**BERT Classification Approach**

Identified there are many languages other that English. So used Polyglot to detect language. Polyglot package detected 37 languages in “Description” & “Short Description” fields. Later, converted the Non-English like “Description” & “Short Description” to English using Googletrans package. There were some restrictions in this translate API that limited the number of translation calls. So split the file into 3, ran the translation by 3 members and later combined the translated and English like files into 1.

BERT - **BERT (Bidirectional Encoder Representations from Transformers)** is a recent [paper](https://arxiv.org/pdf/1810.04805.pdf" \t "_blank) published by researchers at Google AI Language.

BERT makes use of Transformer, an attention mechanism that learns contextual relations between words (or sub-words) in a text. In its vanilla form, Transformer includes two separate mechanisms — an encoder that reads the text input and a decoder that produces a prediction for the task. Since BERT’s goal is to generate a language model, only the encoder mechanism is necessary. The detailed workings of Transformer are described in a [paper](https://arxiv.org/pdf/1706.03762.pdf" \t "_blank) by Google.

**BERT (Fine-tuning)**

BERT can be used for Classification Task by only adding a small layer to the core model: Classification tasks such as sentiment analysis are done similarly to Next Sentence classification, by adding a classification layer on top of the Transformer output for the [CLS] token.

Tried BERT Language Model using

1. Given not translated file & BERT Base Model uncased (**Validation Accuracy – 95%**)



1. Translated file & BERT Base Model uncased (Validation Accuracy – 68%)



1. Given not translated file & BERT Multi-Lingual Base Model Case (Validation Accuracy – 80%)



The **not translated file with & BERT Base Model uncased** gave a better accuracy.

no\_upsampling\_grp = ['GRP\_0', 'GRP\_8', 'GRP\_24', 'GRP\_12', 'GRP\_9']

Upsample minority classes to 250 counts

tokenizer = BertTokenizer.from\_pretrained('bert-base-uncased',

do\_lower\_case=True)

encoded\_data\_train = tokenizer.batch\_encode\_plus(

df[df.data\_type=='train'].CD.values,

add\_special\_tokens=True,

return\_attention\_mask=True,

pad\_to\_max\_length=True,

max\_length=128,

return\_tensors='pt'

)

model = BertForSequenceClassification.from\_pretrained("bert-base-uncased",

num\_labels=len(label\_dict),

output\_attentions=False,

output\_hidden\_states=False)

GPU batch\_size = 3

optimizer = AdamW(model.parameters(),

lr=1e-5,

eps=1e-8)

epochs = 5

Final Epoch Results:

6428/6428 [11:59<00:00, 8.94it/s, training\_loss=0.001]

Epoch 4 Training loss: 0.09553318860604121

Validation loss: 0.3230386202712979

F1 Score (Weighted): 0.9449826509911491

Accuracy: 0.9462555066079296

Final Validation Results BERT – not translated

Accuracy: 0.95

Validation Loss: 0.32