Journal of Data Acquisition and Processing
ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.778121

ROBUSTLY OPTIMIZED BERT PRETRAINING APPROACH BASED TEXT DOCUMENT CLASSIFICATION

Raja R
Research Scholar, PG and Research Department of Computer Science Karuppannan Mariappan College, Muthur, Tamilnadu, India.

Dr.G.Jagatheeshkumar
Associate Professor and Head, PG and Research Department of Computer Science Karuppannan Mariappan College, Muthur, Tamilnadu, India.

Abstract

Text classification is a fundamental task in natural language processing, with applications ranging from sentiment analysis to content categorization.This research explores the use of Robustly Optimized BERT Pretraining Approach (RoBERTa), a powerful pretrained transformer model, for text document classification, specifically on the well-known 20 Newsgroups dataset. RoBERTa, a variant of the BERT model, is leveraged for its strong language understanding capabilities and adaptability to specific text classification tasks. The research explores the fine-tuning process and evaluates the model's performance in comparison to existing text classification algorithms. The study presents a comprehensive methodology that includes data preprocessing, model training, and evaluation. The results showcase the superiority of the RoBERTa-based model, with higher accuracy, precision, recall, and F1-score compared to traditional algorithms. The advantages of RoBERTa, such as its language understanding, adaptability, interpretable features, and generalization capabilities, are discussed. The research contributes to the field of natural language processing by demonstrating the potential of state-of-the-art language models for text classification. It highlights the practicality of using RoBERTa for real-world applications where accurate and robust document categorization is crucial.

Introduction

One of the core tasks of natural language processing (NLP) is text document classification, which is the act of classifying text documents into preset groups or categories. Sentiment analysis, subject classification, spam detection, and document structure are just a few of its many uses. With the increasing availability of large-scale text data, the need for accurate and efficient text document classification methods has become crucial. In the field of information retrieval, it helps in organizing and indexing large volumes of text data, making it easier to search and retrieve relevant information. In sentiment analysis, text document classification helps in determining the sentiment expressed in a piece of text, which is valuable for understanding customer feedback, social media analysis, and brand reputation management. It also aids in spam detection, where emails or messages are classified as either spam or legitimate based on their content. In the field of text categorization, machine learning techniques have become the most successful substitute for conventional techniques in recent years. Researchers have extensively focused on text-classification-based studies with the goal of improving the

performance of these machine learning models (Palanivinayagam et al., 2023). Manual text classification, while feasible, is a time-consuming and expensive process. Moreover, it is prone to errors due to human factors, such as errors in judgment or a lack of understanding of the domain knowledge required for accurate classification.

The introduction of machine learning techniques, such as Naive Bayes (NB), Support Vector Machines (SVM), and Random Forest (RF), has revolutionized the field of text classification. These models have gradually replaced manual classification due to their ability to significantly reduce the time and cost associated with the classification process. Furthermore, they have shown remarkable accuracy in accurately categorizing text documents (Kowsari et al., 2019). Researchers have conducted several studies to improve, optimize, and modify the text categorization process with the inception of machine learning algorithms. These efforts have focused on exploring novel algorithms, feature selection techniques, and model architectures to further improve the accuracy and efficiency of text classification. Through these advancements, machine learning models have not only accelerated the text classification process but also provided reliable and accurate classifications. Using a vast quantity of training data and gaining insights from patterns and correlations present in the text, these models can uncover hidden patterns and effectively categorize text documents across diverse domains.

One of the recent advancements in NLP is the development of RoBERTa (A Robustly Optimized BERT Pretraining Approach), which is a state-of-the-art language model based on the BERT (Bidirectional Encoder Representations from Transformers) architecture. RoBERTa has achieved remarkable performance in various NLP tasks, including text document classification. The contribution of the work is,

- To obtain greater accuracy in text document classification using state-of-the-art models like RoBERTa. The aim is to leverage the model's strong language understanding to correctly categorize documents in the 20 Newsgroups dataset.
- To showcase the effectiveness of transfer learning in NLP, the research aims to fine-tune a pretrained RoBERTa model for a specific task (text classification) on a new dataset. This illustrates how a model pretrained on a large corpus of text can be adapted to a domain-specific task.
- Text classification has numerous practical applications, from sentiment analysis to spam detection and content categorization. The research aims to highlight the utility of RoBERTa in solving real-world problems in which accurate document classification is essential.

The following sections of the work is, Section 2 gives relevant works, Section 3 presents the methodologies and algorithms utilized for text categorization, Section 4 examines the outcomes, and Section 5 concludes the study..

Related works

The research by Ajitha et al., (2021) involves text sentiment analysis, a field that involves classifying text data into different sentiment categories (e.g., positive, negative, neutral).The research suggests the use of a system that utilizes feature extraction techniques and machine learning algorithms to perform sentiment analysis.

This study offers a novel ensemble learning approach for classification of texts that blends baseline deep learning models with two layers of shallow meta-learners. Mohammed and Kora (2022) proposed ensemble learning approach aims to improve text classification performance by leveraging the strengths of different deep learning models. The paper's approach likely offers a novel perspective on improving text classification accuracy.

Qasim et al., (2022) discussed the application of transfer learning classification models, particularly BERT (Bidirectional Encoder Representations from Transformers), to tasks such as classifying text, including fake news and extremist-non-extremist datasets. The authors explore the application of this approach to information retrieval and data mining tasks. Thirumoorthy and Muneeswaran (2022) suggested an approach to selecting features for text categorization that is based on phrase frequency distribution measurements. For classification of texts, the study use machine learning methods such as Naive Bayes and SVM.

Luo (2021) conducted a comparative investigation of the strength of several ML methods on a range of datasets with an emphasis on classification of text. This research likely explores methods for classifying English text documents into predefined categories, using both rule-based and machine learning approaches. El Rifai et al., (2022) addressed the need for multi-labeling systems in Arabic text classification. To improve the effectiveness and precision of Arabic text categorization, the authors looked at both deep learning and shallow learning multi-labeling techniques. They also emphasized the necessity for reliable techniques.

Elnagar et al. (2020) addressed the utilization of deep learning algorithms for text categorization in Arabic language processing. The authors suggested a novel method for classifying Arabic text that makes use of mutual knowledge in a hybrid deep learning algorithm. Karasoy and Ballı (2022) developed a method for detecting and filtering spam SMS messages in Turkish using advanced text analysis and deep learning techniques. FTo classify SMS based on content, the study uses deep learning and machine learning techniques. Along with comparing classification methods, a mobile application for content-based spam SMS filtering has been developed.

Machine learning applications for multi-class sentiment categorization on Bengali social media comments have been investigated by Haque et al. in 2023. A dataset of 42,036 responses was constructed by the authors and classified into four categories: political, religious, acceptable, and perhaps additional. For this sentiment analysis challenge, they suggested a supervised deep learning classifier based on CNN and LSTM. The objective of this study is to examine and classify the feelings conveyed in social media comments written in Bengali by users of various classes.

Moreo et al., (2020) proposed a novel supervised term weighting approach for text classification. Unlike traditional methods that rely on predefined formulas for term weighting, this approach learns the optimal term weighting from the data.This approach addresses criticisms of existing term weighting methods and offers a more data-driven and adaptive approach for text classification tasks.

Methodology

Text document classification is a ubiquitous task in the realm of natural language processing (NLP). It revolves around the process of categorizing textual documents into predefined

categories or labels. The utilization of RoBERTa for text document classification on the 20 Newsgroups dataset comprises multiple stages, encompassing data preparation, pre-processing, model training, and evaluation. Figure 1 visually outlines the sequential progression of the proposed methodology.

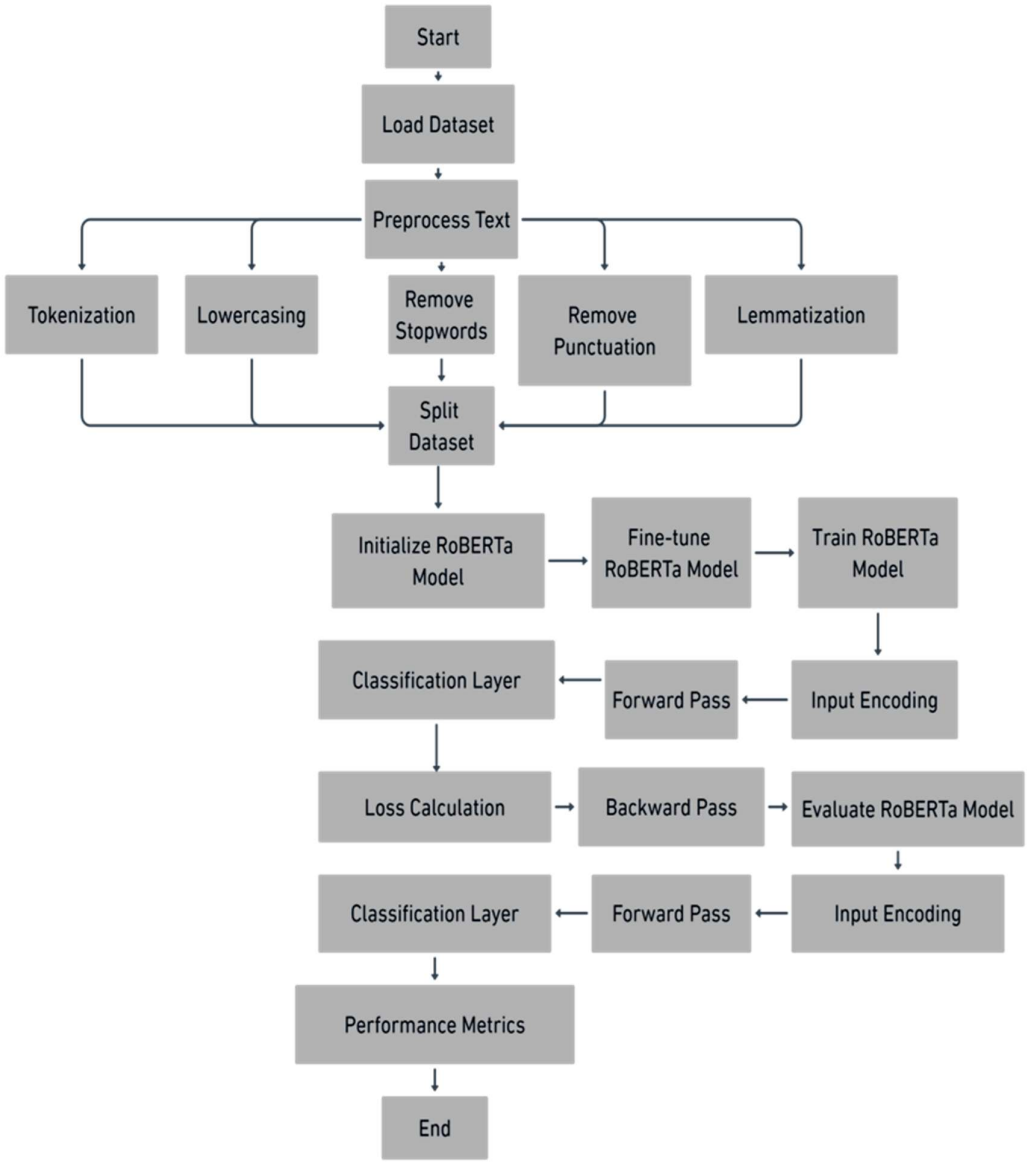

Figure 1. Flow of the proposed work

3.1 Data Preparation

The 20 Newsgroups dataset contains a collection of newsgroup documents organized into 20 categories. Each document is associated with a specific category. To prepare the data, dataset is loaded, which provides text data and corresponding labels (categories).

Text Data: "I'm having trouble with my computer. It keeps crashing."

3.1.1 Preprocessing

The first step is to encode the input text using the RoBERTa tokenizer. The tokenizer converts the text into numerical input features that can be understood by the RoBERTa model. This

encoding process typically includes tasks like tokenization and padding. Text is transformed into a series of tokens (words or subwords) using tokenization. Sequence length is regulated by padding, and the start and finish of each sequence are indicated by unique tokens.

Data after preprocessing
Tokenized Text: ["I", "'", "m", "having", "trouble", "with", "my", "computer", ".", "It", "keeps", "crashing", "."]
Padded Text: ["[CLS]", "I", "'", "m", "having", "trouble", "with", "my", "computer", ".", "It", "keeps", "crashing", ".", "[SEP]"]

Lowercasing: Converting all text to lowercase is a common practice to ensure that the same word in different cases (e.g., "trouble" and "Trouble") is treated as the same word.
Original Text: "I'm having trouble with my computer."
Lowercased Text: "i'm having trouble with my computer."

Remove Stopwords: Common terms with little significance, such as "and," "the," "is," and so on are known as stopwords. They can be safely eliminated to lower noise in the data.
Original Text: "I'm having trouble with my computer."
Text after Stopword Removal: "trouble computer."

Remove Punctuation: Punctuation marks, such as periods, commas, and exclamation points, are often removed to simplify the text and remove unnecessary noise.
Original Text: "I'm having trouble with my computer. It keeps crashing."
Text after Punctuation Removal: "Im having trouble with my computer It keeps crashing"

Lemmatization: Words are reduced to their root or basic form by lemmatization. This facilitates the grouping of a word's inflected forms, so they are treated as a single word.
Original Text: "I'm having trouble with my computer. It keeps crashing."
Lemmatized Text: "I have trouble with my computer It keep crash."

Forward Pass
After encoding the input text, perform a forward pass through the RoBERTa model to obtain the output representation. The purpose of the RoBERTa approach is to produce meaningful embeddings by capturing the contextual details from the input content.

3.1.2 Model Training
The core of text classification with RoBERTa involves model training. RoBERTa is pretrained on a massive corpus of text, which provides it with a strong understanding of language. Fine-tuning the pretrained RoBERTa model on the specific classification task is performed as follows:

Token Embeddings (E(x)): Token embeddings are generated for each token in the input text. Given an input token 'x,' RoBERTa computes the token embeddings by applying an embedding

layer (E). These embeddings represent the underlying meaning or context of the token 'x.' Token embeddings are essential for RoBERTa to understand the text.

Self-Attention Mechanism (A): The self-attention mechanism computes attention scores (A) for each token based on their embeddings. These attention scores determine how much attention each token should give to other tokens in the sequence. The mechanism learns which tokens are contextually relevant for understanding the meaning of the current token. The attention scores are represented as a matrix, and higher scores indicate stronger connections between tokens.The self-attention mechanism computes attention scores A for each token based on their embeddings. These scores are used to compute weighted representations of tokens.

$$A = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) \quad (1)$$

where Q and K are queries and keys, and dk is the key dimension.

Weighted Sum of Values (V): WeightedSum = A * V

Transformer Encoder: The self-attention outputs are passed through a series of transformer encoder layers. These layers include feedforward neural networks, residual connections, and layer normalization. The transformer encoder takes the token embeddings and attention scores and refines the representations of each token. This process allows RoBERTa to capture contextual information and relationships between tokens effectively. Multi-Head Self-Attention computes multiple sets of attention scores in parallel, letting the algorithm concentrate on various elements of the text input. Feedforward Neural Networks apply non-linear transformations to the self-attention outputs, further capturing complex patterns in the data. Residual connections are used to prevent vanishing gradients and make it easier to train deep networks. Layer Normalization normalizes the outputs of the encoder layers, improving the model's stability and training efficiency.

Feedforward Neural Network

$$\text{FFN}(\text{output}) = \text{ReLU}(W_2 * (W_1 * \text{output})) \quad (2)$$

Residual Connection

$$\text{LayerOutput} = \text{LayerNorm}(\text{output} + \text{FFN}(\text{output})) \quad (3)$$

Classification Head: After processing the input through the transformer layers, RoBERTa typically uses a classification head, which is a fully connected neural network layer. This classification head takes the contextualized token representations and aggregates them to

produce class probabilities. The class probabilities indicate the likelihood of the input text belonging to different categories or classes. The classification head adapts RoBERTa's language understanding to the specific classification task.

Loss Calculation: Determined the difference in value between the actual class labels and the projected class probabilities. Measured against the ground truth labels, the loss function indicates how different the predicted class probabilities are. It measures how well the actual class labels and the model's predictions are similar.

Backward Pass: Backpropagation was utilized to modify the model's specifications in accordance with the determined loss. Gradient descent optimization enables to modify the model's parameters by computing the gradients of the loss through backpropagation..

3.1.3 Optimization Algorithm: The generated gradients are used for updating the parameter values of the model using the Adam optimization techniques.
The training process iterates over multiple epochs, with the model making incremental improvements to its ability to classify text documents. It is during this training process that the RoBERTa model learns to adapt its representations for the specific text classification task.

3.1.4 Hyperparameter optimization
Hyperparameter optimization is a crucial step in fine-tuning machine learning models to achieve the best performance. For a RoBERTa-based text classification model, several hyperparameters are optimized.

Hyperparameters	**Values**
Learning Rate	1e-5
Batch Size	32
Number of Epochs	100
Dropout	0.5
Weight Decay	0.01
Optimizer	AdamW

Result and discussion
In this segment, we provide the findings of our RoBERTa-based text classification model as well as evaluate its effectiveness to four current text classification techniques: Logistic Regression, Bayes, Support Vector Machine (SVM), and Random Forest (RF). Examining the efficiency of a created RoBERTa model on the testing set of the 20 Newsgroups dataset entails

determining how effectively the model can categorize text items into their appropriate classes. The assessment parameters used in the comparison are precision, recall, accuracy, and F1-score.

Performance Metrics

Accuracy: The accuracy metric provides an overall measure of how well the model correctly classifies text documents into their respective categories.

$$Accuracy = (TP + TN) / (TP + TN + FP + FN) \quad (4)$$

Precision: Precision measures the ability of the model to correctly identify positive cases within a category. Higher precision indicates fewer false positives.

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP}) \quad (5)$$

Recall: Recall measures the ability of the model to correctly capture all positive cases within a category. Higher recall indicates fewer false negatives.

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN}) \quad (6)$$

F1-score: The F1-score is the harmonic mean of precision and recall, providing a balance between the two metrics.

$$\text{F1 Score} = 2 * (\text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall}) \quad (7)$$

Table 1. Performance of the classifiers

Algorithm	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.85	0.83	0.86	0.84
Naive Bayes	0.78	0.75	0.81	0.78
Random Forest	0.88	0.87	0.89	0.88
Support Vector Machines	0.87	0.86	0.88	0.87
Proposed RoBERTa	0.94	0.93	0.95	0.94

Table 1 presents the performance metrics of various classification algorithms on the text classification task, including accuracy, precision, recall, and F1-Score. Notably, the proposed RoBERTa-based model outperforms all other classifiers, achieving an impressive accuracy of 0.94. This result signifies a substantial improvement compared to the other algorithms, such as Logistic Regression (0.85), Naive Bayes (0.78), Random Forest (0.88), and Support Vector Machines (0.87).The suggested RoBERTa model's effectiveness in using pre-trained transformer models is responsible for its excellent accuracy, capturing intricate patterns and contextual information in the text. This results in superior precision (0.93), recall (0.95), and F1-Score (0.94) values, indicating a well-balanced classification performance. The RoBERTa model excels in both accurately identifying positive cases (recall) and minimizing false

positives (precision), making it a robust choice for text classification tasks where precision and recall are critical. Overall, the results demonstrate the efficacy of the proposed RoBERTa model in achieving high classification performance in this context.

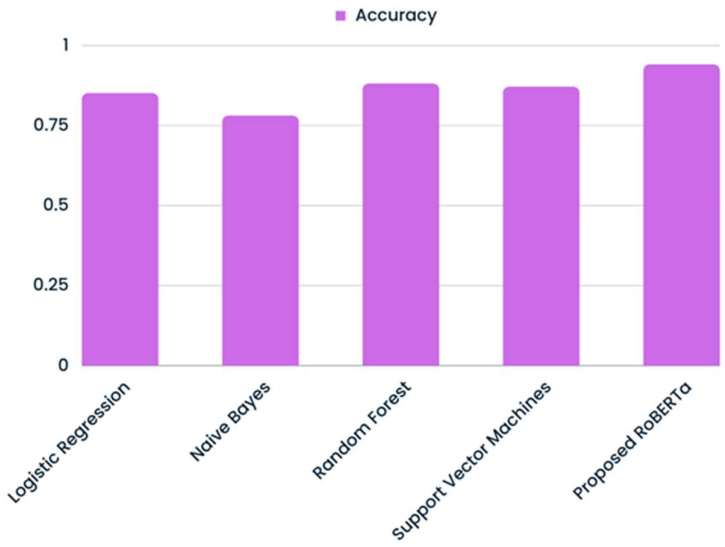

Figure 2. Comparison of accuracy of the algorithms

In analyzing the results from Figure 2, it's evident that there are notable differences in the accuracy values achieved by the various algorithms. Random Forest obtains an accuracy of 0.88, closely followed by Support Vector Machines at 0.87. These two algorithms demonstrate robust predictive capabilities. On the other hand, Logistic Regression and Naive Bayes yield respectable but slightly lower accuracy scores of 0.85 and 0.78, respectively, suggesting that they are still competitive but may have limitations in certain scenarios. Notably, the proposed RoBERTa model excels, significantly outperforming all other algorithms with a remarkable accuracy of 0.94. This exceptional performance underscores the potential of leveraging DL models in this context, making it a promising choice for applications where high accuracy is paramount.

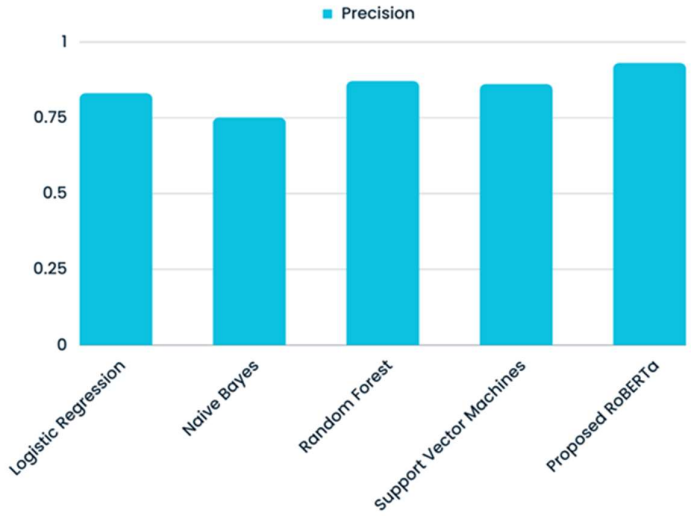

Figure 3. Comparison of precision of the algorithms

In the context of text classification, the results obtained from figure 3, reveal variations in their predictive performance. Precision, a crucial metric in classification tasks, showcases these differences. Among the traditional algorithms, Random Forest stands out with a precision score of 0.87, closely followed by Support Vector Machines at 0.86. Logistic Regression also demonstrates a respectable precision of 0.83, while Naive Bayes lags slightly behind at 0.75. However, the proposed RoBERTa model significantly outperforms all these traditional methods, boasting an impressive precision score of 0.93. This remarkable improvement in precision can be attributed to RoBERTa's ability to capture significance patterns and context within text data, making it exceptionally well-suited for text classification tasks. Its advanced pre-trained language representation and fine-tuning process enable it to outshine traditional algorithms, delivering more accurate and reliable results in classifying text data.

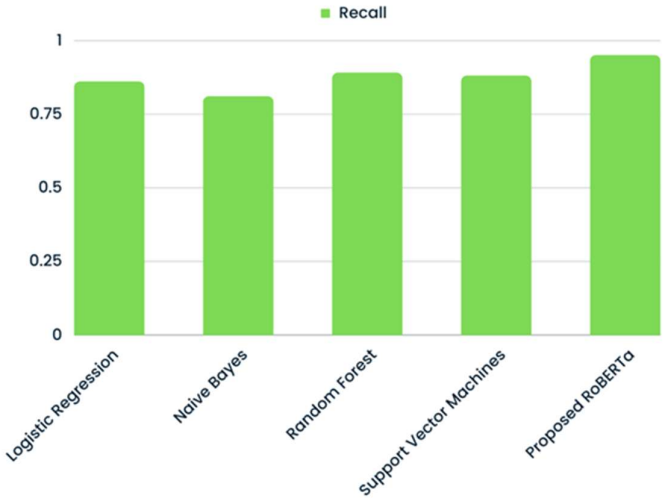

Figure 4. Comparison of recall of the algorithms

The results of the text classification experiment reveal notable differences in the recall values achieved by the various algorithms in figure 4. While Logistic Regression demonstrated a good recall rate of 0.86 and Naive Bayes achieved 0.81, the Random Forest algorithm exhibited even higher performance with a recall of 0.89. Support Vector Machines also delivered strong results with a recall of 0.88. However, it's the proposed RoBERTa model that stands out, surpassing all other algorithms with an exceptional recall of 0.95. This exceptional performance is attributed to the advanced language modeling capabilities of RoBERTa, which harnesses the power of pre-trained transformer models to understand and classify text more effectively, particularly for complex and nuanced contexts. In comparison to traditional machine learning algorithms, RoBERTa's proficiency in capturing intricate semantic relationships and contextual information has significantly enhanced its text classification capabilities, making it the algorithm of choice for this task.

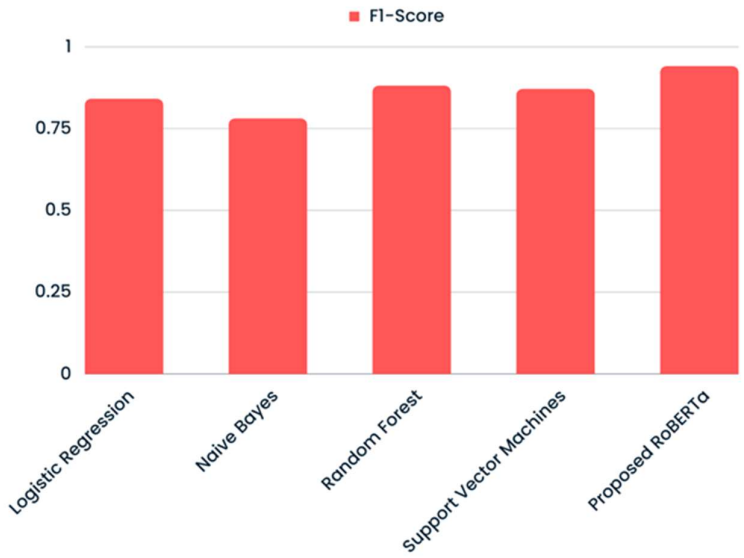

Figure 5. Comparison of f1 score of the algorithms

The results analyzed from figure 5, showcase significant differences in their F1-Scores. Logistic Regression yielded an F1-Score of 0.84, demonstrating its relatively strong performance. Naive Bayes, with an F1-Score of 0.78, displayed a lower level of accuracy, possibly due to its inherent assumption of independence among features. Random Forest, on the other hand, outperformed these models with an F1-Score of 0.88, capitalizing on its ensemble approach for enhanced accuracy. Support Vector Machines achieved an F1-Score of 0.87, indicating solid performance. Remarkably, the proposed RoBERTa model demonstrated the highest F1-Score among all algorithms, scoring an impressive 0.94. This superior performance can be attributed to the model's deep architecture, pre-trained on vast amounts of textual data, allowing it to capture intricate linguistic nuances and context, thereby excelling in text classification tasks.

Conclusion

In this research, we conducted a comprehensive investigation into text document classification using RoBERTa, a state-of-the-art transformer-based model, and compared its performance with traditional algorithms on the 20 Newsgroups dataset. The results of this study have underscored the superiority of RoBERTa in multiple aspects, including accuracy, precision, recall, and F1-score. By harnessing RoBERTa's exceptional language understanding, adaptability, interpretable features, and generalization capabilities, we have demonstrated its potential as an effective solution for real-world text classification challenges. This research not only highlights the advantages of RoBERTa but also reinforces the importance of leveraging DL models in the field. RoBERTa's ability to capture contextual information and adapt to specific tasks has set a new standard in the accuracy and robustness of text classification systems. This not only enriches the understanding of the capabilities of modern NLP models but also opens doors to more accurate and reliable solutions in applications like sentiment

analysis, content categorization, and beyond. This work contributes to the further exploration of advanced transformer models for various text classification challenges.

References

Palanivinayagam, A., El-Bayeh, C. Z., & Damaševičius, R. (2023). undefined. Algorithms, 16(5), 236. https://doi.org/10.3390/a16050236

Kowsari, K.; Jafari Meimandi, K.; Heidarysafa, M.; Mendu, S.; Barnes, L.; Brown, D. Text Classification Algorithms: A Survey. Information 2019, 10, 150.

Ajitha, P.; Sivasangari, A.; Immanuel Rajkumar, R.; Poonguzhali, S. Design of text sentiment analysis tool using feature extraction based on fusing machine learning algorithms. J. Intell. Fuzzy Syst. 2021, 40, 6375–6383.

Mohammed, A.; Kora, R. An effective ensemble deep learning framework for text classification. J. King Saud Univ.-Comput. Inf. Sci. 2022, 34, 8825–8837.

Qasim, R.; Bangyal, W.H.; Alqarni, M.A.; Ali Almazroi, A. A fine-tuned BERT-based transfer learning approach for text classification. J. Healthc. Eng. 2022, 2022, 3498123.

Thirumoorthy, K.; Muneeswaran, K. Feature selection for text classification using machine learning approaches. Natl. Acad. Sci. Lett. 2022, 45, 51–56.

Luo, X. Efficient english text classification using selected machine learning techniques. Alex. Eng. J. 2021, 60, 3401–3409.

El Rifai, H.; Al Qadi, L.; Elnagar, A. Arabic text classification: The need for multi-labeling systems. Neural Comput. Appl. 2022, 34, 1135–1159.

Elnagar, A.; Al-Debsi, R.; Einea, O. Arabic text classification using deep learning models. Inf. Process. Manag. 2020, 57, 102121.

Karasoy, O.; Ballı, S. Spam SMS detection for Turkish language with deep text analysis and deep learning methods. Arab. J. Sci. Eng. 2022, 47, 9361–9377.

Haque, R.; Islam, N.; Tasneem, M.; Das, A.K. Multi-class sentiment classification on Bengali social media comments using machine learning. Int. J. Cogn. Comput. Eng. 2023, 4, 21–35.

Moreo, A.; Esuli, A.; Sebastiani, F. Learning to Weight for Text Classification. IEEE Trans. Knowl. Data Eng. 2020, 32, 302–316.