Evaluation Metrics for Text-Based Image Retrieval

In this project, the evaluation of the text-to-image retrieval system relies on the following key metrics:

1. Cosine Similarity (Similarity Ranking)

2. Top-K Retrieval (Recall@K-like behavior)

Below is a detailed explanation of these metrics along with the corresponding code snippets.

1. Cosine Similarity (Similarity Ranking)

Definition

Cosine similarity measures the similarity between two vectors based on the cosine of the angle between them. For this project, the cosine similarity is calculated between the textual query embeddings and the image feature embeddings.

The formula for cosine similarity is:

```
cosine_similarity(A, B) = (A . B) / (||A|| ||B||)
```

Where:

- A: Textual query feature vector.

- B: Image feature vector.

The closer the cosine similarity is to 1, the more similar the two vectors are.

Code Snippet

The similarity is computed as follows in the project:

```
# Compute similarities between text features and image features
similarity = torch.matmul(text_features, image_features.T)
```

Here:

- text_features contains the query text embedding.

- image_features contains the image feature embeddings.

- torch.matmul computes the dot product between the text and image embeddings.

Purpose

Cosine similarity ranks images based on their similarity to the input query in a shared feature space. It is computationally efficient and ideal for retrieval tasks.

2. Top-K Retrieval (Recall@K-like Behavior)

Definition

Top-K Retrieval focuses on selecting the K most similar images based on their cosine similarity scores. Although the formal Recall@K requires ground truth, this approach implicitly evaluates the ability of the model to rank relevant images higher.

Code Snippet

The top-K retrieval is performed using the topk function:

```
# Get top-K scores and indices
top_k_scores, top_k_indices = all_similarities.topk(k)
```

Here:

- all_similarities is a concatenated tensor of cosine similarity scores computed across all image batches.

- topk(k) returns:

  - top_k_scores: The highest K similarity scores.

  - top_k_indices: The indices of the corresponding images.

These indices are then used to retrieve the top-K images:

```
# Retrieve top-K images and scores
top_k_images = [(all_images[idx.item()], top_k_scores[i].item()) for i, idx in enumerate(top_k_indices)]
```

Purpose

Top-K retrieval evaluates the ranking capability of the model. It ensures that the most relevant images appear within the top-K positions, which is critical in real-world retrieval systems.

Output

The function outputs a list of top-K images and their corresponding similarity scores:

```
# Example output:
[
    (image_tensor_1, 0.95),  # Most similar image
    (image_tensor_2, 0.90),  # Second most similar image
    ...
]
```

In this output:

- image_tensor represents the image retrieved.

- The similarity score reflects how closely the image matches the query.

Conclusion

The current implementation uses Cosine Similarity to compute the similarity between query and image features and employs Top-K Retrieval to evaluate the quality of the ranked results. These metrics are effective for measuring the system's ability to return relevant images based on textual queries.