import dask.dataframe as dd  
import pandas as pd  
from sklearn.feature\_extraction.text import TfidfVectorizer  
from sklearn.metrics.pairwise import cosine\_similarity

amazon\_df = dd.read\_parquet('nswAmz.parquet')  
tiktok\_df = dd.read\_parquet('nswTik.parquet')  
  
amazon\_titles = amazon\_df['Search\_Term'] # Assuming 'Search\_Term' is the correct column name  
tiktok\_text = tiktok\_df['description']  
  
amazon\_titles = [str(item) for item in amazon\_titles]  
tiktok\_text = [str(item) for item in tiktok\_text]

list1 = tiktok\_text  
list2 = amazon\_titles

def calculate\_cosine\_similarity(list1, list2):  
   
  
 # Create a single list for TF-IDF  
 combined\_list = list1 + list2  
  
 # Initialize the vectorizer  
 vectorizer = TfidfVectorizer()  
  
 # Vectorize the text using TF-IDF  
 tfidf\_matrix = vectorizer.fit\_transform(combined\_list)  
  
 # Calculate cosine similarity  
 cos\_sim\_matrix = cosine\_similarity(tfidf\_matrix[:len(list1)], tfidf\_matrix[len(list1):])  
  
 return cos\_sim\_matrix

def save\_to\_csv(similarity\_matrix, list1, list2, dates\_collected, reporting\_dates, output\_file):  
 # Prepare data for DataFrame  
 data = []  
 for i in range(len(list1)):  
 # Get top 1 indices based on similarity  
 top\_indices = similarity\_matrix[i].argsort()[-1:][::-1]  
 for index in top\_indices:  
 # Append data for every pair, include similarity score even if it's 0  
 # Include dates collected and reporting dates  
 data.append([  
 list1[i],   
 dates\_collected[i], # Assuming dates\_collected aligns with list1  
 list2[index],   
 reporting\_dates[index], # Corrected to use reporting\_dates assuming it's aligned with list2  
 similarity\_matrix[i][index]  
 ])  
 # Creating a DataFrame with new columns for dates  
 df = pd.DataFrame(data, columns=[  
 'Hashtag Description',   
 'Date Collected',   
 'SearchTerm',   
 'Cosine Similarity', # Assuming each row can have a different reporting date  
 'Reporting Date'  
 ])  
   
 # Saving the DataFrame to a CSV file  
 df.to\_csv(output\_file, index=False)  
  
# Ensure you have list1, list2, dates\_collected, reporting\_dates, similarity\_matrix, and output\_file defined before calling this function.

# Example file and columns  
  
  
# Calculate cosine similarities  
similarity\_matrix = calculate\_cosine\_similarity(list1, list2)  
  
# Save the results to a CSV file

# Extract 'date\_collected' from tiktok\_df DataFrame  
dates\_collected = list(tiktok\_df['date\_collected'])  
  
# Extract 'Reporting Date' from amazon\_df DataFrame  
reporting\_dates = list(amazon\_df['Reporting Date'])

# save\_to\_csv function with these lists  
save\_to\_csv(similarity\_matrix, list1, list2, dates\_collected, reporting\_dates, output\_csv)  
  
# Confirm that the results have been saved  
print(f"Results saved to {output\_csv}")

# Define the size of each chunk  
chunk\_size = 12000  
  
# Run the loop for 12 times  
for i in range(12):  
 # Calculate start and end index for slicing list1  
 start\_index = i \* chunk\_size  
 end\_index = start\_index + chunk\_size  
 if end\_index > len(list1): # Make sure we don't go past the end of list1  
 end\_index = len(list1)  
  
 # Slice list1 for the current chunk  
 current\_list1\_chunk = list1[start\_index:end\_index]  
  
 # Calculate cosine similarities for the current chunk  
 similarity\_matrix = calculate\_cosine\_similarity(current\_list1\_chunk, list2)  
  
 # Define the output CSV file name for the current chunk  
 output\_csv = f'cosine\_similarity\_output\_part\_{i+1}.csv'  
  
 # Call the function to save the results to a CSV file  
 save\_to\_csv(similarity\_matrix, current\_list1\_chunk, list2, dates\_collected, reporting\_dates, output\_csv)  
  
 print(f"Results saved to {output\_csv}")

# Define the size of each chunk  
chunk\_size = 12000  
  
# Run the loop for 12 times  
for i in range(13, 20):  
 # Your code here  
 start\_index = i \* chunk\_size  
 end\_index = start\_index + chunk\_size  
 if end\_index > len(list1): # Make sure we don't go past the end of list1  
 end\_index = len(list1)  
  
 # Slice list1 for the current chunk  
 current\_list1\_chunk = list1[start\_index:end\_index]  
  
 # Calculate cosine similarities for the current chunk  
 similarity\_matrix = calculate\_cosine\_similarity(current\_list1\_chunk, list2)  
  
 # Define the output CSV file name for the current chunk  
 output\_csv = f'cosine\_similarity\_output\_part\_{i+1}.csv'  
  
 # Call the function to save the results to a CSV file  
 save\_to\_csv(similarity\_matrix, current\_list1\_chunk, list2, dates\_collected, reporting\_dates, output\_csv)  
  
 print(f"Results saved to {output\_csv}")

# Initialize an empty list to store DataFrames  
import pandas as pd  
dataframes = []  
  
# Loop through the file parts 1 to 24  
for i in range(1, 20):  
 # Read each file part  
 df = pd.read\_csv(f'cosine\_similarity\_output\_part\_{i}.csv', encoding='UTF-8')  
 # Append the DataFrame to the list  
 dataframes.append(df)  
  
# Concatenate all DataFrames in the list  
combined\_df = pd.concat(dataframes, ignore\_index=True)

combined\_df = combined\_df.rename(columns={'Cosine Similarity': 'Reporting Date'})  
  
# The title of these two columns get reversed.

df5 = pd.read\_csv('combined\_output.csv', encoding='UTF-8')

df5 = df5.rename(columns={'Cosine Similarity': 'Reporting Date'})

df5 = df5.rename(columns={'Cosine Similarity.1': 'Cosine Similarity'})

# Save the combined DataFrame to a CSV file  
df5.to\_csv('combined\_output.csv', index=False)  
  
print("All files have been merged into combined\_output.csv")