

a) This model loads a pre-trained BERT encoder from HF transformers and fine-tunes it for text classification (learns a classification layer on top of the encoder). The first step is to tokenize and vectorize the data(texts) and to do this, we will again use previously created tokenizers through the simple AutoTokenizer class. The AutoTokenizer.from_pretrained function can load the tokenizer associated with any of the large number of models found in the Hugging Face(HF) models repository. We'll load the appropriate model according to the task and language tokenizer. For English, "bert-base-cased" model is loaded. We also truncate the tokenized data up to a max length if necessary. Then a model is built using a pretrained BERT encoder which takes in sequence as input and outputs labels by building a classification layer on top of the encoder.

AutoModel and ForSequenceClassification in transformers.AutoModelForSequenceClassification.from_pretrained() are important factors in this case.

```
import torch

import transformers

model_name = "bert-base-cased"

Model=transformers.AutoModelForSequenceClassification.from_pretrained(model_name, num_labels=2)
```

Hugging Face(HF) [AutoModel](#) is a generic model class that will be instantiated as one of the base model classes of the library when created with the AutoModel.from_pretrained(...). A pre-trained language model may be quickly adjusted for a text classification job using Transformers AutoModelForSequenceClassification. The pre-trained model is loaded after supplying the number of output labels and the model name (such as "bert-base-cased"). As it features a classification head that can be readily trained using the base model on top of the model outputs, AutoModelForSequenceClassification allows multi-label classification. Then the transformer model will be configured and trained like how it previously occurred with CNN and RNN.

b) This model has a way higher result than the CNN and RNN notebooks with an accuracy of 90.25%. But the CNN models had accuracy between 68-82% and the RNN model had accuracy between 50-72%.