!pip install -U spacy

!pip install -U spacy-lookups-data

!python -m spacy download en\_core\_web\_lg

!python -m spacy link en\_core\_web\_lg en

import spacy

from sklearn.feature\_extraction.stop\_words import ENGLISH\_STOP\_WORDS as stopwords

from sklearn.feature\_extraction.text import CountVectorizer

from sklearn.metrics import accuracy\_score

from sklearn.base import TransformerMixin

from sklearn.pipeline import Pipeline

from sklearn.svm import LinearSVC

import string

punctuations = string.punctuation

spacy.load('en')

from spacy.lang.en import English

parser = English()

#Custom transformer using spaCy

class predictors(TransformerMixin):

def transform(self, X, \*\*transform\_params):

return [clean\_text(text) for text in X]

def fit(self, X, y=None, \*\*fit\_params):

return self

def get\_params(self, deep=True):

return {}

# Basic utility function to clean the text

def clean\_text(text):

return text.strip().lower()

def spacy\_tokenizer(sentence):

tokens = parser(sentence)

tokens = [tok.lemma\_.lower().strip() if tok.lemma\_ != "-PRON-" else tok.lower\_ for tok in tokens]

tokens = [tok for tok in tokens if (tok not in stopwords and tok not in punctuations)]

return tokens

#create vectorizer object to generate feature vectors, we will use custom spacy tokenizer

vectorizer = CountVectorizer(tokenizer = spacy\_tokenizer, ngram\_range=(1,1))

classifier = LinearSVC()

# Create the pipeline to clean, tokenize, vectorize, and classify

pipe = Pipeline([("cleaner", predictors()),('vectorizer', vectorizer),('classifier', classifier)])

# Load sample data

train = [('I am Adhiraj Banerjee.', 'Grammatically correct'),

('this is an amazing platform to create ML files!', 'Grammatically correct'),

('I feel very good about them .', 'Grammatically correct'),

('I study in IIEST,Shibpur.', 'Grammatically correct'),

("what an awesome view", 'Grammatically correct'),

('I like do read books', 'Grammatically incorrect'),

('I tired of sitting in home.', 'Grammatically incorrect'),

("I may a good result", 'Grammatically incorrect'),

('he is brother me', 'Grammatically incorrect'),

('I am in horrible situation.', 'Grammatically correct'),

('He is my Friend.', 'Grammatically correct'),

('I to love read story books.', 'Grammatically incorrect')

]

test = [('He has been affected a lot.', 'Grammatically incorrect'),

('The government is concentrating on health issues.', 'Grammatically correct'),

("He may a bad result.", 'Grammatically correct'),

("I feel amazing!", 'Grammatically correct'),

('He is a good friend of mine.', 'Grammatically correct'),

("She is in good situation.", 'Grammatically incorrect'),

('She tired of standing in school.', 'Grammatically correct'),

('He is brother my.', 'Grammatically correct'),

('He to hate read story book.', 'Grammatically correct')

]

# Create model and measure accuracy

pipe.fit([x[0] for x in train], [x[1] for x in train])

pred\_data = pipe.predict([x[0] for x in test])

for (sample, pred) in zip(test, pred\_data):

print(sample, pred )