EFFECT OF VOCABULARY SIZE AND CLASS IMBALANCE ON LSTM-BASED TEXT CLASSIFICATION MODELS [*]

Youngkee Kim
AI Research 13th
MODU LABS
Seoul, South Korea
zerokee98@naver.com

ABSTRACT

Recurrent neural networks (RNNs), including LSTM, BiLSTM, and GRU, have shown notable success in natural language processing tasks such as text classification. However, their performance is greatly influenced by preprocessing choices—particularly vocabulary size and class imbalance. In this study, we conduct a comparative analysis of these models on the Reuters-21578 dataset under three conditions: baseline, class-weighted training, and data augmentation. By systematically varying the vocabulary size, we find that stable and reliable performance—measured by both accuracy and macro-averaged F1-score—emerges only when the vocabulary covers at least 45% of the total token distribution. Among the models, BiLSTM consistently performs best across all scenarios. Our findings offer practical insights into preprocessing strategies that enhance performance stability and fairness in imbalanced multi-class NLP tasks.

***K*eywords** Text Classification · Class Imbalance · Vocabulary Size · LSTM

1 Introduction

Recurrent Neural Networks (RNNs), including LSTM [1], BiLSTM [2], and GRU [3], are widely used in NLP tasks for their ability to model sequential data. However, their performance is sensitive to preprocessing factors such as vocabulary size, which affects model capacity, training speed, and overfitting risk.

Additionally, real-world datasets like Reuters-21578 [4] often exhibit class imbalance, causing models to favor dominant labels. In such settings, macro F1-score is a more informative metric than accuracy [5, 6].

This study compares LSTM, BiLSTM, and GRU models on the Reuters dataset while varying vocabulary size. We evaluate classification performance using accuracy and F1-score, and test strategies such as class weighting and data augmentation to improve robustness under imbalance.

1.1 Related Work

Recurrent neural networks (RNNs) have been widely adopted in sequence modeling tasks due to their ability to process variable-length input and capture temporal dependencies. Hochreiter and Schmidhuber [1] introduced the Long Short-Term Memory (LSTM) architecture to address the vanishing gradient problem in standard RNNs by incorporating memory cells and gating mechanisms. To simplify the architecture while maintaining performance, Cho et al. [3] proposed the Gated Recurrent Unit (GRU), which replaces the complex LSTM cell with a more compact structure using only update and reset gates. Schuster and Paliwal [2] further extended this line of work by proposing Bidirectional RNNs, which combine forward and backward passes to exploit both past and future context, later commonly implemented as Bidirectional LSTMs (BiLSTMs).

[*] *Citation*: **Authors. Title. Pages.... DOI:000000/11111.**

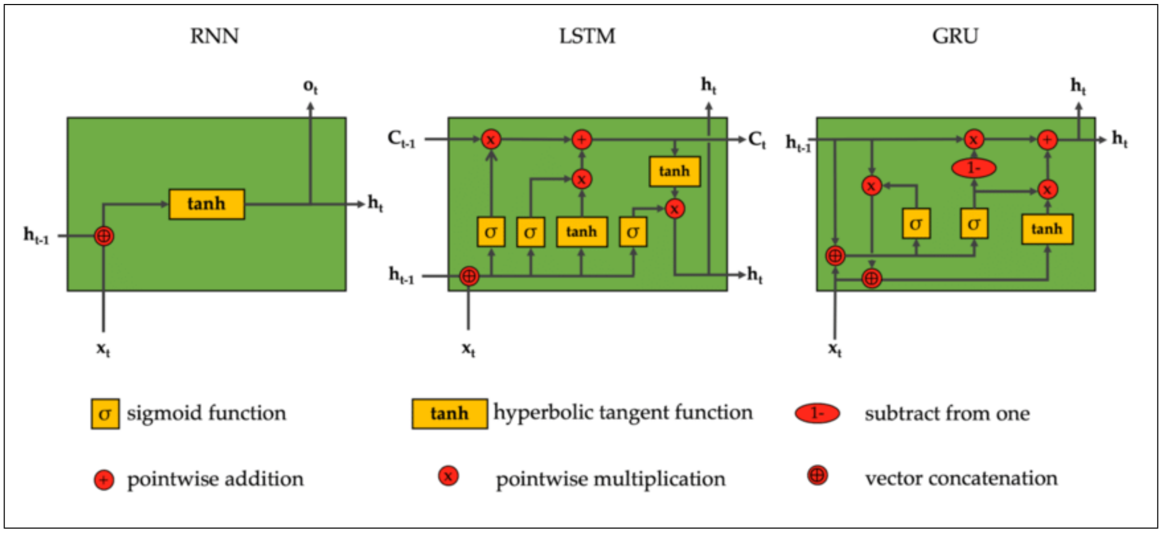


Figure 1: Comparison of RNN, LSTM, and GRU architectures. The diagram illustrates the internal operations of each model, including activation functions and gating mechanisms.

These RNN-based models have been applied to a wide variety of NLP tasks, including text classification, where capturing sequential dependencies and semantic context is critical. However, model performance is often sensitive to preprocessing parameters such as the vocabulary size. Moreover, text classification tasks frequently suffer from class imbalance, where the distribution of labels is skewed, leading to biased model predictions. Hossain et al. [5] explored strategies to mitigate the effects of imbalance in text classification, emphasizing the importance of fair evaluation metrics such as F1-score. Yildiz [6] proposed a data redistribution approach to improve classification accuracy on imbalanced datasets using deep learning models.

In addition, the Reuters-21578 dataset [4] has served as a benchmark corpus for evaluating text categorization systems. It presents a realistic challenge due to its multi-class nature and unbalanced label distribution. The dataset remains a standard in assessing the effectiveness of various machine learning and deep learning methods.

Building upon these studies, our work compares the performance of LSTM, BiLSTM, and GRU models on the Reuters dataset while systematically varying the vocabulary size. Furthermore, we examine how class imbalance affects model performance and apply complementary evaluation metrics to provide a more comprehensive understanding of each architecture's behavior under different data conditions.

2 Dataset and Preprocessing

2.1 Class Imbalance

The Reuters-21578 dataset consists of 11,367 newswire articles labeled across 46 topic categories. However, the class distribution is highly skewed. A small number of categories, such as "earn" and "acq", account for a substantial portion of the dataset, while many others contain fewer than 50 examples. This imbalance poses significant challenges for training deep learning models, as the model may disproportionately learn patterns from dominant classes and ignore minority ones.

Figure 2 illustrates the number of samples per class, highlighting the severity of the imbalance. For example, the largest class contains over 3,000 samples, while several classes have as few as 10–20 samples.

To address this issue, we implemented two complementary strategies:

- **Class weighting:** During training, the loss function was modified to incorporate inverse-frequency-based class weights, ensuring greater contribution from minority classes.
- **Data augmentation (oversampling):** Rare classes were synthetically oversampled in the training set to balance class frequencies.

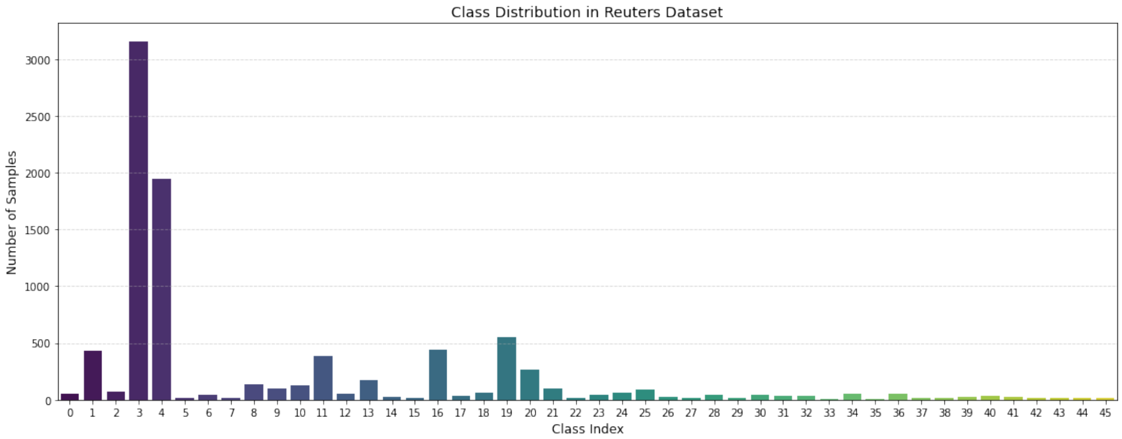


Figure 2: Class-wise sample distribution in the Reuters-21578 dataset. A long-tail pattern is evident, with a few classes dominating and many others underrepresented.

We compared these two settings — class weights only vs. class weights plus augmentation — to evaluate how each method affects model performance and robustness, particularly in F1-score for minority categories.

2.2 Document Length Distribution

Understanding the length of each text sample is critical for determining the appropriate input sequence length for RNN-based models. We measured the number of tokens in each document and plotted their distribution in Figure 3. The distribution exhibits a heavy-tail pattern resembling a logarithmic decay: while most documents are concise, a small number are exceptionally long.

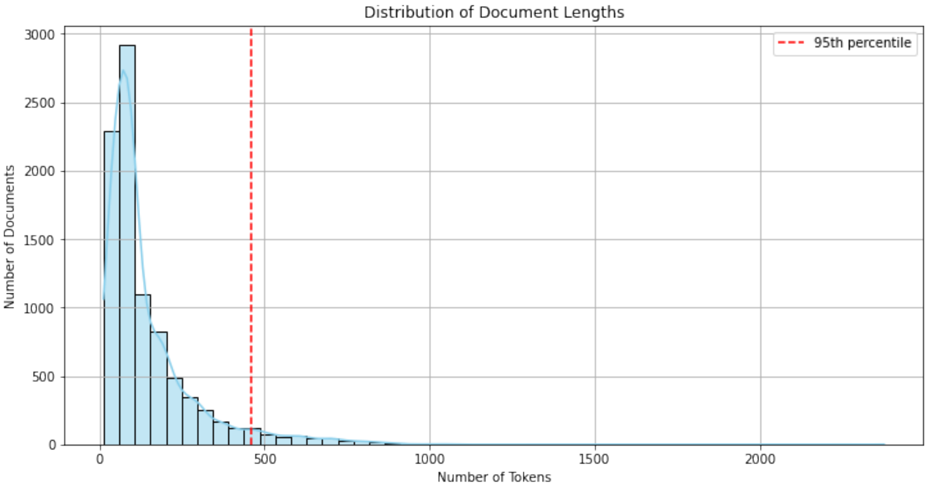


Figure 3: Histogram of document lengths in terms of token counts. The red dashed line marks the 90th percentile.

Analysis shows that 95% of all documents are shorter than 459 words. Therefore, we fixed the maximum input sequence length $L = 459$, applying zero-padding to shorter sequences and truncation to longer ones. This decision strikes a balance between preserving most of the content and controlling computational overhead. Furthermore, keeping the sequence length moderate reduces vanishing gradients and training time in LSTM-based architectures.

2.3 Vocabulary Coverage Analysis

Vocabulary size is a key hyperparameter that affects both performance and memory consumption in NLP models. A small vocabulary may lose important nuances, while a large vocabulary increases parameter space and overfitting risk.

To analyze the coverage–size trade-off, we plotted the cumulative token coverage against vocabulary size in Figure 4. The result reveals a power-law distribution: the most frequent 2,000 words cover over 90% of the tokens in the corpus, and 4,229 words cover 95%. This observation suggests that a small subset of high-frequency words dominates the dataset — a characteristic typical of natural language corpora.

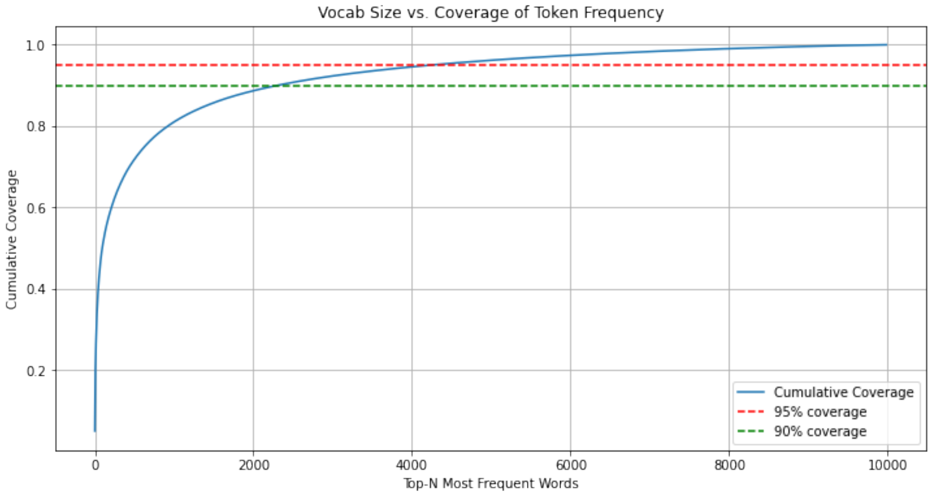


Figure 4: Cumulative token coverage as a function of vocabulary size. The dashed lines indicate 90% and 95% coverage thresholds.

Rather than experimenting with arbitrary vocabulary sizes, we defined our vocabularies based on coverage thresholds. Table 1 presents the mapping from coverage percentage to vocabulary size. This principled approach allows us to systematically investigate how much linguistic information is needed for effective classification.

Coverage (%)	Vocab Size (top-k words)
10%	2
20%	7
30%	20
40%	43
50%	93
60%	209
70%	436
80%	920
90%	2,298
95%	4,229
100%	9,981

Table 1: Vocabulary size required to reach various token coverage levels.

This coverage-driven vocabulary selection allows the model to operate under low-resource settings while maintaining semantic fidelity. It also enables controlled experiments to observe how vocabulary constraints affect generalization and convergence.

3 Model Architecture and Experimental Setup

3.1 4.1 Model Architecture

All models share a common neural architecture pattern designed for text classification tasks using recurrent layers. We implemented three variants: LSTM, BiLSTM, and GRU, all following a two-layer structure. The goal was not to compare model types themselves, but to evaluate how they respond to variations in preprocessing, specifically vocabulary size and class imbalance handling.

Each model begins with an **embedding layer** that projects input token indices into a dense vector space. This is followed by two recurrent layers (either LSTM, BiLSTM, or GRU) with 64 and 32 hidden units respectively. Dropout layers with a rate of 0.1 are applied after each recurrent block to prevent overfitting. A fully connected dense layer with ReLU activation (64 units) is added before the final output layer, which uses a softmax activation to produce probability distributions over 46 classes.

3.2 4.2 Experimental Settings

The experiments were designed to analyze how changes in **vocabulary size** and **class imbalance processing strategy** affect model performance. Specifically, we compared the following conditions:

- Vocabulary sizes were adjusted according to token coverage thresholds, as summarized in Table 1.
- For class imbalance, we evaluated both **weighted loss only** and **weighted loss + oversampling**.

Importantly, the model architecture and training hyperparameters remained fixed throughout to ensure fair comparison.

Fixed Hyperparameters

- Maximum sequence length: $L = 459$
- Number of classes: $C = 46$
- Number of epochs: 30
- Batch size: 64

3.3 4.3 Evaluation Metrics

Model performance was evaluated using two standard metrics:

- **Accuracy** — the overall percentage of correct predictions.
- **Macro-averaged F1-score** — to account for class imbalance by averaging precision and recall across all classes equally.

These metrics were chosen to assess both general performance and robustness on rare categories.

4 Results and Analysis

This study compared classification performance under three preprocessing settings (None, Augmentation, and Weighting) applied to three model architectures (LSTM, BiLSTM, GRU). The evaluation metrics used were **Accuracy** and **F1 Score**, and the results are visualized in Figures 5 and 6. In addition, we analyzed the stability of each model's performance with respect to varying vocabulary sizes.

4.1 Accuracy Analysis

Without preprocessing (None), the BiLSTM model consistently recorded the highest accuracy. Notably, BiLSTM maintained an accuracy above 0.74 when the vocabulary size exceeded 436, indicating robustness to increasing vocabulary size. This corresponds to approximately 43.6% of the total vocabulary. Even at larger vocab sizes, the performance did not degrade. In contrast, the GRU model exhibited relatively low accuracy, and when only class weighting was applied, performance decreased sharply in certain vocab size ranges.

Data augmentation significantly improved the accuracy of BiLSTM and GRU models, with BiLSTM benefiting the most. However, LSTM showed minimal improvement from augmentation and generally had lower accuracy compared to the other models.

4.2 F1 Score Analysis

In terms of F1 Score, BiLSTM again achieved the best performance, particularly when data augmentation was applied. This suggests that BiLSTM is effective in classifying minority classes even under data imbalance. On the other hand, LSTM and GRU models showed relatively low F1 Scores, and applying only class weights did not result in significant improvement.

In summary, BiLSTM was the most stable model and responded best to preprocessing techniques. Among these, data augmentation consistently improved both Accuracy and F1 Score, making it a valuable strategy for imbalanced datasets.

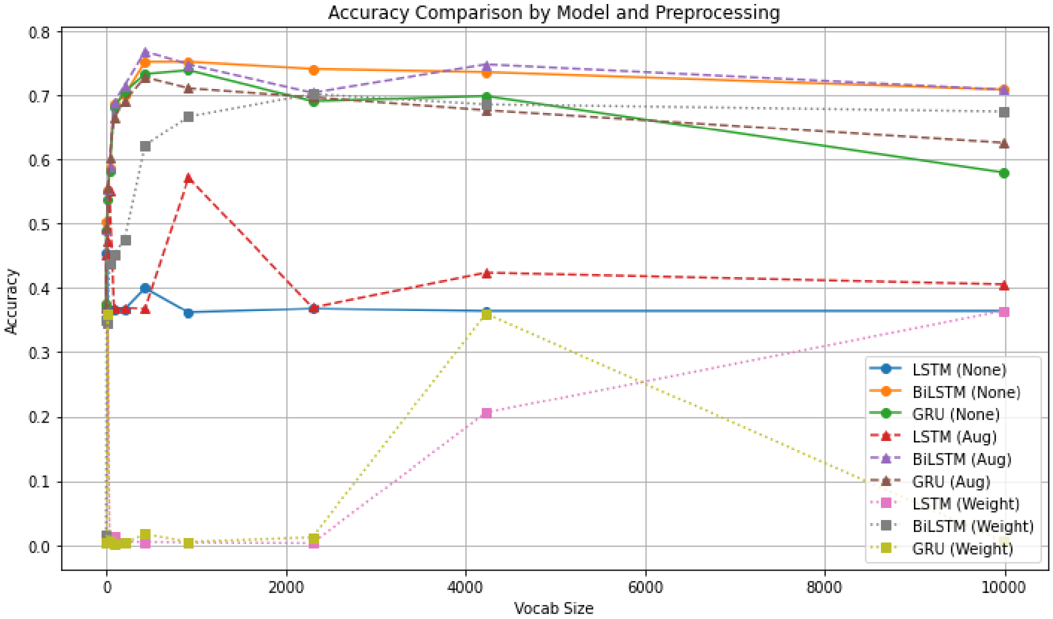


Figure 5: Accuracy comparison by model and preprocessing method

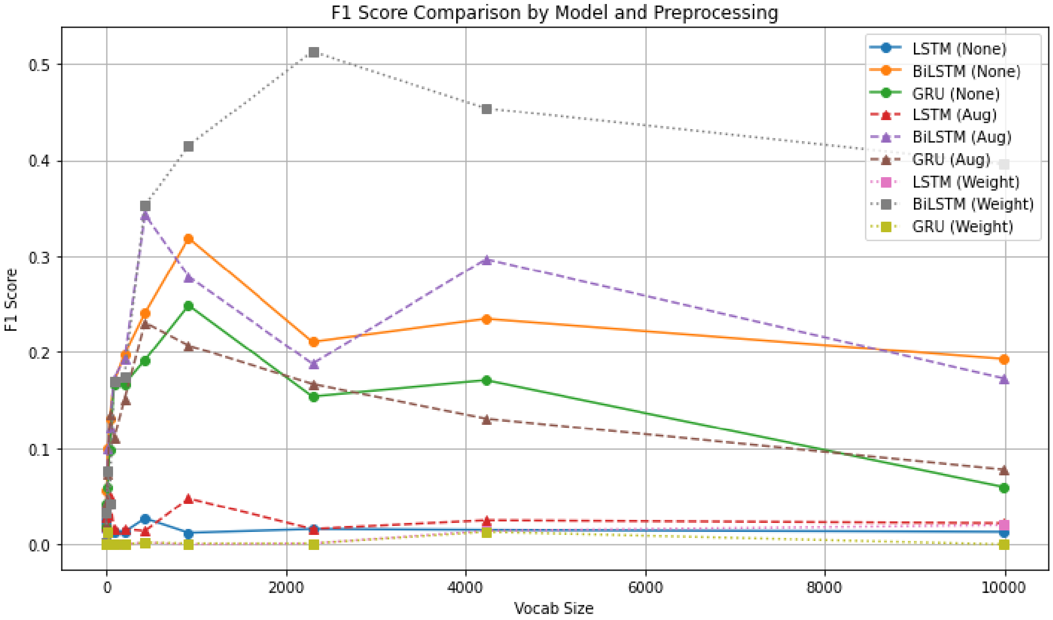


Figure 6: F1 Score comparison by model and preprocessing method

4.3 Model Stability Analysis

To analyze stability, we define the following F1-based stability index:

$$\text{Stability}_{F1} = \frac{\mu_{F1}}{\sigma_{F1}} \tag{1}$$

Here, μ_{F1} and σ_{F1} represent the average and standard deviation of the F1 scores across different vocabulary sizes. A higher value indicates more stable performance.

Among the models, BiLSTM yielded the highest Stability_{F1} value, suggesting relatively consistent performance across vocabularies. In particular, stability appeared empirically robust when the vocabulary size exceeded 436 (approximately 43.6% of total token coverage), continuing up to 9981 (95%). While this trend highlights a potential threshold for effective generalization, it should be interpreted as a dataset-specific observation rather than a universal rule.

Table 2: Comparison of BiLSTM Stability Scores

Range	Stability Score (μ/σ)	Relative Improvement (%)
Entire Vocab Range (2–9981 words)	1.37	–
Stable Range (436–9981 words)	1.89	+38%

Due to extremely low F1 Scores in most cases, LSTM and GRU models were not suitable for robust stability analysis. The stability-based evaluation further supports BiLSTM as the most dependable model across vocabulary settings.

All model-wise metrics, trend graphs, and statistical evaluation formulas are provided in the Appendix.

5 Conclusion

Summary of Findings

This study investigated how different preprocessing techniques and vocabulary sizes affect text classification performance across LSTM, BiLSTM, and GRU models, using the Reuters-21578 dataset. Three preprocessing strategies—no processing, data augmentation, and class weighting—were evaluated across varying vocabulary sizes (2 to 9981).

Among the models, BiLSTM consistently outperformed others in both Accuracy and F1 Score, especially when data augmentation was applied. A vocabulary size of 436 (approximately 43.6%) or higher led to performance stabilization, and a stability index defined as μ_{F1}/σ_{F1} confirmed that BiLSTM was the most robust model against vocabulary fluctuations. These results highlight the importance of preprocessing and vocabulary selection in improving model generalization on imbalanced datasets.

Limitations

Despite promising results, this study has several critical limitations:

- **Dataset-specific imbalance structure:** The observed gains from augmentation and class weighting may heavily depend on the label distribution of the Reuters dataset, limiting generalizability to other domains.
- **Model architecture scope:** Only classic RNN variants (LSTM, GRU, BiLSTM) were explored. No comparison was made with transformer-based architectures such as BERT or RoBERTa, which are state-of-the-art in NLP.
- **Vocabulary-performance causality:** The correlation between increased Vocab Size and performance is not necessarily causal. Improvements may result from broader lexical coverage rather than genuine semantic learning.
- **Limited evaluation metrics:** Only Accuracy and F1 Score were used. More imbalance-sensitive metrics like Macro-F1, AUC, or Precision@K were not considered.
- **Reproducibility:** If augmentation relied on randomized processes or fixed seeds were not used, reproducibility of results may be compromised. The augmented dataset is also not publicly released.

Future Work

To extend this research, we propose the following directions:

- Benchmark the approach across multiple datasets and domains, including multilingual corpora.
- Incorporate transformer-based models and assess their robustness under similar preprocessing and Vocab Size constraints.
- Explore advanced augmentation techniques such as synonym substitution, back-translation, and adversarial example generation.
- Analyze vocabulary pruning and subword tokenization as alternative strategies for reducing model complexity without sacrificing performance.
- Evaluate with broader metrics tailored to class imbalance and practical deployment (e.g., Macro-F1, Recall@K, model size vs. accuracy trade-offs).

These directions will enhance the generality, interpretability, and deployment potential of text classifiers in real-world imbalanced settings.

References

[1] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[2] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

[3] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734. Association for Computational Linguistics, 2014.

[4] David Hayes and Stephen Weinstein. Reuters-21578 text categorization test collection. `https://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html`, 1990. Accessed: 2025-06-04.

[5] T. Hossain, H. Z. Mauni, and R. Rab. Reducing the effect of imbalance in text classification. *Computing and Informatics*, 41(1):98–113, 2022.

[6] Beytullah Yildiz. Efficient text classification with deep learning on imbalanced data improved with better distribution. *DergiPark*, 2022.

Appendix

A.1 Data Augmentation

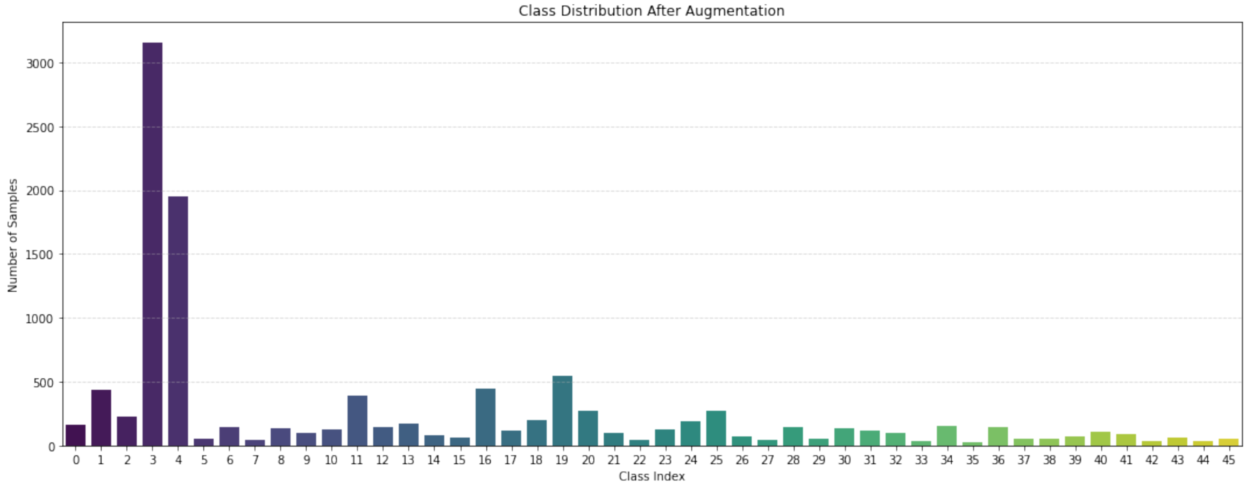
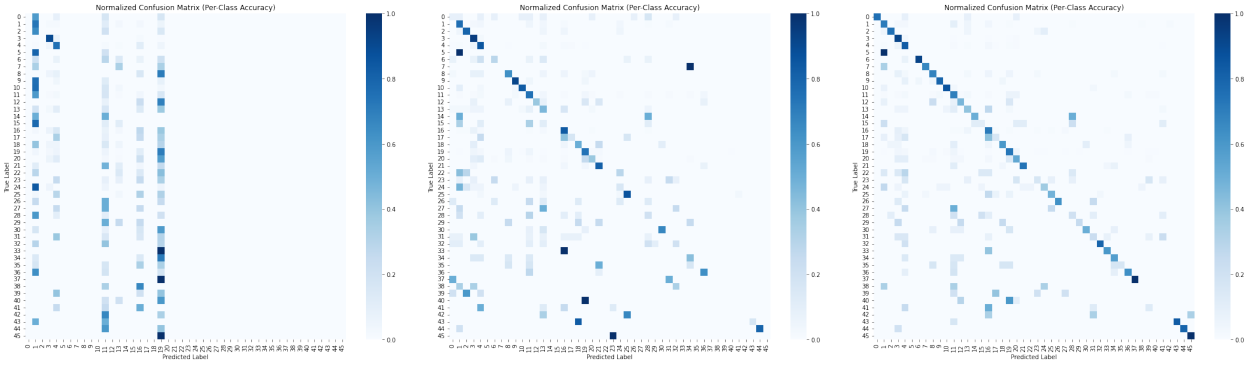


Figure 7: Class Distribution After Augmentation. Although augmentation increases the size of rare classes, significant imbalance remains.

A.2 Confusion Matrices



(a) No Preprocessing

(b) Augmentation

(c) Class Weighting

Figure 8: Normalized Confusion Matrices for Each Preprocessing Strategy

A.3 Full Evaluation Tables

Table 3: Accuracy by Model and Preprocessing Method

Vocab_Size	LSTM (None)	BiLSTM (None)	GRU (None)	LSTM (Aug)	BiLSTM (Aug)	GRU (Aug)	LSTM (Weight)	BiLSTM (Weight)	GRU (Weight)
2	0.371	0.375	0.376	0.369	0.370	0.371	0.009	0.015	0.004
7	0.455	0.502	0.490	0.452	0.491	0.495	0.003	0.349	0.004
20	0.367	0.552	0.538	0.473	0.554	0.556	0.360	0.345	0.360
43	0.367	0.586	0.582	0.551	0.589	0.603	0.005	0.439	0.007
93	0.367	0.687	0.681	0.369	0.689	0.665	0.013	0.453	0.004
209	0.367	0.700	0.705	0.369	0.715	0.691	0.005	0.475	0.003
436	0.400	0.752	0.733	0.368	0.768	0.728	0.005	0.622	0.017
920	0.362	0.752	0.739	0.572	0.748	0.711	0.004	0.667	0.005
2298	0.368	0.741	0.691	0.370	0.704	0.697	0.003	0.702	0.012
4229	0.364	0.736	0.699	0.424	0.748	0.677	0.206	0.686	0.360
9981	0.365	0.709	0.580	0.406	0.709	0.626	0.364	0.675	0.006

Table 4: F1 Score by Model and Preprocessing Method

Vocab_Size	LSTM (None)	BiLSTM (None)	GRU (None)	LSTM (Aug)	BiLSTM (Aug)	GRU (Aug)	LSTM (Weight)	BiLSTM (Weight)	GRU (Weight)
2	0.021	0.018	0.017	0.019	0.018	0.019	0.001	0.002	0.001
7	0.026	0.056	0.043	0.028	0.042	0.042	0.000	0.033	0.001
20	0.013	0.100	0.060	0.032	0.100	0.074	0.013	0.076	0.013
43	0.013	0.131	0.099	0.050	0.121	0.135	0.000	0.043	0.000
93	0.013	0.169	0.166	0.016	0.173	0.110	0.001	0.454	0.000
209	0.013	0.198	0.167	0.016	0.193	0.151	0.000	0.175	0.000
436	0.027	0.241	0.192	0.014	0.343	0.231	0.002	0.353	0.002
920	0.012	0.319	0.249	0.048	0.279	0.207	0.000	0.415	0.001
2298	0.016	0.211	0.154	0.016	0.188	0.167	0.001	0.513	0.001
4229	0.015	0.235	0.171	0.025	0.297	0.131	0.014	0.454	0.013
9981	0.013	0.193	0.060	0.022	0.173	0.078	0.020	0.396	0.000