import numpy as np

import tensorflow as tf

from tensorflow.keras.preprocessing.text import Tokenizer

from tensorflow.keras.preprocessing.sequence import pad\_sequences

from tensorflow.keras.layers import Embedding, Conv1D, GlobalMaxPooling1D, Dense, Dropout, Input

from tensorflow.keras.models import Model

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import LabelEncoder

# Example sentences organized by topics

topic\_sentences = {

"Technology": [

"This is a sentence about technology.",

"Artificial intelligence is revolutionizing industries across the globe.",

"The latest smartphone features cutting-edge technology for enhanced user experience.",

"Advancements in artificial intelligence are reshaping industries.",

"Advancements in pyhton lead to increase in jobs in DataScienece fields",

# Add more technology-related sentences

],

"Furniture": [

"Garden Furniture & Water Features Limited stock and Amazing prices.",

"Kitchen Gardeners International has 317 followers on Google+",

"Pack x6 English Lavender Angustifolia 'Hidcote' Perennial Garden Plug Plants by plugplants.net",

"FLEXIBLE EXPANDABLE HOSE PIPE LIGHT WEIGHT NON KINK WATER SPRAY NOZZLE. (Blue, 100 ft) by Generic",

"Garden beds for sale 10sft at 50 falt off hurry now",

"Good Furniture at good price vist our store before sales end!!",

# Add more technology-related sentences

],

"Electronic Gadgets": [

"Oral-B Pro 4000 CrossAction Electric Rechargeable Toothbrush Powered by Braun by Oral B",

"Apple Ipad Air Wifi64GB Siver MD790B/A by Apple",

"Samsung UE32J5100 Full HD 1080p 32-Inch LED TV (2015 Model) by Samsung",

"Smartphone offers carphonewarehouse.com",

"Save up to 50 percent on Selected Cameras",

"The new DSLR Camera by Sony-Reinvent relive ",

"The new BOSCH tool drill set",

"There are many cameras in the shop",

# Add more environment-related sentences

],

"Daily Accessories": [

"Cute Wigs $9.70 Only",

"Simplified Style Tote Elegant Shoulder Bag Gold Chain Metal Tassel Handbag Canvas Purse by PARA",

"H&M - Leather belt - Black - Ladies",

"Gillette Mach 3 Manual Razor Blades Pack of 4 by Gillette",

"Buy Second-hand, New and Antique Books | Oxfam's Online",

"Containers For Paint - Plastic Buckets, Pails, Pots, Tubs",

# Add more environment-related sentences

],

"Beauty Accesorries": [

"Beauty Protector Beauty Cream Body Lotion | Birchbox",

"This rich cream nourishes our skin, leaving it hydrated, soft",

"Dove Exfoliating Beauty Cream Bar ",

"Maybelline- Maybe she's born with it",

# Add more environment-related sentences

],

"Environment": [

"The environment is a critical global topic.",

"Climate change poses a significant threat to our planet's ecosystems.",

"Renewable energy sources like solar and wind power are essential for a sustainable future.",

"Plant more tree's save climate",

# Add more environment-related sentences

],

"Healthcare": [

"Healthcare systems are undergoing significant changes.",

"Medical breakthroughs in gene editing hold promise for treating genetic disorders.",

"Telemedicine is improving access to healthcare services, especially in remote areas.",

"Yoga everyday keeps your body fit",

"Green veggies helps to boost health",

"Green veggies are good for health",

"Healthy lifestyle is supported by yoga",

"Exercies keeps our body fit",

# Add more healthcare-related sentences

],

"Food": [

"Plant-based diets are gaining popularity for their health and environmental benefits.",

"Genetically modified organisms (GMOs) have sparked debates about their safety and impact on food production.",

"Food safety standards are crucial to prevent foodborne illnesses.",

"Burgers at flat 50 percent off buy now",

"Cheese n Macaroni Pizza get youselves 2",

"Coke at KFC",

"pizza good food",

"Burgers are delicious",

# Add more food-related sentences

],

# Add more topics and sentences

}

# Combine sentences and topics

sentences = []

topics = []

for topic, sentences\_list in topic\_sentences.items():

sentences.extend(sentences\_list)

topics.extend([topic] \* len(sentences\_list))

# Preprocess sentences

max\_words = 1000 # Maximum number of words in the vocabulary

tokenizer = Tokenizer(num\_words=max\_words)

tokenizer.fit\_on\_texts(sentences)

sequences = tokenizer.texts\_to\_sequences(sentences)

word\_index = tokenizer.word\_index

max\_sequence\_length = 100 # Set a reasonable maximum sequence length

padded\_sequences = pad\_sequences(sequences, maxlen=max\_sequence\_length)

# Encode topics

label\_encoder = LabelEncoder()

encoded\_topics = label\_encoder.fit\_transform(topics)

num\_classes = len(label\_encoder.classes\_)

# Split data into train and test sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(padded\_sequences, encoded\_topics, test\_size=0.2, random\_state=42)

# Build an enhanced CNN model

embedding\_dim = 100

input\_layer = Input(shape=(max\_sequence\_length,))

embedding\_layer = Embedding(input\_dim=max\_words, output\_dim=embedding\_dim)(input\_layer)

conv1\_layer = Conv1D(filters=128, kernel\_size=3, activation='relu')(embedding\_layer)

conv2\_layer = Conv1D(filters=128, kernel\_size=5, activation='relu')(embedding\_layer)

pool1\_layer = GlobalMaxPooling1D()(conv1\_layer)

pool2\_layer = GlobalMaxPooling1D()(conv2\_layer)

merged\_layer = tf.keras.layers.concatenate([pool1\_layer, pool2\_layer], axis=-1)

dropout\_layer = Dropout(0.5)(merged\_layer)

output\_layer = Dense(num\_classes, activation='softmax')(dropout\_layer)

model = Model(inputs=input\_layer, outputs=output\_layer)

model.compile(optimizer='adam', loss='sparse\_categorical\_crossentropy', metrics=['accuracy'])

# Train the model

num\_epochs = 50

batch\_size = 16

model.fit(X\_train, y\_train, epochs=num\_epochs, batch\_size=batch\_size, validation\_data=(X\_test, y\_test))

# Now you can use the trained model to predict topics for new sentences

new\_sentences = [

"Advancements in artificial intelligence are reshaping industries.",

"Climate change is a growing concern.",

"Buy 2 burgers at the cost of one",

"Yoga makes your body fit",

"Eat more green vegetables to boost immunity",

"Artificial Intelligence is taking world by storm",

"Italian Pizza made from dough and wheat",

"New camera showroom opened",

# Add more new sentences here

]

new\_sequences = tokenizer.texts\_to\_sequences(new\_sentences)

new\_padded\_sequences = pad\_sequences(new\_sequences, maxlen=max\_sequence\_length)

predictions = model.predict(new\_padded\_sequences)

predicted\_topics = label\_encoder.inverse\_transform(np.argmax(predictions, axis=1))

for sentence, topic in zip(new\_sentences, predicted\_topics):

print(f"Sentence: {sentence} - Predicted Topic: {topic}")