MSc Artificial Intelligence and Data Science

Module 771767- Applied Artificial Intelligence

Natural Language Processing Project Report

By

Student ID – 202403820 | Samuel Datubo Jaja

1.Definition

Text/topic classification is a core problem in the domain of Natural Language Processing (NLP) where textual data is categorized into different categories. In this project, we classify news articles from into different categories: World, Sports, Business, and Sci/Tech. This task is critical in automating content organization, summarization, and retrieval, which are essential for efficient information access.

2. Scope

This project represents a meaningful contribution to NLP by addressing the need for automated systems capable of accurately categorizing news articles. The scope includes implementing traditional machine learning methods and deep learning approaches, comparing their performance, and refining them to achieve high classification accuracy. The insights from this study can be extended to other text-based domains, such as sentiment analysis or topic modeling.

3. Importance

The significance of this task lies in its real-world applicability in the sense that news platforms generate vast volumes of articles daily. Manual categorization is impractical, necessitating automated solutions. Accurate classification improves user experience by personalizing news feeds and enhancing search relevance. Despite advancements in NLP, traditional and deep learning models often face trade-offs in speed, accuracy, and computational resources, making this an area worthy of exploration.

4. Background Review

This section reviews five studies discussing their techniques, pros and cons, and the results achieved.

Recent advances in text classification have shown diverse approaches to handling unstructured data. In "**Application Research of Text Classification Based on Random Forest Algorithm**," Sun et al. (2020) proposed a Random Forest-based approach integrating BERT embeddings with a novel tr-k method. While achieving enhanced classification accuracy and demonstrating robustness to overfitting through improved feature representation, their method faced challenges with computational costs and preprocessing complexity.

Addressing document clustering, Kumbhar et al. (2020) in "**Text Document Clustering Using K-means Algorithm with Dimension Reduction Techniques**" combined K-means clustering with dimensionality reduction techniques (SVD/NMF) and TF-IDF vectorization. Their approach proved computationally efficient with reduced dimensionality, though

showing sensitivity to initial centroid selection. Their results demonstrated NMF with K-means outperforming traditional approaches in clustering accuracy.

In "**A Survey of Text Classification with Transformers**," Fields, Chovanec and Madiraju (2024) evaluated transformer-based classification using models like BERT, GPT, and RoBERTa. While excelling at long-term dependencies and enabling transfer learning, these models faced challenges with computational costs and bias. Their analysis across 358 datasets confirmed superior performance in various classification tasks, despite resource intensity.

Sunagar et al. (2024) in "**Hybrid RNN Based Text Classification Model for Unstructured Data"** introduced a hybrid neural architecture combining RNN, BiLSTM, and GRU with GloVe embeddings. Their model balanced bidirectional context capture with computational efficiency, particularly for smaller datasets, though facing challenges with out-of-vocabulary words. The architecture achieved a notable F1-score of 0.7585 on Twitter data, demonstrating effective handling of long-term dependencies.

In "**Recurrent Neural Networks: A Comprehensive Review of Architectures, Variants, and Applications**," Mienye, Swart and Obaido (2024) provided a comprehensive review of RNN variants, focusing on LSTM and GRU implementations with attention mechanisms. Their analysis showed state-of-the-art performance in text tasks using hybrid models, despite being computationally intensive and slower than pure transformer approaches. The study particularly highlighted the effectiveness of attention-enhanced RNNs in text summarization and classification.

5. SMART Objectives

Specific: Compare Naive Bayes, Random Forest, LSTM, and Bi-LSTM models for AG News topic classification.

Measurable: Aim to achieve accuracy of at least 70% for traditional models and 75% for deep learning models, significantly outperforming the 25% baseline, with stretch goals of reaching the 80-85% performance levels reported in current literature.

Achievable: Previous research demonstrates successful application of these models in similar text classification tasks with comparable accuracy.

Relevant: Enhances information retrieval systems through improved news categorization capabilities.

Time-bound: Complete all phases within 12-week trimester timeframe.

6. Dataset

The AG News dataset is publicly available through the Hugging Face platform. It comprises 120,000 designated for training and 7,600 for testing. Having a size 120K and shape (120, 000rows and 2columns). Each sample includes a news article title and description, accompanied by a label indicating one of four categories: World, Sports, Business, or Sci/Tech. This dataset is particularly suitable for text classification problems due to its balanced class distribution; each category contains an equal number (30,000) of samples (25%), which helps prevent model bias toward any single class. Additionally, the dataset's structure comprising textual data facilitates efficient preprocessing and model training. The AG News dataset's balance and clear structure makes it an excellent choice for developing and evaluating NLP models focused on news topic classification.

7. Exploratory Data Analysis

Exploratory Data Analysis (EDA) involved generating a word cloud as seen from figure 1a and 1b to visualize the most frequent terms and verifying class balance across the dataset categories plotting bar charts as seen in figure 2 below. Following EDA, a shared preprocessing class was implemented, supporting Traditional ML models specifically. Text cleaning included lowercasing, removing punctuation, and eliminating stop words using Natural Language Tool Kit (NLTK). Lemmatization standardized words to their base forms. For ML models, TF-IDF vectorization transformed text into numerical features, emphasizing term importance. Principal Component Analysis (PCA) was applied specifically for Random Forest to aid dimensionality reduction. For DL models, tokenization converted text into word sequences, followed by padding for uniform input length. All DL models utilized the preprocessed data from a common python class as seen from the Jupyter notebook, these steps were critical in ensuring data readiness and robust model performance.

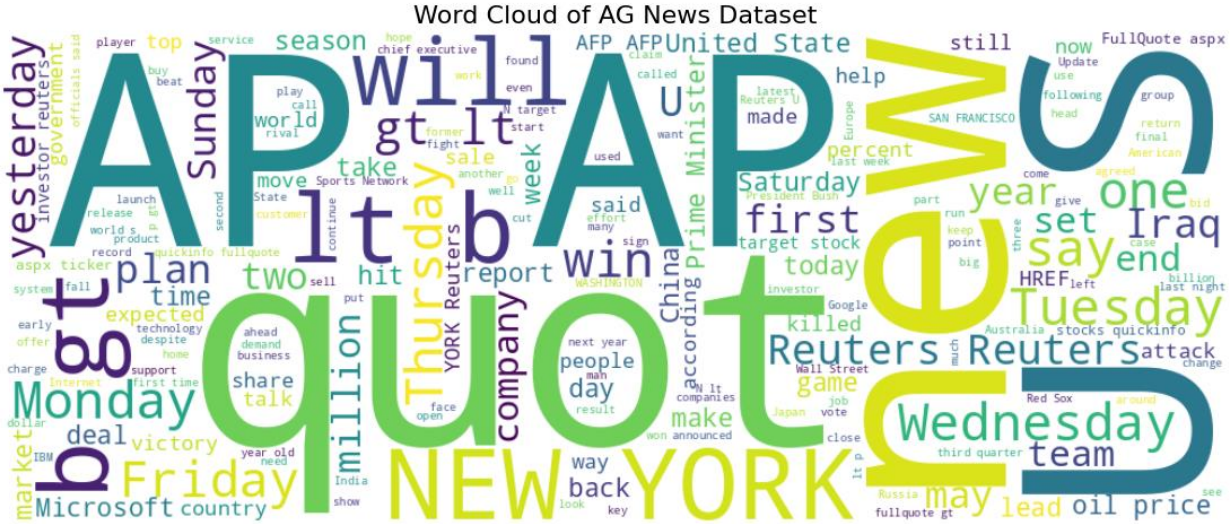

Figure 1a: AG News Word Cloud

Figure 1b: Categorized AG News Word Cloud

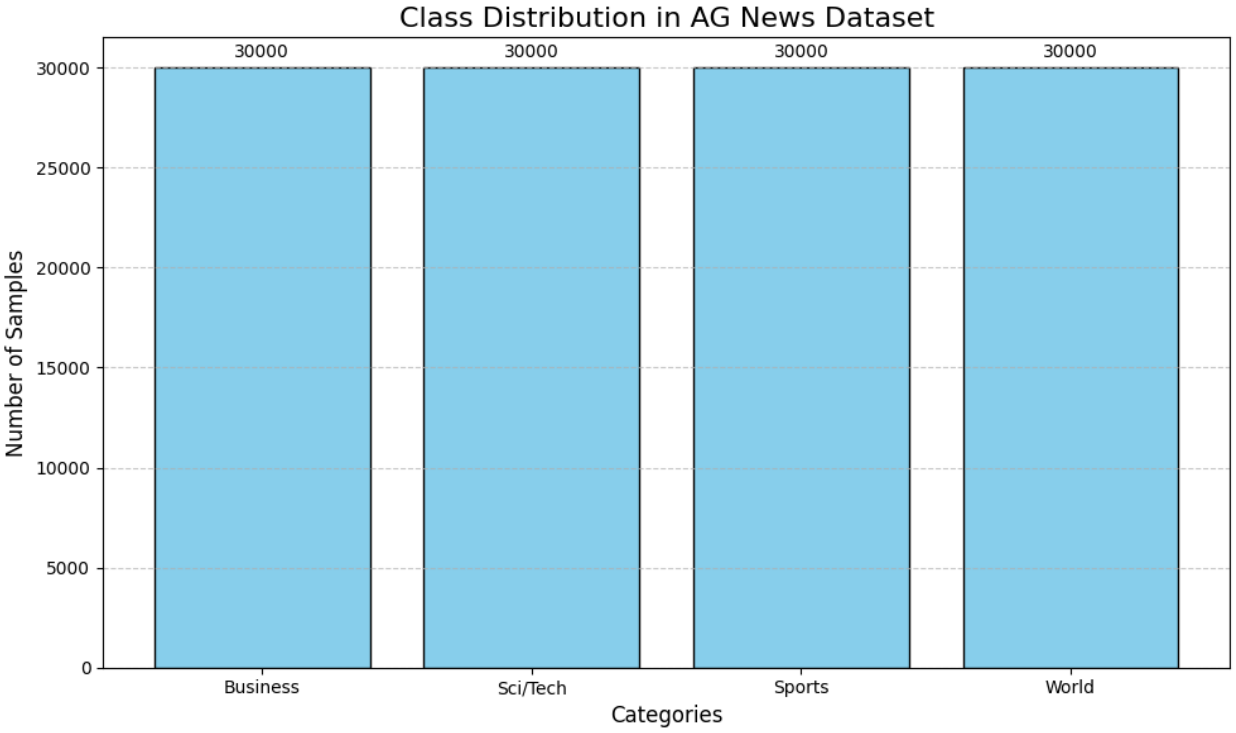

Figure 2: Balanced Class Distribution in AG News Dataset

Baseline Selection: I chose a random chance baseline of 25% for this classification task. This baseline is mathematically derived from the perfectly balanced dataset structure of four categories (Business, Sci/Tech, Sports, World), each containing 30,000 samples. A random guessing model would have a 1/4 probability of correct classification, making 25% a clear, justifiable minimum performance threshold. Any implemented model should demonstrate performance significantly above this baseline to prove its effectiveness in learning meaningful patterns from the text data

8. Traditional Machine Learning Methods

Naive Bayes (Chosen): Computationally efficient classifier ideal for text classification due to its rapid training and prediction capabilities. While effective with high-dimensional data, its feature independence assumption may limit performance with interrelated text features (Mienye, Swart & Obaido, 2024).

K-Means Clustering: Unsupervised algorithm that reveals data patterns through cluster formation. Valuable for topic exploration but lacks direct alignment with predefined categories. Requires expertise for optimal cluster selection (Singh, 2013).

Support Vector Machines (SVM): Effective with high-dimensional, sparse text data through optimal hyperplane separation. Strong resistance to overfitting but computationally intensive for large datasets (Mienye, Swart & Obaido, 2024).

Logistic Regression: Simple, interpretable baseline model. Limited by linear relationship assumptions between features and outcomes, affecting performance in complex text classification tasks (Mienye, Swart & Obaido, 2024).

Random Forest (Chosen): Ensemble method combining decision trees, excelling in handling complex text feature interactions. While computationally demanding, its robust performance and reduced overfitting make it ideal for topic classification (Mienye, Swart & Obaido, 2024).

Justification for Choices:

- Naive Bayes: Selected for efficiency and proven effectiveness in text classification
- Random Forest: Chosen for superior handling of feature interactions and ensemble benefits in topic classification

9. Deep Learning Methods

Standard RNN: Basic sequential model with hidden states for temporal dependencies. Limited by vanishing gradients affecting long-term pattern recognition (Sherstinsky, 2020).

LSTM: Enhanced RNN with memory cells and gates for better long-term dependency capture. Effective for text classification despite higher computational costs (Mienye, Swart & Obaido, 2024; Sherstinsky, 2020).

BiLSTM: Processes sequences bidirectionally for enhanced context understanding. Superior accuracy but more resource-intensive than unidirectional LSTMs (Mienye, Swart & Obaido, 2024; Sherstinsky, 2020).

GRU: Simplified LSTM variant with combined gates for faster training. Trades some flexibility for computational efficiency (Sherstinsky, 2020).

Transformer: Parallel processing architecture using self-attention for superior pattern recognition. State-of-the-art performance but requires substantial computational resources (Vaswani et al., 2017; Devlin et al., 2019).

Justification for Choices:

- LSTM: Selected for robust handling of long-term dependencies in text.
- BiLSTM: Chosen for superior accuracy through bidirectional context processing.

10. Implementation and Refinement

Implementation and Refinement of Naive Bayes

Libraries Used: I utilized MultinomialNB from the scikit-learn library for implementing the Naive Bayes classifier.

Procedures Followed: Preprocessed the text data using a TextPreprocessor class, which included vectorization with parameters such as max features of 1000. Through the initialization of the Naive Bayes custom class with alpha value, the model was trained on the processed training data and evaluated on the test set with different performance metrics like Precision-Recall curve, ROC Curve and test accuracy.

Fine-Tuning Strategies:

Adjusted the smoothing parameter alpha parameter to 0.8 in the MultinomialNB classifier. Smoothing helps handle zero probabilities in categorical data by assigning a small prior probability to unseen words, thereby improving model robustness.

Justification:

- The alpha parameter controls the smoothing applied to word probabilities.
- Tuning alpha helps balance the model's bias-variance trade-off, enhancing its ability to generalize to unseen data.
- Setting alpha=0.8 was determined to provide optimal performance for this specific dataset.

Increased the max_features parameter in the vectorizer to 10,000.

Justification:

- Expanding max_features allows the model to consider a broader vocabulary, capturing more nuances in the text data.
- This enhancement can lead to improved model performance by incorporating a wider range of words, thereby enhancing the model's ability to distinguish between different classes.

Implementation and Refinement of Random Forest

Libraries Used: I utilized the Random Forest Classifier from the scikit-learn library to implement the Random Forest model.

Procedures Followed:

I preprocessed the text data using the TextPreprocessor class, setting max_features to 10,000 for vectorization. To enhance computational efficiency, I applied Principal Component Analysis (PCA) for dimensionality reduction. I configured the Random Forest Classifier with 50 trees (n_estimators=50), a maximum depth of 10 for each tree (max_depth=10), and set random state=42 to ensure reproducibility. After training the model on the processed training data, I evaluated its performance by predicting labels for the test set, calculating the accuracy score, generating a classification report, and visualizing the confusion matrix to assess classification performance across different news categories.

Fine-Tuning Strategies:

Adjusted n_estimators (number of trees) to 1000:

Justification:

- Increasing the number of trees enhances model complexity, allowing the Random Forest to capture more nuances in the data.
- A larger number of trees improves stability and overall accuracy while managing computational efficiency.

Set max_depth to 30:

Justification:

- Increasing tree depth allows the model to better capture complex patterns in the data.
- This configuration prevents underfitting while maintaining generalization to unseen data.

Applied PCA:

Justification:

- Reducing feature dimensionality with PCA mitigates the curse of dimensionality.
- This optimization improves model training speed and performance by focusing on the most relevant features.

Implementation and Refinement of LSTM Model

Libraries Used:

TensorFlow and Keras libraries were utilized to implement the LSTM model, offering flexibility in dynamic layer addition and robust handling of sequential data.

Procedures Followed: I tokenized and padded sequences to a uniform length of 50 using the tokenizer class, with a vocabulary size of 10,000 (max words=10000). Then added embedding layer to convert words into dense vector representations. I added an LSTM layer with 128 units to capture temporal dependencies in text data. Added a dense layer with 64 units, activated by ReLU, for feature extraction. Compiled the model with the Adam optimizer (learning rate=0.001) and sparse categorical cross-entropy loss. Then trained the model for 15 epochs using early stopping to prevent overfitting.

Fine-Tuning Strategies:

Increased Number of Epochs to 40

Justification:

- Allowing more epochs enables the model to learn intricate patterns in the data, especially when using early stopping to halt training if validation performance stagnates.

Decreased Learning rate: Set to 0.0001

Justification:

- A lower learning rate ensures gradual updates to weights, preventing overshooting the optimal minima and refining the model's performance during later epochs.

Dropout and Recurrent Dropout: Applied at 0.5 and 0,4 to prevent overfitting by randomly disabling neurons during training with batch normalization.

Justification:

- Dropout mitigates overfitting by randomly deactivating neurons, forcing the model to rely on more robust patterns. Recurrent dropout specifically targets LSTM layers, enhancing the generalization of temporal dependencies.

Added GloVe Embeddings with trainable set to false (glove.6B.100d.txt).

Justification:

- Pre-trained embeddings provide rich, contextualized word representations, enabling the model to start with meaningful features and reducing training time for better performance on text data.

Implementation and Refinement for Bidirectional-LSTM

Libraries Used:

TensorFlow and Keras for Bi-LSTM implementation and model fine-tuning.

Procedures Followed: A similar process to LSTM was followed with key difference in integrating a BiLSTM layer with 128 units to capture bidirectional dependencies in text sequences. Included a dense layer as well and trained for 15 epochs with early stopping. Evaluated using prediction confidence and misclassification analysis and test accuracy as seen in the evaluation section.

Fine-Tuning Strategies:

Batch size 64 with epoch set to 50
Justification:

- This batch size balances computational efficiency and model stability, while allowing sufficient iterations for convergence.

Bi-LSTM Layer Dropout and Recurrent Dropout: Applied at 0.5 and 0,4 to prevent overfitting by randomly disabling neurons during training.
Justification:

- These dropout rates provide optimal regularization without significantly compromising model performance, as evidenced by the high confidence scores in correct predictions.

Dense Layer: 128Unit and its dropout rate 0.3
Justification:

- 128 units provide sufficient complexity for feature representation while the 0.3 dropout maintains model generalization, as shown by the low misclassification rates between categories.

Added GloVe Embeddings with Trainable set to True (glove.6B.100d.txt).

Justification:

- Trainable pre-trained embeddings allow the model to fine-tune word representations specific to the news classification task, improving the model's understanding of context-specific word meanings.

11. Evaluation

Naive Bayes Evaluation Metrics

- Accuracy score
- Also generated and visualized the ROC Curve
- Precision-Recall curve to assess model performance comprehensively.

Effect of Fine-Tuning: Fine tuning the model improved model accuracy from 85.33% to 89.70% and improved the ROC and Precision-Recall curves to assess model visual performance seen below on figure 3 and figure 4 respectively. The ROC curve shows the balance between true positive rate and false positive rate, while the PR curve demonstrates the relationship between precision and recall. The model's ROC curve approaching the top-left corner and PR curve staying near the top-right corner indicate strong classification performance. These metrics confirm the Naive Bayes model effectively classifies AG News topics while maintaining high precision and recall.

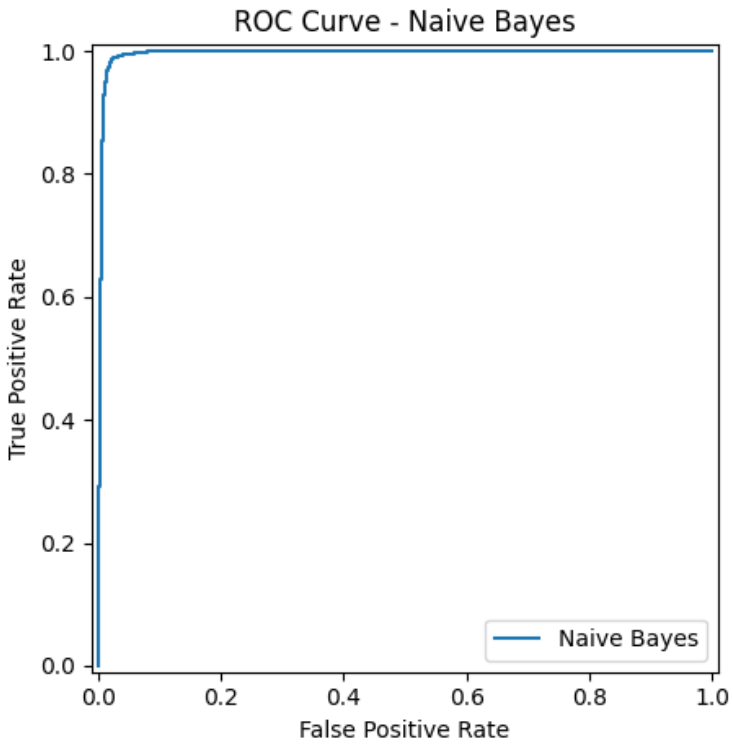

Figure 3: Naïve Bayes ROC Curve

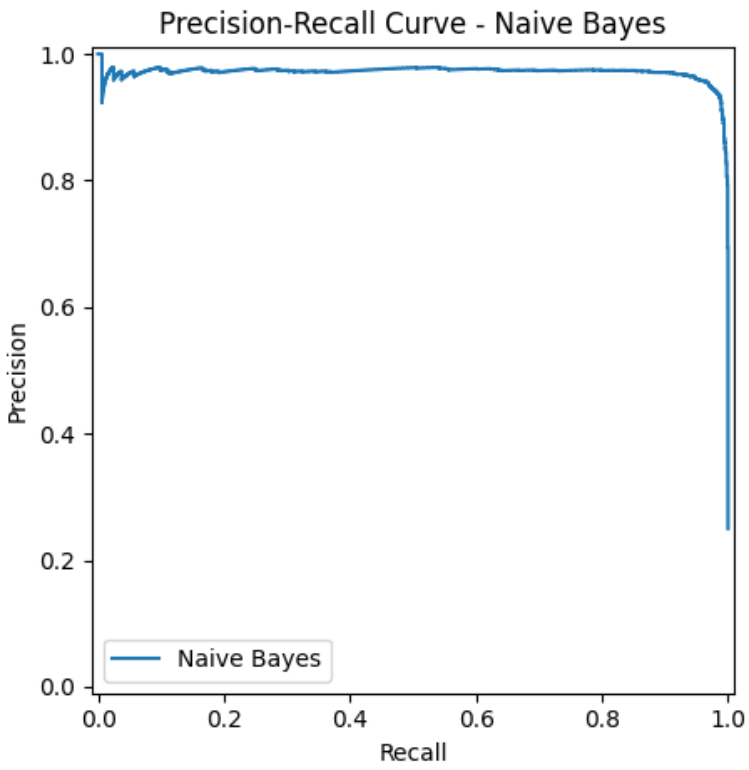

Figure 4: Naïve Bayes Precision-Recall Curve

Random Forest Evaluation Metrics

- Classification Report
- Confusion Matrix.
- Accuracy Score.

Effect of Fine-Tuning: After fine tuning as seen below on table 1 and figure 5 respectively, the performance of the Random Forest model is summarized in its classification report and confusion matrix. The model accuracy improved from 78.17% of 84.14%, with macro and weighted averages for precision, recall, and F1-score at 0.84. The highest precision (0.88) and recall (0.96) were observed in the "World" and "Sports" categories, respectively, highlighting strong classification performance in these domains. However, the "Business" and "Sci/Tech" categories showed lower recall values (0.80 and 0.77), indicating room for improvement in correctly identifying these classes. The confusion matrix illustrates a relatively balanced performance but reveals occasional misclassifications, particularly between "Business" and "Sci/Tech." The model's overall performance reflects its robustness, attributed to fine-tuned hyperparameters, including 1000 trees (n_estimators) and a maximum depth of 30 (max_depth).

Table 1: Random Forest Classification Report

	Precision	Recall	F1-Score	Support
World	0.88	0.83	0.85	1900
Sports	0.84	0.96	0.89	1900
Business	0.84	0.77	0.81	1900
Sci/Tech	0.81	0.80	0.81	1900
Accuracy	-	-	0.84	7600
Macro Avg	0.84	0.84	0.84	7600
Weighted Avg	0.84	0.84	0.84	7600

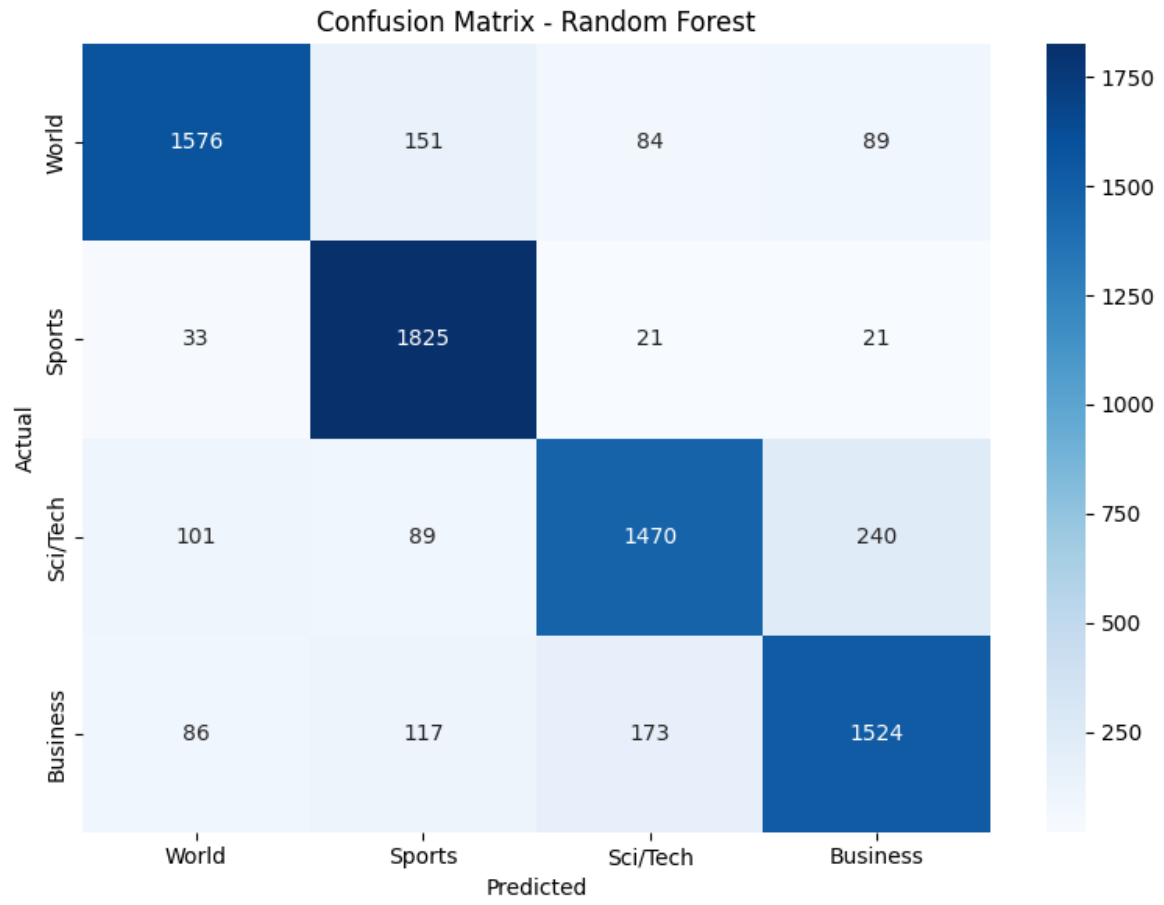

Figure 5: Random Forest Confusion Matrix

Deep Learning Model Evaluation Metrics

LSTM Evaluation Metrics:

- Training and Validation Curves: Highlighted consistent improvement in training and validation accuracy, with minimal overfitting.
- ROC-AUC Curve
- Accuracy Score

Effect of Fine-Tuning:

The fine-tuned LSTM model generalized well, capturing long-term dependencies while avoiding overfitting. This is evident from the high AUC scores and balanced classification performance across the AG News dataset. From figure 6a, the training and validation accuracy curve shows the model effectively learning and generalizing. Training accuracy steadily increases, reaching over 93% by epoch 14. Validation accuracy closely follows, plateauing around epoch 10, indicating optimal generalization. The minimal gap between the curves confirms the model is well-trained with no significant overfitting. While from figure 6b, the training and validation loss curve shows a consistent decrease in training loss, indicating effective learning. Validation loss initially decreases alongside training loss, stabilizing after epoch 4 with minor fluctuations. The close alignment between the curves suggests the model generalizes well with no significant overfitting. Also my test accuracy score increased from 91.61% to 92.75%.

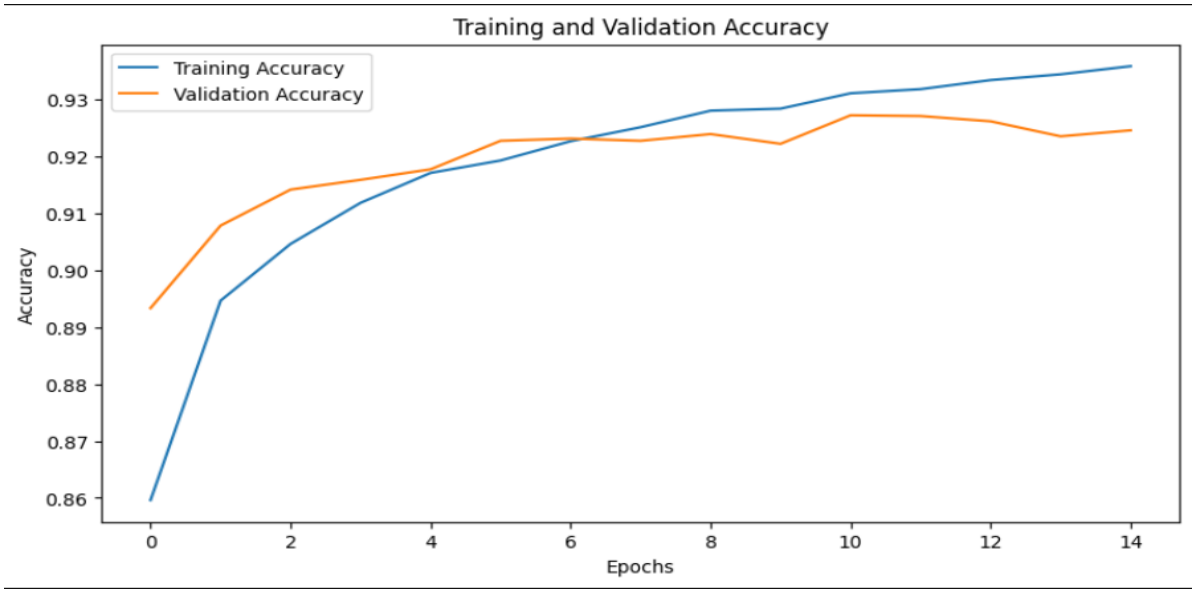

Figure 6a: LSTM Training and Validation Accuracy

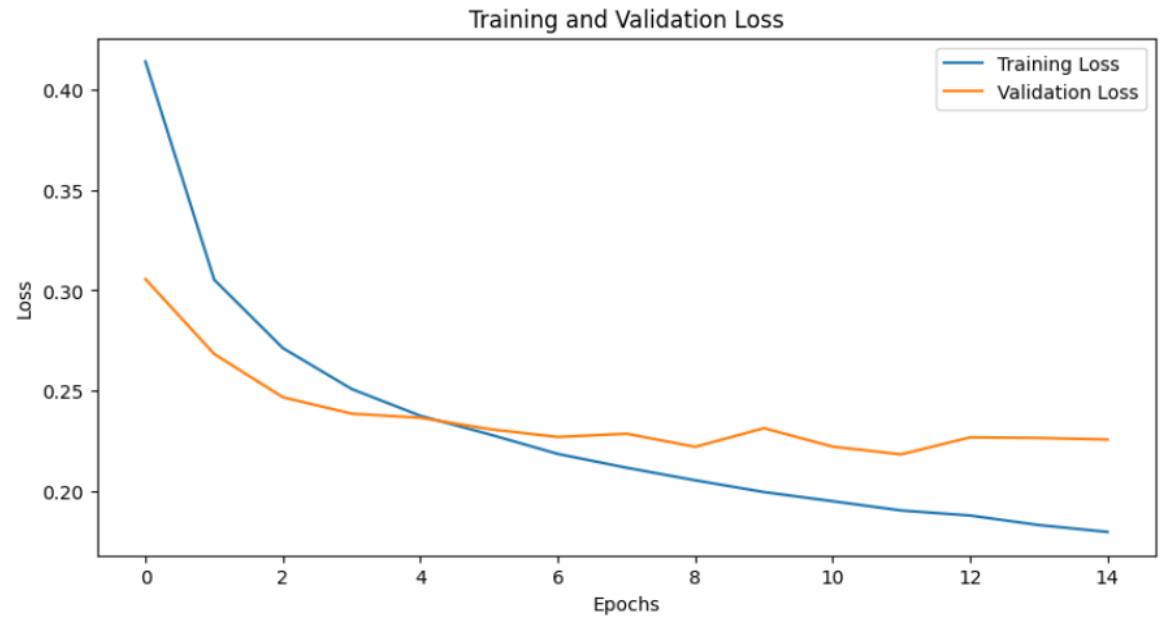

Figure 6b: LSTM Training and Validation Loss

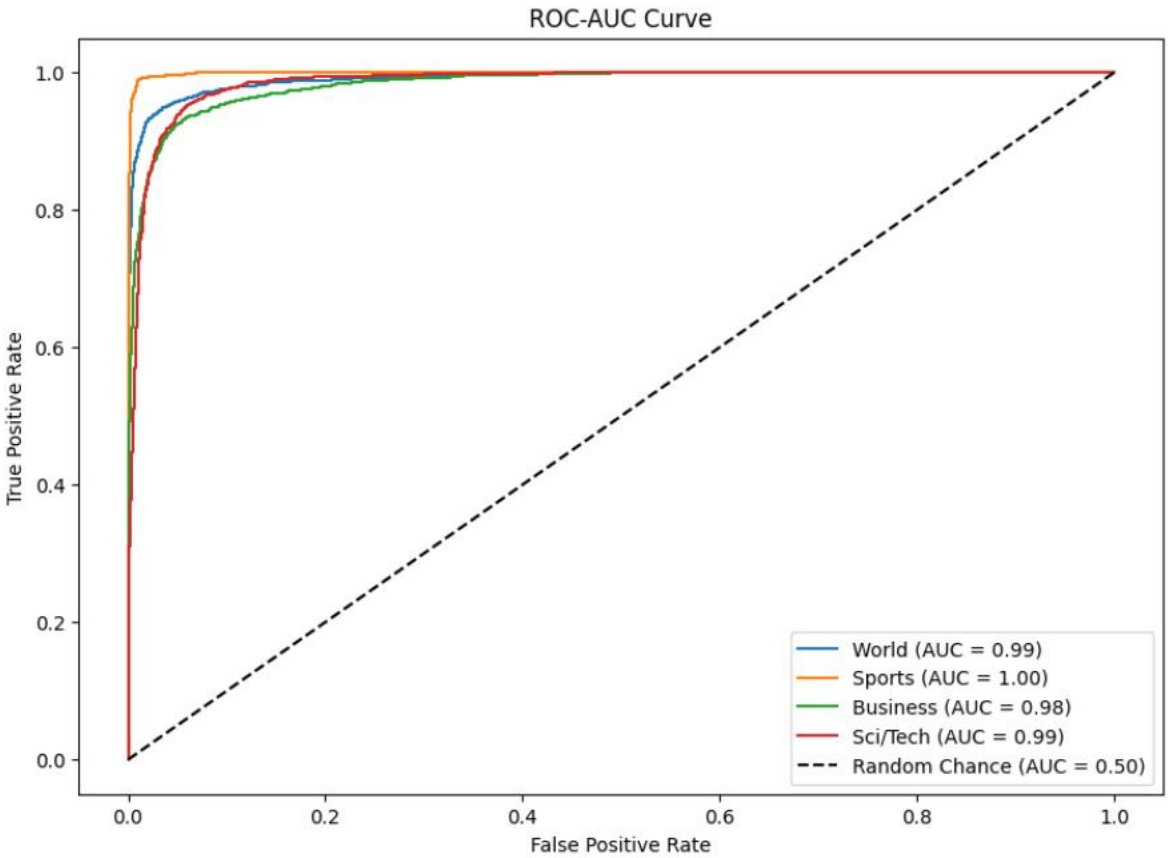

Figure 7: LSTM ROC-AUC Curve

From figure 7, LSTM ROC-AUC Curve illustrates the model's performance in distinguishing between classes. AUC values for all categories are high: Sports (1.00), World (0.99), Sci/Tech (0.99), and Business (0.98). These scores demonstrate excellent discriminative ability, with the curves closely approaching the top-left corner, indicating a high true positive rate and low false positive rate.

Bi-LSTM Evaluation Metrics:

- Misclassification Analysis Heatmap
- Prediction Confidence Distribution
- Accuracy Score

Effect of Fine-Tuning:

The misclassification heatmap in figure 9 shows error patterns across four categories: World, Sports, Business, and Sci/Tech. The most significant misclassifications occur between Business and Sci/Tech categories, with Business being misclassified as Sci/Tech 7% of the time, and Sci/Tech being misclassified as Business 6% of the time. Other misclassification rates are relatively low, ranging from 0-3%. While Bi-LSTM Prediction Confidence Distribution in figure 10 shows the model's high reliability with most correct predictions (green) concentrated near confidence scores of 1.0, reaching frequencies over 5000. Incorrect predictions (red) remain minimal and occur at lower confidence levels, demonstrating strong classification performance, with improved test accuracy score from 91.55% to 93.11%.

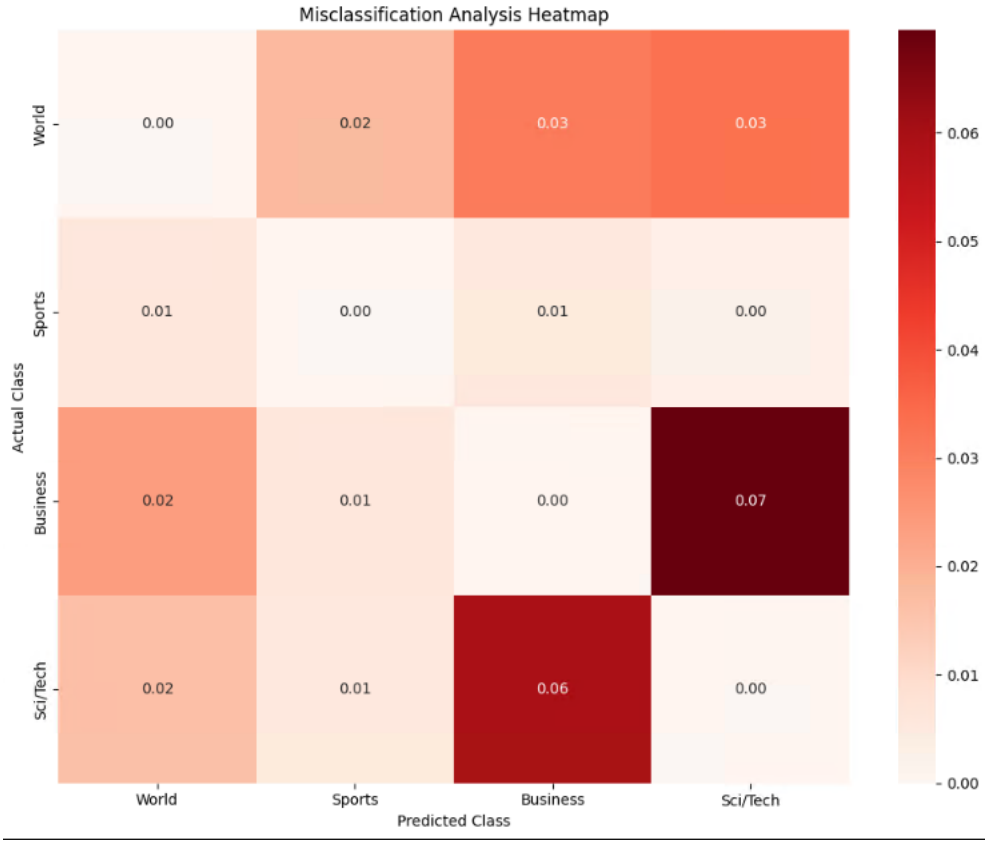

Figure 8: Bi-LSTM Misclassification Analysis Heatmap

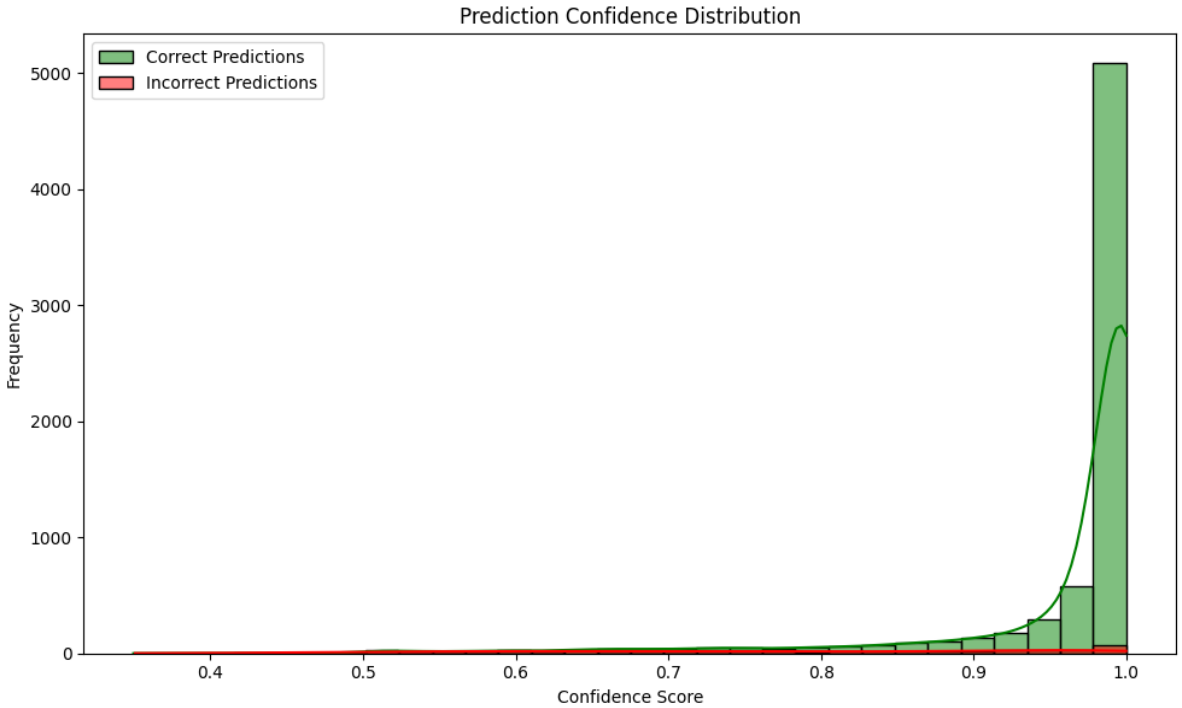

Figure 9: Bi-LSTM Prediction Confidence Distribution

12. Conclusion

In this report, I have explored various models for text classification, including traditional machine learning methods and deep learning approaches. Each model presents unique advantages and challenges, and the choice of model should align with the specific requirements of the task at hand. The deep learning models performed better than the traditional models and all the models generally outperformed the base line targets.

Citation styles standardize how sources are referenced in academic writing, varying by discipline and publication. MLA (Modern Language Association) is prevalent in humanities, using author-page in-text citations (e.g., Smith 23) and a "Works Cited" list. APA (American Psychological Association) is common in social sciences, featuring author-date citations (e.g., Smith, 2020) and a "References" list. Chicago style offers two systems: notes and bibliography, favored in humanities, and author-date, used in sciences. Harvard style, like APA, employs author-date citations and is widely used in various disciplines and I utilized the Harvard citation style. Vancouver style is numbered, often used in medical and scientific fields, with in-text numbers corresponding to a reference list (Wordvice KH, 2022).

References

Hsu, B.-M. (2020) 'Comparison of Supervised Classification Models on Textual Data', Mathematics, 8(5), p. 851. Available at: https://www.mdpi.com/2227-7390/8/5/851

Kadhim, A.I., Cheah, Y.-N. and Ahamed, N.H. (2014) 'Text Document Clustering Using K-means Algorithm with Dimension Reduction Techniques', in Proceedings of the 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation. IEEE, pp. 315–320. Available at: https://ieeexplore.ieee.org/document/7351815 (Accessed: [Date]).

Sherstinsky, A. (2020) 'Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network', Physica D: Nonlinear Phenomena, 404, p. 132306. Available at: https://doi.org/10.1016/j.physd.2019.132306 .

Mienye, I.D., Swart, T.G. and Obaido, G. (2024) 'Recurrent Neural Networks: A Comprehensive Review of Architectures, Variants, and Applications', Information, 15(9), p. 517. Available at: https://www.mdpi.com/2078-2489/15/9/517 .

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I. (2017) 'Attention Is All You Need', Advances in Neural Information Processing Systems, 30, pp. 5998–6008. https://doi.org/10.48550/arXiv.1706.03762

Devlin, J., Chang, M.W., Lee, K. and Toutanova, K. (2019) 'BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding', Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 1, pp. 4171–4186. https://doi.org/10.48550/arXiv.1810.04805

Wordvice KH (2022) Citation Styles: APA vs MLA, Vancouver vs Chicago Style [Blog post]. *Wordvice*. 23 September. https://www.wordvice.com/citation-styles-apa-vs-mla-vancouver-vs-chicago-style [Accessed 28th Dec 2024].

Sun, Y., Zeng, Q., Li, Y. and Bian, Y., 2020. Application Research of text classification based on random forest algorithm. In: 2020 3rd International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE). IEEE, pp.370-374. https://doi.org/10.1109/AEMCSE50948.2020.00086

Kumbhar, R., Mhamane, S., Patil, H., Patil, S. and Kale, S., 2020. Text Document Clustering Using K-means Algorithm with Dimension Reduction Techniques. In: 2020 Fifth International Conference on Communication and Electronics Systems (ICCES). IEEE, pp.1222-1228.

Fields, J., Chovanec, K. and Madiraju, P., 2024. A Survey of Text Classification with Transformers: How Wide? How Large? How Long? How Accurate? How Expensive? How Safe? IEEE Access, 12, pp.6518-6531.

Sunagar, P., Sowmya, B.J., Pruthviraja, D., Supreeth, S., Mathew, J., Rohith, S. and Shruthi, G., 2024. Hybrid RNN Based Text Classification Model for Unstructured Data. SN Computer Science, 5(726), pp.1-13. https://doi.org/10.1007/s42979-024-03091-x