Fine-Tuning DistilBERT for POS Tagging on the Universal Dependencies Dataset

Introduction

This report provides an overview of the process and results for fine-tuning the DistilBERT model for Part-of-Speech (POS) tagging using the Universal Dependencies (UD) dataset. The focus is on the Urdu language subset of the UD dataset, detailing the data preparation, tokenization, model configuration, and training results.

Dataset Information

The Universal Dependencies (UD) dataset is a widely used resource for training and evaluating NLP models, particularly for tasks like Part-of-Speech (POS) tagging, dependency parsing, and more. It provides consistent annotations across multiple languages using a common set of linguistic annotations.

Official Link: https://universaldependencies.org/

Huggingface dataset link: https://huggingface.co/datasets/universal-dependencies/universal_dependencies

Dataset Preparation

The Universal Dependencies dataset for Urdu was utilized for this task. To manage the size of the dataset and expedite the training process, a random subset of 100 samples was selected from each dataset split (train, validation, and test). This subset selection was performed to ensure that the training and evaluation phases were conducted with a manageable number of examples.

Tokenization and Label Alignment

The DistilBERT tokenizer was employed to preprocess the dataset. The tokenization process involved converting the list of tokens into token IDs while ensuring that the labels align correctly with the tokenized inputs. Special attention was given to handling cases where tokens were split into subword tokens during the tokenization process. Each token in the input text was assigned a corresponding label, and special tokens were given a label of -100 to differentiate them from actual tokens.

Model and Training Configuration

Three different fine-tuning approaches were implemented to assess their impact on model performance:

1. **Full Fine-Tuning**: The entire DistilBERT model was fine-tuned for 10 epochs. This approach adjusts all layers of the model to better fit the POS tagging task.
2. **Frozen Embedding Layers**: The embedding layers of the DistilBERT model were frozen, and only the remaining layers were fine-tuned for 20 epochs. This strategy aims to retain the pre-trained knowledge in the embedding layers while adjusting the higher layers for the specific task.
3. **Frozen Embedding Layers and First 3 Transformer Layers**: The embedding layers and the first three transformer layers were frozen, with only the remaining layers fine-tuned for 20 epochs. This approach further restricts the number of layers being adjusted, focusing on the last layers for task-specific learning.

The training configurations were as follows:

- **Output Directory**: Results were saved in the specified output directory.
- **Evaluation Strategy**: Evaluation was performed at the end of each epoch.
- **Learning Rate**: Set to 2e-5.
- **Batch Size**: A batch size of 8 was used for both training and evaluation.
- **Number of Epochs**: 10 epochs for full fine-tuning and 20 epochs for the other two approaches.
- **Weight Decay**: Applied with a strength of 0.01.

Results

The results from the different fine-tuning strategies are as follows:

- **Full Fine-Tuning**: Achieved a test set accuracy of 0.6477 after 10 epochs.
- **Frozen Embedding Layers**: With the embedding layers frozen and fine-tuning for 20 epochs, the model attained a test set accuracy of 0.6913.
- **Frozen Embedding Layers and First 3 Transformer Layers**: This model, with the embedding layers and the first three transformer layers frozen, achieved a test set accuracy of 0.6434 after 20 epochs.

Analysis of Layer Impact on POS Tagging

Part-of-Speech (POS) tagging primarily relies on syntactic features of language. The early layers of a model, such as DistilBERT, are known to capture these syntactic features effectively. This is because these initial layers are adept at identifying and understanding grammatical structures and patterns.

Given that DistilBERT has been pretrained on a vast corpus of text data, it already possesses substantial knowledge about syntactic relationships and structures. This pretrained knowledge makes it feasible to fine-tune only the last few layers of the model for specific tasks, like POS tagging, rather than retraining the entire network.

Resources Utilized

1. ChatGPT
2. Huggingface Bert for Token = Classificationhttps://colab.research.google.com/github/huggingface/notebooks/blob/main/examples/token_classification.ipynb
3. Huggingface Bert for Token (Tensorflow) = https://colab.research.google.com/github/huggingface/notebooks/blob/main/examples/token_classification-tf.ipynb
4. Huggingface Bert = https://huggingface.co/docs/transformers/en/model_doc/bert