# **Advanced Machine Learning - Assignment Report**

**Submitted by:**   
Miryam Horovicz - 313246704, Yaniv Tal - 031431166

## Notebook 1 Overview

Our submitted notebook consists of three sections:

1. **General section** – handles the logistics:
   1. Imports
   2. Mount Google drive
   3. Download data
   4. Preprocess data i.e. arrange full text, prepare data frame
2. **Task part 1 –** Comparison model training by LDA topic modelling
3. **Task part 2** – Comparison model Analysis and Evaluation

## Part 1 Comparison model training

Comparison model training consists of three sections:

1. **Preprocess data** – prepare data for topic modelling:
2. Tokenizing the text
3. Removing stop words
4. Convert the tokenized object into a corpus and dictionary by bag of words method.
5. **Topic model training -** train LDA model applying following architecture

Diagram

Description automatically generated

Trained model results can be shown by words and probabilities in each topic as Shown below

Text

Description automatically generated

1. **Similarity function -** function that get two documents and return similarity measure between them:
2. Apply bag of words for each document
3. Apply the trained LDA model and get representation vector
4. Return cosine similarity between the two vectors.

## Part 2 – Comparison model Analysis and Evaluation

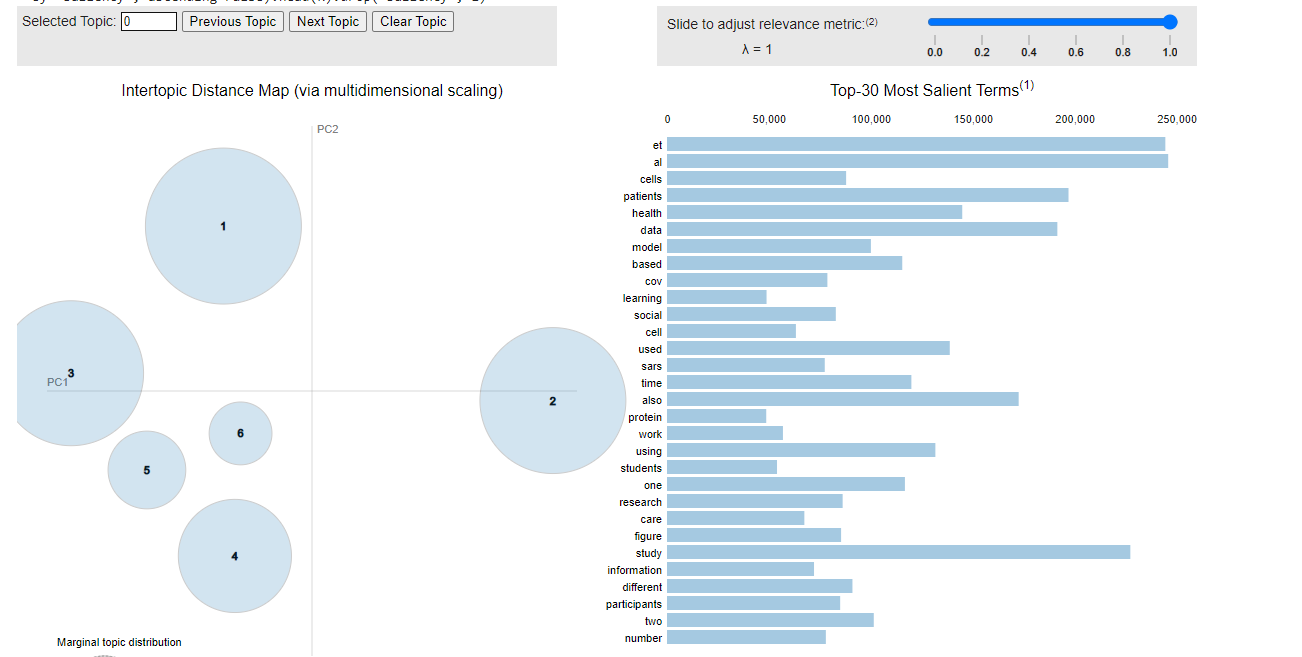
For LDA model analysis and evaluation we used two techniques:

1. **Word Cloud** – Show most frequent words in each topic, bigger font means higher word probability in this topic:

Qr code

Description automatically generated with medium confidence

1. **Interactive Visualization -** interactive visualization in the notebook that shows topics views. User can navigate between the different topics and see the words distribution.



## Part 3 – Trump Tweets classifier

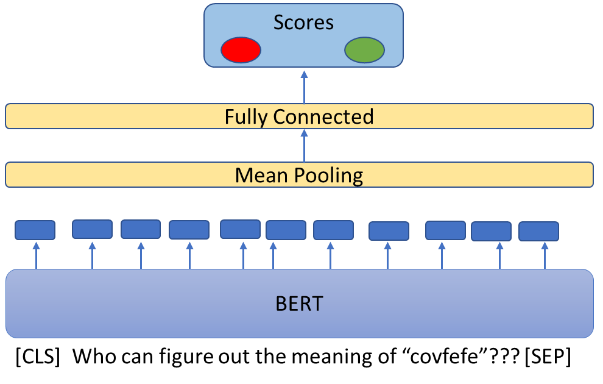
### General

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 with only a language model (BERT), 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.



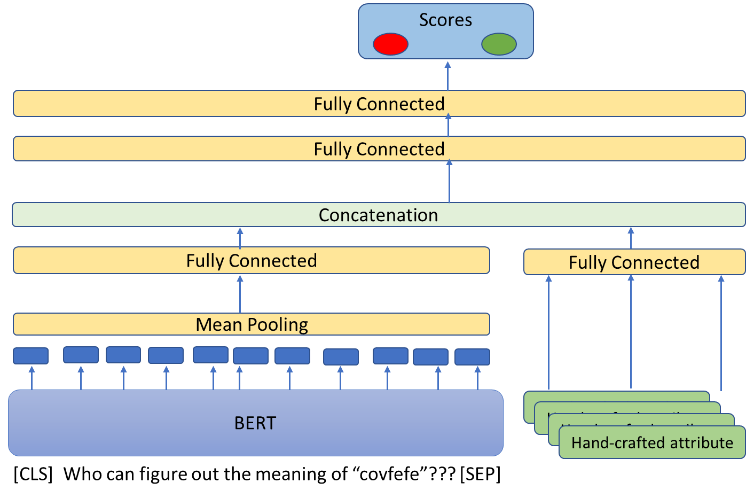
**Train / Validation performance:**

## 

### 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:**

