Explanation of Model Performance Evaluation (evaluate_model)

This section explains the function used to evaluate the performance of a model on a given dataset.

```
def evaluate_model(model, dataloader, data_type="image"):
    model.eval()
    correct, total = 0, 0
    with torch.no_grad():
        for data, target in dataloader:
            data, target = data.to(device), target.to(device)
            if data_type == "sequence":
                data = data.long()
                outputs = model(data)
            _, predicted = torch.max(outputs, 1)
            total += target.size(0)
            correct += (predicted == target).sum().item()
            accuracy = 100 * correct / total if total > 0 else 0
    return accuracy
```

Line-by-line explanation:

- model.eval(): Sets the model to evaluation mode (disables dropout, batchnorm, etc.).
- correct, total = 0, 0: Initializes counters for correct predictions and total samples.
- with torch.no_grad(): Disables gradient calculation for efficiency during evaluation.
- for data, target in dataloader: Iterates over batches of data and labels from the dataloader.
- data, target = data.to(device), target.to(device): Moves data and labels to the selected device (CPU or GPU).
- if data_type == 'sequence': If the data is a sequence (e.g., text), converts it to long integer type and

passes it to the model.

- outputs = model(data): Gets the model's predictions for the batch.

- _, predicted = torch.max(outputs, 1): Gets the predicted class for each sample in the batch.

- total += target.size(0): Increments the total sample count.

- correct += (predicted == target).sum().item(): Increments the correct prediction count.

- accuracy = 100 * correct / total if total > 0 else 0: Calculates accuracy as a percentage.

- return accuracy: Returns the computed accuracy.

Purpose:

- This function is used to evaluate how well a model performs on a dataset, returning the accuracy as a percentage.

- It supports both image and sequence (text) data types.