Delivery 1

Quora Objective:

Make a basic model to solve the Quora challenge https://www.kaggle.com/c/quora-question-pairs

The deliverable should contain a simple solution and a 'improved solution'.

- Try a simple solution
  - What problems/limitations do you think the model has?
  - What type of errors do you get ?
  - What type of features can you build to improve the basic naive solution?
- Improve your simple solution:
  - Construct features for each input and use them to compute distances between the inputs.
  - Investigate and code a feature vector or a distance between two strings. Used you implementation to define a feature used to capture the similarity between two documents.
    - Implement from scratch the feature vector or the distance function for two input documents
    - Split implementations between members in the group (do not code the same thing twice).
    - Explain the implemented code in `main.pdf`.

Format and delivery rules

- The project can be done in groups up to 3 people.
- The project has to be sent by email by a single member of each group (in case it is too heavy send a link to dropbox or similar) to davidbuchaca@ub.edu
  - The Date for the project is monday 12 April 12 pm.
- The project needs to be in a zip file containing all the code to reproduce the results.
- The zip file has to be self contained and with the following form: name1_name2_name3.zip Where name1,name2,... are the names of the members of each group. Please write your full name in CamelCase form.
  - If Elisenda Grau and John Snow make a team the zip filename has to be `ElisendaGrauJohnSnow.zip`

Deliverable format

- The Zip DOES NOT have to include train or test data
- The Zip file **must contain**:
    - `main.pdf`: A description of your work
    - `train_models.ipynb`
        - Notebook with the code needed to train and store models to disc.
        - This Notebook has to be clean (do not define functions here, do them in an external `utils.py` and import them).
        - This notebook has to be reproducible (if you run it twice, the same output has to be displayed and stored to disk).
    - `reproduce_results.ipynb`
- The zip file **can contain (OPTIONAL)**
    - A notebook `utils_name_k.ipynb` containing the functions from `utils.py` that person `name_k` wrote.
    - This can be used to show/explain the usage of the functions in `utils.py`
    - Only person `name_k` should write `utils_name_k.ipynb` .
- The code done for each member in a group has to be explained by its corresponding author of the code. Each person in the group should build at least one feature vector or distance for input documents and should take the responsibility for explaining it in `main.pdf`.

Notes on `train_models.ipynb`

This notebook should...

- be done between all memebrs of a group
- use the code done by each member in the group to generate features for the challange.

This is a Kaggle challange: There is no validation/test data with labels.

Therefore you have to create the following split in order to share the same train validation and test splits across teams:

```
train_df = pd.read_csv(os.path.join(path_folder_quora, "quora_train_data.csv"))

A_df, te_df = sklearn.model_selection.train_test_split(train_df,
                                                       test_size=0.05,
                                                       random_state=123)

tr_df, va_df = sklearn.model_selection.train_test_split(A_df,
                                                        test_size=0.05,
                                                        random_state=123)

print('tr_df.shape=',tr_df.shape)
print('va_df.shape=',va_df.shape)
print('te_df.shape=',te_df.shape)
```

```
tr_df.shape= (291897, 6)
va_df.shape= (15363, 6)
te_df.shape= (16172, 6)
```

Notes on `Reproducible_results.ipynb`

- If there are `random` parts in the code, make sure to have seeds to make your results reproducible.

This notebook should...

- be done between all memebers of a group.
- contain TRAIN, VALIDATION results of ROC AUC (`sklearn.metrics.roc_auc_score`).
- Optional: TEST results can be obtained sending results to Kaggle

Note that I will only load and run this notebook.

- This notebook does not have to train anything.
- It should be relatively fast to execute (probably less than 10 minutes since there is no training).
- This notebook should only load from disk trained models, make predictions and compute metrics.