Load a dataset from the Hub

Finding high-quality datasets that are reproducible and accessible can be difficult. One of 🤗 Datasets main goals is to provide a simple way to load a dataset of any format or type. The easiest way to get started is to discover an existing dataset on the Hugging Face Hub - a community-driven collection of datasets for tasks in NLP, computer vision, and audio - and use 🤗 Datasets to download and generate the dataset.

This tutorial uses the rotten_tomatoes and MInDS-14 datasets, but feel free to load any dataset you want and follow along. Head over to the Hub now and find a dataset for your task!

Load a dataset

Before you take the time to download a dataset, it's often helpful to quickly get some general information about a dataset. A dataset's information is stored inside DatasetInfo and can include information such as the dataset description, features, and dataset size.

Use the load_dataset_builder() function to load a dataset builder and inspect a dataset's attributes without committing to downloading it:

```
>>> from datasets import load_dataset_builder
>>> ds_builder = load_dataset_builder("cornell-movie-review-data/rotten_tomatoes")

# Inspect dataset description
>>> ds_builder.info.description
Movie Review Dataset. This is a dataset of containing 5,331 positive and 5,331 negative processed

# Inspect dataset features
>>> ds_builder.info.features
{'label': ClassLabel(names=['neg', 'pos']),
 'text': Value('string')}
```

If you're happy with the dataset, then load it with load_dataset():

```
>>> from datasets import load_dataset

>>> dataset = load_dataset("cornell-movie-review-data/rotten_tomatoes", split="train")
```

Splits

A split is a specific subset of a dataset like `train` and `test`. List a dataset's split names with the get_dataset_split_names() function:

```
>>> from datasets import get_dataset_split_names

>>> get_dataset_split_names("cornell-movie-review-data/rotten_tomatoes")
['train', 'validation', 'test']
```

Then you can load a specific split with the `split` parameter. Loading a dataset `split` returns a Dataset object:

```
>>> from datasets import load_dataset

>>> dataset = load_dataset("cornell-movie-review-data/rotten_tomatoes", split="train")
>>> dataset
Dataset({
    features: ['text', 'label'],
    num_rows: 8530
})
```

If you don't specify a `split`, 🤗 Datasets returns a DatasetDict object instead:

```
>>> from datasets import load_dataset

>>> dataset = load_dataset("cornell-movie-review-data/rotten_tomatoes")
DatasetDict({
    train: Dataset({
        features: ['text', 'label'],
        num_rows: 8530
    })
    validation: Dataset({
        features: ['text', 'label'],
        num_rows: 1066
    })
    test: Dataset({
        features: ['text', 'label'],
        num_rows: 1066
    })
})
```

Configurations

Some datasets contain several sub-datasets. For example, the MInDS-14 dataset has several sub-datasets, each one containing audio data in a different language. These sub-datasets are known as *configurations* or *subsets*, and you must explicitly select one when loading the dataset. If you don't provide a configuration name, Datasets will raise a `ValueError` and remind you to choose a configuration.

Use the get_dataset_config_names() function to retrieve a list of all the possible configurations available to your dataset:

```
>>> from datasets import get_dataset_config_names

>>> configs = get_dataset_config_names("PolyAI/minds14")
>>> print(configs)
['cs-CZ', 'de-DE', 'en-AU', 'en-GB', 'en-US', 'es-ES', 'fr-FR', 'it-IT', 'ko-KR', 'nl-NL', 'pl-PL'
```

Then load the configuration you want:

```
>>> from datasets import load_dataset

>>> mindsFR = load_dataset("PolyAI/minds14", "fr-FR", split="train")
```