Named Entity Recognition (NER)

1. Introduction

Problem Statement

- Named Entity Recognition (NER) is a fundamental task in Natural Language Processing (NLP) that involves identifying and classifying entities in text into predefined categories such as names of people, organizations, locations, and other entities. The goal of this project is to develop a robust NER model using the **BERT-based** architecture to accurately identify and classify entities in text.

Objectives

- To build an effective NER model using **BERT** and fine-tune it on the **CoNLL-2003 dataset**.
- To evaluate the performance of the model through metrics as **precision**, **recall**, **F1 score**, and **accuracy**.

2. Data Description

Dataset

Use the **CoNLL-2003 dataset**.

- **Source**: The dataset is publicly available and can be loaded using the **datasets** library.
- **Number of Instances:**
  - **Training Set**: Approximately 14,000 sentences.
  - **Validation Set**: Approximately 3,000 sentences.
  - **Test Set**: Approximately 3,000 sentences.
- **Entity Types:**
  - **LOC** (Location)
  - **ORG** (Organization)
  - **PER** (Person)
  - **MISC** (Miscellaneous)

Data Splits

- **Training Set**: Used to train the model.
- **Validation Set**: Used for hyperparameter tuning and model evaluation.
- **Test Set**: Used to evaluate the final model.

3. Baseline Experiments

Goal

- Establish a baseline model using a **pre-trained BERT** model for token classification.

Initial Experiments

- **Model**: BERT-based model
- **Tokenizer**: AutoTokenizer from Hugging Face Transformers
- **Training Configuration**: The model was trained using default hyperparameters due to insufficient resources on Colab to select different ones.

4. Advanced Experiments

1- Fine-Tuning BERT for NER

- **Goal**
  - Improve model performance by fine-tuning the BERT model on the CoNLL-2003 dataset.
- **Methodology**
  - **Preprocessing**: Tokenized text data and aligned NER tags with tokens.
  - **Model Configuration**: Unfreezing additional layers of the BERT model for fine-tuning.
  - **Training Parameters**: Learning rate of 3e-5, 10 epochs, and batch size of 8.

2- Hyperparameter Tuning

- **Goal**
  - Optimize model performance through hyperparameter tuning.
- **Methodology**
  - Hyperparameters Adjusted: Learning rate, batch size, weight decay.
  - Scheduler: Polynomial decay learning rate scheduler.
- **Conclusion**
  - Adjusting hyperparameters further improved model performance, particularly in handling different entity types.

3- Evaluation and Comparison

- **Goal**
  - Evaluate the model on test data and compare results with existing benchmarks.
- **Methodology**
  - **Metrics Used**: Precision, recall, F1 score, and accuracy.
- **Results**
  - Test Accuracy: **68%**
- **Conclusion**
  - The model's performance was competitive with existing benchmarks, demonstrating its effectiveness in NER.

4- Performance Issues

It is important to note that the performance of the model was impacted by resource limitations. Due to the constraints of the free plan on Google Colab, which only provides limited GPU resources, the training process was constrained. As a result, the model did not achieve the expected performance levels and was not fully optimized.

5- Overall Conclusion

The project successfully developed a BERT-based NER model that performs reasonably well on the CoNLL-2003 dataset. Despite resource constraints affecting the final performance, the model showed improvement through fine-tuning and hyperparameter adjustments. The model's performance was competitive with existing benchmarks, demonstrating its potential for effective NER tasks.

6- Additional Requirements

- **Tools and Libraries**
  - **Libraries**
    - pandas
    - numpy
    - datasets
    - transformers
    - tensorflow
    - spacy
    - matplotlib
  - **Tools**
    - Colab
    - TensorFlow
    - Hugging Face Transformers
    - spaCy
- **External Resources and Pre-trained Models**
  - **Pre-trained Models**: BERT-Base, Cased from Hugging Face Transformers
  - **External Resources**: CoNLL-2003 dataset

8- Reflection Questions

2. What was the biggest challenge you faced in implementing Named Entity Recognition?

The biggest challenge was the resource limitations encountered during the project. These constraints affected the training process and overall model performance. The limited availability of GPU resources on the free plan of Google Colab restricted the ability to fine-tune the model effectively and perform extensive hyperparameter tuning, which are crucial steps for achieving high accuracy in Named Entity Recognition tasks.

2. What insights did you gain about NLP and NER through this project?

This project offered valuable insights into the complexities of token classification and the need to fine-tune pre-trained models for specific datasets. It showcased the effectiveness of BERT-based models in managing contextual information for NER tasks and emphasized the influence of computational resources on model performance. Additionally, it explored various techniques to enhance NER effectiveness and power by reviewing Kaggle notebooks, articles, and research papers.

9- Recourses

- **Gemini** in Colab helps me resolve errors during project development.
- **ChatGPT** assists me with documentation writing, grammar, and clear explanations of information.
- https://huggingface.co/datasets/eriktks/conll2003
- I made developments and enhancements in this notebook.
    - https://www.kaggle.com/code/alfathterry/named-entity-recognition-bert/input