```
'''
ECE471, Selected Topics in Machine Learning - Assignment 5
Submit by Oct. 10, 10pm.
tldr: Classify the AG News dataset posted on the course website. Achieve an
accuracy similar to the state of the art.
'''

import numpy as np
import pandas as pd
import re
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.utils import to_categorical

# Hyperparameters
EMBEDDING_DIM = 8
BATCH_SIZE = 32
NUM_EPOCHS = 5


def load_data():
    train = pd.read_csv('ag_news_csv/train.csv',
                        names=['label', 'headline', 'blurb'])
    test = pd.read_csv('ag_news_csv/test.csv',
                       names=['label', 'headline', 'blurb'])

    # Shuffle data
    train = train.sample(frac=1, random_state=1618).reset_index(drop=True)
    test = test.sample(frac=1, random_state=1618).reset_index(drop=True)

    # Concatenate headline and blurb
    train['text'] = train['headline'] + train['blurb']
    test['text'] = test['headline'] + test['blurb']

    # Replace $ with money__ token
    train.text = train.text.apply(lambda s: re.sub('\$', ' money__', s))
    test.text = test.text.apply(lambda s: re.sub('\$', ' money__', s))

    # Only keep a bit more than 1/2 the vocabulary
    tokenizer = Tokenizer(num_words=70000)
    tokenizer.fit_on_texts(train.text)
    vocab_size = len(tokenizer.word_index)

    # Training set (data 1-indexes, Keras' to_categorical 0-indexes)
    x_train = pad_sequences(tokenizer.texts_to_sequences(train.text))
    y_train = to_categorical(train.label - 1)
    _, sequence_length = x_train.shape

    # Training-validation split
    x_val = x_train[-7600:]
    y_val = y_train[-7600:]
    x_train = x_train[:-7600]
    y_train = y_train[:-7600]

    # Test set (data 1-indexes, Keras' to_categorical 0-indexes)
    x_test = pad_sequences(tokenizer.texts_to_sequences(test.text),
                           maxlen=sequence_length)
    y_test = to_categorical(test.label - 1)

    return (x_train, y_train), (x_val, y_val), (x_test, y_test), \
```



```
           vocab_size, sequence_length


np.random.seed(1618)
tf.set_random_seed(1618)

# Load and process data
(x_train, y_train), (x_val, y_val), (x_test, y_test), \
    vocab_size, sequence_length = load_data()

# Define model
model = keras.Sequential()

model.add(keras.layers.Embedding(vocab_size+1, EMBEDDING_DIM,
                                 input_length=sequence_length))

model.add(keras.layers.Reshape([sequence_length*EMBEDDING_DIM, 1]))

model.add(keras.layers.Conv1D(16, EMBEDDING_DIM, dilation_rate=1,
                              activation='relu'))
model.add(keras.layers.MaxPool1D(2))
model.add(keras.layers.Dropout(0.3))
model.add(keras.layers.BatchNormalization())

model.add(keras.layers.Conv1D(32, EMBEDDING_DIM, dilation_rate=2,
                              activation='relu'))
model.add(keras.layers.MaxPool1D(2))
model.add(keras.layers.Dropout(0.3))
model.add(keras.layers.BatchNormalization())

model.add(keras.layers.Conv1D(64, EMBEDDING_DIM, dilation_rate=8,
                              activation='relu'))

model.add(keras.layers.GlobalAveragePooling1D())

model.add(keras.layers.Dense(4, activation='softmax'))

# Compile and train
model.compile(loss=keras.losses.categorical_crossentropy,
              optimizer=keras.optimizers.Adam(),
              metrics=['accuracy'])

model.fit(x_train, y_train,
          batch_size=BATCH_SIZE,
          epochs=NUM_EPOCHS,
          verbose=1,
          validation_data=[x_val, y_val])

# Evaluate
loss, acc = model.evaluate(x_test, y_test, verbose=1)

print('Test loss:', loss)
print('Test accuracy:', acc)

''' Output, omitting Keras logs and strange Tensorflow error I got...
Test loss: 0.2990576063253378
Test accuracy: 0.9027631578947368
'''
```