Enhancing Multi-Label Text Classification using CNN-BiLSTM and CNN-BiGRU through GloVe Embeddings

Piyush Yadav, Aditay Pote, Prakhar Kumar Sinha
2020IMT-068, 2020IMT-069, 2020IMT-070
ABV-IITM Gwalior, M.P., India

***Abstract*—This study addresses the challenge of multi-label text classification within the Natural Language Processing (NLP) domain, spotlighting the inherent difficulties in parsing and categorizing the huge volume of text data typical of digital platforms. Traditional machine learning techniques often miss the semantic interconnections among words, whereas deep learning solutions suffer from significant computational and training overheads. To navigate these obstacles, we used 2 deep learning models that utilises the Global Vectors for Word Representation (GloVe). We first verified using GloVe embeddings with a CNN-BiLSTM model and got a significant increase in validation accuracy as compared to the original CNN-BiLSTM model. We also show that our CNN-BiGRU model performs better with GloVe embeddings than when used without GloVe embeddings. Central to our experimental validation is the application of this model to the well-established 20 Newsgroup dataset, a choice that underscores the model's versatility and efficiency across a broad spectrum of topics and text variations. Our findings illuminate a path forward for refining text classification methodologies, emphasizing the model's potential applicability in real-world scenarios and laying groundwork for future explorations in NLP.**

***Index Terms*—Multi-label Text Classification, Natural Language Processing, GloVe Embeddings, CNN-BiLSTM, Machine Learning, Deep Learning**

I. Introduction

Text classification techniques are primarily utilized to identify which category news articles, online content, and stories fall under based on pre-established classifications. The goal of this process is to determine if a new piece of text aligns with a specific category. The ongoing advancements in information technology have led to a significant increase in data volume. One effective strategy to manage and make sense of this vast amount of data is by organizing it into distinct categories. The surge in data not only makes classification more complex but also demands considerable time and effort for manual sorting. These challenges highlight the necessity and value of automating the process of classifying electronic textual data. Deep learning approaches are becoming popular due to their capability to identify intricate features automatically, reducing the need for manually crafted features and lessening the reliance on specialized domain knowledge. Efforts are now focused on creating neural network designs that can efficiently process and understand textual information [3] [4].

Convolutional neural networks (CNNs) use layers with filters that are applied to local features of the input data. Initially developed for computer vision tasks, CNNs have since been shown to be effective for natural language processing (NLP) as well. They have achieved impressive results in various NLP tasks, such as semantic parsing, search query retrieval, sentence modeling, and other traditional NLP problems, without the need for explicit feature engineering.

Long Short-Term Memory (LSTM) is an improved variant of Recurrent Neural Networks (RNNs) that employs a gating mechanism to address the issue of long-term information preservation and the vanishing gradient problem encountered by traditional RNNs. The gating mechanism, which consists of an input gate, forget gate, and output gate, helps the LSTM model determine whether data from the previous state should be retained or forgotten in the current state. This unique capability of LSTM makes it well-suited for extracting advanced text information and has led to its widespread application in text classification tasks.

In recent years, the scope of LSTM applications has expanded rapidly, with researchers proposing various ways to further improve its accuracy. One such enhancement is the Bidirectional LSTM (Bi-LSTM) neural network, which is composed of LSTM units that operate in both forward and backward directions. This bidirectional approach allows the Bi-LSTM to incorporate both past and future context information, enabling it to learn long-term dependencies without retaining duplicate context information. As a result, Bi-LSTM has demonstrated excellent performance in sequential modeling problems and is widely used for text classification tasks.

Unlike the standard LSTM network, which has a single layer that propagates in a single direction, the Bi-LSTM network has two parallel layers that propagate in forward and backward directions, respectively. This dual-directional propagation allows the Bi-LSTM to capture dependencies in two different contexts, further enhancing its effectiveness in sequential modeling applications.

Gated Recurrent Unit (GRU) and Bidirectional GRU (Bi-GRU) have emerged as formidable alternatives to LSTM and

979-8-3503-6052-3/24/$31.00 ©2024 IEEE

Bidirectional LSTM (Bi-LSTM) in the realm of sequential data processing. While LSTM and Bi-LSTM have been pivotal in addressing the shortcomings of traditional recurrent neural networks, GRU and Bi-GRU offer streamlined architectures and enhanced computational efficiency without compromising performance.

GRU's simplified design, achieved by amalgamating the forget and input gates into a single update gate and introducing a reset gate, streamlines the flow of information within the network. This consolidation of gates reduces the computational burden compared to LSTM, making GRU particularly appealing for applications where computational resources are limited or efficiency is paramount.

Moreover, Bi-GRU takes the advancements of GRU a step further by incorporating bidirectional processing. By leveraging information from both past and future contexts simultaneously through two parallel layers of GRU units, Bi-GRU excels in capturing long-range dependencies and nuanced patterns in sequential data. This bidirectional approach not only enhances the model's understanding of context but also enables it to anticipate future inputs, leading to superior performance in tasks such as speech recognition, machine translation, and sentiment analysis.

In recent years, GRU and Bi-GRU have gained prominence in various fields, ranging from natural language processing and time-series analysis to speech recognition and protein structure prediction. Their versatility, efficiency, and effectiveness in handling sequential data have made them indispensable tools for researchers and practitioners alike, paving the way for advancements in AI-driven applications and systems. As the demand for sophisticated sequential modeling continues to grow, GRU and Bi-GRU are poised to play pivotal roles in shaping the future of artificial intelligence and deep learning.

Word embeddings are a favored method for converting text into numerical data for deep machine learning applications. This process involves breaking down text into individual words or "tokens" and assigning each a fixed-length vector of real numbers, known as an "embedding." Unlike other methods such as one-hot encoding, word frequency counts, or TF-IDF, which assign a single number to each token, word embeddings provide a richer representation. This is because a single number can only convey limited information, whereas vectors can capture a broader range of meanings. This capability allows word embeddings to better represent the varied meanings a word can have in different contexts and its relationships with other words. GloVe (Global Vectors for Word Representation) is a type of unsupervised learning algorithm developed to create word embeddings by analyzing large text corpora. The resulting embeddings capture deep linguistic meanings and relationships [2]. By using pre-trained GloVe embeddings, machine learning projects with smaller datasets can bypass the time-consuming and resource-intensive process of training embeddings from scratch, leveraging the rich semantic information GloVe provides.

Our research focuses on the improvement provided by the pre-trained GloVe embeddings in multi-class text classification using neural networks. We employed two models: a convolution neural network (CNN) with a Bidirectional Long Short Term Memory (BiLSTM), inspired by the paper [1] [5], and our own model- CNN with a Bidirectional Gated Recurrent Unit (BiGRU). The fusion of these technologies allows for a robust text representation that captures deep semantic relationships efficiently, significantly enhancing classification accuracy without necessitating the extensive training typically associated with deep neural models .

II. Literature Review

Traditionally, rule-based models and SVMs were used for tasks such as question classification in query systems [6]. The advent of deep learning approaches has revolutionized text classification. RNNs have gained popularity due to their ability to capture sequential dependencies [7]. Variants of RNNs, such as LSTMs, have been particularly successful in tasks like sentiment analysis and language translation [4].

Attention mechanisms have emerged as a powerful tool for improving the performance of RNN-based models by enabling them to focus on relevant parts of the input sequence [8]. This attention mechanism has been successfully incorporated into various architectures, including transformers, which have achieved state-of-the-art results in a wide range of NLP tasks [9].

In addition to RNNs, CNNs, initially designed for image processing, have been adapted for text classification tasks [10]. CNNs operate by applying convolutional filters over the input text to capture local patterns, making them effective for tasks like sentiment analysis and document categorization.

The use of pre-trained word embeddings, such as Word2Vec [11], has further enhanced the performance of deep learning models. Word embeddings provide dense, distributed representations of words, enabling models to capture semantic relationships between words.

Hybrid models that combine the strengths of different architectures, such as RNNs and CNNs, have shown promise in achieving superior performance in text classification tasks [12]. By leveraging both sequential and local information, these hybrid models can effectively capture both short and long-range dependencies in text data.

Despite the advancements in deep learning for text classification, challenges remain. The increasing complexity of neural network architectures can lead to difficulties in training, such as high time complexity and overfitting [5].

This paper focuses on a text classification model that combines CNNs and BiLSTMs and a new model that combines CNNs and BiGRUs. It also incorporates the GloVe word embedding model in both architectures to enhance the original text features, leading to promising classification performance, as demonstrated in the experiments.

III. Methodology

A. Dataset

To ensure fair evaluation and comparison between the CNN-BiLSTM and CNN-BiGRU models, we conducted preprocess-

TABLE I
SUMMARY OF LITERATURE REVIEW

Paper Author	Techniques Used	Achievements	Limitations
[6] J. Silva, L. Coheur	Rule-based models, SVMs	Effective in structured query classification.	Lack adaptability
[7] Z. Li, J. Han	RNNs	Improved accuracy in text classification	High computational cost
[8] X. Zheng, L. Ding	RNNs with Attention mechanisms	Enhanced performance in capturing context	Complexity in implementation
[9] Z. Liu, X. Wang, H	Word2Vec with LSTM networks	High accuracy in query classification	Dependency on large datasets
[10] Y. Kim	CNNs	Adaptation for text classification	Difficulty in capturing sequential information
[11] C. Hao, H. Qiu	CNNs with Word2Vec	Promising results in multi-label classification	Sensitivity to hyperparameters
[12] W. Tu, K. Yuan	Hybrid models combining RNNs and CNNs	Successful in single-label classification	Increased model complexity

ing steps to standardize the input data. Given that nearly 97 percent of both the training and testing documents in the 20 newsgroups dataset contained less than 1000 words, we truncated all documents to the first 1000 words. This truncation strategy allowed us to maintain a balance between computational efficiency and retaining relevant information for classification tasks.

Furthermore, assessing the class distribution in the training dataset is crucial for understanding potential biases and ensuring that the model's predictions are not skewed towards certain categories. We performed a thorough analysis of the class distribution and found that the dataset exhibited balanced proportions across the 20 predefined category labels. This balance was verified through visual inspection, as illustrated in Figure 1, where each category label exhibited comparable frequencies within the dataset.

B. Pre-processing

To enhance the dataset's cleanliness and eliminate extraneous textual elements, we employed several preprocessing techniques. Initially, we converted all tweets to lowercase and removed leading and trailing whitespaces. Subsequently, we replaced all spaces with newline characters. Punctuation marks were then stripped from each word, followed by the removal of all special characters. Stop words were also removed, and each token underwent lemmatization to normalize based on contextual usage.

A tokenizer was utilized to process the text data, assigning a unique identifier to each distinct token (word) in the dataset. This process resulted in a total vocabulary size of 148,442 unique tokens. The training and test data were converted into sequences of token IDs. These sequences, varying in length, were then either padded or truncated to a fixed length of 300, ensuring a consistent input matrix for the subsequent deep learning model. [2], [4]

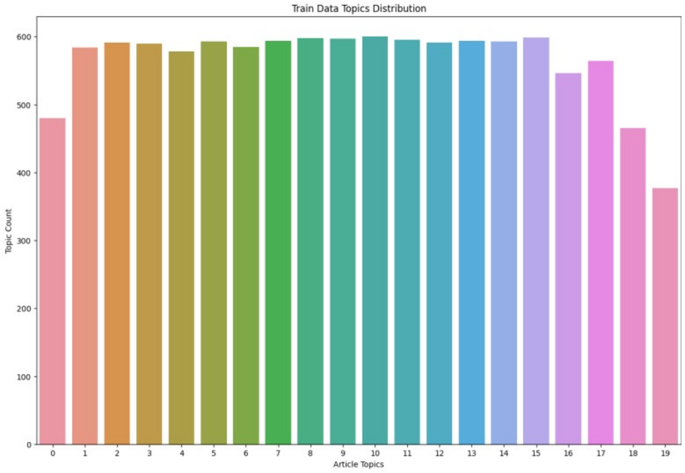

Fig. 1. Topic Class Distribution of Training Dataset

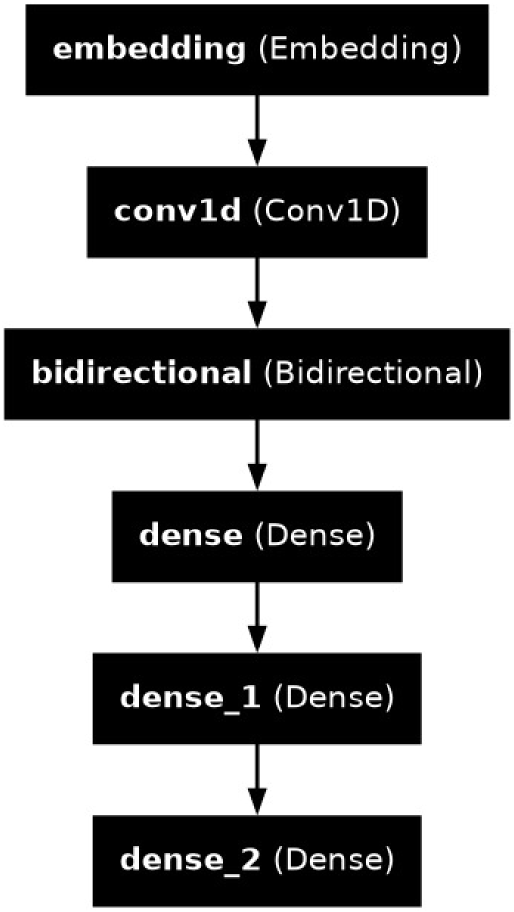

Fig. 2. Architecture of the Base CNN-Bidirectional Model.

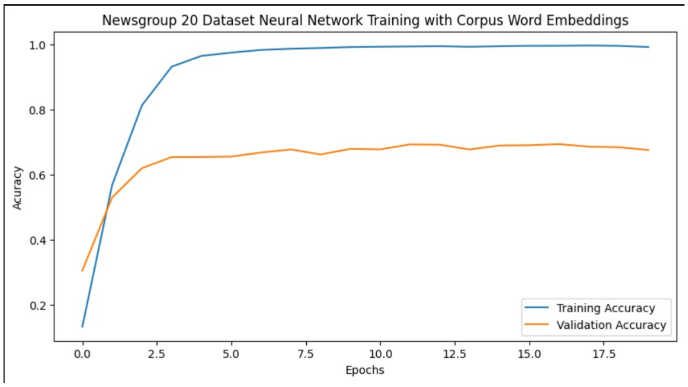

Fig. 3. Training and Validation Accuracy during training of CNN-BiLSTM model.

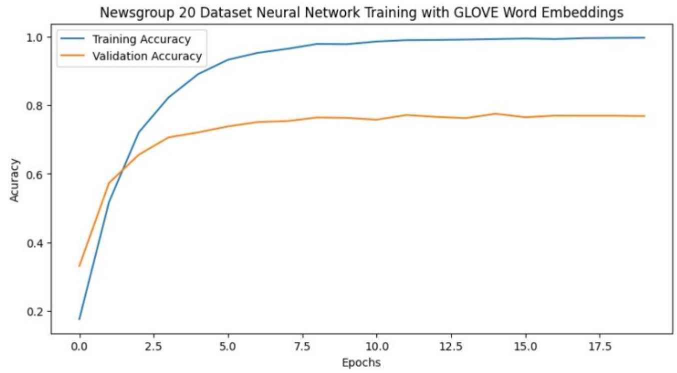

Fig. 4. Training and Validation Accuracy during training of CNN-BiLSTM model with GloVe embeddings.

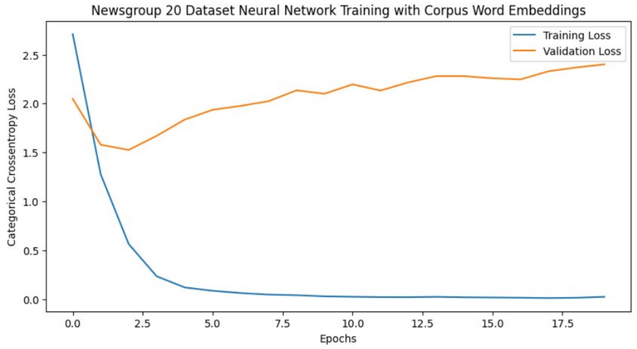

Fig. 5. Training and Validation Loss during training of CNN-BiLSTM model without GloVe embeddings.

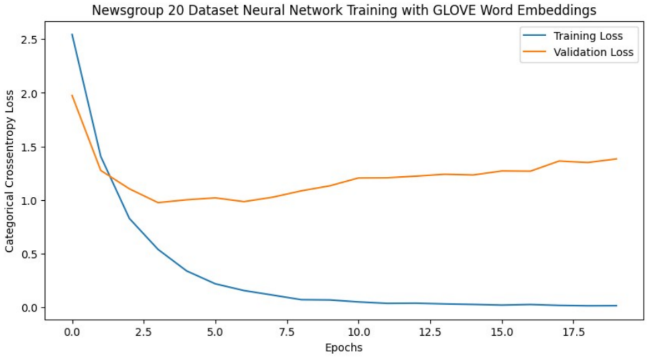

Fig. 6. Training and Validation Loss during training of CNN-BiLSTM model with GloVe embeddings.

C. Model Architecture

1) CNN-BiLSTM Without Glove Embeddings: Our first model employs a CNN-BiLSTM architecture, as illustrated in Figure 2. This architecture integrates a Bidirectional LSTM layer, facilitating bidirectional processing of the input sequence.

We trained custom word embeddings using Embedding layers on the news training data. This was followed by a 1D Convolutional layer to extract local features from the input sequence. Subsequently, a Bidirectional LSTM layer was employed to capture contextual dependencies by processing the sequence both forwards and backwards. The LSTM's output was then passed through two fully connected Dense layers, enabling the learning of higher-level representations. Finally, the model concluded with a Dense output layer utilizing a Softmax activation, suitable for multi-class classification tasks with 20 possible output classes.

The total number of trainable parameters amounted to 15,683,032, which were also the total parameters utilized during training.

2) CNN-BiLSTM With Pre-Trained Glove Embeddings: We first created an embedding matrix using the pre-trained glove embeddings. The embedding matrix was then used as pre-trained weights in the same CNN with bidirectional model architecture (Figure 2.).

The total number of parameters and trainable parameters were equal and same as the first model (CNN-BiLSTM withoul glove embeddings).

3) CNN-BiGRU Without Glove Embeddings: Our third model is a CNN-BiGRU.The Bidirectional LSTM layer was replaced by Bidirectional Gated Recurrent Unit layer in the Bidirectional layer in Figure 2.

Similar to the first model, We trained our own word embeddings.

The number of trainable parameters are 15,552,632. The total number of parameters used during training are same. it can be observed that the number of parameters were less than CNN-BiLSTM model.

IV. Experiments and Results

1) CNN-BiGRU With Pre-Trained Glove Embeddings: Our final model uses the same base architecture as CNN-BiGRU, in addition to using pre-trained GloVe embedding matrix as pre-trained weights for the model.

Both, the total number of parameters and the number of trainable parameters came out to be equal to 15,552,632.

A. Experimental Setup and Training

We ran our experiments in a Python 3 environment on system supported by 2 T4 GPUs. A learning rate of 0.0002,

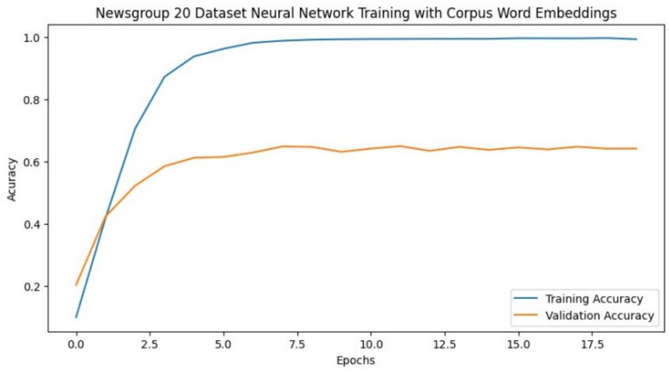

Fig. 7. Training and Validation Accuracy during training of CNN-BiGRU model.

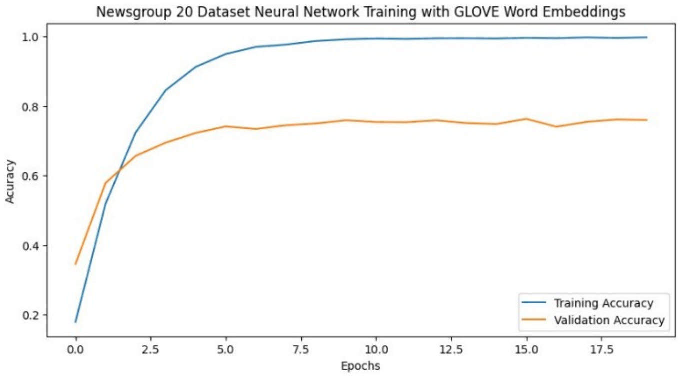

Fig. 8. Training and Validation Accuracy during training of CNN-BiGRU model with GloVe emebddings.

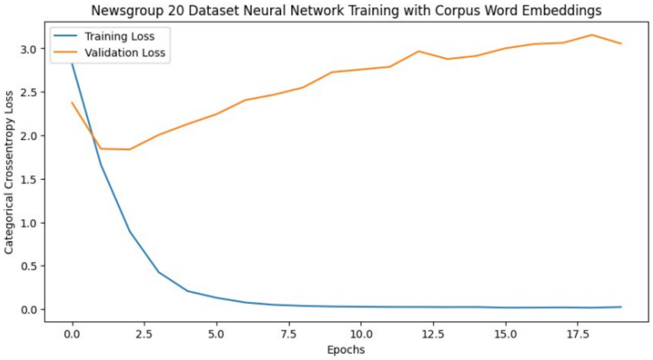

Fig. 9. Training and Validation Loss during training of CNN-BiGRU model without GloVe embeddings.

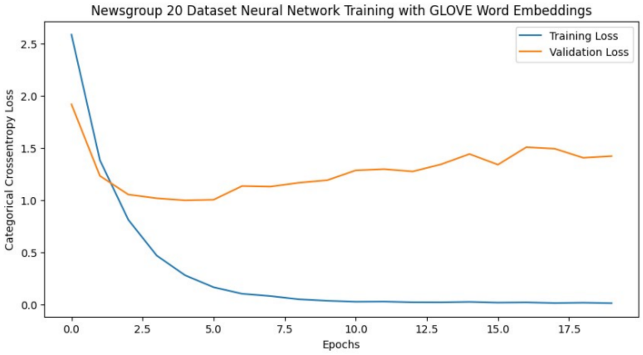

Fig. 10. Training and Validation Loss during training of CNN-BiGRU model with GloVe embeddings.

a batch size of 128, and 20 training epochs were our hyper-parameters.

B. Model Training and Comparative Analysis

First, we observed that the final training accuracy after 20 epochs of the CNN-BiLSTM model without GloVe embeddings was approximately 99.2 percent and the validation accuracy was 67.5 percent (refer Figure 3.). we observeed a training loss of 2.4 percent and a validation loss of 240 percent as seen in Figure 9. We saw a substantial improvement in our pre-trained GloVe embeddings CNN-BiLSTM model where the training accuracy was 99.6 percent while the validation accuracy was 76.8 percent (refer Figure 4.). An increase of 9.3 percent in validation accuracy was observed after using pre-trained GloVe embeddings in the original model. The training loss and validation loss came out to be 1.2 percent and 138 percent, respectively (Figure 6). A decrement of 102 percent observed in improved validation loss.

The final training and validation accuracy came out to be 99.5 percent and 64.7 percent, respectively, in our CNN-BiGRU model without GloVe embeddings (refer Figure 7.). Training accuracy for CNN-BiGRU model with GloVe embeddings came out to be 99.7 percent and its validation accuracy was approximately 76 percent. An increase of 11.3 percent in validation accuracy was observed in CNN-BiGRU model after using GloVe embeddings (refer Figure 8.). The training and validation loss were 2 and 306 percent respectively in CNN-BiGRU without GloVe embeddings model (Figure 9) whereas a training loss of 1 percent and and a validation loss of 142 percent, 164 percent less than the original model, was observed in CNN-BiGRU with GloVe implemented model as observed in Figure 10.

Conclusion of Experimental Analysis: These results underline the potential of integrating GloVe embeddings with CNN-BiLSTM and CNN-BiGRU architectures for advancing multi-class text classification.

V. Conclusion and Future Work

Our study clearly demonstrates the benefits of using GloVe embeddings in neural network models for text classification. By incorporating GloVe, we saw a consistent improvement in accuracy across all models tested. This shows how important it is to use embeddings that understand deep word relationships for better text classification.

Overfitting in the results could be curbed by introducing methods like early-stopping and dropout layers. Training the models for larger epochs and using a GloVe embedding with more tokens or dimensions could further improve upon the obtained results.

Looking ahead, there are several ways we can build on this research. Exploring other pre-trained embeddings like

FastText or BERT might give us even better results, as each has its unique way of understanding words. Tackling overfitting with advanced techniques will also be key to making our models not just accurate but also reliable on unseen data. Additionally, trying out new optimization methods could help our models learn faster and more effectively. Finally, adapting our approach to specific fields or languages could show how versatile and useful it can be across different areas of natural language processing.

References

[1] Hongren Wang, ”Multi-label Text Classification using GloVe and Neural Network Models,” arXiv preprint arXiv:2312.03707, 2023.

[2] Pennington, J., Socher, R., and Manning, C. D., ”Glove: Global vectors for word representation,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[3] H. H. Mohammed, E. Dogdu, A. K. Görür, and R. Choupani, “Multi-Label Classification of Text Documents Using Deep Learning,” *2020 IEEE International Conference on Big Data (Big Data)*, pp. 4681-4689, 2020.

[4] S. Wang, Y. Yang, and X. Meng, “Research on Multi-Label Text Classification Based on Multi-Channel CNN and BiLSTM,” *2022 International Conference on Artificial Intelligence of Things and Crowdsensing (AIoTCs)*, pp. 498-503, 2022.

[5] S. Nazar and R. Rajan, “Multi-label Comment Classification Using GloVe-RNN Framework,” *2022 IEEE 19th India Council International Conference (INDICON)*, pp. 1-4, 2022.

[6] J. Silva, L. Coheur, A. C. Mendes, and A. Wichert, “From symbolic to sub-symbolic information in question classification,” *Artificial Intelligence Review*, vol. 35, no. 2, pp. 137-154, 2011. doi: 10.1007/s10462-010-9188-4.

[7] Z. Li, J. Han, W. E, and Q. Li, “Approximation and optimization theory for linear continuous-time recurrent neural networks,” *The Journal of Machine Learning Research*, vol. 23, no. 1, pp. 1997-2081, 2022.

[8] X. Zheng, L. Ding, and R. Wan, “User and product Attention mechanism based hierarchical BiGRU model,” *Computer Engineering and Applications*, vol. 54, no. 11, pp. 145-152, 2018. doi: 10.3778/j.issn.1002-8331.1701-0337.

[9] Z. Liu, X. Wang, H. Wu, H. Wang, and T. Xu, “Research on rice question and sentence similarity matching method based on BiLSTM-CNN,” *Journal of Chinese Agricultural Mechanization*, vol. 43, no. 12, pp. 125-132, 2022. doi: 10.13733/j.jcam.issn.2095-5553.2022.12.019.

[10] Y. Kim, “Convolutional neural networks for sentence classification,” *arXiv preprint arXiv:1408.5882*, 2014.

[11] C. Hao, H. Qiu, Y. Sun, and C. Zhang, “Research Progress of Multi-label Text Classification,” *Computer Engineering and Applications*, vol. 57, no. 10, pp. 48-56, 2021. doi: 10.3778/j.issn.1002-8331.2101-0096.

[12] W. Tu, K. Yuan, and K. Yu, “Neural Network Models for Text Classification,” *Computer Systems and Applications*, vol. 28, no. 7, pp. 145-150, 2019.