Exercise 4 – NLP models

Part 1 – comparison model training

The world is changing as we speak. The COVID-19 epidemic has been causing dramatic changes to how we live our lives. In this exercise, we will focus on data related to the COVID-19 and will build unsupervised machine learning models.

The data that you will play with is an extensive corpus of academic papers, all dealing with COVID-19. The data is part of a worldwide research challenge provided by Kaggle and is well described here. It is the same dataset used for HW3!

Your mission is divided into two parts:

Design and implement **comparison capability between academic papers**. The comparison capability should be based on natural-language-processing tools and algorithms (as opposed to the compression method we have used in HW3).

Your comparison model should rely upon a **representing** schema of each document. You can use the following **representation** logics (but not limited only to those):

a. Bag of words
b. LDA (Topic modeling)
c. Word embeddings (e.g., Glove, Word2Vec)
d. Transformers embeddings

You are required to experiment with **at least** one of the representations.

Your model should have the ability to get as input two documents (Covid19 papers in our case) and return a similarity measure between the two.

Indeed a learning phase is required to develop such capability, but note that you do not have a tagged corpus.

Part 2 – comparison model analysis and evaluation

a. Analyze the representation of your method. The analysis depends on the method you chose to implement your comparison model. For example – in case you chose the LDA algorithm as the representation schema - a distribution analysis of the different topics is an interesting and relevant analysis.
b. Suggest a (suitable!) evaluation method to evaluate your model's performance from part 1 and report the results you obtained.

Feel free to add any analysis and bottom-line conclusions you find along the way.

Comments:

1. As in HW3 - you are expected to use only the latest 20K papers from the corpus.
2. Think about data preprocessing. Each algorithm approach might require a different data preprocessing approach.

3. Part 3 – classification algorithm

Politicians and other public figures usually have assistants and staffers that manage most of their social media presence. However, like many other norm-defying actions, Donald Trump, the former President of the United States, is taking pride in his untamed use of Twitter. At times during the presidential campaign in 2016, it was hypothesized that Donald Trump was being kept away from his Twitter account in order to avoid unnecessary PR calamities. Trump's tweets are not explicitly labeled (Hilary Clinton, for example, used to sign tweets composed by her by an addition of '-H' at the end of the tweet while unsigned tweets were posted by her staffers). It is known, however, that Trump was using an android phone while the staffers were most likely to use an iPhone. Luckily, the device information is part of the data available via the Twitter API, hence it can be used as an authorship label.

In this part of the HW, you are required to try several **supervised machine learning classifiers** in order to validate the hypothesis about Trump's tweeting habits.
You may use any algorithm you wish in order to build your classifier.

- Data - you can find a .tsv file ("train.tsv") in the course website with the content of the tweets + the label (device type). The data contains 3K tweets
- One day before the submission deadline, we will publish the test dataset (~850 tweets), without the label and will ask you to submit your prediction for this dataset. Your prediction should be binary ('iphone' or 'android').
- Please follow **the exact guidelines** we will publish regarding the submitted predictions.
- You must submit the code you designed and used for the actual predictions.
- We ask you not to be sophisticated with the way you come up with final predictions over the test set. We assume data leakage of the test dataset exists over the web - hence, your predictions should be based on a ML model you built. Methods like crawling Twitter datasets for finding the test instances are not legitimate.

Submission instructions

- The submission must contain your code (written in Python 3).
- You should submit 3 files:
    - Two jupyter notebooks (the first one for part 1 and 2, the second one for part 3). You can use additional .py files for the code to run (including some of the utilities you built). We recommend you to use the google collab platform and share with us the final product of your work.
      The code must be documented in a reasonable way and contain explanations throughout the notebooks regarding the process you implemented. If you use special Python packages (on top of what Anaconda provides) – please document it well.

    - A short PDF document (**up to 6 pages**) describing the work you did (over all parts of the HW), answers to the questions, and central insights/conclusions. If space (6 pages) is limited due to a large number of figures/tables – use the notebook to document all work and present only the highlights in the submitted PDF.

- The exercise has to be done in pairs. **Only one** of the students has to submit the HW. **Make sure to write your full names and IDs in the header of the PDF file**.
- Submission due date: by 19.5.2022, 23:30.