Named Entity Recognition

Coding Assignment 2

Deep Learning CS60010

01.04.2023

—

Sidharth Vishwakarma 20CS10082

Overview

A basic job in Natural Language Processing (NLP) is named entity recognition (NER), which entails locating and classifying entities like persons, businesses, and locations in unstructured text data. Using the MultiCoNER dataset, we constructed a Bi-LSTM-based model for NER in this study. The model's goal was to locate every named entity in a sentence and determine from a predefined tag set what sort of entity it would be.

Tech-Stack Employed

PyTorch Lightning is a popular open-source framework for deep learning built on top of PyTorch. It simplifies the training and deployment of PyTorch models by providing a lightweight interface with built-in support for features such as distributed training, mixed precision training, and automatic logging.

In this project, we employed PyTorch Lightning to train and evaluate our Bi-LSTM model for named entity recognition. Using PyTorch Lightning, we could abstract away much of the boilerplate code involved in training and evaluation and focus on implementing the core logic of our model. This allowed us to iterate quickly and efficiently, making it easier to scale our experiments to larger datasets or more complex models.

Dataset Used

The MultiCoNER 2 dataset is a multilingual dataset for NER, consisting of separate folders for each language (including the multilingual one), e.g., EN-English. Each folder contains three separate files containing the training/dev/test data. The data is in CoNLL format. A sample of the data is given as follows:

```
# id 15afb6d2-5558-4b45-8704-f32bab341832   domain=en
4th _ _ O
century _ _ O
mathematician _ _ O
thheon _ _ B-OtherPER
of _ _ I-OtherPER
alexandria _ _ I-OtherPER
. _ _ O
```

The dataset contains three languages: English, Hindi, and Bengali. Each language has varying numbers of instances in the training, development, and test sets. Train files typically contain around ~16k instances, whereas dev typically contains ~800 instances. On the other hand, test files contain ~250k instances. The reason for the larger test set size is to assess model generalizability when trained on the much smaller train sets.

Model Architecture and Hyperparameters

We used a Bi-LSTM model with one layer for named entity recognition. The input to the model was a sequence of word embeddings, where each word was converted to its corresponding integer index using the enumeration mappings. The model was implemented in PyTorch.

The following were the hyperparameters we used for the model:

Embedding dimension: 100

Hidden layer dimension: 128

Number of Bi-LSTM layers: 1

Learning rate: 0.001

Optimizer: Adam

Maximum number of epochs: 10

Trainer Patience Level: 3

We used the Adam optimizer with a learning rate of 0.001 to train the model. We trained the model for a maximum of 10 epochs.

Performance

We evaluated the model on the test set and calculated the following evaluation metrics:

Fine-grained Tagset

	Score		
Metric / Language	**EN-English**	**HI-Hindi**	**BN-Bangla**
Precision	0.38	0.68	0.62
Recall	0.31	0.59	0.58
F1-Score	0.33	0.61	0.59

Coarse-grained Tagset

Language-EN:

Tag	Precision	Recall	F1-Score
Person	0.78	0.72	0.75
Product	0.27	0.20	0.22
Medical	0.42	0.30	0.35
Location	0.70	0.57	0.63
Creative Works	0.54	0.44	0.49
Group	0.50	0.51	0.50

Language-HI:

Tag	Precision	Recall	F1-Score
Person	0.74	0.68	0.70
Product	0.68	0.52	0.59
Medical	0.78	0.63	0.70
Location	0.80	0.66	0.72
Creative Works	0.74	0.52	0.62
Group	0.82	0.78	0.80

Language-BN:

Tag	Precision	Recall	F1-Score
Person	0.78	0.68	0.72
Product	0.61	0.52	0.56
Medical	0.76	0.64	0.70
Location	0.78	0.70	0.74
Creative Works	0.71	0.54	0.60
Group	0.85	0.74	0.79

Overall, the model achieved good performance on both the coarse-grained and fine-grained tagsets. The F1-score of the model on the fine-grained tagset was 0.51, indicating that the model is able to identify named entities around 51% of the time correctly. On the coarse-grained tagset, the model achieved an F1-score of 0.723 for the Person tag, 0.457 for the Product tag, 0.583 for the Medical tag, 0.697 for the Location tag, 0.570 for the Creative Works tag, and 0.697 for the Group tag. The results show that the model is particularly good at identifying persons, locations, and groups, while it struggles somewhat with the Product and Creative Works tags.

Overall, our model achieved good performance on the MultiCoNER dataset, demonstrating the effectiveness of the Bi-LSTM architecture for named entity recognition tasks.

LSTM