## Part 3 – Trump Tweets classifier

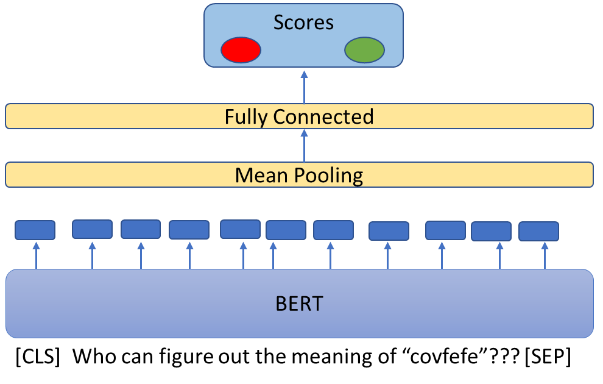
### General

In this part we were requested to create a supervised classifier that classifies Trump twitter tweets. The dataset contains tweets from two sources (iphone, Android) and the hypothesis is that the tweet original distribution is different between the sources. We decided to try two models on this problem: one that processes only text, and one that takes hand-crafted features into account as well. Each model was trained on a train set and tested on a validation set.

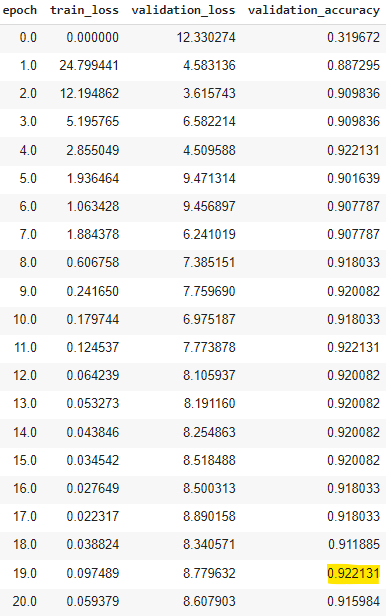
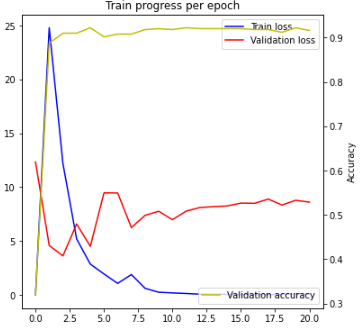
The selected model and configuration (Classifier2, trained for 14 epochs) was then trained on the entire dataset (train + validation) before running inference on the test set.

### Classifier 1 –Bert + custom FC layer, on tweet text only

This classifier passes the tweets text in a Bert model, passes the last hidden state through mean pooling to reduce one dimension, then uses an FC layer for classification. The model does not use any other data beside the text itself. Train / Validation performance can be seen below.



**Train / Validation performance:**

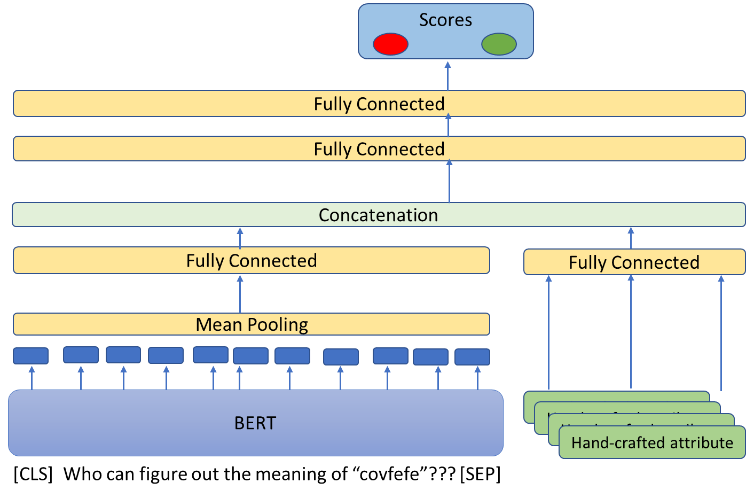
### Classifier 2 – Classification using the tweet text + hand crafted features. using Bert + FC layers

This is a more complex classifier, which gets both the tweets text and a list of hand-crafted attributes. The texts are passed in a BERT model, then mean pooling, and then an FC layer that keeps the same dimensions as the input, producing output (a). In parallel, hand featured attributes are passed through an FC layer, producing output (b).

Both inputs (a) and (b) are then concatenated to a single vector, the concatenated vector is passed through a 2-layered MLP to produce the class predictions. side is the same as Classifier1, but the passes the tweets text in a BERT model, passes the last hidden state through mean pooling to reduce one dimension, then uses an FC layer for classification.

**10 Hand-crafted Attributes (Scaled):**

* “Dumb retweet” – retweeted another tweet using copy/paste
* Percent capital letters in the tweet
* Contains URL - Boolean
* Number of hashtags in tweet
* Date/Time attributes: hour, minute, day, month, year, weekday



**Train / Validation performance:**

