Named Entity Recognition Using CNN-BiLSTM and GloVe Embeddings

Name: Soochan Andrew Lee

1. Introduction

Named Entity Recognition (NER) is a fundamental task in Natural Language Processing (NLP) that involves identifying and classifying specific entities in text. These entities typically belong to predefined categories such as persons (PER), organizations (ORG), locations (LOC), and miscellaneous (MISC). The goal of NER is to detect and label these entities accurately within unstructured text data, which is essential for information extraction and knowledge representation.

NER plays a crucial role in numerous NLP applications by enabling machines to extract structured information from unstructured text. It is widely used in search engines to enhance query understanding, in chatbots and virtual assistants to improve entity recognition in conversations, and in medical text analysis to extract relevant terms from clinical documents. Additionally, NER is valuable in financial news analysis, where it identifies companies, stocks, and market entities, and in legal document processing, where it helps extract relevant names, laws, and contractual information.

Traditional NER approaches, such as rule-based systems and statistical models, face several limitations. These methods struggle with ambiguity in entity classification, where words like “Apple” can refer to both a company and a fruit. Additionally, variations in text formats, including abbreviations, typos, and informal language, make recognition more complex. Another major challenge is limited generalization, as rule-based systems often fail to adapt to new domains and unseen entity names. Furthermore, data sparsity remains an issue, as certain entity types lack sufficient labeled examples for robust learning.

To address these challenges, I propose a hybrid CNN-BiLSTM model enhanced with GloVe embeddings for NER:

- CNN (Convolutional Neural Networks): Extracts local features from words, capturing subword patterns and contextual dependencies.
- BiLSTM (Bidirectional Long Short-Term Memory): Captures long-range dependencies and sequential patterns in text by processing it in both forward and backward directions.
- GloVe Embeddings: Pre-trained word representations provided a rich understanding of word meanings based on their co-occurrence in large corpora, improving the model’s ability to generalize.

Deep learning techniques, including RNNs and CNNs, have replaced traditional rule-based and statistical methods for NER. While RNNs effectively model sequential dependencies, CNNs excel at extracting local patterns. By combining both, the proposed CNN-BiLSTM model leverages their strengths to achieve improved accuracy and robustness in real-world NER applications such as information retrieval, question answering, and machine translation.

2. Dataset Curation

2.1 CoNLL-2003 Dataset

The dataset used for this project is the CoNLL-2003 dataset, which is widely recognized for NER tasks. The dataset consists of text sequences labeled with four entity types: PER (person names), ORG (organizations), LOC (locations), and MISC (miscellaneous entities such as nationalities and events).

Dataset Distribution:

Split	Number of Sentences	Ratio (%)
Training	14,041	67.7%
Validation	3,250	15.7%
Test	3,453	16.6%

The dataset is structured with tokens (`tokens`), POS tags (`pos_tags`), chunking tags (`chunk_tags`), and NER labels (`ner_tags`). However, we only used `tokens` and `ner_tags` for this project.

3. Methodology

3.1 Word Embeddings

Utilized pre-trained GloVe embeddings (300D) to represent words as dense vectors to improve model performance. The embeddings were sourced from glove.6B.300d.txt file, containing a vocabulary size of 400,000 with an embedding dimension of 300. Additionally, introduced a padding token (`<PAD>`) with an all-zero vector to handle variable-length sequences.

3.2 Model Architecture: CNN-BiLSTM

Designed a hybrid CNN + BiLSTM model that effectively captures both local and long-range dependencies in text sequences. The model consists of the following layers:

1. Embedding Layer: Maps words to dense vector representations using pre-trained GloVe embeddings

2. 1D CNN Layer: Captures local word dependencies by applying convolutional filters over the input embeddings.
3. BiLSTM Layer: Extracts contextual relationships from both past and future words using bidirectional processing.
4. Fully Connected Layer: Maps LSTM outputs to NER labels through a linear transformation.
5. Dropout (0.5): Regularizes the model to prevent overfitting by randomly setting a portion of neurons to zero.
6. Softmax Activation: Outputs probabilities for each word's NER label, ensuring a valid classification result.

3.3 Training Process

The training process was designed to optimize the CNN-BiLSTM NER model for efficient learning while preventing overfitting. This involved dynamic padding, batch processing, loss function selection, and regularization techniques to ensure stable training and generalization.

3.3.1 Loss Function and Optimization

The model was trained using Categorical Cross-Entropy Loss, which is a standard loss function for multi-class classification tasks. This loss function measures the discrepancy between predicted label distributions and true labels, ensuring accurate classification.

For optimization, the Adam optimizer was used with a learning rate of 0.001. Adam was chosen for its adaptive learning rate and momentum-based updates, which accelerate convergence while preventing large oscillations in weight updates.

3.3.2 Batch Processing and Dynamic Padding

Since sentences in the dataset vary in length, dynamic padding was applied using `pad_sequence()`, ensuring that all sequences within a batch were padded to match the longest sentence while preserving meaningful structure.

A batch size of 32 was selected, balancing computational efficiency and convergence stability. The model was trained using mini-batch gradient descent, updating weights iteratively to optimize performance.

3.3.3 Regularization and Weight Initialization

To prevent overfitting, dropout regularization (0.5) was applied in the fully connected layer, randomly deactivating neurons during training to enhance generalization.

The embedding layer was initialized with pre-trained GloVe embeddings (300D), ensuring meaningful word representations from the start of training.

3.3.4 Training Procedure and Performance Monitoring

The model was trained for 5 epochs, as empirical analysis showed that loss values plateaued beyond this point, indicating diminishing returns from additional training. Each training iteration followed these steps:

1. Forward Pass: Input sequences were processed through the embedding layer, CNN, BiLSTM, and fully connected layers to generate label predictions.
2. Loss Computation: The Cross-Entropy Loss was computed between predicted and true labels.
3. Backpropagation: Gradients were calculated using autograd, and model weights were updated via Adam optimizer.
4. Dropout Regularization: Applied to the fully connected layer to prevent overfitting.
5. Performance Logging: After each epoch, training and validation loss were recorded to monitor convergence.

The model was trained for a fixed 5 epochs without early stopping, as validation performance remained stable.

4. Experimental Results and Evaluation

The model was trained for 5 epochs, achieving a consistently decreasing loss:

Training Loss per Epoch:

Epoch	Loss
1	0.13922
2	0.03998
3	0.02754
4	0.01934
5	0.01451

4.1 Performance Metrics

After training the CNN-BiLSTM NER model for 5 epochs, I evaluated its performance on the test set using standard NER evaluation metrics: precision, recall, and F1-score. The model successfully classified named entities across four categories: Persons (PER), Organizations (ORG), Locations (LOC), and Miscellaneous (MISC), computed via `seqeval.metrics.classification_report` module.

Classification Report:

Entity	Precision	Recall	F1-Score	Support
LOC	0.83	0.90	0.86	1668
MISC	0.72	0.67	0.70	702
ORG	0.81	0.73	0.77	1661
PER	0.86	0.90	0.88	1617
Overall	0.82	0.82	0.82	5648

The model performed best on PER (Persons) and LOC (Locations), achieving F1-scores of 0.88 and 0.86, respectively, indicating strong identification of well-defined entities like people's names and locations. In contrast, the MISC category had the lowest F1-score (0.70), likely due to data sparsity, making it difficult to recognize less structured entities such as nationalities and creative works. The ORG (Organizations) category had a slightly lower recall (0.73), suggesting some organization names were misclassified or missed, possibly due to overlapping entity types (e.g., companies vs. locations). Despite these challenges, the model achieved an overall micro-F1 score of 0.82, demonstrating strong performance in Named Entity Recognition (NER) tasks.

5. Analyses and Experiments

To further evaluate the effectiveness of the CNN-BiLSTM NER model, I conducted additional experiments focusing on hyperparameter tuning and ablation studies. These experiments provided insights into how different configurations impacted model performance and helped identify areas for potential improvement.

5.1 Hyperparameter Tuning

- Batch Size: The model was trained with batch sizes of 16, 32, and 64. A batch size of 32 achieved the best balance between computational efficiency and stable convergence, while larger batch sizes led to less stable training.
- Learning Rate: I experimented with 0.001, 0.0005, and 0.0001. A learning rate of 0.001 resulted in the most stable training process and the fastest convergence, whereas a lower learning rate prolonged training without significant improvements.
- Dropout Rate: Different dropout rates (0.3, 0.5, and 0.7) were tested to assess their effect on overfitting. A dropout rate of 0.5 provided the optimal trade-off, preventing overfitting while maintaining useful feature learning.

5.2 Ablation Study

To understand the contribution of different components in the model, I performed an ablation study by removing specific elements and comparing their impact on the F1-score across different entity types.

Model Variant	LOC F1	MISC F1	ORG F1	PER F1	Overall F1
Full Model (CNN + BiLSTM + GloVe)	0.86	0.70	0.77	0.88	0.82
Without CNN (Only BiLSTM)	0.85	0.69	0.74	0.89	0.81
Without BiLSTM (Only CNN)	0.84	0.70	0.69	0.84	0.78
Without Pretrained GloVe	0.75	0.57	0.57	0.62	0.64

- Removing CNN had a small but noticeable impact on the model's ability to classify organization and location entities, showing that CNN helps capture local word dependencies that aid in distinguishing proper nouns. However, overall performance remained strong, indicating that BiLSTM alone is capable of modeling most sequence dependencies.
- Removing BiLSTM had a larger impact, particularly on organization and location categories. Since BiLSTM captures long-range dependencies, its removal caused difficulty in recognizing named entities that rely on contextual understanding, reducing performance across the board.
- Removing Pretrained GloVe embeddings resulted in the most significant drop in performance, with the overall F1-score decreasing from 0.82 to 0.64. The impact was especially severe for MISC and ORG categories, likely due to the inability of randomly initialized embeddings to capture meaningful semantic representations.

These findings confirm that the hybrid CNN-BiLSTM architecture with pre-trained GloVe embeddings is optimal for achieving high accuracy in NER tasks.

6. Lessons & Experience Learned

This project provided valuable insights into deep learning-based Named Entity Recognition (NER) and highlighted various challenges associated with NLP tasks. One of the most significant lessons was the impact of pre-trained word embeddings on model performance. The use of GloVe embeddings greatly enhanced the model's ability to recognize entities, particularly well-defined categories like persons (PER) and locations (LOC). Without pre-trained

embeddings, the model struggled to generalize, leading to increased misclassification, especially for rare or ambiguous entities.

In terms of model architecture, the project demonstrated the strengths and trade-offs between convolutional and recurrent neural networks. The CNN component was effective in capturing local word dependencies and subword patterns, while the BiLSTM layer excelled at understanding long-range contextual information. The combination of both architectures provided a balanced approach, ensuring the model effectively processed both word-level and sentence-level dependencies. The ablation study confirmed that removing either CNN or BiLSTM resulted in decreased performance, further reinforcing the importance of this hybrid design.

The challenges in NER became more apparent throughout the project. One of the primary difficulties was entity ambiguity, where certain words could belong to multiple entity classes. For example, organization names were sometimes misclassified as locations due to their resemblance to geographical terms. Additionally, the MISC category posed a unique challenge because of its broad and loosely defined nature, leading to lower recall compared to the other categories. Another challenge was handling variable-length sequences, which was mitigated by implementing dynamic padding to ensure stable training without excessive computational overhead.

Overall, this project underscored the importance of selecting the right model components, handling real-world data complexities, and critically analyzing performance beyond standard metrics. The experience gained from this implementation will be valuable for future research and applications in NLP, particularly in optimizing deep learning models for sequence labeling tasks.