CMPSC 448 - Final Project Report
News Classification with Neural Networks: Embracing a Comprehensive Approach

Vedant Sahai — vzs5356@psu.edu
Sri Krishna Chaitanya — vzs5369@psu.edu
Hrishikesh Shenai — hms6207@psu.edu

1 Introduction

In the ever-evolving digital age, the proliferation of news content across various platforms has led to an overwhelming influx of information. This presents a unique challenge in efficiently categorizing and understanding this vast amount of data. Our project, "News Classification with Neural Networks: Embracing a Comprehensive Approach," aims to address this challenge by leveraging deep-learning techniques. The focus is on developing a robust model that can accurately classify news articles into predefined categories based on their headlines and short descriptions. This endeavor is not only crucial for organizing content but also for enhancing accessibility and readability for end-users.

At the heart of our methodology is the application of Long Short-Term Memory (LSTM) networks, a concept pioneered by Hochreiter & Schmidhuber in their 1997 paper "Long Short-Term Memory" [3]. Their research presented a groundbreaking solution to the limitations of traditional Recurrent Neural Networks (RNNs), specifically addressing the vanishing gradient problem that hindered the learning of long-term dependencies. LSTMs introduced the novel concept of memory cells, capable of maintaining information over extended periods, thereby revolutionizing sequence modeling. In our project, we adapt this innovation in the form of Bidirectional LSTMs (BiLSTMs) to process the sequential nature of language in news headlines and descriptions effectively. The BiLSTM architecture enhances the LSTM's capabilities by analyzing data in both forward and backward directions, ensuring a richer understanding of context, which is essential in accurately categorizing news content.

Complementing the BiLSTMs, our framework also incorporates Convolution Neural Networks (CNNs), inspired by the groundbreaking work of LeCun et al. in their 1998 paper "Gradient-Based Learning Applied to Document Recognition" [6]. Their research demonstrated the powerful capability of CNNs in pattern recognition, particularly in spatial hierarchies of features. In our context, CNNs are employed for their proficiency in recognizing and extracting spatial patterns in textual data, a crucial aspect when dealing with the varied formats and structures of news content. The integration of CNNs allows our model to discern and categorize news articles with greater accuracy, capturing the subtle nuances often present in textual data.

A pivotal component of our project is the inclusion of the XLNet model, as introduced by Yang et al. in their 2019 paper "XLNet: Generalized Autoregressive Pre-training for Language Understanding" [5]. XLNet's innovation lies in its ability to capture a deep, contextual understanding of text by utilizing a permutation-based training approach. This approach allows XLNet to outperform its predecessors in understanding the intricacies and nuances of language, making it particularly adept for complex text classification tasks like news categorization. By incorporating XLNet, our model gains an enhanced capability to process and categorize news articles, considering the intricate contextual relationships within the text.

The integration of these diverse neural network architectures—each contributing its unique strengths — enables our project to address the multifaceted challenge of news classification in the digital age. By building upon these foundational research works, our project not only tackles the immediate challenge of classifying news content but also contributes to the evolving field of neural network applications in text analysis.

1.1 Task and Motivation

The central task of our project is to develop a deep-learning model that can automatically classify news articles into specific categories based on their headlines and short descriptions. This task addresses the challenge of managing and filtering the massive influx of news content generated daily. In the digital era, where news outlets and content creators incessantly publish diverse articles, readers often find it overwhelming to navigate and locate information relevant to their interests. Our model aims to simplify this process by categorizing articles into distinct genres like politics, sports, entertainment, and more.

This automated classification system is not only a tool for efficient content management for publishers but also a means to enhance the user experience by delivering tailored content. In an online world cluttered with information, the ability to quickly access news of interest is invaluable for users. Furthermore, this model has broader implications in various fields. In the realm of information retrieval, it can refine search engine capabilities, making it easier to find relevant articles. For digital marketing, it allows for more targeted advertising by aligning ad content with corresponding news categories. Lastly, in content recommendation systems, this classification can aid in suggesting articles that align with user preferences, thereby increasing engagement and satisfaction.

Our project is guided by several key objectives:

- **Accuracy and Efficiency:** Develop a model that accurately classifies news articles into the correct categories with high efficiency. Minimize misclassification to ensure reliable categorization.
- **Scalability and Robustness:** Create a scalable solution to accommodate the vast and diverse nature of news content. Maintain high performance regardless of the volume of data processed.
- **Generalization Capability:** Ensure the model generalizes well to unseen data, demonstrating its applicability to real-world scenarios beyond the confines of the training dataset.
- **Content-Centric Approach:** Emphasize the quality and relevance of classified content without explicitly considering user interests and preferences. Focus on optimizing content classification to enhance the overall content consumption experience.
- **Insightful Data Analysis:** Conduct thorough exploratory data analysis to gain deep insights into the nature of news data. Use these insights to inform the feature engineering and model selection process.

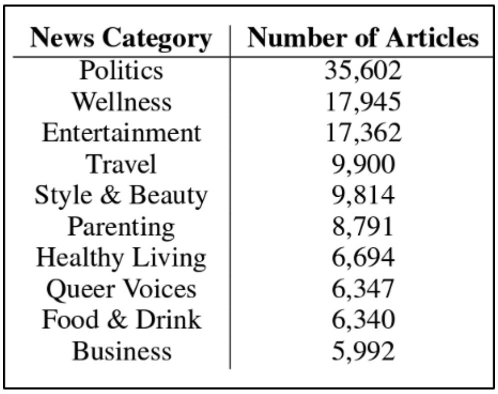
2 Dataset

In this section, we provide an overview of the data in our News Category Dataset [1]. This dataset is unique in that it provides high-quality text content along with fine-grained category information for a significant number of articles. The dataset, which consists of 210,294 news headlines, contains news articles from the Huffington Post that were published between 2012 and 2022. The following characteristics are present in every record in our dataset:

- **Category:** The category in which the article was published.
- **Headline:** The headline of the news article.
- **Short Description:** An abstract of the news article.

Interestingly, the dataset includes articles from 42 different categories, offering a rich and extensive source for investigation and analysis. Based on the number of related articles, Fig 1 displays the top 10 categories in the dataset.

A thorough analysis of the article lengths conducted during the Exploratory Data Analysis (EDA) phase indicated a noteworthy insight of around 200 words in length, as indicated in Fig 2 for most of the articles irrespective of the category. Nonetheless, it was clear that some of the articles had lengths of less than 100 words, and there were others with lengths of more than 350 words, demonstrating the dataset's diversity in content lengths. One significant observation during the EDA was the presence of class imbalance among the top 14 categories, as depicted in Fig 3. Notably, the ”Politics” category exhibited the highest number of articles, totaling approximately 31,000, while the ”Home & Living”



News Category	Number of Articles
Politics	35,602
Wellness	17,945
Entertainment	17,362
Travel	9,900
Style & Beauty	9,814
Parenting	8,791
Healthy Living	6,694
Queer Voices	6,347
Food & Drink	6,340
Business	5,992

Figure 1: Top-10 Categories [1]

category had a much lower count of around 4,000 articles. Recognizing this class imbalance proved crucial in subsequent stages of our modeling process, influencing decisions regarding model architecture and emphasizing the need for strategies to handle uneven class distribution.

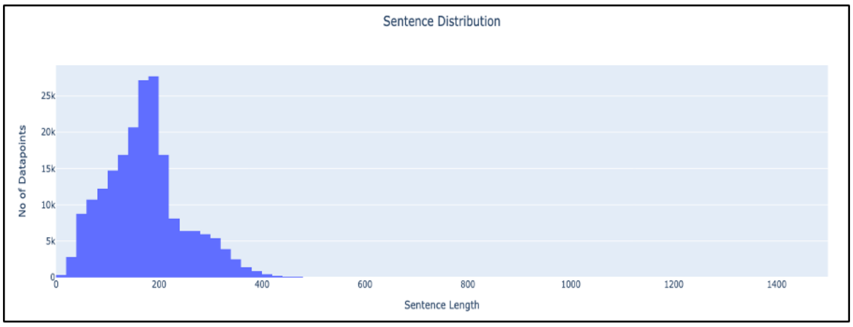


Figure 2: Distribution of Length of Sentences across Articles

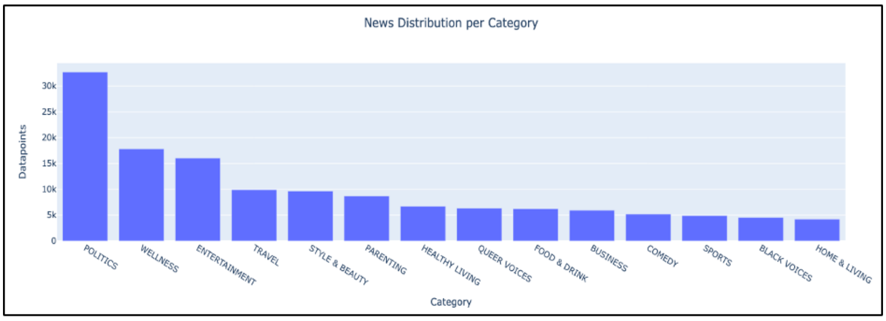


Figure 3: Distribution of Articles per Category

To delve deeper into the dataset characteristics, Fig 4 provided insights into the average article lengths within each category. Understanding the average length of articles for each category is pivotal for modeling decisions, as it can influence the selection of appropriate features and model parameters tailored to the characteristics of each class.

Moreover, the word cloud shown in Fig 5 visually highlights the most frequent words in the dataset, offering insights into prevalent themes. It serves as a valuable tool for identifying recurring topics and understanding the dataset's content distribution. This visual representation aids in feature selection and informs decisions related to class imbalances, providing a concise summary of key textual patterns.

These findings from the EDA phase not only offer a deeper understanding of the dataset's content distribution but also serve as a foundation for informed decisions in the subsequent stages of pre-processing.

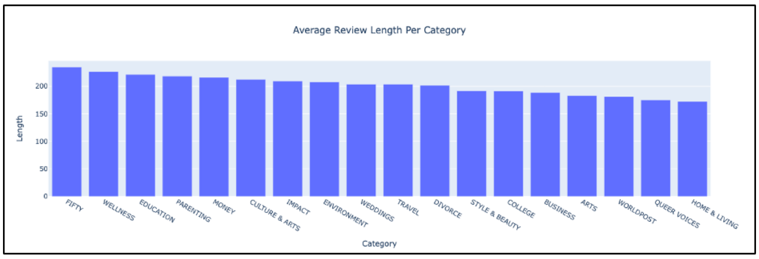


Figure 4: Average length of Articles per Category

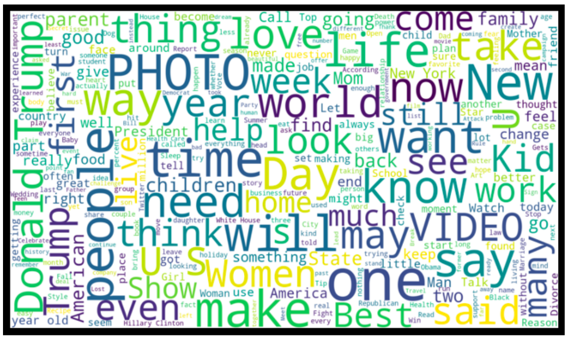


Figure 5: Wordcloud across all Articles

2.1 Pre-Processing

Data pre-processing is required for classification tasks to ensure that the input data is in a format suitable for deep learning models, improving their ability to learn and make accurate predictions. Proper pre-processing improves the classification model's overall performance and generalisation.

- **Text Cleaning:** Stopwords, HTML tags, special characters (emoticons, emojis), punctuation marks, and symbols were removed from raw text data. This enables models to concentrate on the relevant content of news headlines for more accurate analysis and classification.
- **Numbers Removal:** Using digit masking, numerical elements were replaced with '#' characters, simplifying numerical representation and allowing models to capture overall patterns without being influenced by specific numbers.
- **Expanding Contractions:** To provide explicit representation to models and avoid ambiguity, expanded contractions (shortened forms of words or combinations) to their full forms.
- **Word Lemmatization:** Reduced words to their base or root form through lemmatization, treating different variations of words as the same. This reduces vocabulary size and simplifies the model's task.
- **Empty Strings Removal:** Removed empty strings or whitespaces that do not contribute meaningful information to the data.
- **Class Imbalance Mitigation:** Merged similar classes (e.g., Money, Business, Finance to Finance) to address the class imbalance and enhance the model's generalization capacity.
- **Uniformly Pre-processed Data:** The consolidation of similar classes resulted in uniformly pre-processed data, maintaining consistency across all models.

3 Neural Network Architectures

3.1 Convolution Neural Network

CNNs are adept at spatial hierarchies and feature extraction, primarily through convolution filters that capture patterns in spatial data. The architecture 6 excels in image recognition, object detection, and certain natural language processing tasks.

Model Architecture:

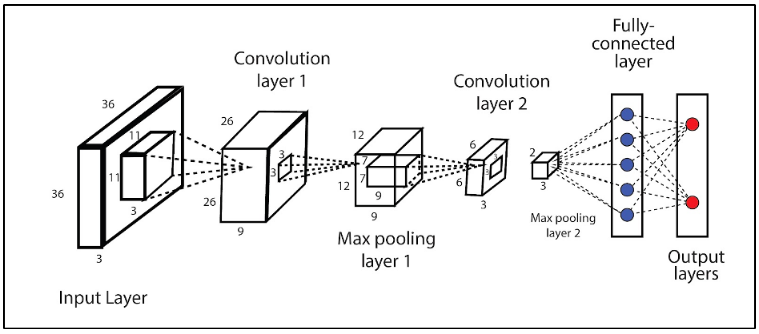


Figure 6: Architecture of a Convolution Neural Network
[17]

Table 1: CNN Model Summary

Layer Name	**Output Shape**	**# Param**
conv0	[1, 50, 300, 1]	17,800
conv0.0.2	[1, 1, 300, 300]	0
conv1	[1, 50, 298, 1]	35,300
conv1.0.2	[1, 1, 300, 300]	0
conv2	[1, 50, 296, 1]	52,800
conv2.0.2	[1, 1, 300, 300]	0
conv3	[1, 50, 294, 1]	70,300
conv3.0.2	[1, 1, 300, 300]	0
conv4	[1, 50, 292, 1]	87,800
conv4.0.2	[1, 1, 300, 300]	0
global_max_pool	[1, 50, 1, 1]	0
dropout	[1, 50, 1, 1]	0
fc	[1, 16]	816
Total Params	**281,800**	

- **Embedding Layer:** Maps 18,000 words into 300-dimensional vectors using pre-trained embeddings.
- **Convolution Blocks:** Five sequential blocks, each with Conv2d, BatchNorm2d, and LeakyReLU activation's. Filters of varying sizes (1x300 to 5x300) extract textual features at multiple scales.
- **Adaptive Max Pooling:** Reduces dimensional while retaining critical features.
- **Dropout and Fully Connected Layer:** A dropout rate of 0.12 prevents over-fitting. The fully connected layer maps feature 16 classification categories.

3.2 Bi-Directional Long Short Term Memory

RNNs shine in sequential data handling, leveraging memory elements to maintain context. Bi-LSTMs shown in fig 7 extend RNN capabilities, analyzing sequences in both directions for a comprehensive understanding, crucial for tasks like sentiment analysis.

Model Architecture:

- **Embedding Layer:** Mirrors the CNN's embedding specifications.
- **Bidirectional LSTM Layers:** A hidden size of 64 captures contextual information from text sequences.
- **Linear Transformation and Dropout:** A linear layer and ReLU activation interpret LSTM features. A dropout rate of 0.30 offers regularization.
- **Output Layer:** Aligns LSTM features to 10 classification categories.

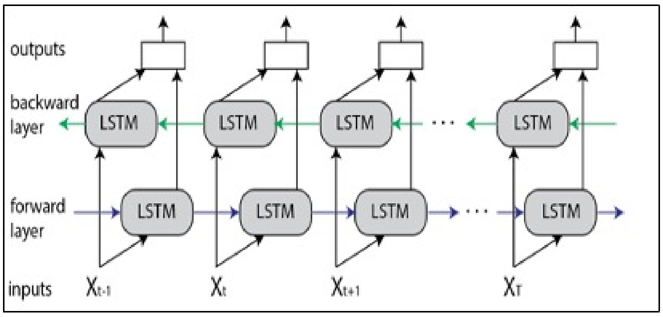


Figure 7: Architecture of Bi-Directional LSTMs [4]

Table 2: Bi-LSTM Model Summary

Layer Name	**Output Shape**	**# Param**
embedding	[1, 300, 300]	0
Astm	[1, 300]	187,392
linear	[1, 300]	16,448
relu	[1, 300]	0
dropout	[1, 300]	0
out	[1, 300]	650
Total Params	**204,490**	

3.3 Transformers

XLNet [5] is a transformer [9]-based language model where ”XL” stands for ”eXtra Long,” emphasizing the model’s ability to capture longer-term dependencies in sequential data. Like BERT, XLNet consists of self-attention mechanisms to capture contextual information from input sequences efficiently

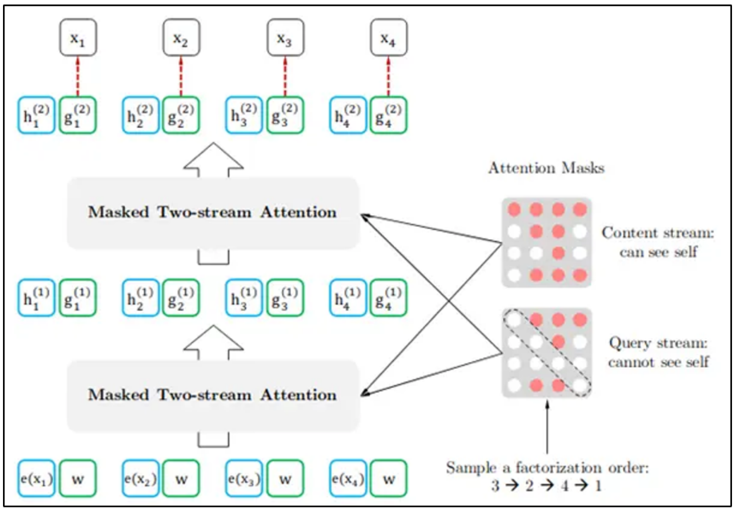

x_1
x_2
x_3
x_4
$h_1^{(2)}$ $g_1^{(2)}$ $h_2^{(2)}$ $g_2^{(2)}$ $h_3^{(2)}$ $g_3^{(2)}$ $h_4^{(2)}$ $g_4^{(2)}$
Attention Masks
Masked Two-stream Attention
Content stream: can see self
$h_1^{(1)}$ $g_1^{(1)}$ $h_2^{(1)}$ $g_2^{(1)}$ $h_3^{(1)}$ $g_3^{(1)}$ $h_4^{(1)}$ $g_4^{(1)}$
Query stream: cannot see self
Masked Two-stream Attention
$e(x_1)$ w $e(x_2)$ w $e(x_3)$ w $e(x_4)$ w
Sample a factorization order: 3 → 2 → 4 → 1

Figure 8: Architecture of XLNet Transformer

Key features of XLNet include:

- **Permutation Language Modeling (PLM):** XLNet considers all permutations of the sequence when predicting the next word. This approach allows the model to capture bidirectional context information and also helps in handling longer-term dependencies.
- **Auto-regressive and Auto-encoding Pre-training:** XLNet combines both auto-regressive and auto-encoding (like BERT) training objectives. This dual training strategy helps XLNet benefit from the strengths of both approaches.
- **Segmented Permutations:** XLNet uses segmented permutations during training, allowing the model to handle different segments of the sequence separately. This is particularly beneficial when dealing

with tasks that involve multiple sentences or segments of text.

Table 3: XLNet Model Summary

Layer Name	**Output Shape**
word_embedding	[1, 768]
XLNetLayer (0-11)	[1, 768]
rel_attn - LayerNorm	[1, 300]
rel_attn - Dropout	[1, 768]
FF - LayerNorm	[1, 768]
FF - Linear 1	[1, 3072]
FF - Linear 2	[1, 768]
FF - Dropout	[1, 768]
FF - GELUActivation	[1, 768]
XLNetModel Dropout	[1, 768]
sequence_summary - Linear	[1, 768]
sequence_summary - Tanh	[1, 768]
logits_proj	[1, 10]
Total Params	**110M**

Model Architecture:

- **Embedding Layer:** Maps 32,000 words into 786-dimensional vectors using pre-trained embeddings.
- **Stack of Transformer blocks:** Each transformer block contains multiple layers of self-attention mechanisms and feed-forward neural networks with GELU [8] Activation.
- **Attention masks:** Applied to the input data during training to control which parts of the sequence the model should attend to.
- **Content Stream and Query Stream:** Each stream processes and attends to content and query respectively.
- **Linear Transformation and Dropout:** A linear layer and Tanh activation interpret model features. A dropout rate of 0.10 offers regularization.
- **Output Layer:** Aligns model features to 10 classification categories.

4 Methodology and Training Details

- **Common Data Pre-processing and Feature Engineering:**
  - **Data Preparation:** 'headline' and 'short-description' columns are combined into 'full-review', removing any null values.
  - **Category Filtering:** Targeted categories with more than 5000 reviews to ensure a balanced dataset.
  - **Tokenization and Padding:** Text data are converted into padded integer sequences using Keras Tokenizer in the case of CNN and RNN and XLNetTokenizer in the case of Transformer.
  - **Word Embeddings:** Implemented 300-dimensional GloVe [7] embedding for enhanced word representation (for CNN and RNN).
- **CNN Model Training:**
  - **Architecture:** The CNN model employs a range of filter sizes and multiple convolution layers, coupled with batch normalization and LeakyReLU [10]activation.
  - **Hyper-parameter Tuning:** Utilizing Optuna [11], key parameters like learning rate (approx. 6.10e-05), number of filters (50), dropout rate (0.12), and batch size (64) were optimized.
  - **Class Weighting:** Addressed class imbalance using class weights inversely proportional to class frequencies.
  - **Loss and Optimization:** Adopted Weighted Cross-Entropy Loss and AdamW optimizer.
  - **Evaluation:** Performance assessed on validation data, monitoring metrics such as loss and accuracy over 10 epochs.
- **RNN (Bi-LSTM) Model Training:**
  - **Architecture:** The BiLSTM model integrates bidirectional LSTM layers with dropout (approx. 0.30) for regularization and ReLU activation.
  - **Hyperparameter Tuning:** Conducted through Optuna, leading to an optimized learning rate of approx. 0.00522 and batch size of 512.
  - **Class Weight Calculation:** Similar to CNN, class weights were computed to balance the dataset.
  - **Training and Validation:** The model underwent rigorous training and validation processes, with performance metrics recorded for each epoch.
  - **Optimization:** Used Adam [12] optimizer and Cross-Entropy Loss, adjusted for class weights.
- **Transformer (XLNet) Model Training:**
  - **Architecture:** The pre-trained model used was “xlnet-base-cased” which comprises XLNetLayer, XLNetFeedForward, and SequenceSummary layers with dropout (approx. 0.10) for regularization and GELU and Tanh activations.
  - **Hyper-parameter Tuning:** Learning rate of 2e-5, batch size of 8, and weight decay of 0.01.
  - **Training and Validation:** The model was trained for 10 epochs on a GPU with evaluations being conducted after every epoch.
  - **Optimization:** Internally, the optimizer used was AdamW.

5 Results

For quantitative analysis in multi-class classification tasks, we used the F1 score, a metric that strikes a balance between recall and precision. Recall measures the model's capacity to capture all pertinent instances of a class, whereas precision indicates the accuracy of positive predictions. Because it is the harmonic mean of recall and precision, the F1 score provides a thorough analysis that is especially useful when class distributions are unbalanced. This is important because under-representation of certain classes makes accuracy on its own unreliable. The F1 score is robust for multi-class classification evaluations because it takes into account both false positives and false negatives, giving a nuanced understanding of a model's effectiveness across a range of class outcomes.

Complementing this, the confusion matrix was also taken into consideration as a versatile tool for assessing the performance of multi-class classification models. It furnishes a detailed breakdown of predictions, showcasing instances correctly or incorrectly classified per class. Derived metrics like precision, recall, and F1 score directly stem from the confusion matrix. Precision measures the accuracy of positive predictions, recall gauges the model's ability to capture actual positives, and the F1 score strikes a balance between the two, proving especially useful in scenarios with imbalanced class distributions. Examining the confusion matrix allows us to pinpoint the strengths or weaknesses in a model, providing a granular understanding of its performance across various classes. This detailed breakdown proves invaluable in the evaluation of multi-class classification models.

5.1 Quantitative Analysis

Table 4: Classification Metrics across the Models

METRICS		CNN	LSTM	XLNet - TRANSFORMER
Top 10 Labels	**F1**	0.74	0.76	**0.84**
	Recall	0.74	0.76	**0.84**
	Precision	0.75	0.76	**0.84**
	Support	47079	47079	**27430**
Top 16 Labels	**F1**	0.77	0.76	0.80
	Recall	0.77	0.76	0.80
	Precision	0.77	0.76	0.80
	Support	36207	36207	27430

The results of all three trained models across the Top-10 and Top-16 labels are shown in Table 4. There is a notable performance discrepancy among the models when considering different label sets, specifically the Top-10 and Top-16 categories. In the context of Top-10 labels, both LSTM and the XLNet Transformer demonstrate superior performance, highlighting their effectiveness in capturing nuances within a more focused range of categories. Conversely, CNN exhibits better performance on the Top-16 labels, indicating its proficiency in handling a broader spectrum of categories. Notably, the XLNet Transformer outperforms both CNN and LSTM by a significant margin, particularly excelling in F1 scores with a notable improvement of +13.5% compared to CNN and +10.5% when compared to the LSTMs. This superior performance could be attributed to the XLNet Transformer's capability to capture intricate contextual dependencies within the data.

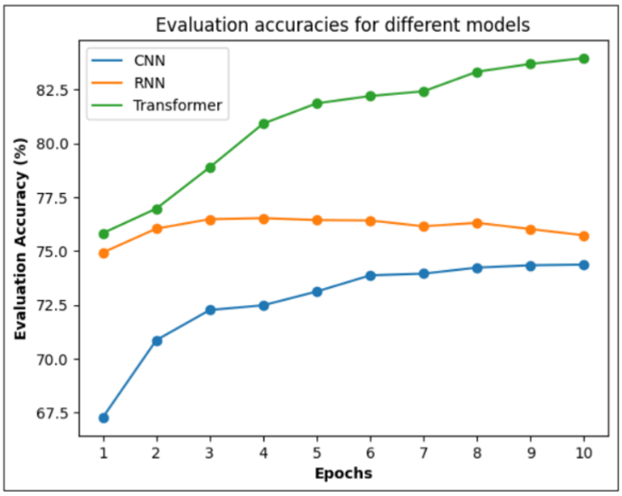
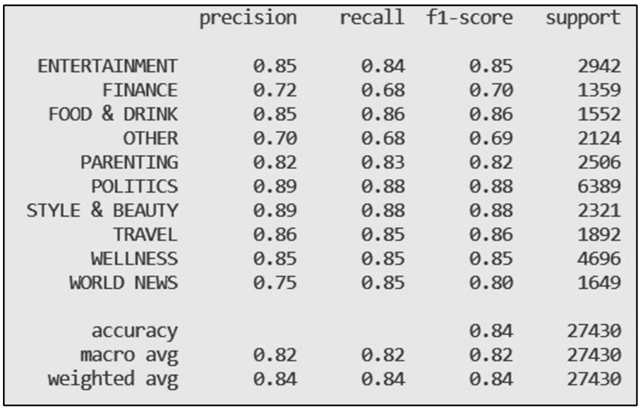


Figure 9: Validation Accuracy for Top-10 Labels per Epoch

Moreover, the validation accuracy per epoch for the Transformers, as shown in Figure 9, is consistently higher than that of the other model implementations. This consistent trend further reiterates the superiority of the XLNet Transformer over CNNs and LSTMs for multi-class text classification tasks. The model's ability to maintain higher accuracy throughout the training process indicates its robust learning capacity and adaptability to the complexities present in the dataset. This underscores the potential of the XLNet Transformer as a powerful tool for tasks requiring nuanced understanding and classification of textual data.

From the table 4, it is evident that the XLNet Transformer performed the best among the models when considering the Top-10 labels. To delve deeper into the analysis, we further examine the per-class classification metrics of transformers. In Figure 10, we observe the performance across individual classes. The analysis of the per-class classification metrics reveals intriguing insights. Specifically, the Transformer model excels in classifying articles related to the "Politics" label, demonstrating the highest performance in this category as also shown in the confusion matrix 11. On the other hand, its performance is comparatively lower when classifying articles labeled as "Other." This discrepancy in performance could be attributed to two main factors: data imbalance and data quality. Firstly, the imbalance in the distribution of data points across different classes might impact the model's ability to generalize equally well across all categories. In the case of the "Politics" label, where the Transformer performs exceptionally well, there may be a more balanced representation of data points, allowing the model to learn and generalize effectively. Secondly, the quality of data within the "Other" category might pose challenges. The presence of noise, outliers, or insufficiently representative samples could hinder the model's ability to discern meaningful patterns, leading to lower classification accuracy.



```
                precision    recall  f1-score   support

ENTERTAINMENT       0.85      0.84      0.85      2942
      FINANCE       0.72      0.68      0.70      1359
 FOOD & DRINK       0.85      0.86      0.86      1552
        OTHER       0.70      0.68      0.69      2124
    PARENTING       0.82      0.83      0.82      2506
     POLITICS       0.89      0.88      0.88      6389
STYLE & BEAUTY      0.89      0.88      0.88      2321
       TRAVEL       0.86      0.85      0.86      1892
     WELLNESS       0.85      0.85      0.85      4696
   WORLD NEWS       0.75      0.85      0.80      1649

     accuracy                           0.84     27430
    macro avg       0.82      0.82      0.82     27430
 weighted avg       0.84      0.84      0.84     27430
```

Figure 10: Classification Report for Top-10 Labels by the XLNet-Transformer

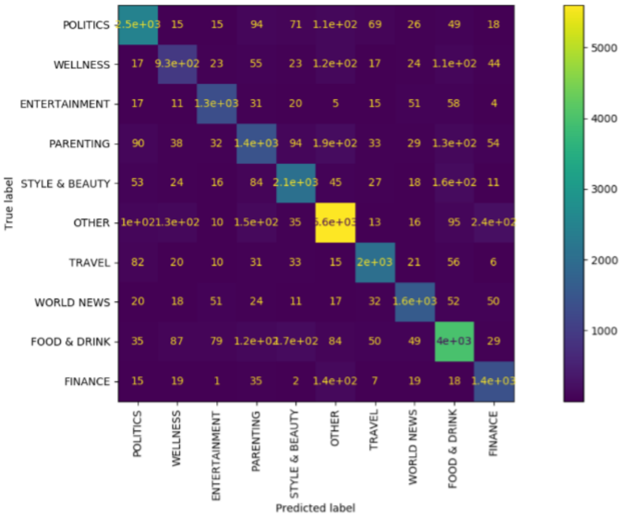


Figure 11: Confusion Matrix for Top 10-Labels by the XLNet-Transformer

5.2 Qualitative Analysis

The table presented in 5 provides a comparison of predictions made by all our models on a set of randomly selected News Headlines and corresponding short descriptions from Google News. As observed from the table, the Transformer consistently outperforms both CNN and RNN in accurately predicting the news category. Further analysis of a larger set of news articles reaffirms the Transformer's superior performance over CNN and RNN models.

Table 5: Test Predictions across the Models

Headline	**Short Description**	**Ground Truth**	**CNN**	**RNN**	**Transformer**
Hamas militants kill people in Jerusalem	Seven people were injured after the two attackers open fire on people in Jerusalem	World News	Other	Other	World News
EUR/USD slips back to 1.0880	The EUR/USD pressured as markets turn back towards the US Dollar.	Finance	Finance	Finance	Finance
Slow start for Tiger Woods at World Series	Tiger Woods trails in the competition carding a 3-over 75.	Sports	Entertainment	Sports	Sports
Respiratory illness surge in China not due to new virus	The surge is concentrated in north China, hospitals have been ”overwhelmed with sick children.”	Health	World News	Health	Health
Disney Honours ’Black Panther’ Chadwick Boseman	Disney has updated the Marvel logo introduction to ’Black Panther’ to honor Chadwick Boseman's 44th birthday	Entertainment	Entertainment	Entertainment	Entertainment

6 Challenges and Obstacles

The identification of class imbalances underscores the importance of employing strategies to address this issue during the modeling phase, ultimately contributing to the development of a more robust and effective model.

- **Imbalanced Dataset Overview:**
  - As seen from the Fig 3, the dataset was highly imbalanced.
  - One class had about 32K samples, while the class with the lowest samples had about 1K samples.
- **Challenges in Dealing with Imbalanced Datasets:**
  - **Biased Model Learning:** The imbalanced distribution can lead the model to prioritize accuracy on the majority class while neglecting minority classes.
  - **Poor Generalization:** Unbalanced datasets can cause models to generalize poorly to new, unseen data as they have been trained on a dataset where those classes are underrepresented.
  - **Difficulty in Capturing Minority Class Patterns:** Models may struggle to capture the underlying patterns within the minority classes due to their limited representation in the training data, resulting in poor predictive performance for these classes.
  - **Model Sensitivity to Noise:** Unbalanced datasets may contain noise or outliers that disproportionately affect the minority classes. Models trained on such data can be sensitive to this noise, leading to sub-optimal performance.

- **Impact on Decision Threshold:** The default decision threshold of classification models is typically set for balanced datasets. In unbalanced datasets, adjusting the decision threshold may be necessary to better balance precision and recall.
- **Difficulty in Model Interpretability:** Models trained on highly unbalanced datasets might produce decision boundaries that are hard to interpret, especially when it comes to understanding the importance of features for minority class prediction.

To address these challenges, we experimented with the following metrics:

- **Under-Sampling:** Strategically undersampled the majority class samples by randomly removing samples from the respective classes. Despite this effort to achieve a more balanced distribution, the anticipated enhancement in model performance was not substantial. This may be attributed to the fact that under-sampling resulted in the loss of valuable information, ultimately diminishing the contextual richness available to the model.

- **Hyper-Parameter Tuning:** Fine-tuned specific model parameters, including Loss Rate, number of filters (in case of CNN), and dropout rate, and experimented with techniques such as the weighted loss function. Although a modest performance improvement was observed, achieving significant gains proved challenging due to the substantial under-representation of the minority classes.

- **Combining Minority Classes:** As mentioned in the pre-training section, similar classes were consolidated into a single class. Merging similar classes enhanced our model's ability to generalize to unseen data by providing a more robust representation for the combined category. Also, fewer classes led to more interpretable and actionable results.

- **Transfer Learning in Transformers:** The XLNet Transformer was pre-trained on various text corpora including Wikipedia, BooksCorpus, Giga5, ClueWeb, and Common Crawl. Hence, the pre-trained XLNet model has already learned rich representations and was able to adapt to the unbalanced dataset, outperforming CNN/RNN-based models.

7 Conclusion

In conclusion, the in-depth examination of various models, including CNN, LSTM, and the XLNet Transformer, sheds light on their effectiveness in a multi-class text classification setting. Significantly, the XLNet Transformer stands out as the top performer, showcasing notable improvements in F1 scores. With an impressive boost of +13.5% over CNN and +10.5% compared to LSTMs, the XLNet Transformer's superior performance underscores its capacity to capture intricate contextual dependencies within the data. This substantial advantage reinforces its robust learning capabilities and adaptability during training, making it the preferred choice for multi-class text classification tasks.

The observed disparities in per-class classification metrics emphasize the importance of considering both data balance and quality when interpreting model performance. Addressing data imbalances and ensuring high-quality data within specific categories may contribute to refining the model's performance across all classes. Overall, the results underscore the efficacy of XLNet Transformer in handling the complexities of multi-class text classification, making it a compelling choice for tasks that demand nuanced understanding and accurate categorization of textual data.

References

[1] Misra, Rishabh. "News Category Dataset." arXiv preprint arXiv:2209.11429 (2022).

[2] MK Gurucharan. "Basic CNN Architecture: Explaining 5 Layers of Convolutional Neural Network." July 27, 2022. https://www.upgrad.com/blog/basic-cnn-architecture

[3] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," in Neural Computation, vol. 9, no. 8, pp. 1735-1780, 15 Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[4] Yichen Zhao. "Complete Guide to RNN, LSTM, and Bidirectional LSTM." March 12, 2023. https://dagshub.com/blog/rnn-lstm-bidirectional-lstm

[5] Yang Zhilin, Dai Zihang, Yang Yiming, Carbonell Jaime G., Salakhutdinov Ruslan, and Le Quoc V.. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In Proceedings of the 33rd International Conference on Neural Information Processing Systems. 5754–5764.

[6] Lecun, Yann & Bottou, Leon & Bengio, Y. & Haffner, Patrick. (1998). Gradient-Based Learning Applied to Document Recognition. Proceedings of the IEEE. 86. 2278 - 2324. 10.1109/5.726791.

[7] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

[8] Hendrycks, Dan, and Kevin Gimpel. 'Gaussian Error Linear Units (GELUs)'. arXiv [Cs.LG], 2023, http://arxiv.org/abs/1606.08415. arXiv.

[9] Vaswani, Ashish, et al. 'Attention Is All You Need'. arXiv [Cs.CL], 2023, http://arxiv.org/abs/1706.03762. arXiv.

[10] Xu, Bing, et al. 'Empirical Evaluation of Rectified Activations in Convolutional Network'. arXiv [Cs.LG], 2015, http://arxiv.org/abs/1505.00853. arXiv.

[11] Akiba, Takuya, et al. 'Optuna: A Next-Generation Hyperparameter Optimization Framework'. arXiv [Cs.LG], 2019, http://arxiv.org/abs/1907.10902. arXiv.

[12] Kingma, Diederik P., and Jimmy Ba. 'Adam: A Method for Stochastic Optimization'. arXiv [Cs.LG], 2017, http://arxiv.org/abs/1412.6980. arXiv.

[13] Y. V. Singh, P. Naithani, P. Ansari and P. Agnihotri, "News Classification System using Machine Learning Approach," 2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), Greater Noida, India, 2021, pp. 186-188, doi: 10.1109/ICAC3N53548.2021.9725409.

[14] F. Ahmed, N. Akther, M. Hasan, K. Chowdhury and M. S. H. Mukta, "Word Embedding based News Classification by using CNN," 2021 International Conference on Software Engineering & Computer Systems and 4th International Conference on Computational Science and Information Management (ICSECS-ICOCSIM), Pekan, Malaysia, 2021, pp. 609-613, doi: 10.1109/ICSECS52883.2021.00117.

[15] T. A. Chowdhury, N. J. Tonoya, T. Maliha, P. Akter and M. S. Mahbub, "Age Based News Classification using LSTM and BERT," 2022 International Conference on Computational Modelling, Simulation and Optimization (ICCMSO), Pathum Thani, Thailand, 2022, pp. 1-6, doi: 10.1109/ICCMSO58359.2022.00014.

[16] L. Deping, W. Hongjuan, L. Mengyang and L. Pei, "News text classification based on Bidirectional Encoder Representation from Transformers," 2021 International Conference on Artificial Intelligence, Big Data and Algorithms (CAIBDA), Xi'an, China, 2021, pp. 137-140, doi: 10.1109/CAIBDA53561.2021.00036.

[17] https://www.analyticssteps.com/blogs/convolutional-neural-network-cnn-graphical-visualization-code-explanation