ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25may1682

Deep Learning Architectures for Text Classification

Chitra Desai

Professor, Department of Computer Science, National Defence Academy, Pune, India

Publication Date: 2025/05/31

Abstract: Text classification is crucial in natural language processing applications such as sentiment analysis, topic tagging, and news categorization. This paper presents a comparative analysis of three deep learning architectures—LSTM, Bidirectional LSTM, and Character-level Convolutional Neural Networks (Char-CNN), for the task of news categorization using the AG News dataset. The models were trained using a unified preprocessing pipeline, including tokenization, padding, and label encoding. Performance was evaluated based on classification accuracy, training time, and learning stability across epochs.

The results show that Bidirectional LSTM outperforms the standard LSTM in capturing long-range dependencies by leveraging both past and future context. The Character-level CNN demonstrates robust performance by learning morphological patterns directly from raw text, making it resilient to misspellings and out-of-vocabulary words. The trade-offs between model complexity, training time, and interpretability has also been analyzed.

This study offers practical insights into model selection for real-world NLP applications and highlights the importance of architectural choices in deep learning-based text classification.

Keywords*: Deep Learning for NLP; Text Classification Models; Bidirectional LSTM Performance; Character-level CNN; AG News Dataset.*

How to Cite: Chitra Desai (2025) Deep Learning Architectures for Text Classification. *International Journal of Innovative Science and Research Technology*, 10(5), 2568-2573. https://doi.org/10.38124/ijisrt/25may1682

I. INTRODUCTION

Text classification has become a cornerstone task in Natural Language Processing (NLP), enabling the automated understanding, organization, and analysis of vast volumes of textual data. Traditionally, rule-based systems and conventional machine learning techniques were employed to tackle this problem. However, these approaches often struggled to capture the rich semantic and syntactic nuances inherent in natural language [1]. The emergence of deep learning has significantly advanced the field, providing architectures capable of learning hierarchical and contextual features directly from raw text without relying heavily on manual feature engineering.

Among the deep learning architectures, Convolutional Neural Networks (CNNs) have demonstrated remarkable success in NLP applications such as sentence classification and semantic parsing. Their strength lies in their ability to extract local n-gram features and exploit hierarchical structures within text, making them highly effective for modeling short- to mid-range dependencies [2]. However, a transformative leap in NLP performance was achieved through the introduction of word embeddings.

Mikolov et al. initially proposed recurrent neural network-based language models that could learn distributed representations of words, preserving their semantic and syntactic properties [3]. This led to the development of the widely adopted Word2Vec model, which refined these embeddings to capture linguistic regularities and compositional patterns in a continuous vector space [4]. Building on this foundational work, recursive neural models were introduced to capture compositional semantics, thereby enhancing the model's ability to understand complex sentence structures and sentiment [5].

These innovations in representation learning and architectural design collectively shifted text classification from a feature-engineering-centric task to one based on end-to-end learning, resulting in significant improvements in accuracy and scalability across real-world applications.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25may1682

Expanding on these advancements, Hierarchical Attention Networks (HAN) were introduced to enhance document-level classification by integrating both word- and sentence-level attention mechanisms [6]. More recently, the advent of transformer-based models has revolutionized how text is represented and processed. The Transformer architecture, proposed by Vaswani et al. [7], laid the groundwork for self-attention mechanisms that capture long-range dependencies more effectively. This innovation gave rise to powerful contextualized models such as BERT [8], RoBERTa [9], and ALBERT [10], which have set new benchmarks across a variety of NLP tasks. Furthermore, lightweight variants like TinyBERT [11] and MobileBERT [12] have addressed the need for efficiency, making these models more suitable for deployment in resource-constrained environments.

These evolving architectures form the technological foundation for building robust, scalable, and high-performance text classification systems. Their continuous development remains central to the progress of NLP.

In this context, the present paper investigates and compares the effectiveness of three deep learning architectures—Long Short-Term Memory (LSTM), Bidirectional LSTM (BiLSTM) [13], and Character-level Convolutional Neural Network (Char-CNN)[14] —using AG News dataset for text classification . While LSTM and BiLSTM operate at the word level and excel at capturing sequential dependencies, Char-CNN works directly with character-level inputs, making it particularly resilient to out-of-vocabulary (OOV) words and noisy textual data.

A key component across all these architectures is the embedding layer, which plays a vital role in transforming high-dimensional, sparse input data into dense, low-dimensional vector representations. Unlike traditional encoding techniques such as one-hot vectors or TF-IDF [15], which treat tokens as independent and orthogonal, embedding layers learn continuous-valued vectors where semantically similar tokens are mapped to nearby points in the space. This capability enhances the model's generalization, reduces computational overhead, and facilitates more nuanced learning of syntactic and semantic relationships. In the LSTM, BiLSTM, and Char-CNN models studied here, embedding layers form the base for modeling context and dependencies, contributing to more effective learning from sequential data.

To ensure a fair and rigorous comparison, a unified preprocessing and training pipeline was implemented across all models. Their performance was evaluated using key metrics such as classification accuracy, training time, and learning stability. The results demonstrate that while BiLSTM outperforms others in terms of accuracy and efficiency, Char-CNN exhibits competitive performance with distinctive strengths in processing raw and noisy text. This comparative study underscores the importance of architectural choices in shaping the practical effectiveness and adaptability of deep learning models for real-world NLP applications.

II. DATASET

The AG News dataset [14] consists of 120,000 training samples and 7,600 test samples, evenly distributed across 4 news categories: 0 World, 1 Sports, 2 Business,3 Science/Technology. The dataset was downloaded from Kaggle.

The samples are listed in Figure 1 that include columns label, title and description (short summary) of a news article, making it ideal for document classification, information retrieval, and deep learning-based NLP tasks.

III. MODELS AND EXPERIMENTAL

This section presents the deep learning models employed for the task of text classification, along with the corresponding experimental setup. The models include Long Short-Term Memory (LSTM), Bidirectional LSTM (BiLSTM), and Character-level Convolutional Neural Network (Char-CNN), each chosen for their unique ability to handle sequential and textual data. The LSTM model captures temporal dependencies in a unidirectional fashion, while BiLSTM enhances this by processing sequences in both forward and backward directions, thus gaining richer contextual understanding. The Char-CNN model, on the other hand, operates at the character level, making it robust to spelling variations and out-of-vocabulary (OOV) words. These models were trained and evaluated using a consistent experimental pipeline to ensure comparability in performance. The setup includes preprocessing techniques, model architectures, hyperparameter tuning, and evaluation metrics, all of which are detailed in the subsequent subsections.

A. A. LSTM-based Text Classification Model

The LSTM-based model for text classification starts with an Embedding layer, which transforms input words into dense 128-dimensional vectors. With a vocabulary limit of 10,000 and a maximum input sequence length of 200 tokens, this layer outputs a tensor of shape (None, 200, 128). It is followed by a unidirectional LSTM (Long Short-Term Memory) layer with 128 units. This layer processes sequences in the forward direction and is well-suited for capturing dependencies in textual data over time. To prevent overfitting, a Dropout layer with a dropout rate of 0.5 is applied. The output is then passed through a Dense layer with 64 units and ReLU activation to enable non-linear feature learning. The final Dense output layer has 4 units with softmax activation to perform multi-class classification corresponding to the four categories in the AG News dataset. This model has about 1.42 million trainable parameters, providing a balanced mix of capacity and simplicity for baseline experimentation.

B. Bidirectional LSTM-based Text Classification Model

The Bidirectional LSTM model expands on the standard LSTM architecture by incorporating context from both past and future tokens within each sequence. It begins with the same Embedding layer as the previous model, producing a (None, 200, 128) tensor. The Bidirectional LSTM layer comprises two LSTMs — one processing the input sequence from left to right and the other from right to left — resulting in an output dimensionality of 256. This bi-directional flow allows the model to better understand the context of each word within a

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25may1682

sentence. Following this is a Dropout layer to mitigate overfitting. The Dense layer with 64 units and ReLU activation helps refine the learned features before classification. The final Dense layer uses softmax activation across 4 units to categorize the input text into one of the four AG News classes. This model has approximately 1.56 million parameters, offering more representational power than the standard LSTM model.

C. Character-level Convolutional Neural Network (Char-CNN)

This character-level CNN model processes text as a sequence of individual characters, capturing fine-grained patterns at the subword level. Unlike traditional word-based models, this architecture begins with an Embedding layer that transforms each character (encoded as an integer) into a dense 128-dimensional vector, allowing the model to learn distributed character-level representations.

	label	Title	Description
0	2	Wall St. Bears Claw Back Into the Black (Reuters)	Reuters - Short-sellers, Wall Street's dwindli...
1	2	Carlyle Looks Toward Commercial Aerospace (Reu...	Reuters - Private investment firm Carlyle Grou...
2	2	Oil and Economy Cloud Stocks' Outlook (Reuters)	Reuters - Soaring crude prices plus worries\ab...
3	2	Iraq Halts Oil Exports from Main Southern Pipe...	Reuters - Authorities have halted oil export\f...
4	2	Oil prices soar to all-time record, posing new...	AFP - Tearaway world oil prices, toppling reco...
...	...	...	...
119995	0	Pakistan's Musharraf Says Won't Quit as Army C...	KARACHI (Reuters) - Pakistani President Perve...
119996	1	Renteria signing a top-shelf deal	Red Sox general manager Theo Epstein acknowled...
119997	1	Saban not going to Dolphins yet	The Miami Dolphins will put their courtship of...
119998	1	Today's NFL games	PITTSBURGH at NY GIANTS Time: 1:30 p.m. Line: ...
119999	1	Nets get Carter from Raptors	INDIANAPOLIS -- All-Star Vince Carter was trad...

120000 rows × 3 columns

Fig 1 AG News Dataset (Sample)

The core of the model comprises multiple stacked 1D convolutional layers with 256 filters each. The initial two convolutional layers use a kernel size of 7 to capture longer character n-grams, followed by max-pooling layers to downsample the feature maps and emphasize the most salient features. Subsequent layers use smaller kernel sizes (3) and continue deep feature extraction, enabling the model to learn increasingly abstract character-level patterns.

A Flatten layer converts the 3D feature maps into a 1D vector, which is then passed through two dense layers with 1024 neurons each and ReLU activations, interleaved with dropout layers (0.5) to mitigate overfitting. Finally, a softmax-activated output layer assigns the input text to one of four target categories.

This architecture is particularly effective in scenarios with noisy, informal, or misspelled text data, as it does not rely on a predefined vocabulary. Instead, it builds its understanding directly from character sequences, offering robustness to out-of-vocabulary issues. Though deeper and more computationally intensive, it is well-suited for large-scale, raw-text classification tasks.

IV. RESULTS

The LSTM model demonstrated consistent but underwhelming performance throughout the training process, as evidenced by the following metrics across five epochs. In the first epoch, the model achieved a training loss of 1.3868 and a training accuracy of 24.8%, with the validation loss at 1.3864 and a validation accuracy of 25.0%. This pattern remained relatively stable throughout the subsequent epochs, with training accuracy fluctuating slightly between 24.76% and 25.09%, and validation accuracy consistently maintaining 25.0%. By the final epoch, the training loss slightly increased to 1.3865, and the validation loss stayed near 1.3863.

Several important observations can be made from these results:

Loss is not decreasing significantly: This suggests that the model is stuck and is not improving. The consistent high loss indicates that the model is not making meaningful progress during training.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25may1682

Accuracy is around 25%: Since the AG News dataset has 4 classes, random guessing would also give 25% accuracy. The model's accuracy being stuck at this level further suggests that it is not learning anything meaningful and is simply making random predictions.

Validation accuracy is not improving: This indicates that the model is either:

Not learning from the data (underfitting), possibly due to an inappropriate model choice, insufficient training, or lack of relevant features.

Facing an incorrect training setup, such as wrong hyperparameters, preprocessing issues, or other underlying problems affecting the model's ability to generalize beyond the training data.

Overall, these points highlight that the LSTM model is failing to capture the necessary patterns in the data, and adjustments to the model architecture, hyperparameters, or training process may be necessary to improve its performance.

The Bidirectional Long Short-Term Memory (BiLSTM) model was trained for 5 epochs, with a total training duration of approximately 8662.24 seconds (144.37 minutes). The model exhibited strong classification performance and stability throughout the training process.

In Epoch 1, the model achieved a training accuracy of 88.45% and a validation accuracy of 91.01%, with corresponding training and validation losses of 0.3366 and 0.2651, respectively, over 1173 seconds. Epoch 2 showed improved performance, reaching 92.51% training accuracy and 91.39% validation accuracy, with training completed in 1118 seconds.

During Epoch 3, the model further advanced to a training accuracy of 93.96% and a validation accuracy of 91.71%, taking 1630 seconds. Epoch 4 maintained high accuracy levels with 95.14% training and 91.24% validation accuracy, trained over 2845 seconds. In Epoch 5, the model achieved its highest training accuracy of 96.08%, with a slightly lower validation accuracy of 91.62%, in 1895 seconds.

The results affirm the BiLSTM model’s strength in capturing sequential dependencies in text, delivering high accuracy with efficient training times.

The training of the Character-level Convolutional Neural Network (Char-CNN) model was carried out over 5 epochs, completing in approximately 15,230 seconds (253.83 minutes). The model demonstrated consistent improvements in classification performance across epochs.

In Epoch 1, the model achieved a training accuracy of 61.37% and a validation accuracy of 83.00%, with corresponding losses of 0.8824 and 0.4750, respectively, over 2644 seconds. Epoch 2 showed significant improvement, reaching a training accuracy of 85.68% and a validation accuracy of 84.78%, completed in 3047 seconds.

Performance continued to rise in Epoch 3, with the training accuracy increasing to 88.28% and validation accuracy to 87.76%, taking 3930 seconds. Epoch 4 recorded the highest validation accuracy of 88.47%, alongside a training accuracy of 89.54%, in 2859 seconds. Finally, Epoch 5 maintained strong metrics with 90.48% training accuracy and 87.41% validation accuracy, trained in 2751 seconds.

These results reflect the effectiveness of character-level feature learning in capturing textual nuances and achieving high classification accuracy with a reasonable training duration.

V. COMPARATIVE ANALYSIS: CHAR-CNN VS. BIDIRECTIONAL LSTM

Both Character-level Convolutional Neural Network (Char-CNN) and Bidirectional Long Short-Term Memory (BiLSTM) models were trained under identical experimental conditions, including batch sizes and a fixed training duration of five epochs.

Despite these uniform settings, the two architectures demonstrated distinct performance profiles in terms of training efficiency, and accuracy progression as shown in Table 1. BiLSTM proved significantly more time-efficient, completing training in approximately 8,662 seconds (~144.37 minutes), which is nearly 43% faster than Char-CNN’s 15,230 seconds (~253.83 minutes). This substantial difference in training time underscores BiLSTM’s computational efficiency despite its recurrent structure.

In terms of accuracy, BiLSTM exhibited a consistently higher validation accuracy across all epochs. It began with a strong initial accuracy of 91.01%, peaked at 91.71% in the third epoch, and maintained a stable performance, closing at 91.62%. In contrast, Char-CNN started at 83.00%, reached its highest accuracy of 88.47% in the fourth epoch, and ended slightly lower at 87.41%. This suggests that while Char-CNN improved steadily, it was unable to match the predictive accuracy of BiLSTM.

Loss values further reinforced BiLSTM’s advantage. The model achieved lower and more stable validation loss, starting at 0.2651 and fluctuating mildly between 0.2576 and 0.3033. Char-CNN, on the other hand, began with a higher loss of 0.4750 and reached its lowest point at 0.3463. These results indicate that BiLSTM not only converged faster but also maintained greater generalization stability throughout training. Collectively, these insights highlight the superiority of BiLSTM in both learning efficiency and model performance for this text classification task, despite Char-CNN’s strengths in handling character-level inputs.

In addition to performance metrics, the models were also analysed for their architectural complexity, particularly the number of trainable parameters as shown in Table 2. The LSTM and BiLSTM models contained approximately 1.4 to 1.8 million parameters, reflecting their relatively lightweight design optimized for sequence modeling using word-level embeddings. In contrast, the Character-level Convolutional Neural Network (Char-CNN) had a significantly higher

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25may1682

parameter count of 11,452,676. This substantial increase stems from the convolutional layers applied over character embeddings and the fully connected layers used for classification. While the larger parameter space of Char-CNN enables it to learn nuanced subword patterns and morphological features, making it resilient to noisy or out-of-vocabulary input, it also contributes to longer training times and greater memory consumption. This comparison underscores the trade-offs between model expressiveness and computational efficiency, particularly in scenarios where deployment resources are limited.

Table 1 Comparison of Model Accuracy and Training Time

Model	***Validation Accuracy***	***Training Time (s)***	***Remarks***
LSTM	~25%	5,800	Failed to learn effectively; minimal gains across epochs.
BiLSTM	91.7%	8,662	Best overall performance; efficient convergence.
Char-CNN	87.8%	15,230	Robust to out-of-vocabulary inputs; longest training time.

Table 2 Comparison of Model Parameters

Model	**Total Parameters**	**Description**
LSTM	1.42 million	Uses word-level embeddings and a single-directional LSTM layer.
BiLSTM	1.56 million	Roughly double the LSTM due to bidirectional connections. Captures both past and future context.
Char-CNN	11,452,676 (11.45 million)	Substantially higher parameter count. Uses convolutional filters over character embeddings, allowing for robust subword and morphological feature learning.

VI. CONCLUSION

The LSTM model underperformed with ~25% validation accuracy, indicating it failed to learn beyond random guessing. The BiLSTM model achieved the best performance with 91.7% accuracy in less time (~8,662s), effectively capturing bidirectional context and sequential dependencies. Char-CNN, while robust to noisy text and achieving a decent 87.8% accuracy, required the longest training time (~15,230s) and the most parameters (11.45M), making it less efficient. BiLSTM's relatively lower parameter count (1.56M) and high accuracy suggest it generalizes well without overfitting. These results highlight BiLSTM as the most practical and scalable choice for real-world text classification tasks.

REFERENCES

[1]. Li, Q., Peng, H., Li, J., Xia, C., Yang, R., Sun, L., Yu, P. S., & He, L. (2021). A Survey on Text Classification: From Traditional to Deep Learning. ACM Computing Surveys, 54(3), 1–35.

[2]. Young, T., Hazarika, D., Poria, S., & Cambria, E. (2018). Recent trends in deep learning based natural language processing. IEEE Computational Intelligence Magazine, 13(3), 55–75.

[3]. T. Mikolov, M. Karafiat, L. Burget, J. Cernock ´ y, and S. Khudanpur, "Recurrent neural network based language model." ` in Interspeech, vol. 2, 2010, p. 3.

[4]. T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in Advances in neural information processing systems, 2013, pp. 3111–3119.

[5]. R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, C. Potts et al., "Recursive deep models for semantic compositionality over a sentiment treebank," in Proceedings of the conference on empirical methods in natural language processing (EMNLP), vol. 1631, 2013, p. 1642.

[6]. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., & Hovy, E. (2016). 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.

[7]. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In Advances in Neural Information Processing Systems (NeurIPS), 30.

[8]. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of NAACL-HLT, pp. 4171–4186.

[9]. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692.

[10]. Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2020). ALBERT: A lite BERT for self-supervised learning of language representations. In International Conference on Learning Representations (ICLR).

[11]. Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., Wang, F., & Liu, Q. (2020). TinyBERT: Distilling BERT for natural language understanding. In Findings of the Association for Computational Linguistics: EMNLP 2020, pp. 4163–4174.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25may1682

[12]. Sun, Z., Yu, H., Song, X., Liu, R., Yang, Y., & Zhou, D. (2020). MobileBERT: a compact task-agnostic BERT for resource-limited devices. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 2158–2170.
[13]. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780.
[14]. Zhang, X., Zhao, J., & LeCun, Y. (2015). Character-level convolutional networks for text classification. In Advances in Neural Information Processing Systems (NeurIPS), 28.
[15]. Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. Information Processing & Management, 24(5), 513–523.