

Advanced Machine Learning - Assignment Report

Submitted by:

Miryam Horovicz - 313246704, Yaniv Tal - 031431166

Notebook 1 Overview

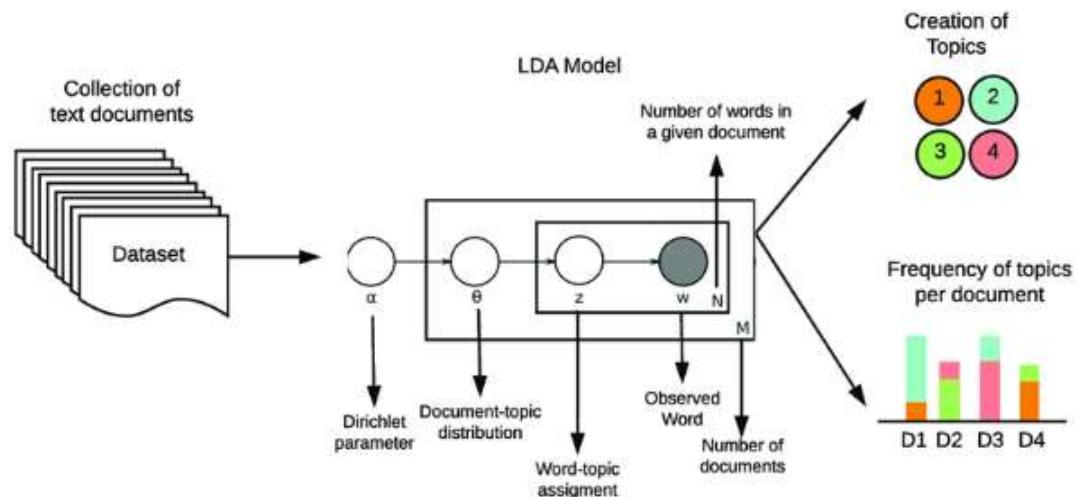
Our submitted notebook consists of three sections:

1. **General section** – handles the logistics:
 - a. Imports
 - b. Mount Google drive
 - c. Download data
 - d. 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:
 - a. Tokenizing the text
 - b. Removing stop words
 - c. Convert the tokenized object into a corpus and dictionary by bag of words method.
2. **Topic model training** - train LDA model applying following architecture



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

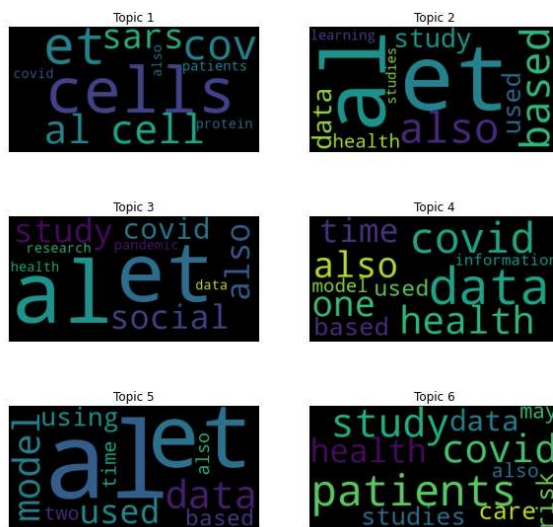
```
[
(0,
'0.007*cells" + 0.005*et" + 0.005*cov" + 0.005*cell" + 0.004*al" + '
'0.004*sars" + 0.004*patients" + 0.004*also" + 0.004*covid" + '
'0.004*protein'),
(1,
'0.008*et" + 0.007*al" + 0.004*also" + 0.004*based" + 0.003*study" + '
'0.003*data" + 0.003*used" + 0.003*health" + 0.003*learning" + '
'0.003*studies'),
(2,
'0.009*al" + 0.008*et" + 0.005*study" + 0.005*social" + 0.004*also" + '
'0.004*covid" + 0.004*pandemic" + 0.003*health" + 0.003*research" + '
'0.003*data'),
(3,
'0.006*data" + 0.004*covid" + 0.004*health" + 0.003*also" + 0.003*one" + '
'0.003*time" + 0.003*based" + 0.003*used" + 0.003*model" + '
'0.003*information'),
(4,
'0.009*al" + 0.009*et" + 0.007*data" + 0.006*model" + 0.005*used" + '
'0.005*using" + 0.004*based" + 0.004*time" + 0.003*two" + 0.003*also'),
(5,
'0.011*patients" + 0.009*study" + 0.009*covid" + 0.007*health" + '
'0.005*data" + 0.004*studies" + 0.004*care" + 0.004*risk" + 0.004*may" + '
'+ 0.003*also')]
```

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

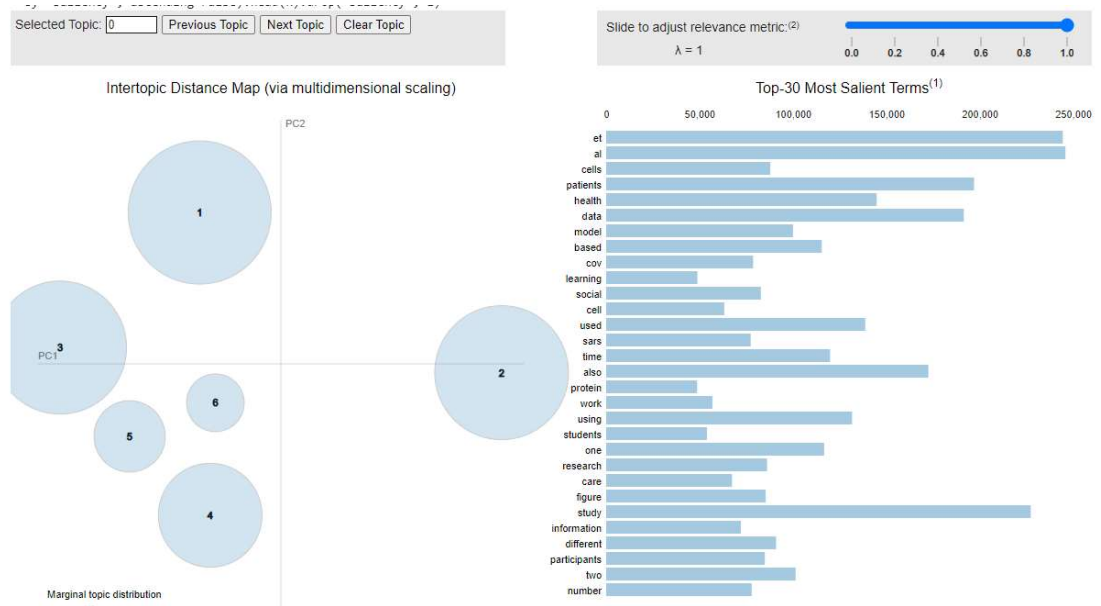
Part 2 – Comparison model Analysis and Evaluation

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:



2. **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

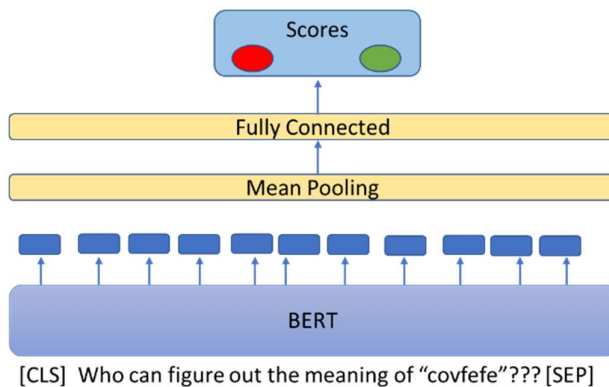
General

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.

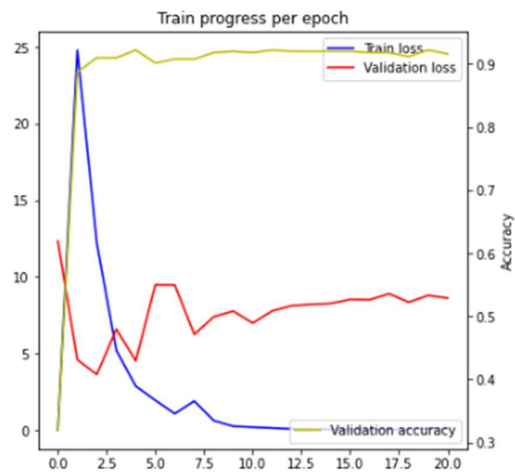
Classifier 1 – Bert + custom FC layer, on tweet text only

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:

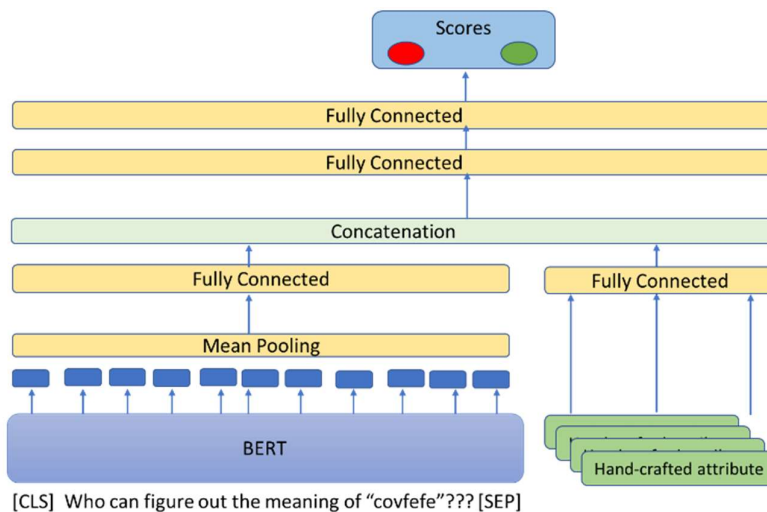
epoch	train_loss	validation_loss	validation_accuracy
0.0	0.000000	12.330274	0.319672
1.0	24.799441	4.583136	0.887295
2.0	12.194862	3.615743	0.909836
3.0	5.195765	6.582214	0.909836
4.0	2.855049	4.509588	0.922131
5.0	1.936464	9.471314	0.901639
6.0	1.063428	9.456897	0.907787
7.0	1.884378	6.241019	0.907787
8.0	0.606758	7.385151	0.918033
9.0	0.241650	7.759690	0.920082
10.0	0.179744	6.975187	0.918033
11.0	0.124537	7.773878	0.922131
12.0	0.064239	8.105937	0.920082
13.0	0.053273	8.191160	0.920082
14.0	0.043846	8.254863	0.920082
15.0	0.034542	8.518488	0.920082
16.0	0.027649	8.500313	0.918033
17.0	0.022317	8.890158	0.918033
18.0	0.038824	8.340571	0.911885
19.0	0.097489	8.779632	0.922131
20.0	0.059379	8.607903	0.915984



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:

epoch	train_loss	validation_loss	validation_accuracy
0.0	0.000000	288.954240	0.686475
1.0	788.628972	29.943416	0.879098
2.0	440.409676	28.899772	0.905738
3.0	289.939790	43.776315	0.897541
4.0	201.878389	26.146926	0.872951
5.0	151.746915	30.934257	0.934426
6.0	126.809202	27.591140	0.922131
7.0	74.091448	35.310097	0.922131
8.0	60.756876	39.713520	0.903689
9.0	58.525434	57.076277	0.924180
10.0	40.620630	53.952453	0.918033
11.0	36.786599	45.682475	0.932377
12.0	43.488554	48.634012	0.932377
13.0	32.806340	44.237678	0.934426
14.0	30.804467	49.742612	0.940574
15.0	25.813664	63.996339	0.932377
16.0	24.701498	62.814026	0.911885
17.0	9.002538	59.220427	0.926230
18.0	13.739529	63.676617	0.905738
19.0	12.084631	73.477135	0.905738
20.0	4.839909	86.119912	0.895492

