ELL-881 - Assignment-3

Name: Guruvu Surya Sai Prakash

Entry No: 2019EE10481

Data Pre-processing:

We first tokenize the words to convert them into smaller units (words, sub words or characters). For this we use **BertTokenizerFast** class from Hugging Face Transformers library which provides a fast implementation of the BERT tokenizer. The BERT tokenizer is specifically designed to work with the BERT model architecture and is used to convert raw text into numerical representations that can be fed into the BERT model for processing. The BertTokenizerFast class is optimized for speed and memory efficiency and is typically used when processing large amount of text.

While tokenizing the text we set the **truncation=True** (This parameter specifies that the input sequences should be truncated to a maximum length if they exceed a certain threshold) and **padding=True** (This parameter specifies that the input sequences should be padded with special tokens to ensure that they all have the same length).

The output of the tokenizer is a dictionary which has **input_ids** (This is a list of token IDs that represent the input sequences) and **attention masks** (This is a list of binary values that indicates which tokens in the input sequence are real tokens (i.e., not padding tokens) and should be attended by the model.

Now we label the sequences. For this we label all the pad tokens, [CLS] and [SEP] tokens to -100, this is to indicate that these tokens should be ignored during the training process and should not be used to calculate the loss or gradient.

Now while labelling the words of the sequences, if the word from which the current sub word is created has O-label or (I-tag)-label, then the sub word is assigned the same label as the original word. If the word as (B-tag)-label, then the first sub word is given (B-tag)-label and the sub words are given (I-tag)-label.

Model Training:

Used Hugging Transformers library that creates a pre-trained BERT model for token classification tasks. BertTokenClassification(bert-base-cased) model was used. This is a specific type of BERT model that has been fine-tuned for token classification tasks, such as Named Entity Recognition (NER) or part-of-speech tagging. It consists of a BERT encoder, which processes the input text and extracts features, and a linear layer on top that performs token classification based on the extracted features.

For training the model used **CrossEntropyLoss**, **Adam Optimizer with weight_decay=1e-5** (tuned) and **Linear Scheduler**.

Results:

Training Results with O-label:
Training weighted F1 Score: 0.9994667784120882
Training avg F1 Score: 0.9945612454345278
Training Accuracy: 99.94669477740288 %

Training Results without O-label:
Training weighted F1 Score: 0.9979346372720741
Training avg F1 Score: 0.96471498843254
Training Accuracy: 99.76291489892688 %

Validation Results with O-label:
Validation weighted F1 Score: 0.9715550956984208
Validation avg F1 Score: 0.8245294558156566
Validation Accuracy: 97.16287048508529 %

Validation Results without O-label:
Validation weighted F1 Score: 0.8734116359260646
Valdiation avg F1 Score: 0.7470327597213384
Validation Accuracy: 85.81085233208383 %

Testing Results with O-label:
Testing weighted F1 Score: 0.9725340305987908
Testing avg F1 Score: 0.8165717364698222
Testing Accuracy: 97.27550867182221 %

Testing Results without O-label:
Testing weighted F1 Score: 0.874528633228816
Testing avg F1 Score: 0.7381831084189785
Testing Accuracy: 85.89234776137914 %

We observe that the test accuracy (without label) is 85.89%, weighted F1 score = 0.874, average F1 = 0.738.

The results of the experiments demonstrate that fine-tuning the BERT model for the NER task can achieve high accuracy and F1-score on the test set. The values were good compared to the LSTM model used in Assignment-2.

One of the key advantages of using BERT for NER is its ability to capture contextual information and dependencies between tokens in the input sequence. This is especially important for NER, as named entities often consist of multiple words and their boundaries may depend on the surrounding context. The attention mechanism of BERT enables the model to attend to relevant tokens in the input and capture their interactions, which improves its ability to identify named entities accurately.

But we can still observe that the average F1 score is low, this because we have few labels which are rare (some might not be present in Training set but might be present in test set) and these are not being learnt.