Named Entity Recognition Using Bidirectional LSTM Networks

Dr. David Raj Micheal
Department of Mathematics
School of Advanced Sciences
Vellore Institute of Technology Chennai
Tamil Nadu – 600127

Keerthana V
Department of Mathematics
School of Advanced Sciences
Vellore Institute of Technology Chennai
Tamil Nadu – 600127

***Abstract*—This project explores the implementation of a Named Entity Recognition (NER) system using a Bidirectional RNN with LSTM layers to identify and categorize named entities within text. By applying the IOB tagging schema, the model classifies tokens into various entity classes, improving the structure and organization of information extraction tasks. The model leverages a bidirectional LSTM to capture contextual information from both past and future tokens, enhancing classification accuracy. Evaluation results show the model's effectiveness in distinguishing entity boundaries and types, offering improvements over traditional rule-based methods.**

***Index Terms*—Embeddings, Knowledge Graphs, Natural Language Processing, Deep Learning, Named Entity Recognition**

I. Introduction

Named Entity Recognition (NER) is a foundational process in natural language processing (NLP) that involves detecting and classifying named entities within text into predefined categories, such as Person, Location, and Organization. NER has widespread applications, including search engines, content categorization, and knowledge base construction, which rely on identifying relevant entities to structure unstructured data. Traditional rule-based and statistical ap-proaches have offered moderate success in NER tasks, but re- cent advancements in deep learning, specifically in sequence- to-sequence learning, have demonstrated superior performance [1]. This study leverages a bidirectional LSTM (BiLSTM) model to capture contextual information in both forward and backward directions, improving accuracy in identifying and classifying named entities within a text.

The main contributions of this work are:

- Development and implementation of a BiLSTM model for the NER task, leveraging IOB tagging for token-level entity classification.
- An analysis of model performance across various entity types, using cross-entropy loss and accuracy metrics [2].
- A comparison of BiLSTM-based NER with traditional rule-based methods, showcasing the advantages of deep contextual learning for NLP applications [3].

II. Data Preprocessing

A. Dataset Overview

The dataset used for this NER model is sourced from entity-annotated corpus [8]. It contains labeled sentences where each token is annotated with an entity tag, following the IOB (Inside-Outside-Beginning) tagging format. Entity categories in this dataset include:

- geo: Geographical Entities
- org: Organizations
- per: Persons
- gpe: Geopolitical Entities
- tim: Time Indicators

Each word in the dataset is labeled as either the beginning of an entity (B), inside an entity (I), or outside any entity (O). This tagging schema facilitates boundary detection and entity classification in a single pass.

B. Data Preprocessing

The dataset is preprocessed to optimize it for sequential input to the BiLSTM model. Key preprocessing steps include:

- **Sentence Grouping:** Each sentence is treated as an independent sequence, and tokens within each sentence are mapped to numerical indices.
- **Tokenization and Index Mapping:** A vocabulary of unique tokens is constructed, with each word and tag assigned a unique index. This mapping facilitates the embedding of words and labels into vector representations.
- **Padding:** Sentences are padded to a fixed length to ensure consistent sequence dimensions across batches, which is essential for efficient batch processing in deep learning frameworks.

The processed dataset has a vocabulary size of 18520 and 17 distinct tags. Sentences were padded to a maximum length of 140 to ensure uniformity across batches.

III. Model Architecture

The NER model architecture is based on a bidirectional LSTM (BiLSTM) network, implemented using Keras with TensorFlow backend. The BiLSTM structure is ideal for capturing the sequence dependencies in both directions, thus improving the context understanding necessary for effective entity recognition [4].

A. Layer Configuration

The model consists of three main layers:

- **Embedding Layer:** The embedding layer transforms each word index into a 50-dimensional dense vector. This layer serves as a lookup table for the vocabulary, mapping each word to a fixed-size vector space.
- **Bidirectional LSTM Layer:** A 64-cell LSTM layer is applied bidirectionally to capture both forward and backward contexts within the sentence. This bidirectionality is crucial for NER tasks, where understanding the full context of a word within a sentence can improve the accuracy of entity classification.
- **TimeDistributed Dense Layer:** This final layer applies a dense layer to each timestep individually, using a softmax activation function to produce a probability distribution over the possible tags.

B. Training Configuration

The model is compiled with categorical cross-entropy loss, using the Adam optimizer and accuracy as the evaluation metric. Key hyperparameters include:

- **Batch Size:** 32
- **Learning Rate:** 0.001, with a ReduceLROnPlateau callback to decrease the learning rate when the validation loss plateaus.
- **Epochs:** 10, with early stopping after 7 epochs if no improvement in validation loss is observed.

IV. Hyperparameter Tuning

To maximize model performance, several hyperparameters were tuned through experimentation. These include:

- **Embedding Vector Dimension:** Set to 50 dimensions, based on experimentation to balance between model complexity and generalization.
- **LSTM Units:** 64 units in the BiLSTM layer, providing a balance between capturing sufficient sequence information and maintaining computational efficiency.
- **Recurrent Dropout:** 0.1, applied to prevent overfitting by randomly dropping units during training.
- **Early Stopping:** Implemented to prevent overfitting, with a patience level of 7 epochs on validation loss.

Hyperparameter tuning was conducted by grid search, systematically adjusting one parameter at a time while observing changes in validation accuracy and loss. This process ensured that the final configuration maximized model accuracy without significantly increasing overfitting risk.

A. Evaluation Metrics

The model's performance was evaluated using accuracy and cross-entropy loss on both training and validation sets. The results were as follows:

1) Model Performance: The model achieved an accuracy of 99.3% on the validation set, demonstrating effective recognition of entities across sentence contexts. Loss convergence indicated stable learning, with minimal overfitting observed due to early stopping and recurrent dropout [5].

TABLE I
RESULTS AND ANALYSIS TABLE

Metric	Result
Training Accuracy	99.7%
Validation Accuracy	99.3%

2) Performance Across Entity Classes: The model's classification accuracy was consistent across most entity types, with slightly lower performance in rare categories, such as Natural Phenomenon entities, due to limited examples.

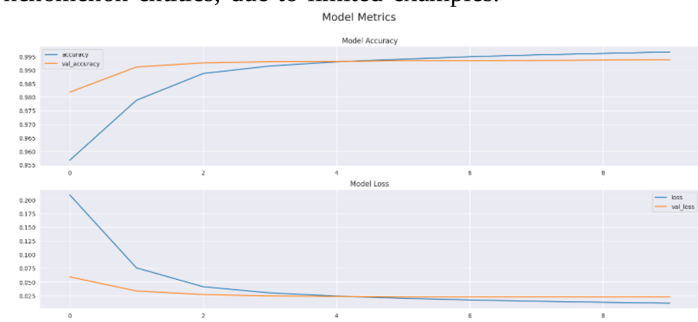

Figure 1: Model Accuracy and Model Loss

V. Future Works

- **Augmenting Dataset:** Incorporating additional labeled data from diverse domains could improve generalization, especially for underrepresented entity types.
- **Using Pretrained Embeddings:** Integrating pretrained embeddings such as GloVe or BERT could potentially improve entity recognition by leveraging semantic knowledge from larger corpora [6].
- **Experimenting with Transformer Architectures:** The use of transformer-based models like BERT or RoBERTa could further enhance the model's performance, as these architectures are particularly well-suited for sequence tagging tasks [7].

References

[1] J. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," in *Proc. of the 2016 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 260-270.

[2] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798-1828, Aug. 2013.

[3] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[4] A. Vaswani et al.,"Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017.

[5] P. J. Liu, M. Saleh, and E. Hovy, "Exploring the effectiveness of CNN-based models for multi-class classification," *IEEE Trans. on Neural Networks and Learning Systems*, 2021.

[6] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. of NAACL-HLT*, 2019.

[7] A. Radford, "Improving language understanding by generative pre-training," *OpenAI*, 2018.

[8] https://www.kaggle.com/datasets/abhinavwalia95/entity-annotated-corpus