

### Step by step summary for execution of CNN and RNN model:

1. Downloading and preparing data is the first stage. Data will be gathered, cleaned up, and maintained in the proper format for the model's training before being used to train the model. Tokenization and vectorization of data are required when working with words. Using the straightforward AutoTokenizer class, we leverage previously generated tokenizers to tokenize and vectorize the strings in our dataset. The datasets' training, testing, and validation parts must be explicitly divided, if not already done so by any built-in library function.
2. Then features either cannot be retrieved by the model or must be extracted manually.
3. Then an appropriate model architecture will be built based on the task and the number of outputs. For the notebooks, a CNN and RNN text classification model was made by setting up its appropriate parameters and hyperparameters like embedding dimension, number of filters which is the number of different convolution filters, embedding\_dim which is the size of the word (i.e. token) embeddings, filter sizes which is the size(length) of convolutional filters, padding, activation function, loss function, etc. These things are essential to configure and instantiate the model. Besides, other factors are vocab\_size which is always the vocabulary size of the tokenizer and num\_labels is the number of unique labels in the data (as here). Some other notable hyperparameters that might be necessary to adjust are learning\_rate which is the step size for weight updates per\_device\_train\_batch\_size which is the number of examples per batch, max\_steps which is the maximum number of steps to train. Also nonlinearity will be needed to be introduced. All these factors will be needed to build the model and the model will be trained for a number of epochs. The hyperparameters and model design vary but most steps remain the same across all models. Model hyperparameters are tuned during training for optimal performance.
4. The development of a metric for measuring performance both during and after training will then be required. One of the several pre-made metrics was loaded using load, and it was then wrapped for the trainer to utilize. Simple multi-class classification was the task at hand, therefore the basic accuracy metric was employed. It is defined as the percentage of properly predicted labels among all labels. The model outputs are compared with the labels to find the model performance.