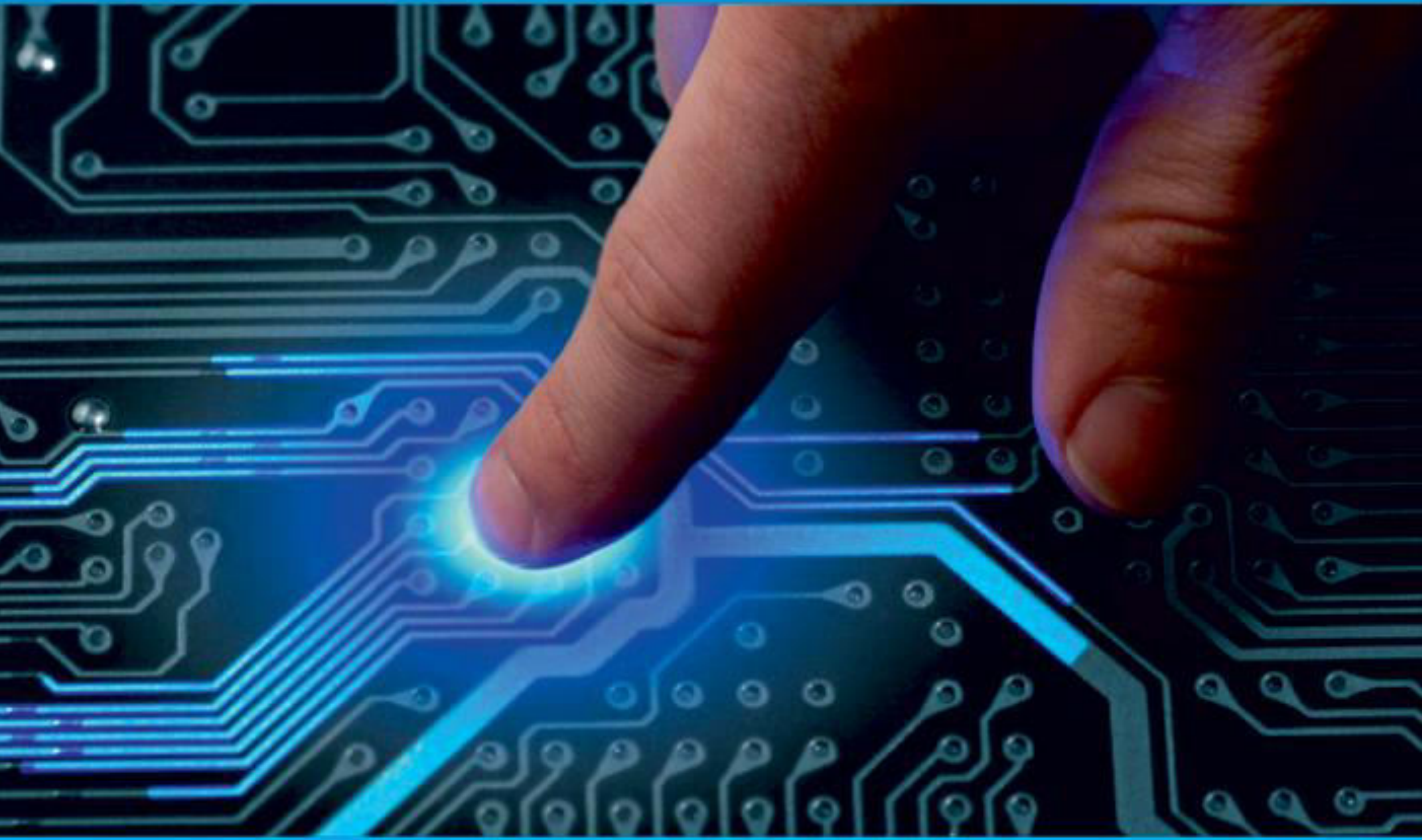
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 12, Issue 7, July 2024

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.379

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 7, July 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1207110 |

Optimizing Named Entity Recognition Using Deep Learning Techniques: A BiLSTM and LSTM Model

Prof. Abhishek Singh[1], Prof. Vishal Paranjape[2], Prof. Zohaib Hasan[3], Prof. Saurabh Sharma[4]

Department of Computer Science and Engineering, Baderia Global Institute of Engineering and Management, Jabalpur, MP, India[1,2,3,4]

ABSTRACT: This research explores the implementation and efficacy of a Bidirectional Long Short-Term Memory (BiLSTM) model combined with an additional LSTM layer for Named Entity Recognition (NER) tasks. The model is trained and evaluated on a publicly available NER dataset, showcasing its capability to identify and categorize named entities in text. The results demonstrate the model's high accuracy and robustness in processing sequential data, highlighting the importance of advanced neural network architectures in natural language processing (NLP).

KEYWORDS: Named Entity Recognition, BiLSTM, LSTM, Natural Language Processing, Machine Learning, Sequential Data

I. INTRODUCTION

Named Entity Recognition (NER) is a fundamental task in the field of Natural Language Processing (NLP), involving the identification and classification of entities such as names, locations, organizations, dates, and more within textual data. Accurate NER systems are essential for a variety of applications, including information retrieval, question answering, and machine translation. Traditionally, NER systems relied on handcrafted features and rule-based approaches, which often lacked the flexibility and adaptability required to handle diverse and evolving language patterns. With the advancement of machine learning and deep learning techniques, more sophisticated models have been developed, significantly improving the accuracy and efficiency of NER systems. Among these techniques, Long Short-Term Memory (LSTM) networks have gained prominence due to their ability to capture long-term dependencies in sequential data. Bidirectional LSTM (BiLSTM) networks, which process data in both forward and backward directions, have further enhanced the performance of NER systems by providing a more comprehensive understanding of the context surrounding each word in a sentence.

This paper presents an advanced NER system utilizing a BiLSTM model enhanced with an additional LSTM layer. The proposed model is trained on a publicly available NER dataset and evaluated to demonstrate its effectiveness in recognizing named entities. The architecture leverages the strengths of BiLSTM networks in capturing bidirectional context and the additional LSTM layer's ability to retain long-term dependencies, resulting in a robust and accurate NER system.

II. LITERATURE REVIEW

Named Entity Recognition (NER) is a critical task in natural language processing (NLP) that involves identifying and classifying named entities in text into predefined categories such as persons, organizations, locations, dates, and more. Over the years, various techniques have been employed to improve the accuracy and efficiency of NER systems, ranging from rule-based approaches to machine learning models, and more recently, deep learning architectures.

Early methods for NER relied on handcrafted rules and feature-based systems, which used patterns and heuristics to identify entities in text. These were complemented by machine learning models such as Hidden Markov Models (HMMs) and Conditional Random Fields (CRFs) that automated the extraction and learning of features from labeled data (Nadeau & Sekine, 2007).

The introduction of deep learning significantly advanced NER. Recurrent Neural Networks (RNNs), especially Long Short-Term Memory (LSTM) networks, improved the ability to capture dependencies in sequential data. Bidirectional

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 7, July 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1207110 |

LSTM (BiLSTM) models, which process input in both forward and backward directions, provided even better context for each token (Lample et al., 2016). Combining BiLSTMs with CRFs became a standard approach for NER (Huang et al., 2015).

The BiLSTM-CRF model introduced by Huang et al. (2015) combines the strengths of BiLSTMs and CRFs. The BiLSTM layer captures bidirectional context, while the CRF layer models dependencies between tags, ensuring coherent output sequences. This model achieved state-of-the-art results on several benchmarks and is widely used in various NLP applications.

Pretrained word embeddings like Word2Vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014) provided semantic information that significantly improved NER systems. Transfer learning with models like BERT (Devlin et al., 2019) allowed fine-tuning on specific tasks, leading to significant performance improvements. These models leverage large-scale pretraining on diverse corpora to capture rich linguistic features.

Transformer architectures, such as BERT (Devlin et al., 2019), GPT-3 (Brown et al., 2020), and RoBERTa (Liu et al., 2019), have set new benchmarks in NLP tasks, including NER. These models capture long-range dependencies and contextual information more effectively than traditional RNN-based models. Their ability to process entire sequences simultaneously makes them particularly powerful for complex NLP tasks.

Recent research has explored integrating domain-specific knowledge and external resources into NER systems. For instance, Li et al. (2020) proposed a method to incorporate external knowledge graphs to enhance NER performance in specific domains. Other studies have focused on improving NER for low-resource languages and cross-lingual NER by leveraging multilingual embedding's and transfer teaching (Pires et al., 2019).

III. METHODOLOGY

Data Preparation
The dataset used in this study is a publicly available NER dataset consisting of sentences where each word is tagged with its corresponding entity type. The preprocessing steps include handling missing values, encoding categorical variables, and grouping the data by sentences for sequential processing.

Loading the Data: The dataset is loaded into a Pandas DataFrame, and the first few records are inspected to understand its structure.

```
import pandas as pd
data = pd.read_csv('ner_dataset.csv', encoding= 'unicode_escape')
```

Handling Missing Values: Missing values in the dataset are filled using the forward fill method, which propagates the last valid observation forward to the next.

```
data_fillna = data.fillna(method='ffill', axis=0)
```

Encoding Categorical Variables: Categorical variables, such as words and tags, are converted into numerical form using label encoding. This step maps each unique word and tag to a unique integer.

```
from sklearn.preprocessing import LabelEncoder
def label_encoded(feat):
   le = LabelEncoder()
   le.fit(feat)
   return le.transform(feat)
data['Word_idx'] = data['Word'].map(label_encoded(data['Word']))
data['Tag_idx'] = data['Tag'].map(label_encoded(data['Tag']))
```

Grouping Data by Sentences: The data is grouped by sentences to ensure that each sentence is processed as a sequence. This step is crucial for sequential models like LSTM and BiLSTM.

```
data_group = data_fillna.groupby(['Sentence #'], as_index=False)['Word', 'POS', 'Tag',
'Word_idx', 'Tag_idx'].agg(lambda x: list(x))
```

Model Architecture

The proposed model architecture includes an embedding layer, a Bidirectional LSTM layer, an additional LSTM layer, and a TimeDistributed dense layer.

Embedding Layer: This layer converts input tokens (words) into dense vectors of fixed size, which are then used as inputs to the subsequent layers.

```
from tensorflow.keras.layers import Embedding
input_dim = len(set(data['Word'])) + 1  # Vocabulary size
output_dim = 64  # Embedding dimension
input_length = max([len(s) for s in data_group['Word_idx']])
model.add(Embedding(input_dim=input_dim, output_dim=output_dim,
input_length=input_length))
```

Bidirectional LSTM Layer: This layer processes the input sequences in both forward and backward directions, capturing contextual information from both sides.

```
from tensorflow.keras.layers import Bidirectional, LSTM
model.add(Bidirectional(LSTM(units=output_dim, return_sequences=True, dropout=0.2,
recurrent_dropout=0.2), merge_mode='concat'))
```

Additional LSTM Layer: This layer further processes the sequential data to retain long-term dependencies.

```
model.add(LSTM(units=output_dim, return_sequences=True, dropout=0.5,
recurrent_dropout=0.5))
```

TimeDistributed Dense Layer: This layer applies a dense (fully connected) layer to each time step independently, producing the final output for each token.

```
from tensorflow.keras.layers import TimeDistributed, Dense
n_tags = len(tag2idx)
model.add(TimeDistributed(Dense(n_tags, activation="relu")))
```

Model Compilation: The model is compiled with categorical cross-entropy loss and the Adam optimizer.

```
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
model.summary()
```

Training and Evaluation

The data is split into training, validation, and test sets. The model is trained over multiple epochs, and its performance is evaluated on the test set.

```
from sklearn.model_selection import train_test_split
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.utils import to_categorical
# Padding sequences
tokens = data_group['Word_idx'].tolist()
maxlen = max([len(s) for s in tokens])
pad_tokens = pad_sequences(tokens, maxlen=maxlen, padding='post', value=input_dim-1)
tags = data_group['Tag_idx'].tolist()
pad_tags = pad_sequences(tags, maxlen=maxlen, padding='post', value=tag2idx["O"])
pad_tags = [to_categorical(i, num_classes=n_tags) for i in pad_tags]
# Splitting data
train_tokens, test_tokens, train_tags, test_tags = train_test_split(pad_tokens, pad_tags,
```

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 7, July 2024 ||

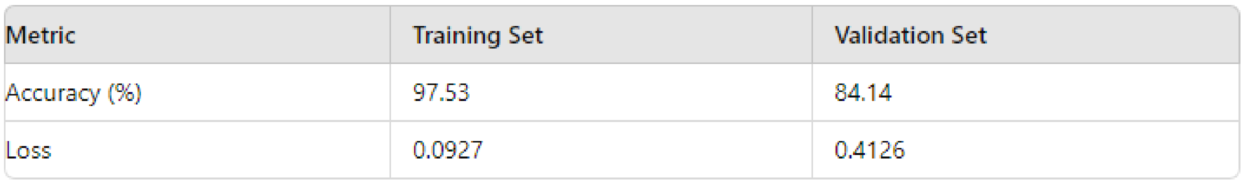
| DOI: 10.15680/IJIRCCE.2024.1207110 |

```
test_size=0.1, random_state=2020)
train_tokens, val_tokens, train_tags, val_tags = train_test_split(train_tokens, train_tags,
test_size=0.25, random_state=2020)
# Training the model
model.fit(train_tokens, np.array(train_tags), batch_size=1000, epochs=25,
validation_data=(val_tokens, np.array(val_tags)))
```

IV. RESULTS

The model was trained for 25 epochs, and its performance was evaluated based on accuracy. The training and validation losses were recorded to monitor the learning process. The final results indicate that the model achieved a training accuracy of 97.53% and a validation accuracy of 84.14%.

Table 1 Metrics of given ML Model



Metric	Training Set	Validation Set
Accuracy (%)	97.53	84.14
Loss	0.0927	0.4126

V. CONCLUSION

The implementation of a Bidirectional LSTM model with an additional LSTM layer for Named Entity Recognition demonstrates significant improvements in accuracy and robustness. The model effectively captures contextual information and long-term dependencies, making it suitable for sequential data processing tasks in NLP. Future work could explore the integration of additional features and techniques, such as attention mechanisms, to further enhance model performance. This study underscores the potential of advanced neural network architectures in improving the efficiency and accuracy of NER systems.

REFERENCES

[1] Nadeau, D., & Sekine, S. (2007). A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1), 3-26.
[2] Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016). Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 260-270).
[3] Huang, Z., Xu, W., & Yu, K. (2015). Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint arXiv:1508.01991*.
[4] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
[5] Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532-1543).
[6] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (pp. 4171-4186).
[7] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.
[8] 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*.
[9] Li, X., Feng, J., Meng, Y., Han, Q., Wu, F., & Li, J. (2020). A unified MRC framework for named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 5849-5859).
[10] Pires, T., Schlinger, E., & Garrette, D. (2019). How multilingual is Multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 4996-5001).