) \tag{4}$$

e. Sequence Padding

To ensure consistent input shapes for deep learning models, sequence padding is applied. This involves padding shorter sequences with zeros to ensure that all sequences have the same length. Uniform sequence length is essential for the proper functioning of deep learning architectures [1]. By standardizing the input sequence length through padding, the model can consistently process textual data, enabling smooth integration into advanced Natural Language Processing models and classifiers.

Through the application of embedding techniques and sequence padding, the disaster Twitter dataset is transformed into a format that is suitable for comprehensive sentiment analysis. These processed numerical representations serve as the foundation upon which advanced Natural Language Processing models and classifiers can be trained and optimized. This ultimately enhances the accuracy and reliability of sentiment analysis predictions in binary text classification scenarios.

3.2. Advanced Natural Language Processing Models

In order to enhance sentiment analysis prediction in binary text classification, it is crucial to integrate advanced Natural Language Processing (NLP) models. The proposed methodology involves the utilization of advanced NLP techniques, leveraging the capabilities of deep learning architectures and transformer-based models to improve the precision and comprehensiveness of sentiment analysis. Recurrent neural networks capture temporal dependencies, while transformer-based architectures excel at long-term dependency capture and semantic understanding [22]. The chosen methods push the boundaries of the precision of sentiment analysis and demonstrate the potential for integrating NLP models with each other:

- *Recurrent Neural Networks (RNNs)*

RNNs are employed due to their sequential nature, making them suitable for processing sequential data, such as text. Specifically, Long Short-Term Memory (LSTM) networks and Gated Recurrent Unit (GRU) networks are utilized to capture long-term dependencies in textual data. This allows the model to comprehend the contextual relationships between words and phrases, which is crucial for achieving accurate sentiment analysis.

- *Convolutional Neural Networks (CNNs)*

CNNs, primarily designed for image recognition, can be adapted for text analysis. By treating words or phrases as local patterns similar to image pixels, CNNs can capture hierarchical features in textual data. CNNs excel at identifying patterns within text, making them valuable for sentiment analysis tasks where specific textual patterns convey sentiments.

- *Ensemble Models*

Ensemble models, which combine predictions from multiple base models, are constructed to leverage diverse learning patterns. Techniques such as bagging and boosting are applied to combine the strengths of different models. This amalgamation ensures a comprehensive and balanced sentiment analysis approach, thereby enhancing the overall accuracy and reliability of predictions. Ensemble methods have drawbacks and challenges such as being computationally expensive and time-consuming due to training and storing multiple models, and combining their outputs, as well as being sensitive to the quality and diversity of data and base models, as they depend on assumptions, limitations, representativeness, and independence of the data samples and features.

3.3. Classifiers and Ensemble Techniques

A. Logistic Regression

For its interpretability and simplicity, Logistic Regression is commonly employed as a fundamental and dependable linear classifier. It serves as a baseline model, offering a standard against which the effectiveness of more complex classifiers can be evaluated. Logistic regression is most helpful when there is a roughly linear relationship between the target variable and the features.

The logistic regression equation models the relationship between a binary dependent variable (*y*) and one or more independent variables (*x*). In the case of a single independent variable, the logistic regression equation takes the form (5) where *P(x)* represents the probability of the dependent variable, b_0 is the intercept term (constant), b_1 is the coefficient associated with the independent variable *x*.

$$P(x) = \frac{1}{1+e^{-(b_0+b_1x)}} \tag{5}$$

B. Support Vector Machines (SVM)

Support Vector Machines (SVMs) are employed because of their ability to handle high-dimensional data and capture complex nonlinear relationships. By utilizing suitable kernels, Support Vector Machines (SVMs) can identify complex patterns within textual data, rendering them highly valuable for sentiment analysis tasks. The flexibility of SVMs in choosing different kernel functions allows for the exploration of various feature spaces.

C. Random Forest

Random Forest, an ensemble learning method based on decision tree classifiers, is utilized for its capability to handle high-dimensional data and capture feature interactions. By aggregating predictions from multiple decision trees, Random Forest mitigates overfitting and provides a robust framework for sentiment analysis. Its inherent parallelism allows efficient processing of large datasets [13].

D. Gradient Boosting Machines (GBM)

GBM constructs a series of decision trees sequentially, with each tree aiming to correct errors made by the previous ones. This iterative learning process allows GBM to focus on misclassified instances, thereby enhancing prediction accuracy. GBM is adept at capturing complex relationships in the data and is particularly useful for improving accuracy in the presence of noisy or ambiguous features.

E. Voting Classifiers

Voting classifiers combine predictions from multiple base classifiers, such as Logistic Regression, SVM, Random Forest, and GBM, using various voting strategies (e.g., majority voting, weighted voting) [23]. By aggregating diverse perspectives, voting classifiers capitalize on the collective intelligence of individual models, resulting in more accurate and robust predictions. Different combinations of base classifiers are explored to identify the optimal ensemble configuration.

F. Stacking

Stacking, a meta-ensemble technique, combines predictions from multiple base classifiers using a meta-learner. Base classifiers, such as Logistic Regression, SVM, Random Forest, and GBM, generate predictions that are subsequently used as inputs for the meta-learner. The meta-learner learns to combine these predictions, optimizing the overall performance of the ensemble. Stacking leverages the strengths of diverse classifiers, enabling the intricate capture of sentiment nuances [23].

3.4. Evaluation Metrics

Evaluation metrics are essential for assessing the performance of the proposed methodology in enhancing sentiment analysis prediction in binary text classification. The following metrics are used to comprehensively evaluate the accuracy, precision, recall, and overall effectiveness of the model [24]:

- Accuracy

Accuracy measures the proportion of correctly classified instances out of the total instances in the dataset. It provides a comprehensive evaluation of the model's accuracy in predicting both positive and negative sentiments.

- Precision

Precision represents the ratio of correctly predicted positive instances to the total instances predicted as positive. High precision indicates that when the model predicts a positive sentiment, it is likely to be correct.

- Recall (sensitivity)

Recall calculates the proportion of correctly predicted positive instances out of the total number of actual positive instances in the dataset. It assesses the model's ability to identify all positive sentiments.

- F1-Score

The F1-score, calculated as the harmonic mean of precision and recall, strikes a balance between these two metrics, accounting for both false positives and false negatives. This metric is especially valuable in scenarios with class imbalances in the dataset.

- Area under the ROC Curve (AUC-ROC)

The ROC (Receiver Operatinsg Characteristic) curve is a graphical representation of a model's discriminative capacity between positive and negative sentiments across various decision thresholds. To quantitatively assess the model's overall performance, the Area Under the Receiver Operating Characteristic (ROC) Curve (AUC-ROC) is employed. The AUC-ROC value functions as a scalar metric, with higher values signifying enhanced discriminatory prowess in distinguishing between positive and negative sentiments.

- Confusion Matrix

The confusion matrix provides a detailed breakdown of the model's predictions, including true positives, true negatives, false positives, and false negatives. It offers insights into specific errors made by the model, aiding in understanding its strengths and weaknesses.

- Receiver Operating Characteristic (ROC) Curve

The sensitivity or true positive rate is plotted versus the false positive rate (also known as 1-specificity) at various classification thresholds using a ROC curve. The trade-off between sensitivity and specificity is visualized, which aids in choosing the best categorization threshold.

The F1 score (6) is computed as the harmonic mean of precision and recall, resulting in a well-balanced metric whereby the model's precision is not prioritized over recall, or vice versa. Consequently, the F1 score exhibits a robust capability in identifying positive instances while simultaneously reducing the occurrence of false positives and false negatives.

$$F_1 = 2 * \frac{precision * recall}{precisoin + recall} \tag{6}$$

The quality of positive predictions is determined by the precision (7) metric. It is calculated by dividing the number of true positive outcomes by the sum of true positive and false positive predictions. The formula employed in the computation of precision is:

$$precison = \frac{TP}{TP + FP} \tag{7}$$

The measure known as recall (8), which is also referred to as sensitivity, examines the model's capacity to accurately identify positive events. Specifically, it represents the percentage of positive events that are correctly predicted out of the total number of positive events that actually occurred. When determining the recall of a classification model, the calculation involves employing the following formula:

$$recall = \frac{TP}{TP + FN} \tag{8}$$

The metric of accuracy (9) assesses the comprehensive accuracy of predictions by dividing the count of accurately predicted positive and negative events by the total count of events. The accuracy can be calculated using the following formula:

$$accuracy = \frac{TN + TP}{TN + FN + TP + FP} \tag{9}$$

The attributes used in equations are as follows:

- True Positive [TP] = prediction = 1, ground truth = 1 – positive prediction, correct.
- False Positive [FP] = prediction = 1, ground truth = 0 – positive prediction, incorrect.
- True Negative [TN] = prediction = 0, ground truth = 0 – negative prediction, correct.
- False Negative [FN] = prediction = 0, ground truth = 1 – negative prediction, incorrect.

By utilizing these comprehensive evaluation metrics, the proposed methodology ensures a thorough assessment of the model's performance. This enables a nuanced understanding of how it enhances sentiment analysis prediction in binary text classification tasks.

4. Experiments and Setup

The basic idea of sentiment analysis is based on predicting the limited categories into which the target object could be divided, as already suggested in the study. Traditional polarized classification differs from advanced sentiment analysis because it may include more than two categories. On the other hand, the more categories are included, the higher the likelihood of errors and inconsistencies, which increases the complexity of the results. This phenomenon results from the input data consisting of expressed textual opinions and the technique that uses tangible scales to recognize and distinguish text tokens.

For instance, within the realm of emotion detection semantic analysis, a well-known aspect is the distinction between positive, negative, or neutral variations of a given passage, commonly known as polarity. On the contrary, more sophisticated classification systems surpass the realm of polarity and encompass emotional states such as fear, surprise, disgust, anger, happiness, and so forth. Similar to the classifiers of naive Bayes, this technique often involves the estimation of a probability distribution across all categories. The attributes of the data will dictate whether and how a neutral class should be employed; if the data is clearly divided into specific language domains, it would be logical to eliminate the neutral language and make the priority of the polarity of positive and negative sentiments. Conversely, this approach may impede a clear differentiation between the two extremes if the data predominantly consists of neutral language with minimal deviations towards positive and negative emotions.

To eliminate the risk of additional error in the comparative analysis of proposed models while retaining the subjectivity of the data, It was decided to focus predominantly on datasets with binary classifications. However, addressing this concern, it is crucial to ensure that the chosen dataset is not only representative but also effectively challenge the model. First of all, the relevance and timeliness of the data, ensuring recent data is crucial for capturing the dynamics of public sentiment during different types of disasters. Second, the dataset should include tweets from a broad geographical range to capture variations in sentiment based on the location of the disaster. As well, the data model should provide details in the features available, sentiment labels, and any preprocessing steps applied. This documentation should cover metadata, variable descriptions, and any potential biases or limitations in the dataset.

Twitter has grown to be a crucial means of emergency communication. People can instantly announce an emergency they're observing because to the widespread use of smartphones. Consequently, an increasing number of agencies—such as news organizations and disaster assistance organizations—are interested in programmatically monitoring Twitter. However, it's not always evident whether a person's words are truly heralding a disaster. Consider the following scenario, where the author employs the term "IGNITED" in a metaphorical sense. While this interpretation is readily apparent to a human observer, particularly with the support of visual context, it presents a greater challenge for automated systems to discern.

In this research, the experiments conducted for building machine learning models on a dataset aimed at predicting whether tweets are about real disasters or not. The dataset consists of 10,000 tweets, each hand-classified into one of two categories: target 1 denotes tweets about actual disasters, and target 0 denotes tweets that are not related to real disasters, as shown in figure 1.

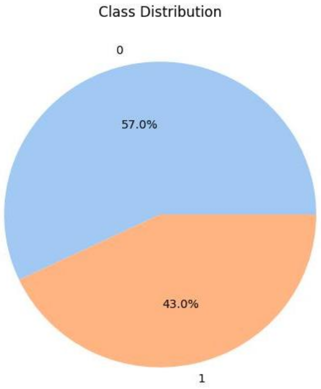

Fig.1. Distribution of training dataset tweets based on the target class

The dataset was sourced from a machine learning task "Natural Language Processing with Disaster Tweets". It consists of two main files: train.csv and test.csv. The training set contains labeled tweets for model training. The test set is used to evaluate the performance of the trained model. The dataset size provides a substantial volume of data for the experiments.

Each sample in both the train and test sets includes the following information: text (the actual content of the tweet, used for analysis), location (the geographical location from which the tweet was sent, which may be blank), keyword (a specific keyword from the tweet, which may be blank), and target (the ground truth for classification during model training).

The primary objective of the experiment was to develop and evaluate machine learning models capable of predicting the relevance of tweets to actual disasters. By utilizing the textual content, along with additional contextual features such as location and keywords, the goal was to accurately classify tweets into the predefined categories (target 0 or target 1).

The text data underwent cleaning processes, including the removal of special characters, stopwords, and irrelevant symbols, to ensure consistency and enhance model accuracy. Proper handling of missing values in the location and keyword columns has been implemented to prevent biases during training. The training data was further divided into training and validation subsets to facilitate model training and performance evaluation. The text, location, and keyword columns were selected as features for model training. The experiments utilized evaluation metrics such as accuracy, precision, recall, F1-score, and possibly the area under the ROC curve (ROC-AUC) to assess the performance of the model.

The experiments were conducted using popular machine learning libraries such as scikit-learn, SciPy, TensorFlow, and PyTorch. Text data was cleaned, tokenized, and transformed into TF-IDF vectors. Each tweet was represented as a sparse vector, where each dimension corresponded to a unique word, and the value indicated the importance of that word in the tweet relative to the entire dataset. For Word2Vec and GloVe embeddings, each word in the text data was replaced by its pre-trained vector representation. The resulting tweet was represented as a matrix of word vectors, capturing semantic information. Tokenization of text was performed using specialized tokenizers provided by the Transformers library. The tokenized input was then fed through the BERT or GPT models, producing contextual embeddings for each token. These embeddings were aggregated (e.g., mean pooling) to obtain fixed-size numerical vectors representing the entire tweet. Data manipulation and analysis were performed using tools like pandas and NumPy. Visualization of results and data exploration were aided by libraries such as Matplotlib or Seaborn.

There are complexities associated with the use of machine learning models and datasets that require careful consideration of potential limitations and biases. To ensure the robustness and reliability of research findings, it is important to acknowledge these inherent challenges. In the context of the models and datasets we select, it is imperative to examine various aspects that may introduce nuances that affect the accuracy and generalizability of our analyses.

One way to add noise and uncertainty to a data set is through ambiguous labeling. The degree to which the labeled data is subject to subjectivity impacts the ground truth reliability for model training. Because tweets are typically short, little context may be provided, making it difficult for machine learning models and annotators to identify the actual nature of the events being referenced. The model may not adequately adapt to a wider range of disaster scenarios if the data collection process places an excessive emphasis on certain types of disasters or relies too heavily on certain sources. Predictions from the model may be biased with respect to certain time periods or geographic areas if the dataset consists primarily of tweets from such time periods or locations.

5. Results and Discussions

In this section, we present the results of our experiments aimed at enhancing sentiment analysis prediction in binary text classification. We utilized a blend of sophisticated natural language processing models and classifiers, utilizing techniques such as TF-IDF, word embeddings (Word2Vec, GloVe), and advanced contextual embeddings (BERT, GPT). The models were evaluated on a dataset of 10,000 tweets, each classified as either related to real disasters (target 1) or not (target 0).

The Table I contains evaluation metrics for various traditional machine learning models applied to binary text classification using TF-IDF as the feature representation method. The evaluation metrics include precision (P), recall (R), F1-score (F1), accuracy (A), and area under the receiver operating characteristic curve (AUC) for two classes (Class 0 and Class 1). These metrics are important for assessing the performance of sentiment analysis prediction algorithms.

The efficacy of conventional models in sentiment analysis may be limited by their inability to handle intricate linguistic patterns and contextual subtleties found in textual data. Complex semantics and long-range dependencies are easily understood by advanced NLP models. In comparison to more sophisticated approaches, the simplicity of traditional models may lead to suboptimal performance.

Table 1. Results comparison. Traditional machine learning models (TF-IDF). The model column represents algorithms such as linear regression (LR), naive bayes (NB), support vector machine (SVM), random forest (RF), extreme gradient boosting (XGB), catboost (CatB), adaboost (AdaB), gradient boosting classifier (GDB), decision tree (DT) and k-nearest neighbors (K-NN)

Model	Evaluation parameters							
	Class 0			Class 1			A	AUC
	P	R	F1	P	R	F1		
LR	0.80	0.82	0.81	0.77	0.74	0.76	0.79	0.86
NB	0.77	0.92	0.84	0.86	0.65	0.74	0.80	0.85
SVM	0.80	0.85	0.83	0.80	0.73	0.76	0.80	0.86
RF	0.77	0.91	0.84	0.85	0.66	0.74	0.80	0.85
XGB	0.76	0.91	0.83	0.84	0.62	0.72	0.78	0.83
CatB	0.77	0.89	0.83	0.83	0.65	0.73	0.79	0.85
AdaB	0.71	0.88	0.79	0.78	0.55	0.64	0.73	0.80
GBC	0.72	0.93	0.81	0.85	0.55	0.67	0.76	0.81
DT	0.77	0.75	0.76	0.69	0.71	0.70	0.73	0.73
K-NN	0.77	0.86	0.81	0.79	0.66	0.72	0.77	0.81

Algorithms such as SVM, LR, NB, and RF achieve relatively high accuracy, indicating their overall effectiveness in binary classification tasks. They also exhibit high AUC values, which indicate strong discriminative power between classes. AdaB and GBC have slightly lower accuracy, indicating room for improvement in correctly classifying tweets. However, despite having lower AUC values, they still demonstrate a reasonable ability to distinguish between disaster-related and non-disaster tweets.

In terms of model selection, the Naive Bayes (NB) algorithm stands out for its excellent recall in identifying disaster-related tweets. This makes it particularly suitable for scenarios where minimizing false negatives is crucial. SVM demonstrates balanced precision, recall, and F1-score, making it a reliable choice for general applications where accuracy is crucial. RF, XGB, and CatB exhibit balanced performance across metrics, making them versatile choices for a wide range of applications. On the other hand, AdaBoost (AdaB) and Gradient Boosting Classifier (GBC), although reasonable, could benefit from further tuning to enhance their performance, particularly in precision and F1-score for class 1. Meanwhile, the DT (Decision Tree) model shows room for improvement, possibly requiring feature engineering or ensemble techniques to enhance its predictive power.

The Table II presents the evaluation metrics for advanced neural network models used in binary text classification. We can see that all models achieve reasonable accuracy, indicating their overall effectiveness in binary classification tasks. However, CNN, RNN, and LSTM show slightly higher accuracy, suggesting their proficiency in distinguishing between disaster-related and non-disaster tweets.

Regarding the AUC parameter, the CNN exhibits the highest AUC value (0.84), indicating a strong discriminative power between classes. RNN, LSTM, and GRU closely follow, highlighting their effective ability to distinguish disaster-related content.

Convolutional Neural Network (CNN) stands out for its high recall in identifying disaster-related tweets, making it well-suited for applications where minimizing missed disaster-related tweets is crucial. Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU) demonstrate balanced performance across various metrics, making them versatile choices for different applications.

Table 2. Results comparison. Deep learning models. The model column represents algorithms such as multi-layer perceptron (MLP), convolutional neural network (CNN), recurrent neural network (RNN), long short-term memory (LSTM), gated recurrent unit (GRU).

Model	**Evaluation parameters**							
	Class 0			**Class 1**			**A**	**AUC**
	P	**R**	**F1**	**P**	**R**	**F1**		
MLP	0.78	0.79	0.78	0.70	0.74	0.72	0.76	0.83
CNN	0.77	0.84	0.80	0.77	0.67	0.72	0.77	0.84
RNN	0.78	0.82	0.80	0.75	0.70	0.73	0.77	0.82
LSTM	0.78	0.81	0.79	0.74	0.70	0.72	0.76	0.81
GRU	0.76	0.78	0.77	0.71	0.69	0.70	0.74	0.81

Even though deep learning models may identify intricate relationships and patterns in text, they might not be easily interpreted. The intricacy of their decision-making procedures makes sentiment estimations less transparent. It is concerning because overfitting training data could make results less transferable to other datasets. Furthermore, huge labeled datasets may be necessary for deep learning models, which may not always be possible in some applications.

The table 3 presents the evaluation metrics for ensemble classifiers (Voting Classifier, Bagging Classifier, and Stacking Classifier) in binary text classification. These metrics provide insights into the performance of ensemble models in identifying disaster-related tweets. All ensemble models exhibit reasonable accuracy, suggesting their effectiveness in binary classification tasks. The Stacking Classifier (StackC) demonstrates superior discriminative power, indicating its strong ability to distinguish between disaster-related and non-disaster tweets.

Stacking Classifier (StackC) stands out for its high recall in identifying disaster-related tweets, making it ideal for applications where minimizing the number of missed disaster-related tweets is crucial. Bagging Classifier (BagC) demonstrates a good balance between precision and recall, indicating its effectiveness in capturing true disaster-related tweets while minimizing false positives. The Voting Classifier (VoteC) demonstrates moderate performance, providing a balanced approach to capturing both false negatives and false positives.

Further fine-tuning and optimizing ensemble strategies could enhance performance, especially in addressing subtle linguistic nuances in disaster-related tweets. Exploring additional ensemble techniques or combining different models may result in a more robust sentiment analysis system, capable of addressing the limitations of individual models. Additionally, it is crucial to underscore the importance of classifiers in addressing the scarcity of annotated data across diverse contexts, as the accurate identification and classification of semantic roles are essential for extracting valuable insights [25].

Table 3. Results comparison. Ensemble classifiers. The model column represents such classifiers as voting classifier (VoteC), bagging classifier (BagC) and stacking classifier (StackC).

Model	**Evaluation Parameters**							
	Class 0			**Class 1**			**A**	**AUC**
	P	**R**	**F1**	**P**	**R**	**F1**		
VoteC	0.78	0.81	0.79	0.74	0.70	0.72	0.76	0.83
BagC	0.77	0.88	0.82	0.80	0.66	0.72	0.78	0.85
StackC	0.80	0.85	0.82	0.79	0.73	0.76	0.80	0.86

Ensemble classifiers perform well in terms of accuracy in deep learning models. The stacking classifier is superior with an average improvement of over 4%. No notable improvement is observed when comparing NB, SVM, and RF, but there is a 7% increase in accuracy for AdaB and DT models. There is a disparity in F1 scores between class 0 and class 1, with class 0 being more viable due to a 7% increase in dataset data. Overall, the discussed models show improvements in F1 score and AUC compared to individual models. BagC and StackC have enhanced recall for Class 1, indicating better identification of positive sentiments. StackC demonstrates balanced performance with improved AUC.

The quality and diversity of the base models are critical to the performance of ensemble classifiers. The ensemble might not increase robustness if these base models share the same flaws or biases. Computational limitations may also arise for ensemble methods when handling large datasets or a wide variety of classifiers. Ensemble models may become more difficult to interpret, making it more challenging to determine how each model contributes to the final forecast.

As shown in Figure 2, logistic regression demonstrates a balanced ROC curve, indicating its ability to effectively discriminate between disaster and non-disaster tweets. NB shows a strong performance with an ROC curve skewed towards the upper left corner, indicating high true positive rates and low false positive rates. NB effectively captures disaster-related tweets while minimizing false positives. SVM exhibits a smooth ROC curve, highlighting its ability to maintain a balance between true positive and false positive rates. Other models showcase ROC curves that indicate reasonable discriminative power. While not as steep as the Disaster Tree (DT), they still effectively differentiate between tweets related to disasters and those that are not.

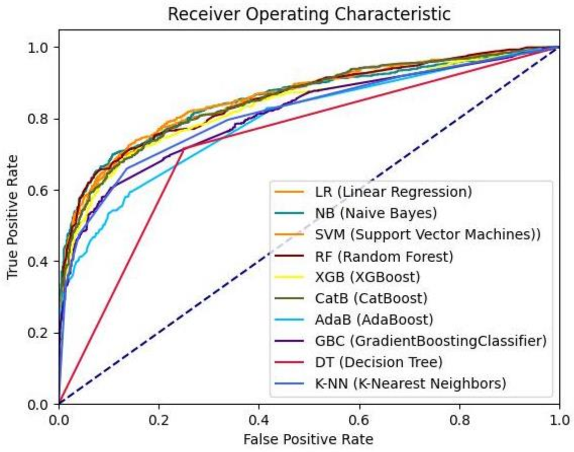

Fig.2. Receiver operating chracteristic for traditional machine learning models

In Figure 3, the CNN and Stacking Classifier models stand out with steep ROC curves, indicating their exceptional ability to accurately classify disaster-related tweets. These models are ideal for applications where minimizing false positives is a top priority, such as emergency response systems. MLP classifier, RNN, LSTM, GRU, Voting Classifier, and Bagging Classifier show reasonable discriminative power, making them versatile choices for a wide range of applications. These models offer a balanced approach between precision and recall, making them suitable for various contexts.

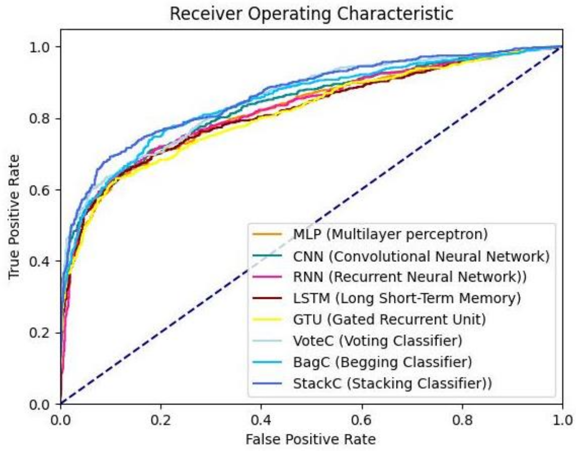

Fig.3. Receiver operating chracteristic for deep learning models and ensemble classifiers

To further analyze the specific errors made by the model, we can refer to Figure 4, which displays the confusion matrices. We can observe that SVM tends to capture some disaster-related tweets, resulting in a significant number of true positives. However, it also misclassifies a significant number of non-disaster tweets as disaster-related (false positives). This indicates a lack of specificity, possibly due to limitations in feature representation.

The model lacks sensitivity, suggesting that there is room for improvement in capturing subtle linguistic cues associated with disasters. Bagging Classifier maintains a balanced false positive and false negative count, achieving a similar trade-off between precision and recall as SVM. However, it misclassifies a significant number of non-disaster tweets as being related to disasters, leading to a high count of false positives. Convolutional Neural Network (CNN) demonstrates a relatively balanced false positive and false negative count and captures more true non-disaster tweets compared to Support Vector Machine (SVM), indicating a higher specificity. However, it struggles to capture subtle patterns in disaster-related language, which affects recall.

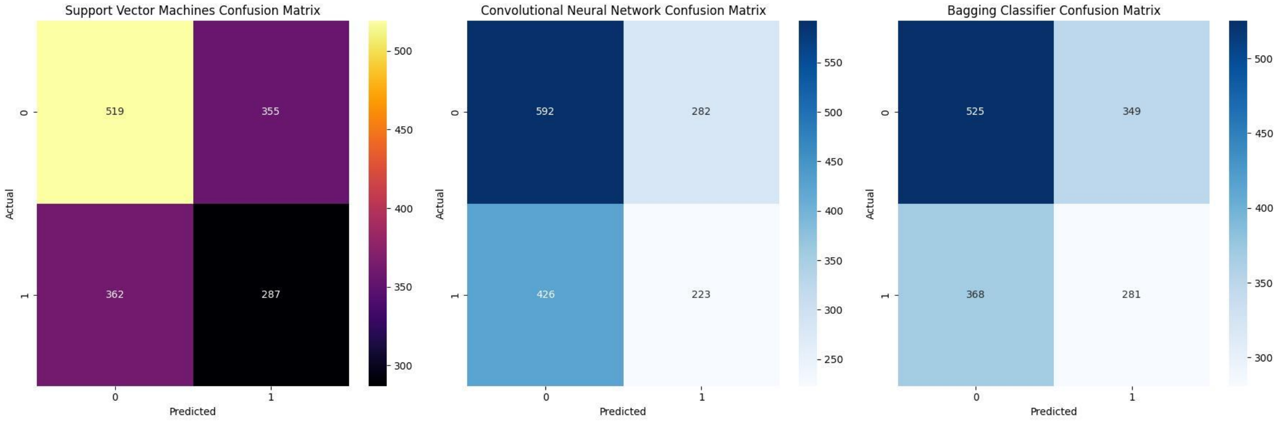

Fig.4. Confusiong matrices for support vector machines, convolutional nueral network and baggging classifer algorithms

Considering the complementary strengths and weaknesses of individual models, an ensemble approach should be explored. Combining the balanced trade-off of Support Vector Machines (SVM), the specificity of Convolutional Neural Networks (CNN), and the structure of the Bagging Classifier may result in a more robust sentiment analysis system that optimizes both precision and recall.

The improvements in robustness and precision have important consequences in fields like marketing, customer relationship management, social media monitoring, and product development. Better sentiment analysis can help businesses understand customer sentiments, tailor marketing strategies, and improve customer satisfaction. In social media, more accurate sentiment predictions can aid in sentiment monitoring, public opinion analysis, and crisis management. Our methodology can also be applied in healthcare to analyze patient reviews and assess the effectiveness of healthcare services. In the legal domain, sentiment analysis can assist in evaluating public sentiments toward legal cases and policy changes. Our approach enables a better understanding of textual sentiments and opens possibilities for diverse applications, highlighting the versatility and practical significance of our research findings.

6. Conclusions

In the ever-expanding landscape of digital communication, understanding sentiment in textual data has become crucial. Our research focused on enhancing sentiment analysis predictions in binary text classification, specifically in the challenging domain of disaster-related tweets. Through a thorough investigation of advanced Natural Language Processing (NLP) models and classifiers, our objective was to improve the precision, sensitivity, and specificity of sentiment analysis in this critical field.

Our investigation illuminated the diverse strengths and weaknesses of different models. Traditional machine learning models, such as Support Vector Machines (SVM), have shown balanced performance but struggle with detecting subtle linguistic cues, resulting in a trade-off between precision and recall. Deep learning models, including Convolutional Neural Networks (CNN), have demonstrated specificity but have struggled with capturing nuanced disaster-related language patterns, which highlights the complexity of the task. Ensemble classifiers, while promising, still struggle with optimizing the delicate balance between true positives and true negatives.

In our pursuit of refining these models, we have identified specific areas for improvement. Fine-tuning strategies, such as adjusting decision thresholds and experimenting with class weights, have emerged as key approaches to optimize the balance between false positives and false negatives. Furthermore, the potential of ensemble methods, which leverage the complementary strengths of individual models, offers an exciting prospect for future research. This approach promises to create a more robust sentiment analysis system.

In conclusion, our research advances our understanding of sentiment analysis in binary text classification, shedding light on both the challenges and opportunities within the domain of disaster-related tweets. By acknowledging the limitations and capitalizing on the strengths revealed through this study, we are paving the way for future advancements in sentiment analysis. This will ensure its continued relevance and effectiveness in deciphering the ever-changing language of the digital age.

References

[1] Y. Wang, G. Huang, J. Li, H. Li, Y. Zhou and H. Jiang, "Refined Global Word Embeddings Based on Sentiment Concept for Sentiment Analysis," in IEEE Access, vol. 9, pp. 37075-37085, 2021. DOI: 10.1109/ACCESS.2021.3062654.

[2] Y. Huang, R. Wang, B. Huang, B. Wei, S. L. Zheng and M. Chen, "Sentiment Classification of Crowdsourcing Participants' Reviews Text Based on LDA Topic Model," in IEEE Access, vol. 9, pp. 108131-108143, 2021.

[3] B. Pang, L. Lee, “Opinion Mining and Sentiment Analysis”, Foundations and Trends® in Information Retrieval, 2(1–2), 1-135, 2008.

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova., “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”, Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, 2019.

[5] Y. Kim, “Convolutional Neural Networks for Sentence Classification”, Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1746–1751, 2014.

[6] W. Song, H. Li, Q. Yu, W. Li, Bingxin Zhang, Qiujuan Zhang, Zhigang Liu, "The Multimedia Sentiment Model Based on Online Homestay Reviews ", International Journal of Engineering and Manufacturing, Vol.10, No.4, pp.13-23, 2020.

[7] J. Hirschberg, C. D. Manning, “Advances in natural language processing”, Science 349, 261-266, 2015.

[8] N. Rizun, Y. Taranenko, W. Waloszek, “Improving the Accuracy in Sentiment Classification in the Light of Modelling the Latent Semantic Relations”, Information 9, no. 12:307., 2018.

[9] S.-J. Wang, A. Mathew, Y. Chen, L.-F. Xi, L. Ma, J. Lee, “Empirical analysis of support vector machine ensemble classifiers”, Expert Systems with Applications, Volume 36, Issue 3, Part 2, Pages 6466-6476, 2009.

[10] L. Zhang, S. Wang, B. Liu, Bing, “Deep Learning for Sentiment Analysis: A Survey”, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2018.

[11] A. Gillioz, J. Casas, E. Mugellini and O. A. Khaled, "Overview of the Transformer-based Models for NLP Tasks", 15th Conference on Computer Science and Information Systems (FedCSIS), pp. 179-183, 2020.

[12] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, “Improving language understanding by generative pre-training”, 2018.

[13] L. Breiman, “Random Forests”, Machine Learning 45, 5–32, 2001. DOI: 10.1023/A:1010933404324

[14] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space”, arXiv preprint, 2013. arXiv:1301.3781.

[15] L. A. Mullen., K. Benoit, O. Keyes, D. Selivanov, J. Arnold, “Fast, Consistent Tokenization of Natural Language Text”, The Journal of Open Source Software 3(23):655, 2018.

[16] J. Pennington, R. Socher, and C. Manning, “GloVe: Global Vectors for Word Representation”, In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1532–1543, 2014.

[17] D. Chen and C. Manning, “A Fast and Accurate Dependency Parser using Neural Networks”, In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 740–750, 2014.

[18] G. Lample, M, Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, “Neural Architectures for Named Entity Recognition”, In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 260–270, 2016.

[19] C. Zhao, S. Sahni, “String correction using the Damerau-Levenshtein distance”, BMC Bioinformatics 20 (Suppl 11), 277, 2019. DOI: 10.1186/s12859-019-2819-0

[20] P. Santoso, P. Yuliawati, R. Shalahuddin, A. Wibawa, “Damerau Levenshtein Distance for Indonesian Spelling Correction”, Journal Informatika. 13. 11, 2019.

[21] I. Gupta., S. Mittal, A. Tiwari, P. Agarwal, A. Kumar, Singh, “TIDF-DLPM: Term and Inverse Document Frequency based Data Leakage Prevention Model”, arXiv.org, 2022.

[22] Y. Goldberg, “A primer on neural network models for natural language processing”, J. Artif. Int. Res. 57, 1, 345–420, 2016. DOI: 10.1613/jair.4992

[23] A. Jurek, Y. Bi, S. Wu, and C. Nugent, “A survey of commonly used ensemble-based classification techniques”, The Knowledge Engineering Review, 29(5), 551–581, 2014.

[24] M. Hossin, and M. N. Sulaiman, “A review on evaluation metrics for data classification evaluations”, International journal of data mining & knowledge management process, 5(2), 1, 2015.

[25] K. Potapova, M. Nalyvaichuk, V. Meliukh, S. Gurynenko, K. Koliada, A. Scherbyna, “Semantic role labelling and analysis in economic and cybersecurity contexts using natural language processing classifiers”, Economic and cyber security. Kharkiv: PC TECHNOLOGY CENTER, 88–122, 2023.

Authors’ Profiles

Zhengbing Hu, Prof., Deputy Director, International Center of Informatics and Computer Science, Faculty of Applied Mathematics, National Technical University of Ukraine “Kyiv Polytechnic Institute”, Ukraine. Adjunct Professor, School of Computer Science, Hubei University of Technology, China. Visiting Prof., DSc Candidate in National Aviation University (Ukraine) from 2019.

Major research interests: Computer Science and Technology Applications, Artificial Intelligence, Network Security, Communications, Data Processing, Cloud Computing, Education Technology.

Ivan Dychka: D.S., Professor, Dean of Faculty of Applied Mathematics, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ukraine.

Research Interests: Computer Systems and Networks Software, Automated Control Systems, Intelligence and Expert Systems, Databases and Knowledge Bases, Information Security Software for Computer Systems and Networks.

Kateryna Potapova: Ph.D, associate professor of the Department of System Programming and Specialized Computer Systems Igor Sikorsky Kyiv Polytechnic Institute, Kyiv, Ukraine.

Research Interests: Control Systems and Information Technology, Computer Systems and Networks Software, Intelligence and Expert Systems, Neural Networks and Image Processing, Artificial Intelligence, Machine Learning.

Vasyl Meliukh was born on March 13, 2001. He received his bachelor’s degree in computer engineering (June 2022) at the System Programming and Specialized Computer Systems Department at National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine. He is currently a master’s degree student in the System Programming and Specialized Computer Systems Department at the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine.

His main research interests are Artificial Intelligence, Machine Learning, Pattern and Sequence Recognition, Data Processing, Semantic Role Labeling, Natural Language Processing.

How to cite this paper: Zhengbing Hu, Ivan Dychka, Kateryna Potapova, Vasyl Meliukh, "Augmenting Sentiment Analysis Prediction in Binary Text Classification through Advanced Natural Language Processing Models and Classifiers", International Journal of Information Technology and Computer Science(IJITCS), Vol.16, No.2, pp.16-31, 2024. DOI:10.5815/ijitcs.2024.02.02