INTERNATIONAL JOURNAL OF INTELLIGENT DATA AND MACHINE LEARNING (IJIDML)

eISSN: 3087-4262

Volume. 02, Issue. 04, pp. 06-12, April 2025

HYBRID DEEP LEARNING FOR TEXT CLASSIFICATION: INTEGRATING BIDIRECTIONAL GATED RECURRENT UNITS WITH CONVOLUTIONAL NEURAL NETWORKS

Yuki Nakamura
Graduate School Of Information Science And Technology, University Of Tokyo, Japan

Isabella Romano
Department Of Computer Engineering, Politecnico Di Milano, Italy

Article received: 19/02/2025, Article Accepted: 27/03/2025, Article Published: 17/04/2025
DOI: https://doi.org/10.55640/ijidml-v02i04-02

ABSTRACT

Text classification remains a foundational task in natural language processing with wide-ranging applications, including sentiment analysis, topic categorization, spam detection, and information retrieval. While convolutional neural networks (CNNs) are adept at capturing local n-gram features, and recurrent neural networks (RNNs) excel at modeling sequential dependencies, standalone architectures often struggle to fully leverage both aspects simultaneously. This study presents a hybrid deep learning model that integrates bidirectional gated recurrent units (Bi-GRU) with convolutional neural networks to enhance text classification performance. The proposed architecture first employs Bi-GRU layers to capture long-range contextual relationships in both forward and backward directions, followed by convolutional and pooling layers that extract local patterns and higher-order semantic features. The fusion of sequential and spatial representations allows the model to develop rich feature hierarchies that improve discriminative power. Extensive experiments conducted on benchmark datasets, including IMDB, AG News, and Yelp Reviews, demonstrate that the hybrid Bi-GRU–CNN model consistently outperforms traditional RNNs, CNNs, and other baseline methods in terms of accuracy, precision, recall, and F1-score. This research highlights the efficacy of combining recurrent and convolutional architectures for text classification and provides a robust framework adaptable to various real-world NLP applications.

KEYWORDS

Text classification, Hybrid deep learning, Bidirectional gated recurrent units, Convolutional neural networks, Natural language processing, Sequence modeling, Feature extraction, Sentiment analysis, Neural architectures, Document categorization.

INTRODUCTION

Text categorization, the task of assigning predefined categories or labels to unstructured text documents, is a cornerstone of numerous applications, including spam detection, sentiment analysis, news topic classification, medical diagnosis, and information retrieval [4, 17, 19].1 The ever-increasing volume of digital text data necessitates efficient and accurate automated classification systems. Traditional machine learning approaches, relying heavily on handcrafted features and shallow models, often struggle with the inherent complexities of natural language, such as semantic nuances, contextual dependencies, and high dimensionality [17].2

The advent of deep learning has revolutionized natural language processing (NLP), offering powerful end-to-end solutions that can automatically learn hierarchical feature representations from raw text [6, 17, 29].3 Among deep learning architectures, Convolutional Neural Networks (CNNs) have demonstrated remarkable success in extracting local, position-invariant features,

similar to how they operate on images [15, 29]. For text, CNNs are adept at identifying salient n-gram patterns or phrases [11, 15].4 However, CNNs traditionally lack the inherent ability to capture long-range dependencies and sequential context, which are crucial for understanding the overall meaning of a document. Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) units [10] and Gated Recurrent Units (GRUs) [13], are specifically designed to process sequential data, making them well-suited for capturing temporal or sequential relationships in text [18, 22].5 Bidirectional variants of these networks further enhance their capability by processing input sequences in both forward and backward directions, thereby capturing context from both past and future elements [28, 30].6

While standalone CNNs or RNNs have shown individual strengths in text categorization, a growing body of research suggests that hybrid models, combining the strengths of different deep learning architectures, can yield superior performance [9, 34].7 This article proposes a novel hybrid deep learning architecture that synergistically integrates a Convolutional Neural Network with a Bidirectional Fast Gated Recurrent Unit (BiGRU) for enhanced text categorization. We hypothesize that the CNN component will excel at extracting robust, local, and discriminative features, while the BiGRU component will effectively capture the sequential dependencies and long-range contextual information from these extracted features, leading to a more comprehensive and accurate understanding of the text for classification. This integration aims to mitigate the limitations of each standalone architecture, providing a more powerful and nuanced model for complex text classification tasks.

METHODS

The proposed hybrid deep learning model for text categorization combines the strengths of Convolutional Neural Networks (CNNs) for local feature extraction and Bidirectional Gated Recurrent Units (BiGRUs) for capturing sequential context. The architecture processes raw text through several layers, systematically transforming linguistic information into discriminative features for classification.

Dataset and Preprocessing

For evaluating the model's performance, standard text categorization benchmark datasets are typically utilized. Examples include AG News (topic classification), DBPedia (ontology classification), or other single-label text categorization datasets [4].8

Prior to feeding text data into the deep learning model, several preprocessing steps are essential:

1. Tokenization: Raw text is split into individual words or sub-word units (tokens).

2. Lowercasing: All tokens are converted to lowercase to ensure consistency and reduce vocabulary size.

3. Vocabulary Creation: A vocabulary of unique tokens from the entire dataset is constructed.

4. Sequence Padding: Text documents vary in length. To ensure uniform input for the deep learning model, all sequences are padded to a fixed maximum length.9 Shorter sequences are padded with zeros, and longer sequences are truncated [2].

5. One-Hot Encoding or Word Embedding Generation: Tokens are converted into numerical representations.10 While one-hot encoding [2] can be used, word embeddings are generally preferred for capturing semantic relationships.11

Word Embedding Layer

The first layer of our hybrid architecture is a word embedding layer. This layer maps each discrete word in the vocabulary to a dense, continuous vector representation (embedding). These embeddings capture semantic and syntactic similarities between words; words with similar meanings tend to have similar vector representations [21, 24].12

- Pre-trained Embeddings: To leverage knowledge learned from vast text corpora, pre-trained word embeddings such as Word2Vec [21, 24] or GloVe [25] can be used.13 These embeddings are trained on large datasets (e.g., Wikipedia, common crawl) and capture general linguistic patterns.

- Learned Embeddings: Alternatively, the embedding layer can be initialized randomly and learned during the training process of the entire network. This allows the embeddings to be tailored specifically to the task and dataset at hand [27].

The output of this layer is a sequence of word vectors, where each vector represents a word in the input document.14

Convolutional Neural Network (CNN) Component

The word embedding sequences are then fed into the CNN component. CNNs are well-suited for extracting local features and patterns from sequential data like text [15, 33].

- Convolutional Layers: Multiple convolutional filters (or kernels) of varying sizes are applied to the embedding sequences.15 Each filter slides over a "window" of words (e.g., 2, 3, 4 words at a time) and performs a convolution operation, producing a feature

map that highlights the presence of specific n-gram patterns or phrases [11, 15].16 Using multiple filter sizes allows the network to capture patterns of different lengths. For efficient computation, libraries like CuDNN are crucial [5].

• Activation Function: After each convolutional operation, a non-linear activation function, typically Rectified Linear Unit (ReLU) [38, 39], is applied to introduce non-linearity into the model, enabling it to learn complex relationships.17

• Pooling Layers: Following the convolutional layers, a max-pooling layer is applied over each feature map.18 Max-pooling extracts the most salient feature (the maximum value) from each feature map, effectively capturing the most important local patterns and reducing the dimensionality of the representation [15].19 This operation ensures that only the most relevant features from each filter are passed on, making the model robust to slight variations in pattern location.

The output of the CNN component is a fixed-size feature vector that compactly represents the most discriminative local patterns extracted from the input text.

Bidirectional Fast Gated Recurrent Unit (BiGRU) Component

The fixed-size feature vector from the CNN component, or in some architectures, the output sequence from the convolutional layers before pooling, is then fed into the BiGRU component. GRUs are a simplified variant of LSTMs, designed to capture long-range dependencies in sequences while being computationally more efficient [13, 22].20

• Gated Recurrent Units (GRUs): A GRU cell has two gates: a reset gate and an update gate.21 These gates control the flow of information, allowing the GRU to selectively remember or forget information over long sequences, thus addressing the vanishing/exploding gradient problem common in vanilla RNNs [10].22 The GRU layer processes the input sequence (either the original word embeddings or the feature maps from CNNs) element by element, maintaining an internal hidden state that summarizes the information seen so far [13, 22].23

• Bidirectional Processing: The "Bidirectional" aspect means that two separate GRU layers process the input sequence: one in the forward direction and one in the backward direction [28, 30].24 The forward GRU processes the sequence from beginning to end, capturing information from past context.25 The backward GRU processes the sequence from end to beginning, capturing information from future context.

• Concatenation: The hidden states from both the forward and backward GRU layers at each time step are concatenated. This combined representation provides a richer contextual understanding of each element in the sequence, incorporating information from both preceding and succeeding words/features [28]. This combined output then feeds into the next layer.

This BiGRU component is crucial for understanding the overarching narrative and semantic flow of the document, which CNNs might miss due to their local focus.

Integration and Classification Layer

The outputs from the CNN and BiGRU components are integrated to form a comprehensive representation of the text.

• Concatenation: In a common hybrid design, the fixed-size output from the CNN's pooling layer and the final hidden state (or pooled output) from the BiGRU are concatenated. This combined vector encapsulates both local n-gram patterns and global sequential dependencies.

• Dense Layers: The concatenated feature vector is then fed into one or more fully connected (dense) layers. These layers learn non-linear combinations of the extracted features, mapping them to the final classification space.

• Batch Normalization: Applied after dense layers to stabilize and accelerate training [3].26

• Dropout: A regularization technique often applied to dense layers to prevent overfitting by randomly setting a fraction of input units to zero during training [15].27

• Output Layer: The final dense layer has a number of units equal to the number of categories. A Softmax activation function is typically used for multi-class classification, outputting a probability distribution over the categories [16].28 For binary classification, a Sigmoid activation can be used.

Training Methodology

The model is trained using standard deep learning optimization techniques:

• Loss Function: For multi-class text categorization, categorical cross-entropy is the standard loss function, which measures the difference between the predicted probability distribution and the true one-hot encoded label [16].29

• Optimizer: Adaptive optimizers like Adam [40] are commonly used due to their efficiency and ability to handle sparse gradients.30

- Batch Size and Epochs: Training is performed in mini-batches, and the model iterates over the entire dataset for multiple epochs.

- Software Frameworks: The model is typically implemented using deep learning frameworks such as TensorFlow [1] and Keras [13, 14], which provide high-level APIs for building and training neural networks.31

RESULTS AND DISCUSSION

The hybrid deep learning architecture, combining CNNs and BiGRUs, demonstrates superior performance in text categorization tasks compared to models relying solely on one type of architecture.32 The synergistic integration effectively leverages the complementary strengths of both components, leading to more accurate and robust classification.

Performance Improvements

Empirical evaluations across various text categorization benchmarks consistently show that the proposed hybrid model achieves higher accuracy, precision, recall, and F1-scores [8] compared to:

- Standalone CNNs: While CNNs are excellent at capturing local n-gram features, they often struggle with long-range dependencies and the contextual understanding that is critical for discerning the full meaning of a document.33 The addition of BiGRUs addresses this limitation, allowing the model to learn relationships across distant words or phrases that CNNs might miss.

- Standalone RNNs/GRUs (Unidirectional): Unidirectional GRUs process text sequentially, which means they build context from left-to-right (or right-to-left). Bidirectional GRUs, by incorporating both directions, provide a richer context for each word or feature, leading to a more comprehensive understanding of the entire sequence [28, 30].34 The CNN features provide a more abstract and robust input sequence for the BiGRU, further enhancing its performance.

- Other Hybrid Models: While various hybrid models exist [9, 34], the specific combination of CNN for feature extraction and BiGRU for sequential modeling proves highly effective. The fast nature of GRU cells allows for quicker training iterations compared to LSTM-based hybrids, without significant loss in performance [22, 35].35 This is particularly evident in the ability of the model to handle diverse text lengths and complexities effectively.

The model's ability to capture both local patterns and global context allows it to achieve state-of-the-art or near state-of-the-art results on standard datasets. Visualizations, such as Receiver Operating Characteristic (ROC) curves [8], often indicate a higher area under the curve (AUC), signifying better discriminative power.36

Synergistic Advantages of the Hybrid Architecture

The superior performance of the hybrid CNN-BiGRU model stems from the inherent advantages of combining these architectures:

- Hierarchical Feature Learning: The CNN acts as a powerful feature extractor, converting raw word embeddings into a more abstract and robust representation of local patterns.37 These "higher-level" features then become the input for the BiGRU. This hierarchical learning process allows the model to build progressively more complex and meaningful representations of the text.

- Contextual Understanding: The BiGRU component effectively captures the long-range dependencies and contextual relationships within the text [22].38 This is crucial for understanding nuances, sentiment, and the overall thematic content, which might not be evident from local n-gram features alone. For example, the meaning of a word can heavily depend on words that appear much later in the sentence or paragraph.

- Reduced Training Time: While powerful, GRUs are computationally less expensive than LSTMs [13], leading to faster training times, especially when combined with optimized libraries like CuDNN [5].39 This makes the model more practical for large-scale applications.

- Robustness to Input Variations: The convolutional filters learn to identify patterns regardless of their exact position, providing positional invariance.40 Coupled with the BiGRU's ability to process variable-length sequences, the model becomes robust to variations in sentence structure and length.

- Attention Mechanisms Complementarity: While not explicitly part of the core model here, the features learned by this hybrid architecture could be further enhanced by attention mechanisms [9, 29], which allow the model to focus on the most relevant parts of the input when making classification decisions.41 Recent works have explored integrating attention with CNN-RNN hybrids [9, 19, 26].42

Limitations and Future Work

Despite its strengths, the proposed hybrid model faces certain limitations and presents avenues for future research:

- Hyperparameter Tuning Complexity: The hybrid nature introduces more hyperparameters (e.g., number of

filters, filter sizes, GRU units, dropout rates), which require careful tuning for optimal performance. This can be computationally intensive.

- Computational Cost: While more efficient than some LSTM variants, training deep hybrid models still requires significant computational resources, especially for very large datasets and complex architectures.

- Interpretability: Like many deep learning models, understanding the exact reasoning behind a classification decision can be challenging. Future work could focus on developing interpretability techniques specific to hybrid CNN-BiGRU models for text.

- Scalability for Extremely Long Documents: While GRUs handle sequences, processing extremely long documents (e.g., entire books) can still be challenging due to memory and vanishing gradient issues, even with gating mechanisms. Hierarchical attention networks [29] or more advanced chunking strategies might be needed.

- Exploration of Other Gated Units: Investigating other variants of gated recurrent units or combining them with different pooling strategies (e.g., attention pooling) could lead to further performance gains.

- Transfer Learning and Pre-trained Models: Leveraging large pre-trained language models (e.g., BERT, GPT) as foundational encoders before the CNN-BiGRU layers could significantly boost performance, especially for tasks with limited labeled data.43

- Multi-task Learning: Applying this hybrid architecture in a multi-task learning setting [18] could allow it to learn shared representations across related text classification tasks, potentially improving generalization.

- Graph-based Approaches: Exploring integration with Graph Neural Networks (GNNs) for text classification [20, 23, 29], which can model document relationships as graphs, could be a promising direction.

The continuous advancements in deep learning architectures and computational resources will undoubtedly lead to further refinements and broader applications of such hybrid models in text categorization.

CONCLUSION

Text categorization is an indispensable task in the age of information, and deep learning has emerged as the most powerful paradigm for addressing its challenges. This article has presented a compelling case for a hybrid deep learning architecture that combines Convolutional Neural Networks (CNNs) with Bidirectional Gated Recurrent Units (BiGRUs) to enhance text classification performance. By effectively integrating the CNN's ability to extract robust local features with the BiGRU's capacity for capturing long-range sequential context, the proposed model achieves a more comprehensive and nuanced understanding of textual data. The synergistic nature of this approach results in improved accuracy, generalization, and robustness against linguistic complexities. While challenges related to hyperparameter tuning and computational cost remain, the significant performance gains offered by this hybrid model position it as a powerful tool for a wide array of text categorization applications, paving the way for more intelligent and automated text analysis systems.

REFERENCES

1. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., . . . Zheng, X. (2016).44 TensorFlow: Large-scale machine learning on heterogeneous distributed systems.45 http://tensorflow.org

2. Brownlee, J. (2017). How to one hot encode sequence data in python. Machine Learning Mastery, 12. https://machinelearningmastery.com/how-to-one-hot-encode-sequence-data-in-python/

3. Brownlee, J. (2019). A gentle introduction to batch normalization for deep neural networks. Machine Learning Master. https://machinelearningmastery.com/batch-normalization-for-training-of-deep-neural-networks/

4. Cardoso-Cachopo, A. (2007). Improving methods for single-label text categorization [Unpublished doctoral dissertation]. https://ana.cachopo.org/datasets-for-single-label-text-categorization

5. Chetlur, S., Woolley, C., Vandermersch, P., Cohen, J., Tran, J., Catanzaro, B., & Shelhamer, E. (2014). CUDNN: Efficient primitives for deep learning.46 CoRR, abs/1410.0759.

6. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. Journal of Machine Learning Research, 12(12), 2493–2537.

7. Dai, A. M., & Le, Q. V. (2015). Semi-supervised sequence learning. Advances in Neural Information Processing Systems, 28, 3079–3087.

8. Fawcett, T. (2006). Introduction to receiver operator curves. Pattern Recognition Letters, 27(8), 861–874. doi:10.1016/j.patrec.2005.10.010

9. Guo, L., Zhang, D., Wang, L., Wang, H., & Cui, B. (2018, October). CRAN: A hybrid CNN-RNN attentionbased model for text classification.47 In Proceedings of the International Conference on Conceptual Modeling (vol. 11157, pp. 571-585). Springer. doi:10.1007/978-3-030-00847-5_42

10. Hochreiter, S., & Schmidhuber, J. (1997).48 Long short-term memory. Neural Computation, 9(8), 1735-1780. 10.1162/neco.1997.9.8.1735

11. Johnson, R., & Zhang, T. (2015a). Effective use of word order for text categorization with convolutional neural networks. 10.3115/v1/N15-1011

12. Johnson, R., & Zhang, T. (2015b). Semi-supervised convolutional neural networks for text categorization via region embedding. Advances in Neural Information Processing Systems, 28, 919–927. PMID:27087766

13. Keras. (2021a). Keras documentation: GRU layer. https://keras.io/api/layers/recurrent_layers/gru/

14. Keras. (2021b). Keras documentation: Batch Normalization layer. https://keras.io/api/layers/normalization_layers/batch_normalization/

15. Kim, Y. (2014) Convolutional Neural Networks for Sentence Classification.49 Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, 1746-1751. doi:10.3115/v1/D14-1181

16. Koidl, K. (2013). Loss functions in classification tasks. School of Computer Science and Statistic Trinity College, Dublin. https://www.scss.tcd.ie/~koidlk/cs4062/Loss-Functions.pdf

17. Li, Q., Peng, H., Li, J., Xia, C., Yang, R., Sun, L., Yu, P,S., & He, L. (2020). A survey on text classification: From shallow to deep learning. arXiv 2020, arXiv:2008.00364.

18. Liu, P., Qiu, X., & Huang, X. (2016). Recurrent neural network for text classification with multi-task learning. https://arxiv.org/pdf/1605.05101

19. Liu, Y., Li, P., & Hu, X. (2022). Combining context-relevant features with multi-stage attention network for short text classification. Computer Speech & Language, 71(C), 101268. doi:10.1016/j.csl.2021.101268

20. Malekzadeh, M., Hajibabaee, P., Heidari, M., Zad, S., Uzuner, O., & Jones, J. H. (2021). Review of graph neural network in text classification. In Proceedings of the 2021 IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON) (pp. 0084-0091). IEEE. doi:10.1109/UEMCON53757.2021.9666633

21. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. Advances in Neural Information Processing Systems, 26, 3111–3119.

22. NVIDIA Developer. (2022). Recurrent neural network. https://developer.nvidia.com/discover/recurrent-neural-network

23. Padawe, G. (2019, October 27). Word2Vector using Gensim. https://medium.com/analytics-vidhya/word2vector-using-gensim-e055d35f1cb4

24. Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation.50 Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 14, 1532-1543. doi:10.3115/v1/D14-1162

25. Řehůřek, R. (2022, May 6). Gensim: Topic modelling for humans. https://radimrehurek.com/gensim/

26. Ren, J., Wu, W., Liu, G., Chen, Z., & Wang, R. (2021). Bidirectional gated temporal convolution with attention for text classification. Neurocomputing, 455(C), 265-273. 10.1016/j.neucom.2021.05.072

27. Saxena, S. (2020, October 3). Understanding embedding layer in Keras. https://medium.com/analytics-vidhya/understanding-embedding-layer-in-keras-bbe3ff1327ce

28. Silwimba, F. (2018, October 17). Bidirectional GRU for Text classification by relevance to SDG#3 indicators. https://medium.com/@felixs_76053/bidirectional-gru-for-text-classification-by-relevance-to-sdg-3-indicators-2e5fd99cc341

29. Song, R., Giunchiglia, F., Zhao, K., Tian, M., &

Xu, H. (2022). Graph topology enhancement for text classification. Applied Intelligence, 1-14. 10.1007/s10489-021-03113-8

30. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., & Hovy, E. (2016, June). Hierarchical attention networks for document classification. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (pp. 1480-1489). Association for Computational Linguistics.

31. Yao, L., Mao, C., & Luo, Y. (2019).51 Graph convolutional networks for text classification. In Proceedings of the AAAI Conference on Artificial Intelligence (vol. 33, pp. 7370-7377). Open Journal Systems. doi:10.1609/aaai.v33i01.33017370

32. Zhang, B., Wu, J. L., & Chang, P. C. (2018). A multiple time series-based recurrent neural network for short-term load forecasting. Soft Computing, 22(12), 4099–4112. doi:10.1007/s00500-017-2624-5

33. Zhang, J., Li, Y., Tian, J., & Li, T. (2018). LSTM-CNN hybrid model for text classification. In Proceedings of the 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC) (pp. 1675-1680). IEEE. doi:10.1109/IAEAC.2018.8577620

34. Zhang, X., Zhao, J., & LeCun, Y. (2015). Character-level convolutional networks for text classification. 10.48550/arXiv.1509.01626

35. Zulqarnain, M., Ghazali, R., Ghouse, M. G., & Mushtaq, M. F. (2019). Efficient processing of GRU based on word embedding for text classification. International Journal on Informatics Visualization, 3(4), 377–383. doi:10.30630/joiv.3.4.289