In [4]:

```
#library needed for text classifier training
import spacy
from spacy.util import minibatch, compounding

#library to load and process data
import os
import copy
import re
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.utils import shuffle
from __future__ import unicode_literals
from operator import itemgetter


#to save trained model
from pathlib import Path
```

In [5]:

```
def clean_string(mystring):
    return re.sub('[^A-Za-z\ 0-9 ]+', '', mystring)



def main(model=None, output_dir=None, n_iter=3):
    if model is not None:
        nlp = spacy.load(model)  # load existing spaCy model
        print("Loaded model '%s'" % model)
    else:
        nlp = spacy.blank('en')  # create blank Language class
        print("Created blank 'en' model")

    # add the text classifier to the pipeline if it doesn't exist
    # nlp.create_pipe works for built-ins that are registered with spaCy
    if 'textcat' not in nlp.pipe_names:
        textcat = nlp.create_pipe('textcat')
        nlp.add_pipe(textcat, last=True)
    # otherwise, get it, so we can add labels to it
    else:
        textcat = nlp.get_pipe('textcat')

    # add label to text classifier
    for i in ['0','1','2','3','4','5','6','7','8','9']:
        textcat.add_label(i)


    os.chdir('/home/bala/Documents/Hackathon/MachineHack/spaCyModel2')
    temp_1 = open("/media/sf_ForVirtualBox/Hackathon/MachineHack/WhoseLineIsItAnywayTRAIN.csv", 'r', encoding='la
    df = pd.read_csv(temp_1)
    df = df[['text', 'author']]
    df = df.dropna()
    df['text'].replace(r'\s+', ' ', regex=True, inplace=True)

    author_values = df['author'].unique()
    labels_default = dict((v, 0) for v in author_values)

    df_train, df_val = train_test_split(df, test_size=0.05)
    df_train = df_train.reset_index(drop=True)
    df_val = df_val.reset_index(drop=True)
    df_val.to_csv("WhoseLineIsItAnywayTRAIN_val.csv")# execute validations from the predictions file
```

```
In [ ]:
```