

```
In [ ]: # !wget "https://github.com/awsaf49/flickr-dataset/releases/download/v1.0/fl
# !unzip -q flickr8k.zip -d ./flickr8k
# !rm flickr8k.zip
# !echo "Downloaded Flickr8k dataset successfully."
```

'wget' is not recognized as an internal or external command,
operable program or batch file.
'unzip' is not recognized as an internal or external command,
operable program or batch file.
'rm' is not recognized as an internal or external command,
operable program or batch file.
"Downloaded Flickr8k dataset successfully."

```
In [2]: import pandas as pd
```

```
# Load captions
df = pd.read_csv('./flickr8k/captions.txt')
print(df.head())
```

	image \	caption
0	1000268201_693b08cb0e.jpg	A child in a pink dress is climbing up a set o...
1	1000268201_693b08cb0e.jpg	A girl going into a wooden building .
2	1000268201_693b08cb0e.jpg	A little girl climbing into a wooden playhouse .
3	1000268201_693b08cb0e.jpg	A little girl climbing the stairs to her playh...
4	1000268201_693b08cb0e.jpg	A little girl in a pink dress going into a woo...

```
In [6]: import re
```

```
def clean_caption(caption):
    caption = caption.lower()
    caption = re.sub(r"^[a-z ]+", "", caption)
    caption = caption.strip()
    return f"<start> {caption} <end>"

# Apply cleaning
df['cleaned_caption'] = df['caption'].apply(clean_caption)
print(df.head())
```

```

                                image \
0  1000268201_693b08cb0e.jpg
1  1000268201_693b08cb0e.jpg
2  1000268201_693b08cb0e.jpg
3  1000268201_693b08cb0e.jpg
4  1000268201_693b08cb0e.jpg

                                caption \
0  A child in a pink dress is climbing up a set o...
1      A girl going into a wooden building .
2  A little girl climbing into a wooden playhouse .
3  A little girl climbing the stairs to her playh...
4  A little girl in a pink dress going into a woo...

                                cleaned_caption
0  <start> a child in a pink dress is climbing up...
1  <start> a girl going into a wooden building <end>
2  <start> a little girl climbing into a wooden p...
3  <start> a little girl climbing the stairs to h...
4  <start> a little girl in a pink dress going in...

```

```
In [7]: df['caption'][0]
```

```
Out[7]: 'A child in a pink dress is climbing up a set of stairs in an entry way .'
```

```
In [8]: df['cleaned_caption'][0]
```

```
Out[8]: '<start> a child in a pink dress is climbing up a set of stairs in an entry
way <end>'
```

```
In [9]: import matplotlib.pyplot as plt
        from PIL import Image

        sample_img = './flickr8k/Images/' + df.iloc[0]['image']
        sample_caption = df.iloc[0]['cleaned_caption']

        img = Image.open(sample_img)
        plt.imshow(img)
        plt.title(sample_caption)
        plt.axis('off')
        plt.show()
```

<start> a child in a pink dress is climbing up a set of stairs in an entry way <end>



```
In [10]: from collections import Counter

# Gather all words from cleaned captions
all_captions = df['cleaned_caption'].tolist()
all_words = []
for caption in all_captions:
    all_words.extend(caption.split())

# Count word frequencies
word_counts = Counter(all_words)
print(f"Total unique words (before filtering): {len(word_counts)}")
print(word_counts.most_common(10)) # Show top 10 words
```

```
Total unique words (before filtering): 8780
[('a', 62986), ('<start>', 40455), ('<end>', 40455), ('in', 18974), ('the', 18418), ('on', 10743), ('is', 9345), ('and', 8851), ('dog', 8136), ('with', 7765)]
```

```
In [11]: # Set threshold for rare words
threshold = 5
vocab = [word for word, count in word_counts.items() if count >= threshold]
print(f"Vocabulary size (after threshold={threshold}): {len(vocab)}")
```

```
Vocabulary size (after threshold=5): 2986
```

```
In [12]: # Add special tokens
vocab = ['<pad>', '<start>', '<end>', '<unk>'] + vocab

# Create mappings
word2idx = {word: idx for idx, word in enumerate(vocab)}
idx2word = {idx: word for word, idx in word2idx.items()}

print(f"Sample word2idx: {list(word2idx.items())[:20]}")
```

```
Sample word2idx: [('<pad>', 0), ('<start>', 4), ('<end>', 18), ('<unk>', 3),  
('a', 5), ('child', 6), ('in', 7), ('pink', 8), ('dress', 9), ('is', 10),  
('climbing', 11), ('up', 12), ('set', 13), ('of', 14), ('stairs', 15), ('a  
n', 16), ('way', 17), ('girl', 19), ('going', 20), ('into', 21)]
```

```
In [13]: def caption_to_indices(caption, word2idx):  
# Check if word is in vocabulary, if not use '<unk>' token's index,  
# making sure it's within vocabulary size  
return [word2idx.get(word, word2idx['<unk>']) if word2idx.get(word, word  
  
df['caption_indices'] = df['cleaned_caption'].apply(lambda cap: caption_to_i
```

```
In [14]: df['caption_indices'][10]
```

```
Out[14]: [4, 5, 24, 19, 56, 7, 57, 58, 7, 59, 14, 5, 60, 61, 37, 28, 62, 7, 5, 63, 1  
8]
```

```
In [15]: import torch.nn as nn  
import torch  
from torch.utils.data import Dataset, DataLoader  
from sklearn.model_selection import train_test_split  
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")  
  
# Use EfficientNet-B0  
from torchvision.models import efficientnet_b7, EfficientNet_B7_Weights  
  
# Load weights + transforms  
weights = EfficientNet_B7_Weights.IMAGENET1K_V1  
efficientnet = efficientnet_b7(weights=weights)  
efficientnet.classifier = nn.Identity() # Remove classifier for feature ext  
efficientnet = efficientnet.to(device)  
efficientnet.eval()  
  
# Use predefined transform  
transform = weights.transforms()  
  
# Feature extraction function with EfficientNet  
def extract_feature_efficientnet(image_path):  
    image = Image.open(image_path).convert('RGB')  
    image = transform(image).unsqueeze(0).to(device)  
    with torch.no_grad():  
        feature = efficientnet(image) # [1, 1280] for EfficientNet-B0  
    return feature.squeeze(0).cpu()
```

```
In [16]: device = torch.device("cuda" if torch.cuda.is_available() else "cpu")  
print(device)
```

cuda

```
In [17]: print(transform)
```

```

ImageClassification(
    crop_size=[600]
    resize_size=[600]
    mean=[0.485, 0.456, 0.406]
    std=[0.229, 0.224, 0.225]
    interpolation=InterpolationMode.BICUBIC
)

```

```

In [18]: import torchvision.transforms as transforms
         from PIL import Image

```

```

In [16]: import os
         import pickle

         features_dict = {}
         image_folder = './flickr8k/Images/'

         for img_name in df['image'].unique():
             img_path = os.path.join(image_folder, img_name)
             features = extract_feature_efficientnet(img_path)
             features_dict[img_name] = features

```

```

In [ ]: import pickle
         # Save extracted features
         with open('features_efficientnet.pkl', 'wb') as f:
             pickle.dump(features_dict, f)

         print("Extracted and saved features for all images using Efficientnet.")

```

Extracted and saved features for all images using Efficientnet.

```

In [19]: import pickle

         # Load pre-extracted features
         with open('features_efficientnet.pkl', 'rb') as f:
             features_dict = pickle.load(f)

         print(f"Total features loaded: {len(features_dict)}")

```

Total features loaded: 8091

```

In [20]: len(features_dict['1000268201_693b08cb0e.jpg'])

```

Out[20]: 2560

```

In [21]: import torch
         import os

         device = torch.device("cuda" if torch.cuda.is_available() else "cpu")

```

```

In [22]: class FlickrDataset(Dataset):
         def __init__(self, dataframe, features_dict, word2idx, max_len=25):
             self.df = dataframe
             self.features_dict = features_dict
             self.word2idx = word2idx
             self.max_len = max_len

```

```

def __len__(self):
    return len(self.df)

def __getitem__(self, idx):
    row = self.df.iloc[idx]
    image_name = row['image']
    caption_idx = row['caption_indices']
    feature = self.features_dict[image_name]

    feature = feature.float()
    caption_idx = torch.tensor(caption_idx, dtype=torch.long)
    return feature, caption_idx

from torch.nn.utils.rnn import pad_sequence

def collate_fn(batch):
    features, captions = zip(*batch)
    features = torch.stack(features, 0)
    captions = pad_sequence(captions, batch_first=True, padding_value=word2idx['<pad>'])
    return features, captions

```

```

In [23]: train_df, val_df = train_test_split(df, test_size=0.2, random_state=42)

train_dataset = FlickrDataset(train_df, features_dict, word2idx)
val_dataset = FlickrDataset(val_df, features_dict, word2idx)

train_loader = DataLoader(train_dataset, batch_size=32, shuffle=True, collate_fn=collate_fn)
val_loader = DataLoader(val_dataset, batch_size=32, shuffle=False, collate_fn=collate_fn)

```

```

In [24]: # Get one batch from the DataLoader
features_batch, captions_batch = next(iter(train_loader))

print(f"Features batch shape: {features_batch.shape}") # Should be (batch_size, num_features)
print(f"Captions batch shape: {captions_batch.shape}") # Should be (batch_size, max_caption_length)

# Let's look at the first feature vector and caption
print(f"First image feature vector (shape): {features_batch[0].shape}")
print(f"First caption indices: {captions_batch[0]}")

```

```

Features batch shape: torch.Size([32, 2560])
Captions batch shape: torch.Size([32, 25])
First image feature vector (shape): torch.Size([2560])
First caption indices: tensor([ 4,  5, 397, 804,  5, 395, 37, 567, 155,
        40, 18,  0,  0,  0,
         0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0])

```

```

In [25]: def decode_caption(indices, idx2word):
    words = [idx2word[idx.item()] for idx in indices if idx.item() not in [word2idx['<pad>']]
    return ' '.join(words)

# Example: Decode first caption in batch
decoded_caption = decode_caption(captions_batch[0], idx2word)
print(f"Decoded Caption: {decoded_caption}")

```

Decoded Caption: <start> a skateboarder attempts a trick with no shirt on <end>

```
In [26]: vocab_size = len(word2idx)
max_idx = 0
min_idx = vocab_size

for i in range(len(df)):
    caption_indices = df.iloc[i]['caption_indices']
    max_idx = max(max_idx, max(caption_indices))
    min_idx = min(min_idx, min(caption_indices))

print(f"Min index in captions: {min_idx}")
print(f"Max index in captions: {max_idx}")
print(f"Vocab size: {vocab_size}")

if max_idx >= vocab_size or min_idx < 0:
    print("✗ Some indices are out of bounds!")
else:
    print("✓ All indices are within valid range!")
```

Min index in captions: 0

Max index in captions: 2987

Vocab size: 2988

✓ All indices are within valid range!

```
In [27]: class Encoder(nn.Module):
    def __init__(self, embed_size):
        super(Encoder, self).__init__()
        self.fc = nn.Linear(2560, embed_size)
        self.bn = nn.BatchNorm1d(embed_size, momentum=0.01)

    def forward(self, features):
        features = self.fc(features)
        features = self.bn(features)
        return features

class Decoder(nn.Module):
    def __init__(self, embed_size, hidden_size, vocab_size, num_layers=1, dr
        super(Decoder, self).__init__()
        self.embed = nn.Embedding(vocab_size, embed_size, padding_idx=0)
        self.lstm = nn.LSTM(embed_size, hidden_size, num_layers, batch_first
        self.fc = nn.Linear(hidden_size, vocab_size)

    def forward(self, features, captions):
        captions = torch.clamp(captions, 0, len(word2idx) - 1)
        embeddings = self.embed(captions[:, :-1])
        inputs = torch.cat((features.unsqueeze(1), embeddings), 1)
        hiddens, _ = self.lstm(inputs)
        outputs = self.fc(hiddens)
        return outputs

class ImageCaptioningModel(nn.Module):
    def __init__(self, encoder, decoder):
        super(ImageCaptioningModel, self).__init__()
        self.encoder = encoder
```

```

        self.decoder = decoder

    def forward(self, features, captions):
        features = self.encoder(features)
        outputs = self.decoder(features, captions)
        return outputs

```

```

In [28]: def calculate_accuracy(outputs, targets):
    _, predicted = outputs.max(2) # [batch_size, seq_len]
    correct = (predicted == targets) & (targets != word2idx['<pad>'])
    total = (targets != word2idx['<pad>']).sum().item()
    correct_total = correct.sum().item()
    return correct_total / total if total > 0 else 0

```

```

In [29]: class EarlyStopping:
    def __init__(self, patience=3, delta=0):
        self.patience = patience
        self.delta = delta
        self.counter = 0
        self.best_loss = None
        self.early_stop = False

    def __call__(self, val_loss):
        if self.best_loss is None or val_loss < self.best_loss - self.delta:
            self.best_loss = val_loss
            self.counter = 0
        else:
            self.counter += 1
            if self.counter >= self.patience:
                self.early_stop = True

    def evaluate_with_acc(model, val_loader, criterion):
        model.eval()
        val_loss = 0
        val_acc = 0

        with torch.no_grad():
            for features, captions in val_loader:
                features, captions = features.to(device), captions.to(device)
                outputs = model(features, captions)

                outputs = outputs[:, :-1, :]
                targets = captions[:, 1:]

                loss = criterion(outputs.reshape(-1, vocab_size), targets.reshape(-1))
                acc = calculate_accuracy(outputs, targets)

                val_loss += loss.item()
                val_acc += acc

            return val_loss / len(val_loader), val_acc / len(val_loader)

    def train(model, data_loader, criterion, optimizer, num_epochs, val_loader=None):

```



```

early_stopping = EarlyStopping(patience=5)

train_loss_history = []
train_acc_history = []
val_loss_history = []
val_acc_history = []

for epoch in range(num_epochs):
    model.train()
    total_loss = 0
    total_acc = 0

    for features, captions in data_loader:
        features, captions = features.to(device), captions.to(device)

        outputs = model(features, captions)
        outputs = outputs[:, :-1, :]
        targets = captions[:, 1:]

        loss = criterion(outputs.reshape(-1, vocab_size), targets.reshape(-1))
        acc = calculate_accuracy(outputs, targets)

        optimizer.zero_grad()
        loss.backward()
        optimizer.step()

        total_loss += loss.item()
        total_acc += acc

    avg_loss = total_loss / len(data_loader)
    avg_acc = total_acc / len(data_loader)
    train_loss_history.append(avg_loss)
    train_acc_history.append(avg_acc)
    print(f"Epoch [{epoch+1}/{num_epochs}], Train Loss: {avg_loss:.4f}, Train Acc: {avg_acc:.4f}")

    if val_loader:
        val_loss, val_acc = evaluate_with_acc(model, val_loader, criterion)
        val_loss_history.append(val_loss)
        val_acc_history.append(val_acc)
        print(f"Validation Loss: {val_loss:.4f}, Validation Acc: {val_acc:.4f}")
        # scheduler.step(val_loss) # <--- Adjust LR based on val_loss
        early_stopping(val_loss)
    else:
        early_stopping(avg_loss)

    if early_stopping.early_stop:
        print("Early stopping triggered.")
        break

# Plot Loss
plt.figure(figsize=(12, 5))
plt.subplot(1, 2, 1)
plt.plot(train_loss_history, label='Train Loss')
plt.plot(val_loss_history, label='Val Loss')
plt.title('Loss Curve')
plt.xlabel('Epoch')
plt.ylabel('Loss')

```

```
plt.legend()

# Plot Accuracy
plt.subplot(1, 2, 2)
plt.plot(train_acc_history, label='Train Acc')
plt.plot(val_acc_history, label='Val Acc')
plt.title('Accuracy Curve')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.legend()

plt.show()
```

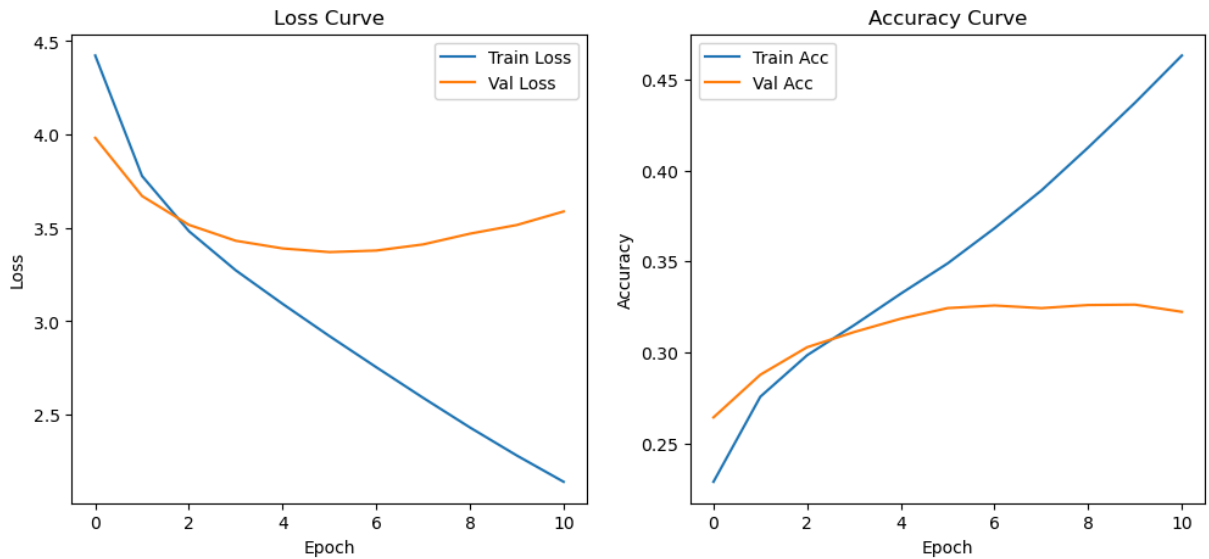
```
In [57]: vocab_size = len(word2idx)
embed_size = 512 # Bigger embedding for more capacity
hidden_size = 1024 # Bigger LSTM hidden size

encoder = Encoder(embed_size).to(device)
decoder = Decoder(embed_size, hidden_size, vocab_size, num_layers=3, dropout
model = ImageCaptioningModel(encoder, decoder).to(device)

criterion = nn.CrossEntropyLoss(ignore_index=word2idx['<pad>'])
optimizer = torch.optim.Adam(model.parameters(), lr=1e-3)
# scheduler = torch.optim.lr_scheduler.ReduceLROnPlateau(optimizer, mode='mi

train(model, train_loader, criterion, optimizer, num_epochs=50, val_loader=v
```

```
Epoch [1/50], Train Loss: 4.4230, Train Acc: 0.2289
Validation Loss: 3.9819, Validation Acc: 0.2643
Epoch [2/50], Train Loss: 3.7776, Train Acc: 0.2757
Validation Loss: 3.6704, Validation Acc: 0.2877
Epoch [3/50], Train Loss: 3.4824, Train Acc: 0.2985
Validation Loss: 3.5161, Validation Acc: 0.3029
Epoch [4/50], Train Loss: 3.2726, Train Acc: 0.3149
Validation Loss: 3.4309, Validation Acc: 0.3112
Epoch [5/50], Train Loss: 3.0931, Train Acc: 0.3323
Validation Loss: 3.3899, Validation Acc: 0.3185
Epoch [6/50], Train Loss: 2.9205, Train Acc: 0.3489
Validation Loss: 3.3702, Validation Acc: 0.3243
Epoch [7/50], Train Loss: 2.7537, Train Acc: 0.3682
Validation Loss: 3.3786, Validation Acc: 0.3258
Epoch [8/50], Train Loss: 2.5899, Train Acc: 0.3889
Validation Loss: 3.4114, Validation Acc: 0.3243
Epoch [9/50], Train Loss: 2.4310, Train Acc: 0.4126
Validation Loss: 3.4689, Validation Acc: 0.3260
Epoch [10/50], Train Loss: 2.2810, Train Acc: 0.4372
Validation Loss: 3.5153, Validation Acc: 0.3262
Epoch [11/50], Train Loss: 2.1401, Train Acc: 0.4631
Validation Loss: 3.5878, Validation Acc: 0.3223
Early stopping triggered.
```



In [123]: `!pip install torchinfo`

```
Collecting torchinfo
  Downloading torchinfo-1.8.0-py3-none-any.whl.metadata (21 kB)
  Downloading torchinfo-1.8.0-py3-none-any.whl (23 kB)
Installing collected packages: torchinfo
Successfully installed torchinfo-1.8.0
```

In [58]: `from torchinfo import summary`
`summary(model)`

```
Out[58]: =====
Layer (type:depth-idx)                   Param #
=====
ImageCaptioningModel                     --
├─Encoder: 1-1                           --
│   └─Linear: 2-1                         1,311,232
│   └─BatchNorm1d: 2-2                   1,024
├─Decoder: 1-2                           --
│   └─Embedding: 2-3                     1,529,856
│   └─LSTM: 2-4                           23,093,248
│   └─Linear: 2-5                         3,062,700
=====
Total params: 28,998,060
Trainable params: 28,998,060
Non-trainable params: 0
=====
```

In [74]: `# Save Encoder and Decoder weights`
`torch.save(encoder.state_dict(), 'encoder.pth')`
`torch.save(decoder.state_dict(), 'decoder.pth')`

`print("Models saved successfully!")`

Models saved successfully!

```
In [75]: # Load weights into models
encoder.load_state_dict(torch.load('encoder.pth'))
decoder.load_state_dict(torch.load('decoder.pth'))

encoder.eval()
decoder.eval()

print("Models loaded successfully!")
```

Models loaded successfully!

```
In [59]: def generate_caption(model, feature, word2idx, idx2word, max_len=20):
    model.eval()
    result_caption = []

    # Encode the feature
    feature = feature.unsqueeze(0).to(device) # [1, 2048]
    with torch.no_grad():
        feature = model.encoder(feature) # [1, embed_size]

    # Start token
    word_idx = torch.tensor([word2idx['<start>']]).to(device)
    caption = [word_idx]

    for _ in range(max_len):
        caption_tensor = torch.tensor(caption).unsqueeze(0).to(device) # [1, seq_len]
        with torch.no_grad():
            embeddings = model.decoder.embed(caption_tensor) # [1, seq_len, embed_size]
            inputs = torch.cat((feature.unsqueeze(1), embeddings), dim=1) # [1, seq_len+1, embed_size+embed_size]
            hiddens, _ = model.decoder.lstm(inputs)
            outputs = model.decoder.fc(hiddens) # [1, seq_len+1, vocab_size]

        # Take the last word's prediction
        outputs = outputs.squeeze(0) # [seq_len+1, vocab_size]
        _, predicted = outputs[-1].max(0) # last time step

        next_word = idx2word[predicted.item()]
        if next_word == '<end>':
            break

        result_caption.append(next_word)
        caption.append(predicted)

    return ' '.join(result_caption)
```

```
In [60]: def generate_caption_beam_search(model, feature, word2idx, idx2word, beam_size):
    model.eval()
    feature = feature.unsqueeze(0).to(device)
    feature = model.encoder(feature)

    sequences = [[list(), 0.0, feature]]
    for _ in range(max_len):
        all_candidates = []
        for seq, score, feature in sequences:
            if len(seq) > 0 and seq[-1] == word2idx['<end>']:
```

```

        all_candidates.append((seq, score, feature))
        continue
    inputs = torch.tensor([seq], dtype=torch.long).to(device)
    if len(inputs[0]) == 0:
        inputs = torch.tensor([[word2idx['<start>']]], dtype=torch.long)
    outputs = model.decoder(feature, inputs)
    outputs = outputs[:, -1, :] # Get the last word prediction
    probs = torch.nn.functional.log_softmax(outputs, dim=-1)
    top_probs, top_idx = probs.topk(beam_size)

    for i in range(beam_size):
        candidate = seq + [top_idx[0][i].item()]
        candidate_score = score - top_probs[0][i].item()
        all_candidates.append((candidate, candidate_score, feature))

    sequences = sorted(all_candidates, key=lambda tup: tup[1][:beam_size])

    best_seq = sequences[0][0]
    caption = [idx2word[idx] for idx in best_seq if idx != word2idx['<start>']]
    return ' '.join(caption)

```

```

In [65]: # Load one sample image and extract its feature
sample_image = './flickr8k/Images/' + df.iloc[80]['image']
sample_feature = extract_feature_efficientnet(sample_image).to(device)

# Generate caption
caption = generate_caption(model, sample_feature, word2idx, idx2word)
print(f"Generated Caption: {caption}")

# Show the image
from PIL import Image
import matplotlib.pyplot as plt

img = Image.open(sample_image)
plt.imshow(img)
plt.title(caption)
plt.axis('off')
plt.show()

```

Generated Caption: boy on edge water while in lake

boy on edge water while in lake



```
In [69]: # Load one sample image and extract its feature
sample_image = './flickr8k/Images/' + df.iloc[150]['image']
sample_feature = extract_feature_efficientnet(sample_image).to(device)

# Generate caption
caption = generate_caption_beam_search(model, sample_feature, word2idx, idx2word)
print(f"Generated Caption: {caption}")

# Show the image
from PIL import Image
import matplotlib.pyplot as plt

img = Image.open(sample_image)
plt.imshow(img)
plt.title(caption)
plt.axis('off')
plt.show()
```

Generated Caption: a a boy eats a food

a a boy eats a food



```
In [47]: import nltk
         from nltk.translate.bleu_score import sentence_bleu, SmoothingFunction
         nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to C:\Users\SONU
[nltk_data]   SONI\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

Out[47]: True

```
In [70]: def calculate_bleu(model, dataframe, features_dict, word2idx, idx2word, max_
         model.eval()
         smoothie = SmoothingFunction().method4 # Helps avoid zero scores for sm

         bleu1_scores = []
         bleu2_scores = []
         bleu3_scores = []
         bleu4_scores = []

         for i in range(min(len(dataframe), max_samples)):
             image_name = dataframe.iloc[i]['image']
             references = dataframe[dataframe['image'] == image_name]['cleaned_ca
             references = [ref.split() for ref in references] # tokenized refere

             # Extract feature for the image
             feature = features_dict[image_name].to(device)

             # Generate prediction
             generated = generate_caption(model, feature, word2idx, idx2word)
             candidate = generated.split() # tokenized candidate caption

             # Calculate BLEU scores
             bleu1 = sentence_bleu(references, candidate, weights=(1, 0, 0, 0), s
             bleu2 = sentence_bleu(references, candidate, weights=(0.5, 0.5, 0, 0,
```

```

    bleu3 = sentence_bleu(references, candidate, weights=(0.33, 0.33, 0.33))
    bleu4 = sentence_bleu(references, candidate, weights=(0.25, 0.25, 0.25, 0.25))

    bleu1_scores.append(bleu1)
    bleu2_scores.append(bleu2)
    bleu3_scores.append(bleu3)
    bleu4_scores.append(bleu4)

    avg_bleu1 = sum(bleu1_scores) / len(bleu1_scores)
    avg_bleu2 = sum(bleu2_scores) / len(bleu2_scores)
    avg_bleu3 = sum(bleu3_scores) / len(bleu3_scores)
    avg_bleu4 = sum(bleu4_scores) / len(bleu4_scores)

    print(f"BLEU-1: {avg_bleu1:.4f}")
    print(f"BLEU-2: {avg_bleu2:.4f}")
    print(f"BLEU-3: {avg_bleu3:.4f}")
    print(f"BLEU-4: {avg_bleu4:.4f}")

```

In [72]: `calculate_bleu(model, val_df, features_dict, word2idx, idx2word, max_samples`

```

BLEU-1: 0.1847
BLEU-2: 0.0569
BLEU-3: 0.0323
BLEU-4: 0.0210

```

In [3]: `!pip install transformers`

Collecting transformers

Downloading transformers-4.51.3-py3-none-any.whl.metadata (38 kB)

Requirement already satisfied: filelock in c:\sonu\anaconda\lib\site-packages (from transformers) (3.13.1)

Collecting huggingface-hub<1.0,>=0.30.0 (from transformers)

Downloading huggingface_hub-0.30.2-py3-none-any.whl.metadata (13 kB)

Requirement already satisfied: numpy>=1.17 in c:\sonu\anaconda\lib\site-packages (from transformers) (1.26.4)

Requirement already satisfied: packaging>=20.0 in c:\sonu\anaconda\lib\site-packages (from transformers) (24.1)

Requirement already satisfied: pyyaml>=5.1 in c:\sonu\anaconda\lib\site-packages (from transformers) (6.0.1)

Requirement already satisfied: regex!=2019.12.17 in c:\sonu\anaconda\lib\site-packages (from transformers) (2024.9.11)

Requirement already satisfied: requests in c:\sonu\anaconda\lib\site-packages (from transformers) (2.32.3)

Collecting tokenizers<0.22,>=0.21 (from transformers)

Downloading tokenizers-0.21.1-cp39-abi3-win_amd64.whl.metadata (6.9 kB)

Collecting safetensors>=0.4.3 (from transformers)

Downloading safetensors-0.5.3-cp38-abi3-win_amd64.whl.metadata (3.9 kB)

Requirement already satisfied: tqdm>=4.27 in c:\sonu\anaconda\lib\site-packages (from transformers) (4.66.5)

Requirement already satisfied: fsspec>=2023.5.0 in c:\sonu\anaconda\lib\site-packages (from huggingface-hub<1.0,>=0.30.0->transformers) (2024.6.1)

Requirement already satisfied: typing-extensions>=3.7.4.3 in c:\sonu\anaconda\lib\site-packages (from huggingface-hub<1.0,>=0.30.0->transformers) (4.11.0)

Requirement already satisfied: colorama in c:\sonu\anaconda\lib\site-packages (from tqdm>=4.27->transformers) (0.4.6)

Requirement already satisfied: charset-normalizer<4,>=2 in c:\sonu\anaconda\lib\site-packages (from requests->transformers) (3.3.2)

Requirement already satisfied: idna<4,>=2.5 in c:\sonu\anaconda\lib\site-packages (from requests->transformers) (3.7)

Requirement already satisfied: urllib3<3,>=1.21.1 in c:\sonu\anaconda\lib\site-packages (from requests->transformers) (2.2.3)

Requirement already satisfied: certifi>=2017.4.17 in c:\sonu\anaconda\lib\site-packages (from requests->transformers) (2024.8.30)

Downloading transformers-4.51.3-py3-none-any.whl (10.4 MB)

```
----- 0.0/10.4 MB ? eta -:-:--
----- 3.4/10.4 MB 18.3 MB/s eta 0:00:0
1
----- 7.3/10.4 MB 18.1 MB/s eta 0:00:0
1
----- 10.4/10.4 MB 18.5 MB/s eta 0:00:
00
```

Downloading huggingface_hub-0.30.2-py3-none-any.whl (481 kB)

Downloading safetensors-0.5.3-cp38-abi3-win_amd64.whl (308 kB)

Downloading tokenizers-0.21.1-cp39-abi3-win_amd64.whl (2.4 MB)

```
----- 0.0/2.4 MB ? eta -:-:--
----- 2.4/2.4 MB 19.9 MB/s eta 0:00:00
```

Installing collected packages: safetensors, huggingface-hub, tokenizers, transformers

Successfully installed huggingface-hub-0.30.2 safetensors-0.5.3 tokenizers-0.21.1 transformers-4.51.3

```
In [4]: from transformers import BlipProcessor, BlipForConditionalGeneration
        from PIL import Image
        import requests

        # Load pretrained BLIP model
        processor = BlipProcessor.from_pretrained("Salesforce/blip-image-captioning-
        model = BlipForConditionalGeneration.from_pretrained("Salesforce/blip-image-

        # Load an image
```

Using a slow image processor as `use_fast` is unset and a slow processor was saved with this model. `use_fast=True` will be the default behavior in v4.5.2, even if the model was saved with a slow processor. This will result in minor differences in outputs. You'll still be able to use a slow processor with `use_fast=False`.

```
preprocessor_config.json: 0%|          | 0.00/287 [00:00<?, ?B/s]
```

c:\sonu\anaconda\Lib\site-packages\huggingface_hub\file_download.py:144: UserWarning: `huggingface_hub` cache-system uses symlinks by default to efficiently store duplicated files but your machine does not support them in C:\Users\SONU SONI\cache\huggingface\hub\models--Salesforce--blip-image-captioning-base. Caching files will still work but in a degraded version that might require more space on your disk. This warning can be disabled by setting the `HF_HUB_DISABLE_SYMLINKS_WARNING` environment variable. For more details, see https://huggingface.co/docs/huggingface_hub/how-to-cache#limitations. To support symlinks on Windows, you either need to activate Developer Mode or to run Python as an administrator. In order to activate developer mode, see this article: <https://docs.microsoft.com/en-us/windows/apps/get-started/enable-your-device-for-development>

```
warnings.warn(message)
```

```
tokenizer_config.json: 0%|          | 0.00/506 [00:00<?, ?B/s]
```

```
vocab.txt: 0%|          | 0.00/232k [00:00<?, ?B/s]
```

```
tokenizer.json: 0%|          | 0.00/711k [00:00<?, ?B/s]
```

```
special_tokens_map.json: 0%|          | 0.00/125 [00:00<?, ?B/s]
```

```
config.json: 0%|          | 0.00/4.56k [00:00<?, ?B/s]
```

```
pytorch_model.bin: 0%|          | 0.00/990M [00:00<?, ?B/s]
```

```
model.safetensors: 0%|          | 0.00/990M [00:00<?, ?B/s]
```

```
In [5]: # Load an image
        img_path = "./flickr8k/Images/" + df.iloc[150]['image'] # change to your in
        image = Image.open(img_path)

        # Preprocess and generate
        inputs = processor(image, return_tensors="pt")
        out = model.generate(**inputs)
        caption = processor.decode(out[0], skip_special_tokens=True)

        print("Generated caption:", caption)

        import matplotlib.pyplot as plt

        img = Image.open(img_path)
        plt.imshow(img)
        plt.title(caption)
        plt.axis('off')
        plt.show()
```

Generated caption: a boy eating a piece of pizza

a boy eating a piece of pizza



In [51]: `!pip install transformers evaluate nltk`

Requirement already satisfied: transformers in c:\sonu\anaconda\lib\site-packages (4.51.3)
Collecting evaluate
 Downloading evaluate-0.4.3-py3-none-any.whl.metadata (9.2 kB)
Requirement already satisfied: nltk in c:\sonu\anaconda\lib\site-packages (3.9.1)
Requirement already satisfied: filelock in c:\sonu\anaconda\lib\site-packages (from transformers) (3.13.1)
Requirement already satisfied: huggingface-hub<1.0,>=0.30.0 in c:\sonu\anaconda\lib\site-packages (from transformers) (0.30.2)
Requirement already satisfied: numpy>=1.17 in c:\sonu\anaconda\lib\site-packages (from transformers) (1.26.4)
Requirement already satisfied: packaging>=20.0 in c:\sonu\anaconda\lib\site-packages (from transformers) (24.1)
Requirement already satisfied: pyyaml>=5.1 in c:\sonu\anaconda\lib\site-packages (from transformers) (6.0.1)
Requirement already satisfied: regex!=2019.12.17 in c:\sonu\anaconda\lib\site-packages (from transformers) (2024.9.11)
Requirement already satisfied: requests in c:\sonu\anaconda\lib\site-packages (from transformers) (2.32.3)
Requirement already satisfied: tokenizers<0.22,>=0.21 in c:\sonu\anaconda\lib\site-packages (from transformers) (0.21.1)
Requirement already satisfied: safetensors>=0.4.3 in c:\sonu\anaconda\lib\site-packages (from transformers) (0.5.3)
Requirement already satisfied: tqdm>=4.27 in c:\sonu\anaconda\lib\site-packages (from transformers) (4.66.5)
Collecting datasets>=2.0.0 (from evaluate)
 Downloading datasets-3.5.1-py3-none-any.whl.metadata (19 kB)
Requirement already satisfied: dill in c:\sonu\anaconda\lib\site-packages (from evaluate) (0.3.8)
Requirement already satisfied: pandas in c:\sonu\anaconda\lib\site-packages (from evaluate) (2.2.2)
Collecting xxhash (from evaluate)
 Downloading xxhash-3.5.0-cp312-cp312-win_amd64.whl.metadata (13 kB)
Collecting multiprocessing (from evaluate)
 Downloading multiprocessing-0.70.18-py312-none-any.whl.metadata (7.5 kB)
Requirement already satisfied: fsspec>=2021.05.0 in c:\sonu\anaconda\lib\site-packages (from fsspec[http]>=2021.05.0->evaluate) (2024.6.1)
Requirement already satisfied: click in c:\sonu\anaconda\lib\site-packages (from nltk) (8.1.7)
Requirement already satisfied: joblib in c:\sonu\anaconda\lib\site-packages (from nltk) (1.4.2)
Requirement already satisfied: pyarrow>=15.0.0 in c:\sonu\anaconda\lib\site-packages (from datasets>=2.0.0->evaluate) (16.1.0)
 Downloading multiprocessing-0.70.16-py312-none-any.whl.metadata (7.2 kB)
Requirement already satisfied: aiohttp in c:\sonu\anaconda\lib\site-packages (from datasets>=2.0.0->evaluate) (3.10.5)
Requirement already satisfied: typing-extensions>=3.7.4.3 in c:\sonu\anaconda\lib\site-packages (from huggingface-hub<1.0,>=0.30.0->transformers) (4.11.0)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\sonu\anaconda\lib\site-packages (from requests->transformers) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in c:\sonu\anaconda\lib\site-packages (from requests->transformers) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\sonu\anaconda\lib\site-packages (from requests->transformers) (2.2.3)

Requirement already satisfied: certifi>=2017.4.17 in c:\sonu\anaconda\lib\site-packages (from requests->transformers) (2024.8.30)
Requirement already satisfied: colorama in c:\sonu\anaconda\lib\site-packages (from tqdm>=4.27->transformers) (0.4.6)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\sonu\anaconda\lib\site-packages (from pandas->evaluate) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\sonu\anaconda\lib\site-packages (from pandas->evaluate) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in c:\sonu\anaconda\lib\site-packages (from pandas->evaluate) (2023.3)
Requirement already satisfied: aiohappyeyeballs>=2.3.0 in c:\sonu\anaconda\lib\site-packages (from aiohttp->datasets>=2.0.0->evaluate) (2.4.0)
Requirement already satisfied: aiosignal>=1.1.2 in c:\sonu\anaconda\lib\site-packages (from aiohttp->datasets>=2.0.0->evaluate) (1.2.0)
Requirement already satisfied: attrs>=17.3.0 in c:\sonu\anaconda\lib\site-packages (from aiohttp->datasets>=2.0.0->evaluate) (23.1.0)
Requirement already satisfied: frozenlist>=1.1.1 in c:\sonu\anaconda\lib\site-packages (from aiohttp->datasets>=2.0.0->evaluate) (1.4.0)
Requirement already satisfied: multidict<7.0,>=4.5 in c:\sonu\anaconda\lib\site-packages (from aiohttp->datasets>=2.0.0->evaluate) (6.0.4)
Requirement already satisfied: yarl<2.0,>=1.0 in c:\sonu\anaconda\lib\site-packages (from aiohttp->datasets>=2.0.0->evaluate) (1.11.0)
Requirement already satisfied: six>=1.5 in c:\sonu\anaconda\lib\site-packages (from python-dateutil>=2.8.2->pandas->evaluate) (1.16.0)
Downloading evaluate-0.4.3-py3-none-any.whl (84 kB)
Downloading datasets-3.5.1-py3-none-any.whl (491 kB)
Downloading multiprocess-0.70.16-py312-none-any.whl (146 kB)
Downloading xxhash-3.5.0-cp312-cp312-win_amd64.whl (30 kB)
Installing collected packages: xxhash, multiprocess, datasets, evaluate
Successfully installed datasets-3.5.1 evaluate-0.4.3 multiprocess-0.70.16 xxhash-3.5.0

In [54]: `nltk.download("all")`

```
[nltk_data] Downloading collection 'all'
[nltk_data] |
[nltk_data] | Downloading package abc to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\abc.zip.
[nltk_data] | Downloading package alpino to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\alpino.zip.
[nltk_data] | Downloading package averaged_perceptron_tagger to
[nltk_data] | C:\Users\SONU SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping taggers\averaged_perceptron_tagger.zip.
[nltk_data] | Downloading package averaged_perceptron_tagger_eng to
[nltk_data] | C:\Users\SONU SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping
[nltk_data] | taggers\averaged_perceptron_tagger_eng.zip.
[nltk_data] | Downloading package averaged_perceptron_tagger_ru to
[nltk_data] | C:\Users\SONU SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping
[nltk_data] | taggers\averaged_perceptron_tagger_ru.zip.
[nltk_data] | Downloading package averaged_perceptron_tagger_rus to
[nltk_data] | C:\Users\SONU SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping
[nltk_data] | taggers\averaged_perceptron_tagger_rus.zip.
[nltk_data] | Downloading package basque_grammars to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping grammars\basque_grammars.zip.
[nltk_data] | Downloading package bcp47 to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Downloading package biocreative_ppi to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\biocreative_ppi.zip.
[nltk_data] | Downloading package bllip_wsj_no_aux to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping models\bllip_wsj_no_aux.zip.
[nltk_data] | Downloading package book_grammars to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping grammars\book_grammars.zip.
[nltk_data] | Downloading package brown to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\brown.zip.
[nltk_data] | Downloading package brown_tei to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\brown_tei.zip.
[nltk_data] | Downloading package cess_cat to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\cess_cat.zip.
[nltk_data] | Downloading package cess_esp to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\cess_esp.zip.
[nltk_data] | Downloading package chat80 to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\chat80.zip.
[nltk_data] | Downloading package city_database to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\city_database.zip.
[nltk_data] | Downloading package cmudict to C:\Users\SONU
```

```
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\cmudict.zip.
[nltk_data] | Downloading package comparative_sentences to
[nltk_data] | C:\Users\SONU\SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\comparative_sentences.zip.
[nltk_data] | Downloading package comtrans to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Downloading package conll2000 to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\conll2000.zip.
[nltk_data] | Downloading package conll2002 to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\conll2002.zip.
[nltk_data] | Downloading package conll2007 to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Downloading package crubadan to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\crubadan.zip.
[nltk_data] | Downloading package dependency_treebank to
[nltk_data] | C:\Users\SONU\SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\dependency_treebank.zip.
[nltk_data] | Downloading package dolch to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\dolch.zip.
[nltk_data] | Downloading package english_wordnet to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\english_wordnet.zip.
[nltk_data] | Downloading package europarl_raw to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\europarl_raw.zip.
[nltk_data] | Downloading package extended_omw to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Downloading package floresta to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\floresta.zip.
[nltk_data] | Downloading package framenet_v15 to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\framenet_v15.zip.
[nltk_data] | Downloading package framenet_v17 to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\framenet_v17.zip.
[nltk_data] | Downloading package gazetteers to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\gazetteers.zip.
[nltk_data] | Downloading package genesis to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\genesis.zip.
[nltk_data] | Downloading package gutenber to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\gutenberg.zip.
[nltk_data] | Downloading package ier to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\ier.zip.
[nltk_data] | Downloading package inaugural to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\inaugural.zip.
```



```
[nltk_data] | Downloading package indian to C:\Users\SONU
[nltk_data] |     SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\indian.zip.
[nltk_data] | Downloading package jeita to C:\Users\SONU
[nltk_data] |     SONI\AppData\Roaming\nltk_data...
[nltk_data] | Downloading package kimmo to C:\Users\SONU
[nltk_data] |     SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\kimmo.zip.
[nltk_data] | Downloading package knbc to C:\Users\SONU
[nltk_data] |     SONI\AppData\Roaming\nltk_data...
[nltk_data] | Downloading package large_grammars to C:\Users\SONU
[nltk_data] |     SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping grammars\large_grammars.zip.
[nltk_data] | Downloading package lin_thesaurus to C:\Users\SONU
[nltk_data] |     SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\lin_thesaurus.zip.
[nltk_data] | Downloading package mac_morpho to C:\Users\SONU
[nltk_data] |     SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\mac_morpho.zip.
[nltk_data] | Downloading package machado to C:\Users\SONU
[nltk_data] |     SONI\AppData\Roaming\nltk_data...
[nltk_data] | Downloading package masc_tagged to C:\Users\SONU
[nltk_data] |     SONI\AppData\Roaming\nltk_data...
[nltk_data] | Downloading package maxent_ne_chunker to
[nltk_data] |     C:\Users\SONU SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping chunkers\maxent_ne_chunker.zip.
[nltk_data] | Downloading package maxent_ne_chunker_tab to
[nltk_data] |     C:\Users\SONU SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping chunkers\maxent_ne_chunker_tab.zip.
[nltk_data] | Downloading package maxent_treebank_pos_tagger to
[nltk_data] |     C:\Users\SONU SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping taggers\maxent_treebank_pos_tagger.zip.
[nltk_data] | Downloading package maxent_treebank_pos_tagger_tab to
[nltk_data] |     C:\Users\SONU SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping
[nltk_data] |     taggers\maxent_treebank_pos_tagger_tab.zip.
[nltk_data] | Downloading package mores_sample to C:\Users\SONU
[nltk_data] |     SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping models\mores_sample.zip.
[nltk_data] | Downloading package movie_reviews to C:\Users\SONU
[nltk_data] |     SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\movie_reviews.zip.
[nltk_data] | Downloading package mte_teip5 to C:\Users\SONU
[nltk_data] |     SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\mte_teip5.zip.
[nltk_data] | Downloading package mwa_ppdb to C:\Users\SONU
[nltk_data] |     SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping misc\mwa_ppdb.zip.
[nltk_data] | Downloading package names to C:\Users\SONU
[nltk_data] |     SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\names.zip.
[nltk_data] | Downloading package nombank.1.0 to C:\Users\SONU
[nltk_data] |     SONI\AppData\Roaming\nltk_data...
[nltk_data] | Downloading package nonbreaking_prefixes to
[nltk_data] |     C:\Users\SONU SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\nonbreaking_prefixes.zip.
```



```
[nltk_data] | Downloading package nps_chat to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\nps_chat.zip.
[nltk_data] | Downloading package omw to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Downloading package omw-1.4 to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Downloading package opinion_lexicon to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\opinion_lexicon.zip.
[nltk_data] | Downloading package panlex_swadesh to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Downloading package paradigms to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\paradigms.zip.
[nltk_data] | Downloading package pe08 to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\pe08.zip.
[nltk_data] | Downloading package perluniprops to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping misc\perluniprops.zip.
[nltk_data] | Downloading package pil to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\pil.zip.
[nltk_data] | Downloading package pl196x to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\pl196x.zip.
[nltk_data] | Downloading package porter_test to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping stemmers\porter_test.zip.
[nltk_data] | Downloading package ppattach to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\ppattach.zip.
[nltk_data] | Downloading package problem_reports to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\problem_reports.zip.
[nltk_data] | Downloading package product_reviews_1 to
[nltk_data] | C:\Users\SONU SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\product_reviews_1.zip.
[nltk_data] | Downloading package product_reviews_2 to
[nltk_data] | C:\Users\SONU SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\product_reviews_2.zip.
[nltk_data] | Downloading package propbank to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Downloading package pros_cons to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\pros_cons.zip.
[nltk_data] | Downloading package ptb to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\ptb.zip.
[nltk_data] | Downloading package punkt to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Package punkt is already up-to-date!
[nltk_data] | Downloading package punkt_tab to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping tokenizers\punkt_tab.zip.
```

```
[nltk_data] | Downloading package qc to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\qc.zip.
[nltk_data] | Downloading package reuters to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Downloading package rslp to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping stemmers\rslp.zip.
[nltk_data] | Downloading package rte to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\rte.zip.
[nltk_data] | Downloading package sample_grammars to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping grammars\sample_grammars.zip.
[nltk_data] | Downloading package semcor to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Downloading package senseval to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\senseval.zip.
[nltk_data] | Downloading package sentence_polarity to
[nltk_data] | C:\Users\SONU SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\sentence_polarity.zip.
[nltk_data] | Downloading package sentiwordnet to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\sentiwordnet.zip.
[nltk_data] | Downloading package shakespeare to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\shakespeare.zip.
[nltk_data] | Downloading package sinica_treebank to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\sinica_treebank.zip.
[nltk_data] | Downloading package smultron to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\smultron.zip.
[nltk_data] | Downloading package snowball_data to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Downloading package spanish_grammars to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping grammars\spanish_grammars.zip.
[nltk_data] | Downloading package state_union to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\state_union.zip.
[nltk_data] | Downloading package stopwords to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Package stopwords is already up-to-date!
[nltk_data] | Downloading package subjectivity to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\subjectivity.zip.
[nltk_data] | Downloading package swadesh to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\swadesh.zip.
[nltk_data] | Downloading package switchboard to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\switchboard.zip.
[nltk_data] | Downloading package tagsets to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
```

```
[nltk_data] | Unzipping help\tagsets.zip.
[nltk_data] | Downloading package tagsets_json to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping help\tagsets_json.zip.
[nltk_data] | Downloading package timit to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\timit.zip.
[nltk_data] | Downloading package toolbox to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\toolbox.zip.
[nltk_data] | Downloading package treebank to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\treebank.zip.
[nltk_data] | Downloading package twitter_samples to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\twitter_samples.zip.
[nltk_data] | Downloading package udhr to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\udhr.zip.
[nltk_data] | Downloading package udhr2 to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\udhr2.zip.
[nltk_data] | Downloading package unicode_samples to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\unicode_samples.zip.
[nltk_data] | Downloading package universal_tagset to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping taggers\universal_tagset.zip.
[nltk_data] | Downloading package universal_treebanks_v20 to
[nltk_data] | C:\Users\SONU SONI\AppData\Roaming\nltk_data...
[nltk_data] | Downloading package vader_lexicon to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Downloading package verbnet to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\verbnet.zip.
[nltk_data] | Downloading package verbnet3 to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\verbnet3.zip.
[nltk_data] | Downloading package webtext to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\webtext.zip.
[nltk_data] | Downloading package wmt15_eval to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping models\wmt15_eval.zip.
[nltk_data] | Downloading package word2vec_sample to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping models\word2vec_sample.zip.
[nltk_data] | Downloading package wordnet to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Downloading package wordnet2021 to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Downloading package wordnet2022 to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
[nltk_data] | Unzipping corpora\wordnet2022.zip.
[nltk_data] | Downloading package wordnet31 to C:\Users\SONU
[nltk_data] | SONI\AppData\Roaming\nltk_data...
```

```
[nltk_data] | Downloading package wordnet_ic to C:\Users\SONU
[nltk_data] |       SONI\AppData\Roaming\nltk_data...
[nltk_data] |       Unzipping corpora\wordnet_ic.zip.
[nltk_data] | Downloading package words to C:\Users\SONU
[nltk_data] |       SONI\AppData\Roaming\nltk_data...
[nltk_data] |       Unzipping corpora\words.zip.
[nltk_data] | Downloading package ycoe to C:\Users\SONU
[nltk_data] |       SONI\AppData\Roaming\nltk_data...
[nltk_data] |       Unzipping corpora\ycoe.zip.
[nltk_data] |
[nltk_data] Done downloading collection all
```

Out[54]: True

In [56]: **import** evaluate

```
# Load BLEU metric
bleu = evaluate.load("bleu")
predictions = ["a man riding a horse by the sea"]
references = ["a man riding a horse on the beach", "person on a horse at th

from nltk.tokenize import word_tokenize

predictions = [gen_caption.lower()] # already string
references = [gt_captions] # list of ground truth captions as stri

results = bleu.compute(predictions=predictions, references=references)
print(results)

{'bleu': 0.5410822690539396, 'precisions': [0.75, 0.5714285714285714, 0.5,
0.4], 'brevity_penalty': 1.0, 'length_ratio': 1.1428571428571428, 'translati
on_length': 8, 'reference_length': 7}
```

In []: