Explanation of Loading Predefined Datasets

This section explains the code used to load and preprocess standard datasets for training and evaluation.

```
transform_mnist = transforms.Compose([
    transforms.ToTensor(), transforms.Normalize((0.1307,), (0.3081,))])
mnist_train = datasets.MNIST(root='./data', train=True, download=True, transform=transform_mnist)
mnist_test = datasets.MNIST(root='./data', train=False, download=True, transform=transform_mnist)
mnist_loader = Dataloader(mnist_train, batch_size=64, shuffle=True)
mnist_test_loader = Dataloader(mnist_test, batch_size=64, shuffle=False)
```

- transform_mnist: Defines a sequence of transformations for MNIST images: converts to tensor and normalizes with mean 0.1307 and std 0.3081.
- mnist_train: Loads the MNIST training set, applies the transform, downloads if not present.
- mnist_test: Loads the MNIST test set, applies the transform, downloads if not present.
- mnist_loader: Creates a DataLoader for the training set with batch size 64 and shuffling enabled.
- mnist_test_loader: Creates a DataLoader for the test set with batch size 64 and shuffling disabled.

```
# Vocabulary for text data
imdb = load_dataset("imdb")
def yeild_tokens(data):
    for text in data:
        yield text.split()
vocab = build_vocab_from_iterator(yeild_tokens(imdb['train']['text']), specials=["<unk>"], max_tokens=10000)
vocab.set_default_index(vocab["<unk>"])
```

- imdb: Loads the IMDB movie review dataset using HuggingFace datasets.

- yeild_tokens: (typo, should be 'yield_tokens') Generator function that splits each text into tokens.

- vocab: Builds a vocabulary from the training texts, adds a special <unk> token, limits to 10,000 tokens.

- vocab.set_default_index: Sets the default index for unknown tokens to <unk>.

Purpose:

- These lines prepare standard datasets and vocabulary for use in model training and evaluation.

- Ensures consistent preprocessing and batching for both image and text data.