

Data Loading

Since we are now done with tokenization and have our cleansed dataset ready, we will move to data loading, i.e. making the dataset fit to be fed into the model directly...

Here's a brief explanation of the roles of each component in `torch.utils.data`:

1. **Dataset:**

- A base class for PyTorch datasets.
- It provides a way to define and customize your dataset by implementing two key methods:
 - `__len__`: Returns the size of the dataset.
 - `__getitem__`: Fetches a data sample by index.
- Example: Create a custom dataset for specific data processing needs.

2. **random_split:**

- Splits a dataset into non-overlapping subsets of given lengths.
- Useful for creating training, validation, and test splits.
- Example: `train_set, val_set = random_split(dataset, [80, 20])`.

3. **DataLoader:**


- Wraps a dataset to provide batch loading, shuffling, and parallel data loading.
- Handles batching, shuffling, and multiprocessing (for faster data loading).
- Example: `train_loader = DataLoader(train_set, batch_size=32, shuffle=True)`.

4. **RandomSampler:**

- Samples elements randomly from a dataset without replacement.
- Useful for shuffling when more control is needed than just using `shuffle=True` in `DataLoader`.
- Example: `sampler = RandomSampler(dataset)`.


5. **SequentialSampler:**

- Samples elements sequentially, i.e., in the order they appear in the dataset.
- Useful for inference or when the order of data matters.
- Example: `sampler = SequentialSampler(dataset)`.

 `Data Loading.ipynb` please go through this colab file to understand how to load the data into a suitable form

Pipeline functions and model configuration

Pipeline functions offer an easy way to use Hugging Face Transformers for a variety of NLP tasks, such as sentiment analysis, text classification, and text generation, with minimal code. In this module, we'll explore common pipeline tasks, show how to customize them with specific models, and dive into GPT-2 model configurations to understand how to fine-tune and optimize them for specific needs. Let's get started!

 Pipeline functions.ipynb

A Bit About the **datasets** Library

The **datasets** library from Hugging Face provides an efficient way to access, load, and process a variety of datasets for NLP tasks. With a few simple commands, you can:

- Load large datasets.
- Explore and customize data processing.
- Prepare data for model training or evaluation.

Refer to the following articles for an overview:

[Introduction - Hugging Face NLP Course](#)

[What if my dataset isn't on the Hub? - Hugging Face NLP Course](#)

[Time to slice and dice - Hugging Face NLP Course](#)