# Bert modeller

entity\_list\_file="D:\\Sola\\DEV\\SC\\NewsCrawler\\bert\_modeller\\data\\entity\_list.csv"

training\_file="D:\\Sola\\DEV\\SC\\NewsCrawler\\bert\_modeller\\data\\training\_data\_1\_0\_3000.csv"

model\_files\_output\_folder="D:\\Sola\\DEV\\SC\\NewsCrawler\\bert\_modeller\\models"

evaluation\_file="D:\\Sola\\DEV\\SC\\NewsCrawler\\bert\_modeller\\data\\evaluation\_data.csv"

evaluation\_output\_file="D:\\Sola\\DEV\\SC\\NewsCrawler\\bert\_modeller\\data\\evaluation\_result.csv"

# import packages

import numpy as np

import pandas as pd

# optional text processing

from utils.text\_processing import text\_processing

# model training

from simpletransformers.classification import ClassificationModel, ClassificationArgs

from sklearn.metrics import classification\_report

from scipy.special import softmax

import nltk

#nltk.download('wordnet')

#nltk.download('stopwords')

#Prepare file for training

df = pd.read\_csv(training\_file, header=0)

data = {}

df = pd.DataFrame(df[['text', 'label']])

df = df.rename({'label': 'labels'}, axis=1)

df['text'] = df.apply(lambda x: text\_processing(x.text,

                lower=True,

                remove\_url=True,

                remove\_punctuation=True,

                remove\_stopwords=False,

                replace\_entity=True,

                replace\_hash=True,

                split\_alphanumeric=False,

                lemmatize=True,

                stem=True), axis=1)

print("Check null: ", df.isnull().sum().sum())

print(df.head(3))

# initialise Model

model\_args = ClassificationArgs(num\_train\_epochs=2, learning\_rate = 5e-5, \

                                output\_dir=model\_files\_output\_folder,overwrite\_output\_dir= True

                                )

model = ClassificationModel(args = model\_args, use\_cuda = False,model\_type = 'roberta', model\_name = 'roberta-base')

# train the model

model.train\_model(df)

#Prepare file for evaluation

df = pd.read\_csv(evaluation\_file, header=0)

data = {}

df = pd.DataFrame(df[['text', 'label']])

df = df.rename({'label': 'labels'}, axis=1)

df['text'] = df.apply(lambda x: text\_processing(x.text,

                lower=True,

                remove\_url=True,

                remove\_punctuation=True,

                remove\_stopwords=False,

                replace\_entity=True,

                replace\_hash=True,

                split\_alphanumeric=False,

                lemmatize=True,

                stem=True), axis=1)

print("Check null: ", df.isnull().sum().sum())

display(df.head(3))

#Evaluate the model

results = pd.DataFrame(columns = ['test\_set', 'precision', 'recall', 'f1'])

# get predictions

pred, raw\_outputs = model.predict(df['text'].tolist())

# append prediction (0 or 1) and probability (prob) to original dataframe

df['pred'] = pred

for i in range(len(df)):

    df.loc[i, 'raw\_output\_0'] = raw\_outputs[i][0]

    df.loc[i, 'raw\_output\_1'] = raw\_outputs[i][1]

probabilities = softmax(raw\_outputs, axis=1)

probabilities\_1 = [x[1] for x in probabilities]

df['prob'] = probabilities\_1

df.to\_csv(evaluation\_output\_file, index=False)

display(df.head(9))

# Simple bert\_model\_consumer

#%pip install simpletransformers

model\_files\_output\_folder="D:\\Sola\\DEV\\SC\\NewsCrawler\\bert\_modeller\\models"

from simpletransformers.classification import ClassificationModel, ClassificationArgs

#read model from file and then use to predict:

model2 = ClassificationModel("roberta", model\_files\_output\_folder,use\_cuda=False)

model2.predict(['new Blockchain plafrm is fraudilent','the economy has collapsed','the economy is in good shape'])

# Bert batch model prediction

#%pip install simpletransformers

model\_files\_output\_folder="D:\\Sola\\DEV\\SC\\NewsCrawler\\bert\_modeller\\models"

input\_data\_file          ="D:\\Sola\\DEV\\SC\\NewsCrawler\\bert\_modeller\\data\\input\_data.csv"

output\_data\_file         ="D:\\Sola\\DEV\\SC\\NewsCrawler\\bert\_modeller\\data\\output\_data.csv"

import pandas as pd

from simpletransformers.classification import ClassificationModel

from utils.text\_processing import text\_processing

#read model from file

model2 = ClassificationModel("roberta", model\_files\_output\_folder,use\_cuda=False)

#Prepare file for evaluation

df = pd.read\_csv(input\_data\_file, header=0)

df = pd.DataFrame(df[['uid','source\_id'

                      ,'source','article\_date'

                      ,'content','url','count'

                      ,'img\_link','entity'

                      ,'author','coin']])

df['content'] = df.apply(lambda x: text\_processing(x.content,

                lower=True,

                remove\_url=True,

                remove\_punctuation=True,

                remove\_stopwords=False,

                replace\_entity=True,

                replace\_hash=True,

                split\_alphanumeric=False,

                lemmatize=True,

                stem=True), axis=1)

print("Check null: ", df.isnull().sum().sum())

display(df.head(3))

pred, raw\_outputs = model2.predict(df['content'].tolist())