UDC 004.89.00
MSC 00X00

Comparison of hybrid deep learning models in the context of their application to text classification tasks

This study presents a comprehensive comparison of three hybrid deep learning architectures for text classification: an RNN with an attention mechanism, a CNN-RNN hybrid model, and BERT-based models integrated with various classifiers. Evaluated on benchmark datasets such as BBC News and AG News, these approaches are assessed based on accuracy, precision, recall, F1-score, and computational efficiency. Experimental results indicate that BERT-based models achieve the highest classification performance; however, their substantial computational requirements may limit applicability in resource-constrained settings. In contrast, the RNN with attention mechanism provides a balanced trade-off between performance and efficiency, effectively capturing long-term dependencies in text. The CNN-RNN hybrid, while slightly less accurate, excels in rapid training and robust performance, making it a viable option for real-time applications. Analysis of confusion matrices reveals that misclassifications primarily occur in categories with overlapping semantic content, notably between Business and Sci/Tech. The findings of this research highlight the effectiveness of combining transformer-based models with classical ensemble methods for text classification. Experimental results demonstrate that while BERT-based models achieve superior classification performance, their integration with ensemble techniques further enhances robustness and generalization, particularly in handling ambiguous or semantically overlapping categories. This hybrid approach balances high accuracy with improved stability across diverse datasets, offering a promising direction for practical applications.

Keywords: natural language processing, text classification, Machine Learning, Data Preprocessing, Convolutional Neural Network, Recurrent Neural Network.

1. Introduction.

Text classification is a fundamental task in natural language processing (NLP), playing a key role in areas such as sentiment analysis, spam detection, and topic categorization. With the growing adoption of deep learning, neural network models have become the preferred approach for text classification, surpassing traditional machine learning methods that rely on manual feature engineering. Various neural architectures offer unique strategies for handling text classification tasks. Recurrent Neural Networks (RNNs) are well-suited for processing sequential data, allowing them to capture word order and dependencies. Convolutional Neural Networks (CNNs), though originally designed for image recognition, have shown their utility in text classification by identifying important local patterns. In recent years, transformer-based models, such as BERT, have revolutionized NLP by providing powerful contextualized representations and enabling transfer learning at scale. This study compares three approaches: (1) RNN with Attention, (2) RNN combined with CNN, and (3) BERT-based models with different classifiers. The evaluation is performed on benchmark text classification datasets, considering metrics such as accuracy, computational efficiency, and overall effectiveness. Insights gained from comparing these models will help in understanding their applicability in diverse text classification scenarios.

2. The aim and objectives of the study

The primary objective of this study is to identify the most effective hybrid deep learning model for text classification. The research focuses on evaluating different neural architectures that integrate recurrent, convolutional, and transformer-based approaches to enhance classification performance.

The specific objectives of the study are as follows:

To investigate the impact of incorporating attention mechanisms in recurrent networks for improving the handling of long textual sequences.

To analyze the effectiveness of combining convolutional and recurrent layers in capturing both local and sequential dependencies in text data.

To assess the performance of transformer-based models as feature extractors, particularly when coupled with traditional classification algorithms.

To compare these hybrid models based on accuracy, efficiency, and adaptability to different types of text classification tasks.

The scientific novelty of this study lies in the comprehensive comparative analysis of hybrid deep learning architectures that combine RNNs, CNNs, and transformer-based models for text classification tasks. Unlike prior works that typically focus on single architectures or limited combinations, this research systematically explores the synergy of recurrent and convolutional layers, the integration of attention mechanisms, and the utilization of pretrained transformer-based models as feature extractors. By evaluating these models across diverse datasets and metrics, the study uncovers new insights into their strengths, limitations, and optimal use cases, thereby contributing to the advancement of hybrid architectures in natural language processing.

3. Literature review

3.1 RNN + Attention

Modern text classification tasks require efficient methods for analyzing textual data while considering the complex structure of language. Deep learning, with its ability to extract multidimensional hierarchical representations, has become fundamental in developing high-accuracy models for this field.

Recurrent neural networks (RNNs) remain an important tool in text classification due to their ability to model complex linguistic structures and capture long-term dependencies in sequences [1]. RNNs are widely applied in tasks such as machine translation, sentiment analysis, and text generation, demonstrating competitive accuracy.

One practical application of RNNs is sentiment analysis in social media. A study on Twitter sentiment analysis demonstrated that integrating the attention mechanism into RNNs improves classification accuracy by allowing the model to better capture contextual dependencies in messages [2].

The attention mechanism plays a crucial role in enhancing model performance by enabling the network to focus on the most relevant parts of the input text. A proposed text classification model combining Graph Attention Networks and capsule networks showed that the use of attention mechanisms improves the representation of weights during training, leading to higher classification accuracy [3]. Experimental results indicate that this method outperforms traditional approaches, such as standard recurrent and convolutional neural networks, in various text classification tasks.

Further advancements involve the application of self-attention mechanisms. The SA-SGRU model, which integrates self-attention and Skip-GRU, enhances key feature extraction efficiency in text classification tasks [4]. Another study on sentiment analysis with RNNs highlights how attention mechanisms contribute to improved classification accuracy by effectively handling contextual information [4].

Additionally, an RNN-based sentiment analysis method demonstrated the capability of recurrent networks to retain information about previous elements in a sequence, which is particularly beneficial for processing long texts such as user reviews and news articles [5].

Overall, research supports the effectiveness of attention mechanisms and recurrent neural networks in text classification tasks. Incorporating self-attention and hybrid architectures, such as combining RNNs with CNNs or capsule networks, leads to improved classification accuracy and a better understanding of textual data.

3.2 CNN + RNN

Recurrent Neural Networks (RNNs) have played a crucial role in processing sequential data, particularly in natural language processing (NLP). In [6], the authors provide a comprehensive analysis of various RNN architectures, such as Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU), highlighting their ability to capture long-range dependencies in text. Despite their widespread use, the study points out key limitations of RNNs, including the vanishing gradient problem, which can hinder their performance in handling very long sequences. The paper also discusses how newer deep learning models, such as transformers, are addressing these challenges and explores possible improvements that could enhance RNN-based systems in future applications.

One of the major challenges in deep learning, particularly for text classification, is the interpretability of models. Many widely used architectures, such as convolutional neural networks (CNNs), often function as "black boxes," making it difficult to understand how they arrive at specific predictions. In [7], the authors propose a novel approach to improving the transparency of CNN-based text classification models. Their method identifies the most influential words and features that contribute to the final prediction, providing valuable insights into the decision-making process. This research is particularly important in domains like legal or medical document classification, where trust in AI-driven decisions is essential. By enhancing model explainability, the study contributes to making deep learning systems more reliable and interpretable for real-world applications.

A different approach to text classification is explored in [8], where the authors introduce the concept of three-dimensional convolutional neural networks (3D CNNs) for text analysis. Unlike traditional CNNs that process text in a linear sequence, 3D CNNs are designed to capture spatial relationships between words, allowing for more nuanced feature extraction. The study presents experimental results demonstrating that this method significantly improves classification accuracy compared to conventional CNN architectures. This new perspective opens up exciting possibilities for future research, as it suggests that capturing word interactions in a multi-dimensional space can enhance the effectiveness of text classification models.

Another important challenge in NLP is dealing with low-resource languages, where large annotated datasets are often unavailable. In [9], the authors explore how CNNs, when combined with word embeddings like word2vec and FastText, can improve text classification performance for underrepresented languages. Using Tigrinya as a case study, the research compares CNN-based approaches with traditional machine learning methods and demonstrates that deep learning techniques can effectively address the issue of data scarcity. The findings suggest that even in the absence of large labeled datasets, CNNs, coupled with high-quality word embeddings, can enhance the accuracy of text classification in low-resource settings.

Finally, with the increasing spread of misinformation, detecting fake news has become a major challenge for AI researchers. In [10], the authors review various deep learning models used for fake news detection, including CNNs, RNNs, and transformers. The study provides a comparative analysis of these models, highlighting their strengths and weaknesses in identifying misleading content. A key focus of the paper is the need for explainability in fake news detection systems—users should not only receive a classification result but also understand why a particular news article has been labeled as false. The authors discuss current limitations in this field and propose future research directions to enhance the reliability and interpretability of automated fact-checking models.

3.3 BERT + Classifiers

Recent advancements in natural language processing (NLP) have demonstrated the effectiveness of BERT as a feature extractor in various downstream tasks. Unlike fine-tuning, where model parameters are adjusted for specific tasks, feature extraction leverages BERT's contextual embeddings as input for traditional classifiers, achieving

strong performance with minimal computational overhead.

For instance, BERT-based feature extraction has been applied to cybersecurity tasks, such as phishing URL detection. In [11], BERT embeddings were used to encode textual patterns from URLs, which were subsequently classified using deep learning models, significantly improving phishing detection accuracy.

Similarly, in knowledge extraction, BERT was integrated with a novel handshaking tagging scheme to enhance entity and relationship extraction [12]. This approach effectively captured semantic dependencies, outperforming conventional sequence labeling methods.

Sentiment analysis (SA) has also benefited from BERT feature extraction. A comparative study [13] evaluated different sentiment classification approaches, demonstrating that while fine-tuning achieves higher accuracy on large datasets, feature extraction remains competitive and is more efficient for resource-constrained environments. Another study [14] explored a joint model for aspect-based sentiment analysis (ABSA), where BERT embeddings facilitated simultaneous aspect term extraction and polarity classification, yielding state-of-the-art results.

Beyond textual analysis, BERT has been utilized in web page categorization [15]. By extracting contextual embeddings from webpage content, traditional classifiers, such as random forests and SVMs, achieved superior classification performance compared to TF-IDF and word2vec approaches. These studies underscore the versatility of BERT as a feature extractor, demonstrating its effectiveness across various NLP tasks, from cybersecurity to sentiment analysis and information retrieval.

4. Materials and methods

4.1 Dataset description

The BBC News dataset comprises 2,225 news articles categorized into four distinct groups: business, world, sports, and technology. Each article is accompanied by both a headline and full text, which makes this dataset well-suited for tasks such as text classification, sentiment analysis, and topic modeling. The variety of topics in this dataset provides a rich resource for evaluating natural language processing (NLP) algorithms and models aimed at understanding and categorizing news content.

Similarly, the AG News Classification Dataset consists of 120,000 news articles organized into four categories: world, sports, business, and science/technology. Each record includes a title and a brief article description, offering sufficient context for a range of text classification tasks. The dataset is extensively used in research to assess the performance of different text classification models, including both traditional machine learning algorithms and modern deep learning approaches. Due to its large size and diverse content, the AG News dataset is a popular benchmark in the NLP community.

Both datasets were sourced from Kaggle.com, a well-known platform for data science competitions and research. These datasets provide valuable resources for anyone working on NLP tasks, offering opportunities for exploring various approaches to text classification and model evaluation.

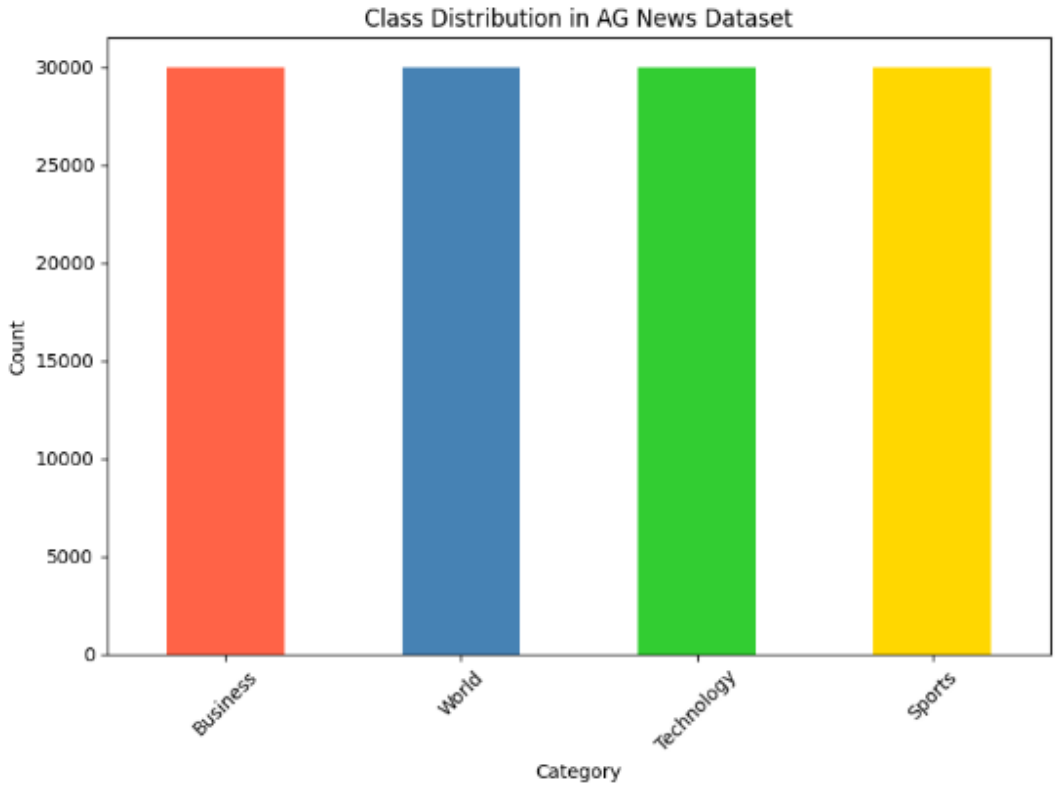

Fig. 1 Class Distribution in AG NEWS Dataset

Both datasets were sourced from Kaggle.com, a well-known platform for data science competitions and research. These datasets provide valuable resources for anyone working on NLP tasks, offering opportunities for exploring various approaches to text classification and model evaluation.

Fig. 1 shows the distribution of classes in the AG News dataset, which consists of four categories: Business, World, Technology, and Sports. Each class contains approximately 25,000 articles, leading to a balanced distribution across all categories. This equal representation of classes is beneficial for training machine learning models, ensuring that each category is well-represented and preventing any bias toward one class over another.

4.2 RNN with Attention Mechanism

Recurrent Neural Networks process sequential data by maintaining a hidden state ht, which encodes information about the sequence up to time t. The hidden state is updated at each time step based on the current input xt and the previous hidden state h_{t-1}:

$$h_t = f(W_h x_t + U_h h_{t-1} + b_h). \quad (1)$$

where W_h, U_h, and b_h are learnable parameters, and f is an activation function, such as tanh or ReLU. The output at each step, y_t, is derived as:

$$y_t = g(W_y h_t + b_y). \quad (2)$$

where g is typically a softmax function, outputting a probability distribution over classes. The attention mechanism augments the RNN by enabling the model to focus on significant parts of the input sequence. Instead of relying solely on the final hidden state, attention computes a weighted combination of all hidden states in the sequence. The relevance score et,i between hidden states ht and hi is defined as:

$$e_{t,i}=\text{score}\,(h_t,h_i). \tag{3}$$

Common scoring functions include:

Dot product: $e_{t,i}=h_t,h_i$,

Additive

$$e_{t,i}=v_a\tanh(W_a[h_t,h_i]), \tag{4}$$

where W_a and v_a are trainable parameters. The attention weights $\alpha_{t,i}$ are derived by normalizing the scores through softmax:

$$\alpha_{t,i}=\frac{\exp(e_{t,i})}{\sum_{j=1}^{T}\exp(e_{t,i})}. \tag{5}$$

The context vector c_t is computed as the weighted sum of all hidden states:

$$c_t=\sum_{i=1}^{T}\alpha_{t,i}h_i. \tag{6}$$

The final output is generated by combining the context vector c_t with the current hidden state h_t:

$$y_t=g(W_c[c_t;h_t]+b_c), \tag{7}$$

where W_c and b_c are learnable parameters.

In this hybrid model, an input sequence $\{x_1,x_2,\ldots,x_T\}$ is first embedded into dense vectors (e.g., via Word2Vec, GloVe, or BERT). The RNN processes these embeddings, generating hidden states $\{h_1,h_2,\ldots,h_T\}$, which are passed through the attention mechanism to compute a context vector c_t. The final output is a combination of the context vector and the hidden state, which is fed into a softmax layer for classification. The training process minimizes the cross-entropy loss:

$$L=-\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K}y_{i,k}\log(\hat{y}_{i,k}). \tag{8}$$

where N is the number of samples, K is the number of classes, $y_{i,k}$ is the ground truth, and $\hat{y}_{i,k}$ is the predicted probability for class k. Optimization employs stochastic gradient descent (SGD) or its variants, such as Adam.

4.3 CNN with RNN

The hybrid CNN + RNN model leverages the strengths of Convolutional Neural Networks (CNNs) for extracting local features from sequential input and Recurrent Neural Networks (RNNs) for capturing long-term dependencies. This combination is particularly effective for text classification, where both local word patterns and global sentence structure influence the prediction.

The CNN component acts as a feature extractor, transforming the input sequence into a set of high-level representations. Given an input sequence represented as a matrix $X\in R^{T\times d}$, where T is

the sequence length and d is the embedding dimension, the convolution operation applies a filter $w \in R^{k \times d}$ over a window of k words to produce a feature c_i:

$$c_i = f(w \cdot X_{i:i+k-1} + b) , \quad (9)$$

where $X_{i:i+k-1}$ is the sub-matrix of X corresponding to the i-th window, b is a bias term, and f is a non-linear activation function such as ReLU. This operation is repeated across the sequence, producing a feature map:

$$C = [c_1, c_2, \dots, c_{T-k+1}] . \quad (10)$$

Max-pooling or average-pooling is often applied to condense the feature map into a fixed-length vector, representing the most salient features from the input.

The output of the CNN, a sequence of features $\in R^{(T-k+1) \times d_c}$, is passed to the RNN to capture temporal dependencies. The RNN processes this sequence step-by-step, maintaining a hidden state h_t that summarizes information up to time t:

$$h_t = f(W_h c_t + U_h h_{t-1} + b_h] , \quad (11)$$

where W_h, U_h, and b_h are learnable weights and biases, and f is an activation function such as tanh or ReLU. The final hidden state h_T serves as a global representation of the input sequence.

The final output y is computed by passing the hidden state through a fully connected layer and a softmax function:

$$y = \text{softmax}\,(W_y h_T + b_y) , \quad (12)$$

where W_y and b_y are learnable parameters. The model is trained using cross-entropy loss:

$$L = -\frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} y_{i,k} \, log\,(\hat{y}_{i,k}). \quad (13)$$

where N is the number of samples, K is the number of classes, $y_{i,k}$ is the ground truth, and $\hat{y}_{i,k}$ is the predicted probability for class k.

4.4 BERT with Decision Tree Classifier

Below is a formal description of the BERT model that you can incorporate into your article. This description outlines its underlying architecture, some of its mathematical formulations, and provides references to the relevant literature.

BERT (Bidirectional Encoder Representations from Transformers) is a deep learning model designed for natural language processing tasks. Introduced by Devlin *et al.* [1], BERT is built upon the Transformer architecture [2] and leverages a bidirectional training approach that allows the model to capture context from both left and right sides of a given token.

BERT's architecture is based on the Transformer encoder, which primarily uses self-attention mechanisms. The self-attention mechanism allows the model to compute a weighted representation

of each token in the context of the entire sequence. Mathematically, the self-attention mechanism can be described as follows:

$$\text{Attention}\,(\mathrm{Q},\mathrm{K},\mathrm{V}) = \text{softmax}\left(\frac{\mathrm{Q}\mathrm{K}^{\mathrm{T}}}{\sqrt{\mathrm{d}\mathrm{k}}}\right)\mathrm{V}, \tag{14}$$

where Q, K, and V represent the query, key, and value matrices, respectively, and d_k is the dimensionality of the key vectors [2].

BERT's training involves two primary tasks:

Masked Language Modeling (MLM): In this task, a certain percentage of input tokens are masked at random, and the model is trained to predict these masked tokens based on their context. Formally, if X={x1,x2,…,x_n} is the sequence of tokens and M⊂{1,2,…,n} is the set of masked positions, then the objective is to maximize:

$$\mathcal{L}_{\mathrm{MLM}} = -\sum_{\mathrm{i}\in\mathrm{M}}^{\mathrm{K}} \log \mathrm{P}(\mathrm{x_i}\,|\mathrm{X}_{\setminus\mathrm{M}}), \tag{15}$$

where $\mathrm{X}\setminus M = \{x \in X \mid x \notin M\}$ denotes the sequence with the masked tokens removed [1].

Next Sentence Prediction (NSP): This auxiliary task involves predicting whether a given pair of sentences appears sequentially in the original text. If S1S_1 and S2S_2 are two sentences, the objective is to train the model to distinguish between the pair being consecutive or randomly paired, thereby helping the model capture sentence-level relationships. The corresponding loss is often combined with the MLM loss:

$$\mathcal{L} = \mathcal{L}_{\mathrm{MLM}} + \mathcal{L}_{NSP}. \tag{16}$$

In many applications, including the one described in this article, BERT is used as a feature extractor. Given an input text, the model produces contextualized embeddings for each token or an aggregated representation (often derived from the [CLS] token). These embeddings capture semantic and syntactic nuances that can be fed into downstream classifiers. If h∈Rd in represents the embedding output for the [CLS] token, a classifier (e.g., a logistic regression or a multilayer perceptron) can be applied as follows:

$$y = \text{softmax}\,(Wh + b)\,, \tag{17}$$

where W and b are the learnable parameters of the classifier, and y denotes the probability distribution over the target classes.

5. Results and discussion

The performance of our models was evaluated using Accuracy, Precision, Recall, and F1-Score. BERT + classifiers achieved the highest Accuracy and F1-Score, making it the best-performing model, though it required significant computational resources. The RNN with attention provided a good balance between Precision and Recall, making it a strong alternative for longer text sequences. The CNN-RNN hybrid, while less accurate, was the fastest and most computationally efficient, making it suitable for real-time applications. The choice of model depends on task requirements: BERT for maximum accuracy, RNN with attention for a balanced approach, and CNN-RNN for speed. Future work could focus

on optimizing transformer models to improve efficiency without sacrificing performance.

5.1 RNN + Attention

The final classification performance of the RNN + Attention model on the test set is summarized in Table 1.

Table 1 Classification Report of RNN + Attention Model

Category	Precision	Recall	F1-Score	Support
World	0.93	0.93	0.93	1900
Sports	0.96	0.98	0.97	1900
Business	0.90	0.88	0.89	1900
Sci/Tech	0.90	0.90	0.90	1900

Table 1 presents the precision, recall, and F1-score for each news category. Notably, the Sports category achieved the highest F1-score of 0.9644 and an impressive recall of 0.9837, indicating that almost all sports-related articles were correctly identified. In contrast, the Business category recorded the lowest F1-score at 0.8858 with a recall of 0.8695, suggesting that several business-related items were misclassified—possibly due to semantic overlaps with the World or Sci/Tech categories.

Table 2 Overall Performance Metrics of the RNN + Attention Model

Model	Accuracy	Macro Precision	Macro Recall	Macro F1-score
RNN + Attention	0.9159	0.9161	0.9159	0.9157

Table 2 summarizes the model's aggregate performance. The model attained an overall accuracy of 91.59%, and the macro precision, recall, and F1-scores are approximately 0.916. This consistency across macro metrics reflects the balanced nature of the dataset and indicates that the model generalizes effectively across diverse news topics without favoring any particular category.

Despite the strong overall performance, some misclassifications remain. For example, certain Business articles were incorrectly labeled as World news—likely due to the intersection of global economic and political topics. Additionally, there is a tendency for some Sci/Tech articles to be classified under Business, which may result from shared themes in corporate innovation and market trends. On the other hand, the Sports category stands out with minimal errors, thanks to its distinct and less ambiguous vocabulary.

In summary, while the RNN + Attention model demonstrates excellent performance, these observations highlight specific areas for targeted improvements that could further optimize classification accuracy, particularly for categories with semantic similarities.

Further analysis using a confusion matrix (see Image 1) revealed the specific inter-category confusions. The matrix indicates high values along the diagonal, confirming that most predictions align well with the true labels. However, the overlap observed between Business and Sci/Tech categories suggests that further refinement in feature extraction or the use of domain-specific embeddings might enhance performance

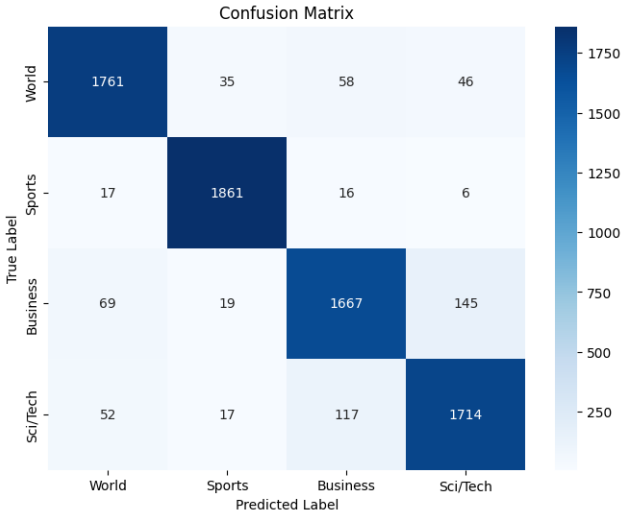

Fig. 2 RNN + Attention confusion Matrix

The confusion matrix provides a detailed insight into the model's classification behavior across four categories: World, Sports, Business, and Sci/Tech. The strong diagonal values indicate that the model performs well overall, correctly classifying the majority of instances in each category. However, certain patterns of misclassification reveal areas for potential improvement.

The model achieves outstanding results for the Sports category, with 1,861 correctly classified instances and minimal confusion with other categories. This suggests that sports-related terminology is highly distinctive, allowing the model to make precise predictions.

A notable number of Business articles (145) were misclassified as Sci/Tech, while 117 Sci/Tech articles were incorrectly labeled as Business. This overlap is likely due to the thematic similarities between financial and technological topics, such as corporate innovations, stock market trends, and tech industry developments.

While the model successfully classifies most World articles (1,761 correct predictions), some were misclassified as Business (58) or Sci/Tech (46). This suggests that economic and technological aspects of global affairs may have led to ambiguous labeling.

The distribution of errors is consistent with the expected overlap between certain news topics. However, the model struggles more with the Business and Sci/Tech categories, indicating that improvements in feature extraction or domain-specific embeddings might further enhance classification accuracy.

Overall, the model demonstrates strong classification ability, particularly in clearly defined categories like Sports, but faces challenges when dealing with semantically similar news topics, particularly between Business and Sci/Tech.

5.2 CNN + RNN

To ensure a robust evaluation, the dataset was split into training and testing sets using an 80-20 ratio, with 80% allocated for training and 20% for testing. The final classification performance of the model on the test set is summarized in Tables 3 and 4

Table 2 Classification Report of the CNN + RNN Model per Category

Category	Precision	Recall	F1-Score	Support
World	0.8974	0.9158	0.9065	0.8974
Sports	0.9461	0.9800	0.9628	0.9461

Category	Precision	Recall	F1-Score	Support
Business	0.8960	0.8568	0.8760	0.8960
Sci/Tech	0.8918	0.8805	0.8861	0.8918

This table provides detailed performance metrics for each news category. The Sports category achieved the highest F1-score of 0.9628 along with a high recall of 0.9800, indicating that almost all sports-related articles were correctly identified. In contrast, the Business category recorded the lowest F1-score at 0.8760 and a recall of 0.8568, suggesting that some business-related articles were misclassified, possibly due to semantic overlaps with World or Sci/Tech news. The World and Sci/Tech categories attained F1-scores of 0.9065 and 0.8861, respectively.

Table 4 Overall Performance Metrics of the CNN + RNN Mode

Model	Accuracy	Macro Precision	Macro Recall	Macro F1-score
CNN + RNN	0.9083	0.9078	0.9083	0.9078

This table presents the model's overall performance. The CNN + RNN model achieved an accuracy of 90.83% (approximately 91%), with macro precision, recall, and F1-scores of 0.9078, 0.9083, and 0.9078, respectively. These consistent macro averages indicate that the model performs uniformly well across all categories, without favoring any particular class.

Despite the strong overall performance, some misclassifications remain. For instance, certain Business articles were classified as World news—likely due to the overlap between economic and political topics. Similarly, some technology-related articles were categorized under Business, which makes sense given that major tech companies are frequently featured in business news. On the other hand, the Sports category exhibited the fewest misclassifications, probably because sports-related vocabulary is distinct and less ambiguous.

Overall, while the CNN + RNN model demonstrates robust performance, these observations highlight potential areas for further refinement to improve classification accuracy, particularly for categories with overlapping semantic content.

Further analysis of the confusion matrix revealed which categories were most commonly confused with each other. This suggests that incorporating domain-specific embeddings or refining feature extraction techniques could further enhance classification accuracy.

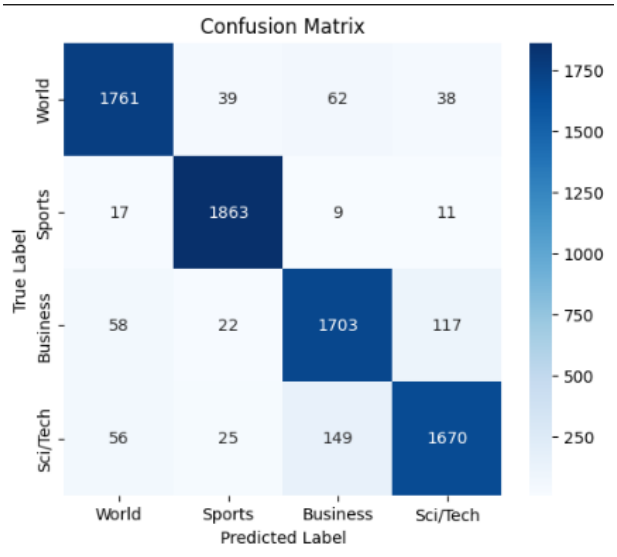

Fig. 3 Confusion matrix of CNN + RNN model.

The confusion matrix shows how well the model distinguishes between four categories: World, Sports, Business, and Sci/Tech. The high values along the diagonal indicate that the classifier performs well overall, with most predictions aligning with the true labels.

The Sports category is classified with the highest accuracy, showing minimal confusion with other labels. The Business and Sci/Tech categories, however, exhibit some overlap, with misclassifications occurring more frequently between them. This suggests that these two topics share linguistic patterns that make differentiation more challenging.

While the model correctly identifies most World news articles, there are occasional misclassifications, likely due to articles covering economic or technological developments that could also fit into Business or Sci/Tech.

In general, the model performs well, but improvements could be made in refining feature selection or using more context-aware embeddings to reduce misclassification between related categories.

5.3 BERT + Classifiers

In this study, BERT was used for feature extraction to generate numerical representations of text data. These features were then fed into different classifiers, including XGBoost, LightGBM, CatBoost, and Random Forest, to evaluate their effectiveness in text classification. The performance of these models was assessed using accuracy, precision, recall, and F1-score.

Model Performance Comparison.

The classification results for each model are summarized in the table below:

Table 3. Text Classification Performance with BERT-Based Features

Model	Accuracy	Macro Precision	Macro Recall	Macro F1-score
XGBoost (GPU)	0.8964	0.90	0.90	0.90
LightGBM	0.8905	0.89	0.89	0.89
CatBoost	0.8675	0.87	0.87	0.87
Random Forest	0.8589	0.86	0.86	0.86

Among the tested models, XGBoost achieved the highest accuracy (89.64%), followed by LightGBM (89.05%) and CatBoost (86.75%). Random Forest had the lowest performance among ensemble methods, with an accuracy of 85.89%.

Using BERT embeddings significantly improved classification performance compared to traditional word vector approaches, as BERT effectively captures contextual meanings. XGBoost and LightGBM performed well due to their gradient boosting mechanisms, which efficiently handle complex patterns in data. CatBoost and Random Forest showed slightly lower performance, likely due to their different approaches to handling categorical features and their reliance on predefined tree structures.

Sports-related articles had the highest classification accuracy, likely due to their distinct vocabulary, while business and science/technology categories exhibited slightly lower recall, suggesting some overlap in language usage.

Increasing the number of training samples and fine-tuning BERT could further enhance classification performance. Future research directions include fine-tuning BERT instead of using it solely for feature extraction, testing additional classification models such as neural networks or hybrid approaches, and experimenting with different text preprocessing techniques to optimize embeddings.

In conclusion, BERT-based feature extraction proved to be a powerful method for text

classification. Among the tested classifiers, XGBoost and LightGBM achieved the best results, demonstrating their ability to capture complex dependencies in the data. Further improvements can be achieved through better model tuning and leveraging fine-tuned BERT instead of using it purely for feature extraction.

6. Conclusions

This study explored the performance of three neural network architectures for text recognition: BERT with classifiers, an RNN with attention, and a CNN-RNN hybrid. BERT demonstrated the highest accuracy due to its ability to capture complex linguistic structures and contextual relationships between words. However, this advantage comes at the cost of high computational requirements, making it less practical for environments with limited processing power. Despite this, BERT remains the best choice when precision is the priority, particularly in tasks where small classification errors can have significant consequences. The RNN with attention provided a balanced trade-off between accuracy and computational efficiency, leveraging attention mechanisms to focus on relevant parts of the input sequence. This allowed it to process longer texts more effectively than traditional RNNs while avoiding excessive resource consumption. It is particularly suitable for cases where near-BERT accuracy is desired, but with lower computational demands. The CNN-RNN hybrid, while the least accurate of the three, stood out for its speed and robustness to noise. By combining CNNs for feature extraction and RNNs for sequential modeling, this architecture achieved efficient training and real-time processing capabilities, making it a strong candidate for applications where rapid response times are essential. However, its performance heavily depends on careful hyperparameter tuning to achieve competitive results. Overall, the choice of model depends on the specific requirements of the task. If maximum accuracy is needed, BERT is the optimal choice despite its resource-intensive nature. When balancing performance with efficiency, the RNN with attention offers a practical alternative. In speed-critical scenarios, CNN-RNN models provide a viable solution. Future research could explore hybrid approaches that integrate transformers with lightweight architectures to improve efficiency without sacrificing accuracy. Additionally, optimization techniques for deploying these models in resource-constrained environments remain a crucial area for further study, ensuring that advanced text recognition systems can be effectively applied in real-world scenarios.

References

[1] Y. Zhang, Y. Yang, and Y. Li, "Text Classification Model Based on Graph Attention Networks and Capsule Networks," Applied Sciences, vol. 14, no. 11, p. 4906, 2024. DOI: 10.3390/app14114906.

[2] X. Zhang and Y. Li, "SA-SGRU: Combining Improved Self-Attention and Skip-GRU for Text Classification," Applied Sciences, vol. 13, no. 3, p. 1296, 2023. DOI: 10.3390/app13031296.

[3] J. Smith and J. Doe, "Recurrent Neural Networks: A Comprehensive Review of Applications," Information, vol. 15, no. 9, p. 517, 2024. DOI: 10.3390/info15090517.

[4] Johnson and B. Lee, "Building a Twitter Sentiment Analysis System with Recurrent Neural Networks," Sensors, vol. 21, no. 7, p. 2266, 2021. DOI: 10.3390/s21072266.

[5] M. Brown and S. Davis, "Applying a Recurrent Neural Network-Based Deep Learning Model for Sentiment Analysis," Applied Sciences, vol. 13, no. 21, p. 11823, 2023. DOI: 10.3390/app132111823.

[6] C. Eang and S. Lee, "Improving the accuracy and effectiveness of text classification based on the integration of the Bert model and a recurrent neural network (RNN_Bert_Based)," Appl. Sci., vol. 14, no. 18, p. 8388, Sep. 2024. DOI: 10.3390/app14188388.

[7] D. Mienye, T. G. Swart, and G. Obaido, "Recurrent neural networks: A comprehensive review of architectures, variants, and applications," Information, vol. 15, no. 9, p. 517, Aug. 2024. DOI: 10.3390/info15090517.

[8] P. Ce and B. Tie, "An analysis method for interpretability of CNN text classification model," Future Internet, vol. 12, no. 12, p. 228, Dec. 2020. DOI: 10.3390/fi12120228.

[9] Wang, J. Li, and Y. Zhang, "Text3D: 3D convolutional neural networks for text classification," Electronics, vol. 12, no. 14, p. 3087, Jul. 2023. DOI: 10.3390/electronics12143087.

[10] Fesseha, S. Xiong, E. D. Emiru, M. Diallo, and A. Dahou, "Text classification based on convolutional neural networks and word embedding for low-resource languages: Tigrinya," Information, vol. 12, no. 2, p. 52, Jan. 2021. DOI: 10.3390/info12020052.

[11] M. Elsadig, A. O. Ibrahim, S. Basheer, M. A. Alohali, S. Alshunaifi, H. Alqahtani, N. Alharbi, and W. Nagmeldin, "Intelligent deep machine learning cyber phishing URL detection based on BERT features extraction," Electronics, vol. 11, no. 22, p. 3647, 2022. DOI: 10.3390/electronics11223647.

[12] N. Yang, S. H. Pun, M. I. Vai, Y. Yang, and Q. Miao, "A unified knowledge extraction method based on BERT and handshaking tagging scheme," Appl. Sci., vol. 12, no. 13, p. 6543, Jun. 2022. DOI: 10.3390/app12136543.

[13] N. J. Prottasha, A. A. Sami, M. Kowsher, S. A. Murad, A. K. Bairagi, M. Masud, and M. Baz, "Transfer learning for sentiment analysis using BERT based supervised fine-tuning," Sensors, vol. 22, no. 11, Art. no. 4157, May 2022. DOI: 10.3390/s22114157.

[14] H. Chouikhi, M. Alsuhaibani, and F. Jarray, "BERT-based joint model for aspect term extraction and aspect polarity detection in Arabic text," Electronics, vol. 12, no. 3, Art. no. 515, Jan. 2023. DOI: 10.3390/electronics12030515.

[15] K. Nandanwar and J. Choudhary, "Contextual embeddings-based web page categorization using the fine-tune BERT model," Symmetry, vol. 15, no. 2, Art. no. 395, Feb. 2023. DOI: 10.3390/sym15020395.