HSMA 6 Session 5B Exercise 3

You should work as a group to undertake this exercise. Try to get through as many of the tasks as possible within the allotted time. You should use the supplied *nlp_ner* virtual environment for this exercise.

In this exercise, you are going to perform some Named Entity Recognition on a public dataset – a million news headlines published over a period of 19 years, sourced from the Australian Broadcasting Corporation (ABC) : A Million News Headlines (kaggle.com). The data has been supplied in .csv format with two columns – the first containing the publication date, and the second containing the text from the headline. The text has all been made lowercase – this is not ideal for Named Entity Recognition, as capitalisation can provide clues as to whether something is a Named Entity or not, so part of this exercise will be exploring how well pre-trained SpaCy models can predict named entities on such data.

The dataset has been provided to you as part of the GitHub repository for this session. Please extract the zip file (headline_archive) and ensure the extracted file (abcnews-date-text.csv) is located in the same directory as the .py file you will create here before beginning this exercise.

As a group, write code as follows :

1. Imports of SpaCy, at least one SpaCy model, displaCy, the csv library, and the random library. It is also recommended that you import tqdm – a library that allows the use of progress bars, and which comes with SpaCy. The import statement required is : `from tqdm import tqdm, trange`
2. You need to read in the headlines from the .csv file and store them in a list of headlines. Remember that there are two columns (only one of which contains the headlines), and the first row contains column headers, which are not required. You may find it useful to refresh your memory on the materials on reading in information from .csv files in session 1F.
3. Load in your chosen SpaCy pre-trained model
4. Shuffle the list of headlines you've created using `random.shuffle()`. As the list of headlines is huge (a million) it will likely be impractical for you to analyse all of them on your computers in a reasonable time for this exercise. Instead, you are going to apply the pre-trained SpaCy model to a proportion of them (and if you shuffle the list first, this will be a random selection). Tip – think a *very small proportion*; Dan has a very fast PC, and he chose 1% of the dataset (which takes about 30 seconds on his computer). Try different proportions of the dataset depending on the speed of your computer. Here's what you need to do : create an empty list that will store your doc objects, and then loop through the chosen proportion of headlines, applying the pre-trained SpaCy model to each, and storing the resultant doc object for that headline in your list of doc objects. *It is strongly recommended you use a progress bar to help identify a workable proportion of the data to use for your computer. To do this using tqdm, simply replace the word* `range` *with the word* `trange` *in a for loop.*
5. Randomly sample 100 of your doc objects for visualisation using displaCy to display the predicted named entities. You can use `random.sample(l, x)` to randomly sample x items from a list l without replacement (ie you pick all unique items). Apply the displaCy entity visualiser to each of your 100 doc objects. Run your code up to this point and look through the visualisations – how well has the model performed? Do you notice any performance improvement using the medium or large SpaCy en_core_web models?

6. **This task is slightly more advanced**. You need to create a dictionary where the keys are the different named entity labels (types) and the value for each key is a list of the predicted named entities of that type across all of the headlines that you analysed (ie the list of doc objects you created from applying the SpaCy model). You can approach this however you like, but here is one approach :
    a. Write a function that will assemble a dictionary entry as described above, given three inputs – a dictionary in which to put the entry, the label (NER type) for which you're creating an entry (and which will become the key for this entry), and a list of doc objects. The function should look at each predicted named entity in each doc object, and add it to a list if the named entity is predicted to match the label supplied. Then that list should be added against the key to form a dictionary entry in the supplied dictionary, and the dictionary passed back.
    b. Create an empty dictionary to store your named entities
    c. Create a list of the named entity labels you want to look at. To save you typing these out, you can use the code in the supplied spacy_ner_types_for_ex3.py file, which contains code setting up a list of all the default named entity labels in SpaCy.
    d. For each label in your list, apply the function you wrote above to create a dictionary entry
    e. Once you've assembled your dictionary, print out one of the dictionary entries to see all of the named entities across your sample of headlines that are predicted to have a certain label (e.g. "LOC")